

MEDIATEK

everyday genius

MT2503D SOC Processor Data Sheet

Version: 1.0
Release date: 2015-12-14

Specifications are subject to change without notice.

© 2015 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc.

Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Document Revision History

Revision	Date	Author	Description
0.1	2015-10-08	Sharon Chu	Initial draft
1.0	2015-12-14	Sharon Chu	

Table of Contents

Document Revision History	2
Table of Contents	3
1 System Overview	12
1.1 Platform Features	16
1.2 MODEM Features.....	18
1.3 GSM/GPRS RF Features	19
1.4 Multimedia Features	20
1.5 Bluetooth Features	22
1.6 FM Features	23
1.7 GPS feature	24
1.8 General Descriptions	26
2 Product Descriptions	28
2.1 Pin Description.....	28
2.1.1 Ball Diagram.....	28
2.1.2 Pin Coordination.....	28
2.1.3 Detailed Pin Description	31
2.1.4 Pin Multiplexing, Capability and Settings	45
2.2 Electrical Characteristics	52
2.2.1 Absolute Maximum Ratings	52
2.2.2 Recommended Operating Conditions	53
2.2.3 Electrical Characteristics under Recommended Operating Conditions	54
2.3 System Configuration	61
2.3.1 Strapping Resistors	61
2.3.2 Mode Selection.....	61
2.4 Power-on Sequence and Protection Logic	61

2.5	Analog Baseband	65
2.5.1	APC-DAC	65
2.5.2	Auxiliary ADC	66
2.5.3	Audio Mixed-Signal Blocks.....	67
2.6	Power Management Unit Blocks	69
2.6.1	LDO	70
2.6.2	BOOST	74
2.6.3	ISINK and KPLED Switches.....	74
2.6.4	STRUP	76
2.6.5	PCHR	76
2.7	GSM/GPRS RF.....	82
2.7.1	General Description.....	82
2.7.2	Functional Block Diagram	83
2.7.3	Electrical Characteristics	83
2.8	Bluetooth.....	87
2.8.1	Block Description.....	87
2.8.2	Functional Specifications	88
2.9	FM RF.....	90
2.9.1	Block Description.....	90
2.9.2	Functional Specifications	91
2.10	Package Information.....	92
2.10.1	Package Outlines	92
2.10.2	Thermal Operating Specifications	95
2.10.3	Lead-free Packaging	95
2.11	Ordering Information.....	95
2.11.1	Top Marking Definition	95
3	Micro-Controller Unit Peripherals.....	96
3.1	Pulse-Width Modulation Outputs (2 Channel).....	96

3.1.1	General Description.....	96
3.1.2	Register Definition	97
3.2	SIM Interface	101
3.2.1	Register Definition	103
3.2.2	SIM Card Insertion and Removal	118
3.2.3	Card activation and Deactivation	118
3.2.4	Answering to Reset Sequence	118
3.2.5	SIM Data Transfer	119
3.3	Keypad Scanner	122
3.3.1	General Description.....	122
3.3.2	Register Definitions	125
3.4	General Purpose Inputs/Outputs.....	132
3.4.1	General Description.....	132
3.4.2	Register Definitions	133
3.5	General-purpose Timer	269
3.5.1	General Descriptions.....	269
3.5.2	Register Definition.....	270
3.5.3	Application Note	273
3.6	MCU OSTIMER	274
3.6.1	Overview	274
3.6.2	Terminology.....	274
3.6.3	Introduction to Wakeup Source.....	274
3.6.4	Register Definition	276
3.7	UART1	286
3.7.1	General Description.....	286
3.7.2	Register Definition	287
3.8	UART2.....	301
3.8.1	General Description.....	301

3.8.2	Register Definition	303
3.9	UART3	318
3.9.1	General Description.....	318
3.9.2	Register Definitions	319
3.10	I2C/SCCB Controller	332
3.10.1	General Description.....	332
3.10.2	Programming Examples	334
3.10.3	Register Definition	335
3.11	I2C/SCCB Controller (I2C_SCCB_Controller_V18)	345
3.11.1	General Description.....	345
3.11.2	Programming Examples	347
3.11.3	Register Definition	348
3.12	Real Time Clock	358
3.12.1	General Descriptions.....	358
3.12.2	Register Definition	358
3.13	Auxiliary ADC Unit	373
3.13.1	Register Definition.....	376
3.13.2	General Programming Guide	384
3.13.3	Usage Programming Guide.....	384
3.13.4	Performance Programming Guide	384
3.13.5	AUXADC PDN.....	385
3.13.6	Notice	385
3.14	USB Device Controller.....	385
3.14.1	General Description.....	385
3.14.2	Terminology.....	386
3.14.3	Register Definition	387
3.14.4	System Integration Guide.....	402
3.15	Accessory Detector	403

3.15.1	General Description.....	403
3.15.2	Pulse Width Modulation	405
3.15.3	Register Definition	405
3.16	SD Memory Card Controller (MSDC0)	415
3.16.1	Introduction.....	415
3.16.2	Overview	415
3.16.3	Register Definition	416
3.17	SD Memory Card Controller (MSDC1)	441
3.17.1	Introduction.....	441
3.17.2	Overview	442
3.17.3	Register Definition	443
3.18	BTIF	468
3.18.1	General Description.....	468
3.18.2	Register Definition	469
4	GPS	475
4.1	RF Part	475
4.1.1	LNA/Mixer.....	475
4.1.2	VCO/Synthesizer.....	475
4.1.3	LPF	475
4.1.4	ADC.....	475
4.2	Digital Part	476
4.2.1	ARM7EJ-S.....	476
4.2.2	Cache	476
4.2.3	Boot ROM.....	476
4.2.4	Real Time Clock (RTC)	476
4.2.5	SMPS	477
4.2.6	Timer function.....	477
4.2.7	GPIO in RTC domain	477

4.2.8	Low power detection	477
4.2.9	Clock module.....	477
4.2.10	Reset controller	478
4.2.11	Host interface	479
4.2.12	Interrupt control unit	479
4.2.13	Flash.....	479
4.2.14	GPIO unit.....	480
4.2.15	PPS	480
4.2.16	ECLK	480
4.2.17	SYNC	480
4.2.18	Power scheme.....	481
4.3	Electrical Characteristics	483
4.3.1	DC characteristics	483
4.3.2	Analog related characteristics	486
4.3.3	RF related characteristics	487
4.4	Interface Characteristics.....	488
4.4.1	JTAG interface timing.....	488
4.4.2	RS-232 interface timing.....	489
4.4.3	SPI interface timing	490
4.4.4	I2C interface timing	490

Lists of Tables and Figures

Table 1. Pin coordinates	28
Table 2. Acronym for pin types	31
Table 3. PIN function description and power domain.....	31
Table 4. Acronym for state of pins	41
Table 5. State of pins.....	41
Table 6. Acronym for pull-up and pull-down types	45
Table 7. Capability of PU/PD, driving and Schmitt trigger.....	45
Table 8. Absolute maximum ratings for power supply.....	52
Table 9. Absolute maximum ratings for voltage input	53
Table 10. Absolute maximum ratings for storage temperature	53

Table 11. Recommended operating conditions for power supply	53
Table 12. Recommended operating conditions for voltage input	53
Table 13. Recommended operating conditions for operating temperature	54
Table 14. Electrical characteristics	54
Table 15. Strapping table	61
Table 16. Mode selection of chip	61
Table 17. APC-DAC specifications	65
Table 18. Functional specifications of auxiliary ADC	66
Table 19. Functional specifications of analog voice blocks	68
Table 20. Functional specifications of analog audio blocks	69
Table 21. LDO types and brief specifications	72
Table 22. Analog LDO specifications	73
Table 23. Digital LDO specifications	73
Table 24. RTC LDO specification	74
Table 25. RTC LDO specification	74
Table 26. ISINKs and KPLED Switches Specification	75
Table 27. Charger detection specifications	80
Table 28. Pre-charge specifications	80
Table 29. Constant current specifications	80
Table 30. Constant voltage and over-voltage protection specifications	81
Table 31. BC1.1 specifications	81
Table 32. DC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)	83
Table 33. Rx AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)	84
Table 34. Tx GMSK AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)	85
Table 35. SX AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)	86
Table 36. DCXO AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)	86
Table 37. Basic data rate – receiver specifications	88
Table 38. Basic data rate – transmitter specification	88
Table 39. Enhanced data rate – receiver specifications	89
Table 40. Enhanced data rate – transmitter specifications	89
Table 41. FM receiver DC characteristics (TA=25°C, VDD=2.8V unless otherwise stated)	91
Table 42. FM receiver AC characteristics	91
Table 43. Time-out condition for answering to reset sequence	119
Table 44. 5*5 double KEY's order number in COL/ROW matrix	132
Table 45. 5*5 triple KEY's order number in COL/ROW matrix	132
Table 46. Abbreviations	274
Table 47. Wakeup sources	275
Table 48. Characteristics of wakeup sources	275
Table 49. Relationship between commands and touch panel control signals	375
Table 50. Sharing of pins for SD memory card controller	415
Table 51. Sharing of pins for SD memory card controller	442
Table 52. IIR[5:0] codes associated with the possible interrupts	470

Figure 1. Typical application of MT2503D.....	15
Figure 2. MT2503D block diagram.....	27
Figure 3. Ball diagram and top view.....	28
Figure 4. IO types in state of pins.....	44
Figure 5. Power-on/off control sequence by pressing PWRKEY and XOSC32_ENB = 0.....	62
Figure 6. Power-on/off control sequence by pressing PWRKEY and XOSC32_ENB = 1.....	63
Figure 7. Block diagram of audio mixed-signal blocks.....	68
Figure 8. PMU system block diagram.....	70
Figure 9. Power domain.....	71
Figure 10. LDO block diagram.....	72
Figure 11. ISINKs and KPLED switches bock diagram.....	75
Figure 12. PCHR block diagram.....	77
Figure 13. Charging states diagram.....	78
Figure 14. Diagram of MT2503D 2G RFSYS.....	83
Figure 15. System diagram of Bluetooth RF transceiver.....	87
Figure 16. Block diagram of hardware top-level architecture.....	91
Figure 17. Outlines and dimension of VFBGA 8.4mm*6.2mm, 215-ball, 0.4 mm pitch package.....	94
Figure 18. Mass production top marking of MT2503D.....	95
Figure 19. PWM waveform.....	96
Figure 20. PWM waveform with register values.....	97
Figure 21. Block diagram of SIM interface.....	101
Figure 22. Timing diagram of SIM interface.....	103
Figure 23. Answering to reset sequence.....	119
Figure 24. 5x5 double keypad matrix (50 keys).....	123
Figure 25. 5x5 triple keypad matrix (75 keys).....	124
Figure 26. 5*5 double keypad scan waveform.....	124
Figure 27. One key pressed with de-bounce mechanism denoted.....	125
Figure 28. (a) Two keys pressed, case 1; (b) Two keys pressed, case 2.....	125
Figure 29. kp timing register.....	131
Figure 30. GPIO block diagram.....	133
Figure 31. OS timer system view.....	274
Figure 32. Wakeup event and Irq_b integration diagram.....	276
Figure 33. Pause command complete and pause request state.....	280
Figure 34. Debug wakeup events.....	286
Figure 35. Block Diagram of UART1.....	287
Figure 36. Block Diagram of UART2.....	302
Figure 37. Block Diagram of UART3.....	319
Figure 38. AUXADC architecture.....	373
Figure 39. Touch panel circuit structure.....	374
Figure 40. Touch panel sampling waveform.....	375
Figure 41. Suggested Accessory Detection Circuit.....	404
Figure 42. The Sate machine between Microphone and Hook-Switch plug-in/out change.....	405
Figure 43. PWM waveform.....	405
Figure 44. PWM waveform with register value present.....	408
Figure 45. Card detection for SD memory card.....	416
Figure 46. Card detection for SD memory card.....	443
Figure 47. Card detection for SD memory card (Scheme 2).....	443
Figure 48. Interface connection between BT and baseband system.....	468
Figure 49. RTC with internal RTC LDO application circuit 1.....	476
Figure 50. RTC with internal RTC LDO application circuit 2.....	477
Figure 51. Power on reset diagram.....	478
Figure 52. Power on/off reset behavior.....	478
Figure 53. Flow diagram of SYNC function.....	480

Figure 54. Power supply connection (low power) 481
 Figure 55. Power supply connection (low cost) 482
 Figure 56. Power supply connection (external LDO) 482
 Figure 57. Power on/off sequence for external LDO mode 483
 Figure 58. Timing diagram of JTAG interface 489
 Figure 59. Timing diagram of RS-232 interface 489
 Figure 60. Timing diagram of SPI interface..... 490
 Figure 61. Timing diagram of HOST I2C interface..... 491

1 System Overview

MT2503D is a monolithic chip integrating leading edge power management unit, analog baseband and radio circuitry based on the low-power CMOS process.

MT2503D is a feature-rich and extremely powerful single-chip solution for high-end GSM/GPRS capability. Based on the 32-bit ARM7EJ-S™ RISC processor, MT2503D's superb processing power, along with high bandwidth architecture and dedicated hardware support, provides a platform for high-performance GPRS Class 12 MODEM application and leading-edge multimedia applications.

MT2503D also features:

- A highly integrated Bluetooth transceiver which is fully compliant with Bluetooth specification v3.0.
- A FM receiver supporting both audio broadcast de-modulation and RDS/RBDS data decoding.

Typical application diagram is shown in **Figure 1**.

Platform

MT2503D is capable of running the ARM7EJ-S™ RISC processor, which provides the best trade-off between system performance and power consumption.

For large amounts of data transfer, high-performance DMA (Direct Memory Access) with hardware flow control is implemented, which greatly enhances the data movement speed while reducing the MCU processing load.

Targeted as a media-rich platform for mobile applications, MT2503D also provides hardware

security digital rights management for copyright protection. For further safeguard and to protect the manufacturer's development investment, hardware flash content protection is provided to prevent unauthorized porting of the software load.

Memory

MT2503D supports serial flash interface with various operating frequencies.

Multimedia

The MT2503D multimedia subsystem provides serial interface for CMOS sensors. The camera resolution is up to VGA size. The software-based codec can be used to process various video types. To take advantage of the high MCU performance, GIF and PNG decoders are implemented by the software.

In addition, MT2503D is implemented with a high-performance audio synthesis technology, as well as a high-quality audio amplifier to provide superior audio experiences.

Connectivity and storage

MT2503D supports UART, USB 1.1 FS/LS, SDIO and SD storage systems. These interfaces provide MT2503D users with the highest level of flexibility in implementing high-end solutions.

To achieve a complete user interface, MT2503D also brings together all the necessary peripheral blocks for a multimedia GSM/GPRS phone. The peripheral blocks include the keypad scanner with the capability to detect multiple key presses, SIM controller, real-time clock, PWM, serial LCD

controller and general-purpose programmable I/Os.

Audio

Using a highly integrated mixed-signal audio front-end, the MT2503D architecture provides easy audio interfacing with direct connection to the audio transducers. The audio interface integrates A/D converters for voice band, as well as high-resolution stereo D/A converters for both audio and voice band.

MT2503D supports AMR codec to adaptively optimize the quality of speech and audio. Moreover, HE-AAC codec is implemented to deliver CD-quality audio at low bit rates.

In addition, an 1.2W audio amplifier is also embedded to save the BOM cost of adopting external amplifiers.

GSM/GPRS radio

MT2503D integrates a mixed-signal baseband front-end in order to provide a well-organized radio interface with flexibility for efficient customization. The front-end contains gain and offset calibration mechanisms and filters with programmable coefficients for comprehensive compatibility control on RF modules. MT2503D achieves outstanding MODEM performance by utilizing a highly dynamic range ADC in the RF downlink path.

MT2503D embeds a high-performance and completely integrated single-ended SAW-less RF transceiver for multi-band GSM cellular system. In this RF transceiver, a quad-band receiving feature with high sensitivity is supported utilizing one RF receiver and a fully integrated channel filter. With ultra-high dynamic range, the off-chip balun and SAW filters on the receiving path can be removed for BOM cost

reduction. In addition, the minimum component count is guaranteed by realizing a highly integrated transmitter, low-spur frequency synthesizer and a Digitally-Controlled Crystal Oscillator (DCXO).

GPS

A high-performance single-chip multi-GNSS solution which includes on-chip CMOS RF, digital baseband, ARM7 CPU and an embedded NOR flash. It is able to achieve the industry's highest level of sensitivity, accuracy and Time-to-First-Fix (TTFF) with the lowest power consumption in a small-footprint lead-free package. Its small footprint and minimal BOM requirement provide significant reductions in the design, manufacturing and testing resource required for portable applications.

With built-in LNA to reach total receiver chain NF to 2.2 dB, you can eliminate antenna requirement and do not need external LNA. It also up to 12 multi-tone active interference cancellers (ISSCC2011 award) offer you more flexibility in system design.

MT2503D acquires and tracks satellites in the shortest time even at indoor signal levels. In addition, MT2503D supports various location and navigation applications, including autonomous GPS, GLONASS, GALILEO, BEIDOU(after ICD released), SBAS ranging (WAAS, EGNOS, GAGAN, and MSAS), QZSS, DGPS (RTCM) and AGPS.

Bluetooth radio

MT2503D offers a highly integrated Bluetooth radio and baseband processor. Only a minimum of external components are required. MT2503D provides superior sensitivity and class 1 output power and thus ensures the quality of the

connection with a wide range of Bluetooth devices.

MT2503D is fully compliant with Bluetooth v3.0 and offers enhanced data rates of up to 3Mbps. It also provides the coexistence protocol with 802.11 system.

MT2503D supports rich Bluetooth profiles, enabling diversified applications that are widely used on the handset with excellent interoperability.

FM radio

The FM radio subsystem provides a completely integrated FM Rx receiver supporting 65 ~ 108MHz FM bands with 50kHz tuning step. In addition to receiving FM audio broadcasting, the digital RDS/RBDS data system is supported as well. The integrated FM transceiver utilizes state-of-the-art digital demodulation/modulation techniques to achieve excellent performance.

In order to achieve high SINAD, good sensitivity and excellent noise suppression, the FM receiver adopts adaptive demodulation scheme to optimize Rx system performance in all ranges of signal quality by reference of a very sophisticated channel quality index (CQI). When the received signal quality is poor, the design not only enhances the ACI rejection capability but also uses a very ingenious skill to soft mute annoying noise so as to provide good perception quality.

The FM radio subsystem supports both long antenna, which is usually an earphone, and auto-calibrated short antenna, which is usually a FPC short antenna or shared antenna with GSM for different application scenarios.

Debugging function

The JTAG interface enables in-circuit debugging of the software program with the ARM7EJ-S™ core. With this standardized debugging interface, MT2503D provides developers with a wide set of options in choosing ARM development kits from different third party vendors.

Power management

A power management is embedded in MT2503D to provide rich features a high-end feature phone supports, including Li-ion battery charger, high performance and low quiescent current LDOs, and drivers for LED and backlight.

MT2503D offers various low-power features to help reduce the system power consumption. MT2503D is also fabricated in an advanced low-power CMOS process, hence providing an overall ultra-low leakage solution.

Package

The MT2503D device is offered in a 8.4mm×6.2mm, 215-ball, 0.4mm pitch, VFBGA package.

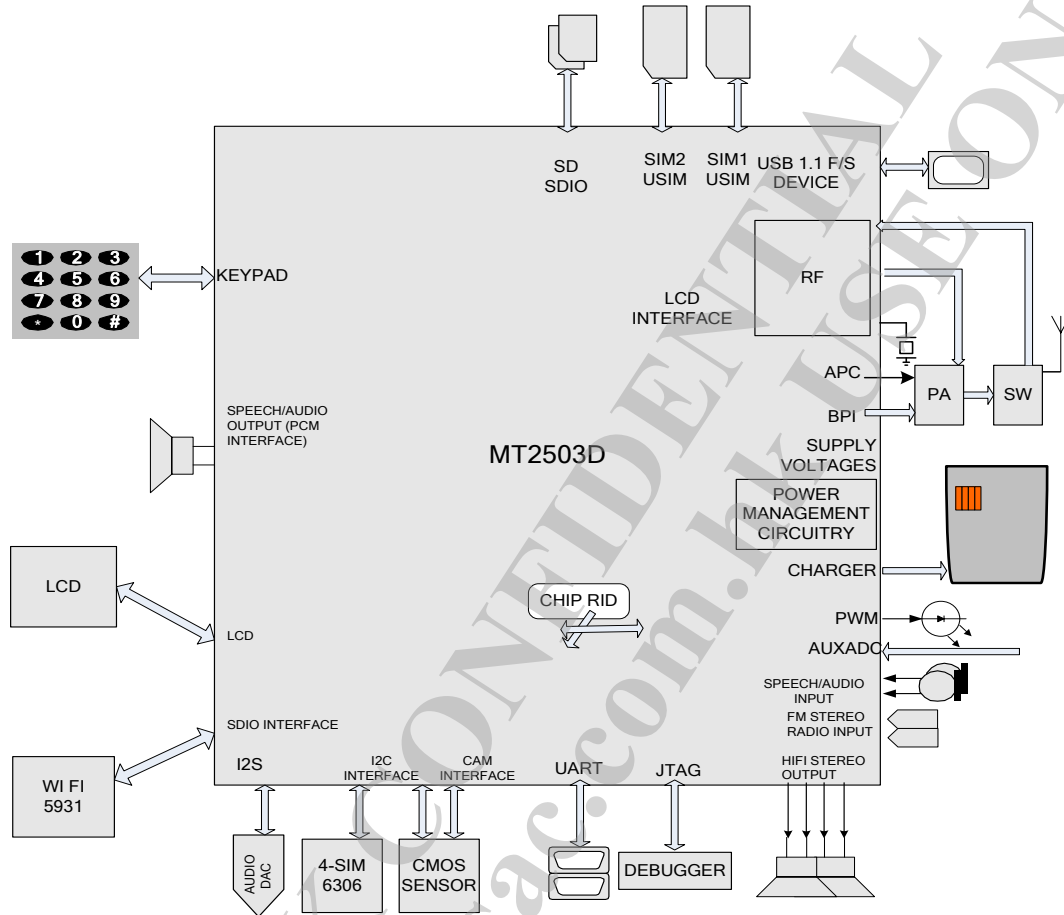


Figure 1. Typical application of MT2503D

1.1 Platform Features

General

- Integrated voice-band, audio-band and base-band analog front-end
- Integrated full-featured power management unit

MCU subsystem

- ARM7EJ-S™ 32-bit RISC processor
- Java hardware acceleration for fast Java-based games and applets
- High-performance multi-layer AHB bus
- Dedicated DMA bus with 16 DMA channels
- On-chip boot ROM for factory flash programming
- Watchdog timer for system crash recovery
- 3 sets of general-purpose timers
- Division coprocessor

User interfaces

- 5-row x 5-column keypad controller with hardware scanner
- Supports multiple key presses for gaming
- Dual SIM/USIM controller with hardware T = 0/T = 1 protocol control
- Real-time clock (RTC) operating with a low-quiescent-current power supply
- General-purpose I/Os (GPIOs) available for auxiliary applications
- 2 sets of Pulse Width Modulation (PWM) output
- 24 external interrupt lines
- 1 external channel auxiliary 10-bit A/D converter

Security

- Supports security key and chip random ID

Connectivity

- 3 UARTs with hardware flow control and supports baud rate up to 921,600 bps
- FS/LS USB 1.1 device controller
- Multimedia card, secure digital Memory Card, host controller with flexible I/O voltage power
- Supports 4-bit SDIO interface for SDIO peripherals as well as WIFI connectivity
- DAI/PCM and I2S interface for audio applications
- I2C master interface for peripheral management including image sensors
- SPI master/slave interface for peripheral management.

Power management

- Li-ion battery charger
- 13 LDOs for the power supply of memory card, camera, Bluetooth, RF, SIM card and other diversified usage
- 1 open-drain output switches to supply/control the LED
- LDO type vibrator
- One NMOS switch to control keypad LED
- Thermal overload protection
- Under-voltage lock-out protection
- Over-voltage protection
- Different levels of power-down modes with sophisticated software control enables excellent power saving performance.

Test and debugging

- Built-in digital and analog loop back modes for both audio and baseband front-end

- DAI port complies with GSM Rec.11.10.
- JTAG port for debugging embedded MCU

1.2 MODEM Features

Radio interface and baseband front-end

- Digital PM data path with baseband front-end
- High dynamic range delta-sigma ADC converts the downlink analog I and Q signals to digital baseband.
- 10-bit D/A converter for Automatic Power Control (APC)
- Programmable radio Rx filter with adaptive gain control
- Dedicated Rx filter for FB acquisition
- 6-pin Baseband Parallel Interface (BPI) with programmable driving strength
- Supports multi-band

Voice and modem CODEC

- Dial tone generation
- Voice memo
- Noise reduction
- Echo suppression
- Advanced sidetone oscillation reduction
- Digital sidetone generator with programmable gain
- Two programmable acoustic compensation filters
- Supports GSM/GPRS modem
- GSM quad vocoders for adaptive multirate (AMR), enhanced full rate (EFR), full rate (FR) and half rate (HR)
- GSM channel coding, equalization and A5/1, A5/2 and A5/3 ciphering
- GPRS GEA1, GEA2 and GEA3 ciphering
- GPRS packet switched data with CS1/CS2/CS3/CS4 coding schemes
- GPRS Class 12

- Supports SAIC (single antenna interference cancellation) technology
- Supports VAMOS(Voice services over Adaptive Multi-user channels on One Slot) technology in R9 spec.

Voice interface and voice front-end

- Microphone input has one low-noise amplifier with programmable gain and Automatic Gain Control (AGC) mechanisms
- Voice power amplifier with programmable gain
- 2nd order Sigma-Delta A/D converter for voice uplink path
- Shares D/A converter with audio playback path
- Supports full-duplex hands-free operation
- Compliant with GSM 03.50

1.3 GSM/GPRS RF Features

Receiver

- Dual single-ended LNAs support Quad band Quadrature RF mixer
- Fully integrated channel filter
- High dynamic range ADC
- 12dB PGA gain with 6dB gain step

Transmitter

- Transmitter outputs support quad bands.
- Highly precise and low noise RF transmitter for GSM/GPRS applications

Frequency synthesizer

- Programmable fractional-N synthesizer
- Integrated wide range RFVCO
- Integrated loop filter
- Fast settling time suitable for multi-slot GPRS applications

Digitally-Controlled Crystal Oscillator (DCXO)

- Two-pin 26MHz crystal oscillator
- On-chip programmable capacitor array for coarse-tuning
- On-chip programmable capacitor array for fine-tuning
- Low power mode supports 32K crystal removal

1.4 Multimedia Features

LCD controller

- Supports simultaneous connection to serial 2 lane LCD modules
- LCM formats supported: RGB565, RGB666, RGB888
- Supports LCD module with maximum resolution up to 320x240
- Per pixel alpha channel
- True color engine
- Supports hardware display rotation
- Capable of combining display memories with up to 4 blending layers

Camera interface

- YUV422 format image input
- Capable of processing image of size up to VGA (MediaTek serial interface)

JPEG decoder

- Baseline JPEG decoding
- Supports various YUV formats, DC/AC Huffman tables and quantization tables

JPEG encoder

- ISO/IEC 10918-1 JPEG baseline mode
- ISO/IEC 10918-2 compliance
- Supports YUV420 and grayscale formats
- Supports EXIF/JFIF
- Standard DC and AC Huffman tables
- Provides 5 levels of encode quality
- Supports zeros shutter delay

MJPEG

- Decode spec: CIF@30fps
- Encode spec: QVGA@15fps

Image data processing

- Supports 4x digital zoom

- High throughput hardware scaler. Capable of tailoring an image to an arbitrary size.
- Horizontal scaling with bilinear interpolation
- Vertical scaling with bilinear interpolation
- YUV and RGB color space conversion
- RGB/YCbCr format thumbnail output

MPEG-4/H.263 CODEC

- Software-based MPEG4 encoder
- Software-based MPEG4 decoder
- ISO/IEC 14496-2 simple profile:
 - Decode spec: 480x320@25fps
 - Encode spec: QVGA@15fps
- ISO/IEC 14496-2 advanced simple profile:
 - Decode @ level 0/1/2/3
 - ITU-T H.263 profile 0 @ level 40
- Supports visual tools for decoder: I-VOP, P-VOP, B-VOP, AC/DC prediction, 4-MV, unrestricted MV, error resilience, short header, global motion compensation, method 1/2 quantization, quarter-pel motion compensation.
- Error resilience for decoder: Slice resynchronization, data partitioning, reversible VLC
- Supports visual tools for encoder: I-VOP, P-VOP, Half-Pel, DC prediction, unrestricted MV, short header

H.264

- ISO/IEC 14496-10 baseline profile
 - Decode spec: QCIF@30fps

2D accelerator

- Supports 32-bpp ARGB8888, 24-bpp RGB888, 16-bpp RGB565, 24-bpp ARGB6666.

- 4 layers overlay with individual color format, window size, source key, constant alpha and rotation
- Rectangle fill with constant
- BitBlt: Capable with 7 rotation types
- Alpha blending with 7 rotation types, per-pixel alpha and pre-multiplied alpha
- Font drawing: Normal font and anti-aliasing font

Audio CODEC

- Supports AAC codec decoding
- Wavetable synthesis with up to 64 tones
- Advanced wavetable synthesizer capable of generating simulated stereo
- Wavetable including GM full set of 128 instruments and 47 sets of percussions
- PCM playback and record
- Digital audio playback

Audio interface and audio front-end

- Supports I2S interface
- High-resolution D/A converters for stereo audio playback
- Voice band A/D converter support
- Stereo to mono conversion

1.5 Bluetooth Features

Radio features

- Fully compliant with Bluetooth specification 3.0
- Low out-of-band spurious emissions support simultaneous operation with GPS and GSM/GPRS worldwide radio systems
- Low-IF architecture with high degree of linearity and high order channel filter
- Integrated T/R switch and Balun
- Fully integrated PA provides 7.5dBm output power
- -95dBm sensitivity with excellent interference rejection performance
- Hardware AGC dynamically adjusts receiver performance in changing environments

Baseband features

- Up to 4 simultaneous active ACL links
- Up to 1 simultaneous SCO or eSCO link with CVSD coding
- Supports eSCO
- Scatternet support: Up to 4 piconets simultaneously with background inquiry/page scan
- Supports sniff mode
- AFH and PTA collaborative support for WLAN/BT coexistence
- Idle mode and sleep mode enables ultra-low power consumption.
- Supports PCM interface and built-in programmable transcoders for linear voice with re-transmission
- Built-in hardware modem engine for access code correlation, header error correction, forward error correction, CRC, whitening and encryption

- Channel quality driven data rate adaptation
- Channel assessment for AFH

Platform features

- Embedded processor for Bluetooth protocol stack with built-in memory system
- Fully verified ROM based system with code patch for feature enhancement

1.6 FM Features

- 65-108MHz worldwide FM bands with 50KHz tuning step
- Supports RDS/RBDS radio data system
- Digital stereo demodulator
- Adaptive FM demodulator for both high- and low-quality scenarios
- Low sensitivity level with superior interference rejection
- Programmable de-emphasis (bypass/50 S/75 S)
- Stereophonic multiplex signal (MPX) signal detection and demodulation
- Superior stereo noise reduction and soft mute volume control
- Audio dynamic range control
- Mono/stereo blending
- Audio sensitivity $3\text{dB}\mu\text{Vemf}$ (SINAD=26dB)
- Audio SINAD $\geq 60\text{dB}$
- Supports Anti-jamming algorithm
- Supports short antenna

1.7 GPS feature

Specifications

- GPS/GLONASS/GALILEO/BEIDOU(after ICD released) receiver
- Supports multi-GNSS incl. QZSS, SBAS ranging
- Supports WAAS/EGNOS/MSAS/GAGAN
- 12 multi-tone active interference cancellers (ISSCC2011 award)
- RTCM ready
- Indoor and outdoor multi-path detection and compensation
- Supports FCC E911 compliance and A-GPS
- Max. fixed update rate up to 10 Hz

Advanced software features

- AlwaysLocate™ advanced location awareness technology
- EPOTM/HotStill™ orbit prediction
- EASY™ self-generated orbit prediction
- Supports logger function
- Supports time service application which can be achieved by PPS VS. NMEA feature.

Reference oscillator

- TCXO
Frequency: 16.368 MHz, 12.6 ~ 40.0 MHz
Frequency variation: ± 2.5 ppm
- Crystal
Frequency: 26 MHz, 12.6 ~ 40.0 MHz
Frequency accuracy: ± 10 ppm

RF configuration

- SoC, integrated in single chip with CMOS process

ARM7EJ-S CPU

- Up to 158 MHz processor clock
- Dynamic clock rate control

Pulse-per-second (PPS) GPS time reference

- Adjustable duty cycle
- Typical accuracy: ± 10 ns

Power scheme

- A 1.8 volts SMPS build-in SOC
- Direct lithium battery connection (2.8 ~ 4.3 volts)
- Self build 1.1 volts RTC LDO, 1.1 volts core LDO, and 2.8 volts TCXO LDO

Build-in reset controller

- Does not need external reset control IC

Internal real-time clock (RTC)

- 32.768 KHz ± 20 ppm crystal
- 1.1 volts RTC clock output
- Supports external pin to wake up

Backup mode

- A Force_On pin to ease backup mode application circuit

Serial interface

- 3 UARTs
- SPI, I2C
- GPIO interface (up to 16 pins)

NMEA

- NMEA 0183 standard V4.1 and backward compliance
- Supports 219 different data

Superior sensitivities

- Acq.: -148 dBm (cold) / -163 dBm (hot)
- Tracking: -165 dBm

Ultra-low power consumption**(GPS+GLONASS)**

- Acquisition: 37 mW
- Tracking: 27 mW
- AlwaysLocate™: 3.0 mW

hardware design

- 9 passive external components
- Single RF Front-End for Multi-GNSS frequency bands

1.8 General Descriptions

Figure 2 is the block diagram of MT2503D. Based on a multi-processor architecture, MT2503D integrates an ARM7EJ-S™ core, the main processor running high-level GSM protocol software as well as multimedia applications, single digital signal processor core, which manages the low-level MODEM and advanced audio functions, an embedded processor running Bluetooth baseband and link control protocol and the Bluetooth radio control.

MT2503D consists of the following subsystems:

- Microcontroller Unit (MCU) subsystem: Includes an ARM7EJ-S™ RISC processor and its accompanying memory management and interrupt handling logics
- Digital Signal Processor (DSP) subsystem: Includes a DSP and its accompanying memory, memory controller and interrupt controller
- MCU/DSP interface: Junction at which the MCU and the DSP exchange hardware and software information
- Microcontroller peripherals: Include all user interface modules and RF control interface modules
- Microcontroller coprocessors: Run computing-intensive processes in place of the microcontroller
- DSP peripherals: Hardware accelerators for GSM/GPRS channel codec
- Multimedia subsystem: Integrates several advanced accelerators to support multimedia applications
- Voice front-end: Data path for converting analog speech to and from digital speech
- Audio front-end: Data path for converting stereo audio from an audio source
- Baseband front-end: Data path for converting a digital signal to and from an analog signal from the RF modules
- Timing generator: Generates the control signals related to the TDMA frame timing
- Power, reset and clock subsystem: Manage the power, reset and clock distribution inside MT2503D.
- Bluetooth subsystem: Includes an embedded processor with embedded ROM/RAM system, baseband processor, and a high-performance radio block
- Power management unit: Self-contained power supply source which also controls the charging and system startup circuitry.

Details of the individual subsystems and blocks are described in the following chapters.

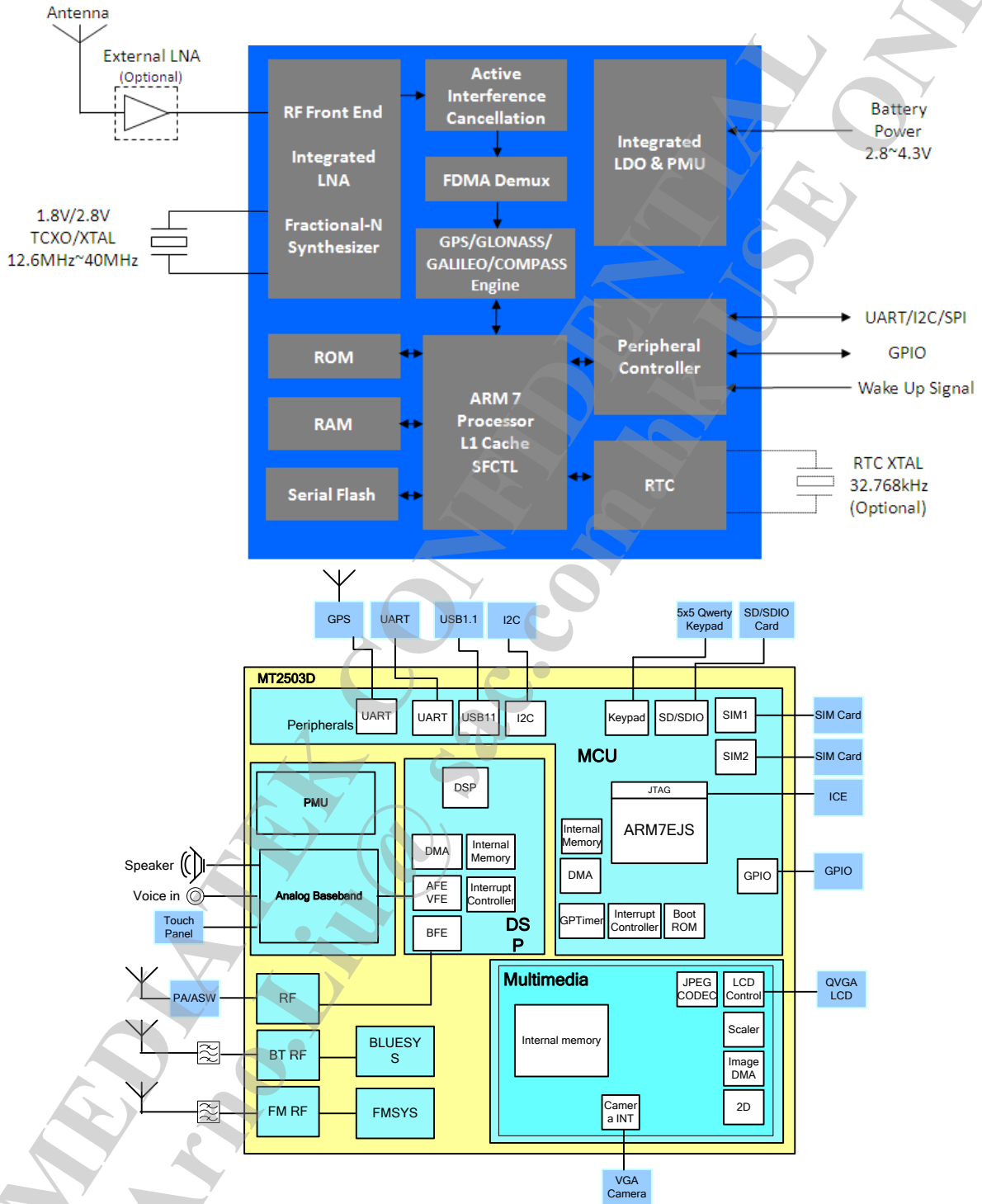


Figure 2. MT2503D block diagram

2 Product Descriptions

2.1 Pin Description

2.1.1 Ball Diagram

For MT2503D, an VFBGA 8.4mm*6.2mm, 215-ball, 0.4mm pitch package is offered. Pin-outs and the top view are illustrated in **Figure 3** for this package.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
A	RXLB	AVSS_2_GHF	SRCLKENAI		AVSS_2_GHF	BT_LNA		VIC28		BPL_BUS2		SCL28	CMRST		CMCLK	CMCSD0	GPS_XIN	GPS_XOUT	GPS_AVDD43_RTC	GPS_AVDD43_DCV	A	
B	RXHB	TX_LB	AVSS_2_GHF	XTAL1	XTAL2	AVSS_2_GHF	AVDD15_BTRF	BPL_BUS1		BPL_BUS0	VDDK	SDA28	GPIO_3	GPIO_2	CMCSD1	CMCK		GPS_DVS_S11_CORE	GPS_DCV_FB	GPS_AVS_S11_CLDO	B	
C		TX_HB	EINT		FREF			KCOL1	KROM1	KROM2			GPIO_0		CMPCDN	GPS_DVDD11_CORE	GPS_FDRCE_ON		GPS_DCV	GPS_AVS_S43_DCV	C	
D	VRF	AVSS_2_GHF	GPIO_10	GPIO_11	BKT_CLK_SEL		KROM0	KROM3	KCOL3		KCOL4	GPIO_4	GPIO_5	GPIO_1	GPIO_6		GPS_DVDD11_RTC	GPS_AVDD11_CLDO			D	
E	VBAT_VA	AVSS44_ALDO	VCAMA	TESTMODE	KCOL0	KCOL2	UTXD1	URXD1	KROM4	GPIO_9	GPIO_8	GPIO_7					GPS_DVDD28_SF	GPS_TX1	GPS_AVDD28_TLDO	GPS_AVDD28_CLDO	E	
F		FM_ANT_P		FMWKKEY	GND	GND							GPS_DVSS11_CORE	GPS_DVSS11_CORE	GPS_DVSS28_SF		GPS_JCK		GPS_AVDDTCKO_SW		F	
G	VREF	AVSS_FM	CHRLDO	BATON	GND	GND	GND	GND	GND	GND			GPS_DVSS11_CORE	GPS_DVSS11_CORE	GPS_DVSS11_CORE	GPS_DVSS11_CORE	GPS_TX2	GPS_RX1	GPS_AVDD43_VBAT	GPS_AVS_S43_MISC	G	
H	BATSNS	ISENSE	VCDT	KPLED	GND	GND	GND	GND	GND	GND			GPS_DVSS11_CORE	GPS_DVSS11_CORE	GPS_DVSS11_CORE	GPS_DVSS11_CORE	GPS_DVSS28_IO	GPS_DVD28_JC1	GPS_VREF		H	
J		DRV		ALXIN4	GND	GND	GND	GND	GND	GND	VSF		GPS_DVSS11_CORE	GPS_DVSS11_CORE			GPS_SC_K1	GPS_DVDD11_CORE2		GPS_AVDD18_CM	GPS_OEC	J
K		ISINK	AVSS44_PMU	AVSS44_PMU	GND		LSCK	MCCM0	MCDA1	MCCK							GPS_RX0	GPS_RX2		GPS_T1P	GPS_T1N	K
L	AVSS44_BOOST	AVSS44_BOOST	AU_HPL	ACCDDET	LSC_E_B	TESTMODE_D	RESETB	LPTE		MCDA0	MCDA3	VMC	VSIM2	GPS_JRS_T				GPS_JDI	GPS_DVD28_JC2	GPS_HRST_B		L
M	AVDD_SPK	AU_HPR	AVSS28_ABB	AU_VIN0_P	AU_VIN0_N		LSA0	LSDA			MCDA2		SIM2_SCLK	SIM2_S_RST			GPS_TX0	GPS_JRCK	GPS_SCS1	GPS_XTES_T	GPS_AVS_S_RF	M
N	SPK_OUTN	AVSS_SPK	AU_HSP	AU_HSN	AU_VIN1_N	AU_MIC_BIAS0	APC	AVSS44_DLDO	V18R	V18S	VIC18	VIC18	SIM2_SIO	SIM1_SIO	SIM1_S_RST	GPS_DTO	GPS_EINT0	GPS_EINT1	GPS_AVS_S_RF	GPS_RFIN	N	
P	SPK_OUTP	AVSS_SPK	VA		AU_VIN1_P		VRTC	VCORE		VBAT_DIGITAL		USB11_DP	USB11_DM		VSIM1	SIM1_SCLK	GPS_JMS		GPS_AVDD18_RXFE	GPS_AVS_S_RF	P	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		

Figure 3. Ball diagram and top view

2.1.2 Pin Coordination

Table 1. Pin coordinates

Pin#	Net name	Pin#	Net name	Pin#	Net name
A1	RXLB	E12	GPIO_7	K9	MCDA1
A2	AVSS_2GHF	E17	GPS_DVDD28_SF	K10	MCCK
A3	SRCLKENAI	E18	GPS_TX1	K16	GPS_RX0
A5	AVSS_2GHF	E19	GPS_AVDD28_TLDO	K17	GPS_RX2
A6	BT_LNA	E20	GPS_AVDD28_CLDO	K19	GPS_T1P

Pin#	Net name	Pin#	Net name	Pin#	Net name
A8	VIO28	F2	FM_ANT_P	K20	GPS_T1N
A10	BPI_BUS2	F4	PWRKEY	L1	AVSS44_BOOST
A12	SCL28	F5	GND	L2	AVSS44_BOOST
A13	CMRST	F6	GND	L3	AU_HPL
A15	CMMCLK	F13	GPS_DVSS11_CORE	L4	ACCDET
A16	CMCSD0	F14	GPS_DVSS11_CORE	L5	LSCE_B
A17	GPS_XIN	F15	GPS_DVSS28_SF	L6	TESTMODE_D
A18	GPS_XOUT	F17	GPS_JCK	L7	RESETB
A19	GPS_AVDD43_RTC	F19	GPS_AVDD_TCXO_SW	L8	LPTE
A20	GPS_AVDD43_DCV	G1	VREF	L10	MCDA0
B1	RXHB	G2	AVSS_FM	L11	MCDA3
B2	TX_LB	G3	CHRLDO	L12	VMC
B3	AVSS_2GHF	G4	BATON	L13	VSIM2
B4	XTAL1	G5	GND	L14	GPS_JRST_
B5	XTAL2	G6	GND	L17	GPS_JDI
B6	AVSS_2GHF	G7	GND	L18	GPS_DVDD28_IO2
B7	AVDD15_BTRF	G8	GND	L19	GPS_HRST_B
B8	BPI_BUS1	G9	GND	M1	AVDD_SPK
B10	BPI_BUS0	G10	GND	M2	AU_HPR
B11	VDDK	G13	GPS_DVSS11_CORE	M3	AVSS28_ABB
B12	SDA28	G14	GPS_DVSS11_CORE	M4	AU_VIN0_P
B13	GPIO_3	G15	GPS_DVSS11_CORE	M5	AU_VIN0_N
B14	GPIO_2	G16	GPS_DVSS11_CORE	M7	LSA0
B15	CMCSD1	G17	GPS_TX2	M8	LSDA
B16	CMCSK	G18	GPS_RX1	M11	MCDA2
B18	GPS_DVSS11_CORE	G19	GPS_AVDD43_VBAT	M13	SIM2_SCLK
B19	GPS_DCV_FB	G20	GPS_AVSS43_MISC	M14	SIM2_SRST
B20	GPS_AVSS11_CLDO	H1	BATSNS	M16	GPS_TX0
C2	TX_HB	H2	ISENSE	M17	GPS_JRCK
C3	EINT	H3	VCDT	M18	GPS_SCS1_
C5	FREF	H4	KPLED	M19	GPS_XTEST
C8	KCOL1	H5	GND	M20	GPS_AVSS_RF
C9	KROW1	H6	GND	N1	SPK_OUTN
C10	KROW2	H7	GND	N2	AVSS_SPK
C13	GPIO_0	H8	GND	N3	AU_HSP
C15	CMPDN	H9	GND	N4	AU_HSN
C16	GPS_DVDD11_CORE1	H10	GND	N5	AU_VIN1_N
C17	GPS_FORCE_ON	H11	GND	N6	AU_MICBIAS0

Pin#	Net name	Pin#	Net name	Pin#	Net name
C19	GPS_DCV	H13	GPS_DVSS11_CORE	N7	APC
C20	GPS_AVSS43_DCV	H14	GPS_DVSS11_CORE	N8	AVSS44_DLDO
D1	VRF	H15	GPS_DVSS11_CORE	N9	VIBR
D2	AVSS_2GHF	H16	GPS_DVSS11_CORE	N10	VUSB
D3	GPIO_10	H17	GPS_DVSS28_IO	N11	VIO18
D4	GPIO_11	H18	GPS_DVDD28_IO1	N12	VIO18
D5	EXT_CLK_SEL	H19	GPS_VREF	N13	SIM2_SIO
D7	KROW0	J2	DRV	N14	SIM1_SIO
D8	KROW3	J4	AUXIN4	N15	SIM1_SRST
D9	KCOL3	J5	GND	N16	GPS_JDO
D11	KCOL4	J6	GND	N17	GPS_EINT0
D12	GPIO_4	J7	GND	N18	GPS_EINT1
D13	GPIO_5	J8	GND	N19	GPS_AVSS_RF
D14	GPIO_1	J9	GND	N20	GPS_RFIN
D15	GPIO_6	J10	GND	P1	SPK_OUTP
D16	GPS_32K_OUT	J11	GND	P2	AVSS_SPK
D17	GPS_AVDD11_RTC	J12	VSF	P3	VA
D19	GPS_AVDD11_CLDO	J13	GPS_DVSS11_CORE	P5	AU_VIN1_P
E1	VBAT_VA	J14	GPS_DVSS11_CORE	P7	VRTC
E2	AVSS44_ALDO	J16	GPS_SCK1	P8	VCORE
E3	VCAMA	J17	GPS_DVDD11_CORE2	P10	VBAT_DIGITAL
E4	TESTMODE	J19	GPS_AVDD18_CM	P12	USB11_DP
E5	KCOL0	J20	GPS_OSC	P13	USB11_DM
E6	KCOL2	K2	ISINK	P15	VSIM1
E7	UTXD1	K3	AVSS44_PMU	P16	SIM1_SCLK
E8	URXD1	K4	AVSS44_PMU	P17	GPS_JMS
E9	KROW4	K5	GND	P19	GPS_AVDD18_RXFE
E10	GPIO_9	K7	LCK	P20	GPS_AVSS_RF
E11	GPIO_8	K8	MCCM0		

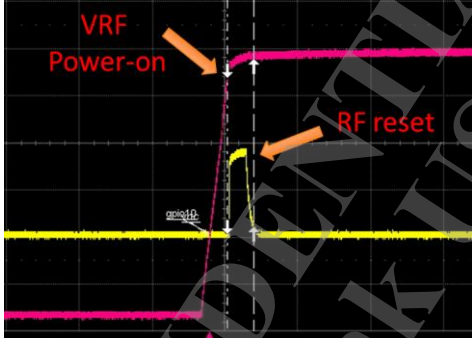
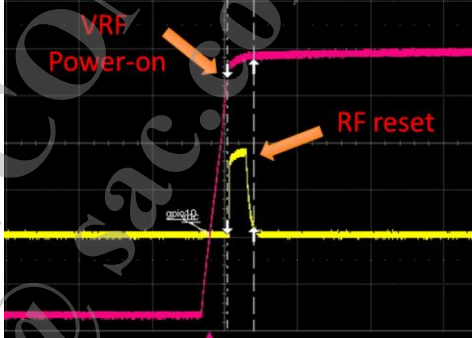
2.1.3 Detailed Pin Description

Table 2. Acronym for pin types

Abbreviation	Description
AI	Analog input
AO	Analog output
AIO	Analog bi-direction
DI	Digital input
DO	Digital output
DIO	Digital bi-direction
P	Power
G	Ground

Table 3. PIN function description and power domain

Pin name	Type	Description	Power domain
System			
RESETB	DIO	System reset	DVDD18_EMI
SRCLKENAI	DIO	26MHz clock request by external devices	VRF
EINT	DIO	External Interrupt	VRF
TESTMODE_D	DIO	Digital Test Mode	DVDD18_EMI
GPIO_0	DIO	General purpose input /output 0	DVDD28
GPIO_1	DIO	General purpose input /output 1	DVDD28
GPIO_2	DIO	General purpose input /output 2	DVDD28
GPIO_3	DIO	General purpose input /output 3	DVDD28
GPIO_4	DIO	General purpose input /output 4	DVDD28
GPIO_5	DIO	General purpose input /output 5	DVDD28
GPIO_6	DIO	General purpose input /output 6	DVDD28
GPIO_7	DIO	General purpose input /output 7	DVDD28
GPIO_8	DIO	General purpose input /output 8	DVDD28
GPIO_9	DIO	General purpose input /output 9	DVDD28

Pin name	Type	Description	Power domain
GPIO_10	DIO	<p>General purpose input /output 10, with analog output function, please be notified that this GPIO may temporarily output high signal after VRF power-on then output low once RF circuit reset done</p> 	VRF
GPIO_11	DIO	<p>General purpose input /output 11, with analog output function, please be notified that this GPIO temporarily output high signal after VRF power-on then output low once RF circuit reset done</p> 	VRF
RF control circuitro			
BPI_BUS0	DIO	RF hard-wire control bus bit 0	DVDD28
BPI_BUS1	DIO	RF hard-wire control bus bit 1	DVDD28
BPI_BUS2	DIO	RF hard-wire control bus bit 2	DVDD28
UART interface			
URXD1	DIO	UART1 receive data	DVDD28
UTXD1	DIO	UART1 transmit data	DVDD28
Keypad interface			
KCOL0	DIO	Keypad column 0	DVDD28
KCOL1	DIO	Keypad column 1	DVDD28
KCOL2	DIO	Keypad column 2	DVDD28
KCOL3	DIO	Keypad column 3	DVDD28
KCOL4	DIO	Keypad column 4	DVDD28
KROW0	DIO	Keypad row 0	DVDD28
KROW1	DIO	Keypad row 1	DVDD28
KROW2	DIO	Keypad row 2	DVDD28

Pin name	Type	Description	Power domain
KROW3	DIO	Keypad row 3	DVDD28
KROW4	DIO	Keypad row 4	DVDD28
Camera interface			
CMRST	DIO	CMOS sensor reset signal output	DVDD28
CMPDN	DIO	CMOS sensor power down control	DVDD28
CMCSD0	DIO	CMOS sensor data input 0	DVDD28
CMCSD1	DIO	CMOS sensor data input 1	DVDD28
CMMCLK	DIO	CMOS sensor pixel clock input	DVDD28
CMCSK	DIO	CMOS sensor pixel clock output	DVDD28
MS/SD card interface			
MCDA0	DIO	SD serial data IO 0/memory stick serial data IO	DVDD33_MSDC
MCDA1	DIO	SD serial data IO 1/memory stick serial data IO	DVDD33_MSDC
MCDA2	DIO	SD serial data IO 2/memory stick serial data IO	DVDD33_MSDC
MCDA3	DIO	SD serial data IO 3/memory stick serial data IO	DVDD33_MSDC
MCKK	DIO	SD serial clock/memory stick serial clock	DVDD33_MSDC
MCCM0	DIO	SD command output/memory stick bus state output	DVDD33_MSDC
SIM card interface			
SIM1_SIO	DIO	SIM1 data input/outputs	VSIM1
SIM1_SRST	DIO	SIM1 card reset output	VSIM1
SIM1_SCLK	DIO	SIM1 card clock output	VSIM1
SIM2_SIO	DIO	SIM2 data input/outputs	VSIM2
SIM2_SRST	DIO	SIM2 card reset output	VSIM2
SIM2_SCLK	DIO	SIM2 card clock output	VSIM2
I2C interface			
SCL28	DIO	I2C clock 2.8v power domain	DVDD28
SDA28	DIO	I2C data 2.8v power domain	DVDD28
LCD interface			
LSCE_B	DIO	Serial display interface chip select output	DVDD18_EMI
LSCK	DIO	Serial display interface clock	DVDD18_EMI
LSDA	DIO	Serial display interface data	DVDD18_EMI
LSA0	DIO	Serial display interface address	DVDD18_EMI
LPTE	DIO	Serial display tearing signal	DVDD18_EMI
FM			
FM_ANT_P	AI	FM input from antenna	VCAMA
Bluetooth			
BT_LNA	AIO	Bluetooth RF single-ended input	DVDD28
2G RF			
RXHB	AI	RF input for highband Rx (DCS/PCS)	VRF

Pin name	Type	Description	Power domain
RXLB	AI	RF input for lowband Rx (GSM900/GSM850)	VRF
TX_HB	AO	RF output for highband Tx (DCS/PCS)	VRF
TX_LB	AO	RF output pin for lowband Tx (GSM900/GSM850)	VRF
FREF	AO	DCXO reference clock output	VRF
XTAL1	AIO	Input 1 for DCXO crystal	VRF
XTAL2	AIO	Input 2 for DCXO crystal	VRF
EXT_CLK_SEL	AIO	DCXO mode selection	VRF
USB			
USB11_DM	AIO	D- data input/output	-
USB11_DP	AIO	D+ data input/output	-
GPS			
GPS_RFIN	RF signal	LNA RF Input pin	
GPS_XIN	Analog input	RTC 32KHz XTAL input	
GPS_XOUT	Analog output	RTC 32KHz XTAL output	
GPS_AVDD43_RTC	Analog power	RTC LDO input	
GPS_AVDD43_DCV	SMPS	SMPS input pin.	
GPS_DVSS11_CORE	Digital ground	Digital 1.1V core ground	
GPS_DCV_FB	SMPS	SMPS feedback pin	
GPS_AVSS11_CLDO	Analog ground	GND pin for core LDO	
GPS_DVDD11_CORE1	Digital power	Digital 1.1V core power input	
GPS_FORCE_ON	1.2V LVTTTL input PPU,PPD, SMT	Logic high to force power on this chip. Default: pull-up	
GPS_DCV	SMPS	SMPS output pin	
GPS_AVSS43_DCV	SMPS	SMPS GND pin	
GPS_32K_OUT	1.2V LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	RTC domain GPIO pin, can be programmed to 32KHz clock output or DR wake-up signal input Default: pull-down Default: 16mA driving	

Pin name	Type	Description	Power domain
GPS_AVDD11_RTC	Analog power	RTC LDO output	
GPS_AVDD11_CLDO	Analog power	Core LDO output pin	
GPS_DVDD28_SF	Digital power	Digital 2.8V serial flash power input	
GPS_TX1	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial output for UART 1 Default: pull-up Default: 8mA driving	
GPS_AVDD28_TLDO	Analog power	TCXO LDO output pin	
GPS_AVDD28_CLDO	Analog power	Core LDO input pin. Always powered by external source or SMPS	
GPS_DVSS28_SF	Digital ground	Digital 2.8V serial flash ground	
GPS_JCK	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial input for UART 1 Default: pull-up Default: 8mA driving	
GPS_AVDD_TCXO_SW	Analog power	TCXO power switch output pin	
GPS_TX2	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial output for UART 2 Default: pull-up Default: 8mA driving Strap pin tcxo_sw_sel 1'b0: AVDD_TCXO_SW output 1.8V 1'b1: AVDD_TCXO_SW output 2.8V	

Pin name	Type	Description	Power domain
GPS_RX1	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial input for UART 1 Default: pull-up Default: 8mA driving	
GPS_AVDD43_VBAT	Analog power	TCXO LDO input pin. Always be powered by external source. UVLO will detect this PIN to check power status.	
GPS_AVSS43_MISC	Analog ground	GND pin for buck controller, TCXO LDO and start-up block	
GPS_DVSS28_IO	Digital ground	Digital 1.8/2.8V IO ground	
GPS_DVDD28_IO1	Digital power	Digital 1.8/2.8V IO power input	
GPS_VREF	Analog	Bandgap output pin. Must add 1uF decoupling cap on EVB.	
GPS_SCK1	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	SPI clock output Default: pull-up Default: 8mA driving Strap pin clk_sel[0] Clk_sel[1:0] Mode 2'b00: XTAL mode 2'b01: External clock mode 2'b10: TCXO mode 2'b11: 16.368MHz TCXO mode	
GPS_DVDD11_CORE2	Digital power	Digital 1.1V core power input	
GPS_AVDD18_CM	RF power	1.8V supply for XTAL OSC, bandgap, thermal sensor and level shifter	
GPS_OSC	Analog signal	Input for crystal oscillator or TCXO	
GPS_RX0	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial input for UART 0 Default: pull-up Default: 8mA driving	

Pin name	Type	Description	Power domain
GPS_RX2	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial input for UART 2 Default: pull-up Default: 8mA driving	
GPS_T1P	Analog signal	RF testing signal	
GPS_T1N	Analog signal	RF testing signal	
GPS_JRST_	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	SPI slave selection 1 Default: pull-up Default: 8mA driving Strap pin clk_sel[1]	
GPS_JDI	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	JTAG interface data input. Default: pull-down Default: 8mA driving	
GPS_DVDD28_IO2	Digital power	Digital 1.8/2.8V IO power input	
GPS_HRST_B	2.8V LVTTTL input SMT	System reset. Active low Default: pull-up	

Pin name	Type	Description	Power domain
GPS_TX0	2.8V, LVTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial output for UART 0 Default: pull-up Default: 8mA driving	
GPS_JRCK	2.8V, LVTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	SPI clock output Default: pull-up Default: 8mA driving Strap pin clk_sel[0] Clk_sel[1:0] Mode 2'b00: XTAL mode 2'b01: External clock mode 2'b10: TCXO mode 2'b11: 16.368MHz TCXO mode	
GPS_SCS1_	2.8V, LVTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	SPI slave selection 1 Default: pull-up Default: 8mA driving Strap pin clk_sel[1]	
GPS_XTEST	2.8V LVTTL input SMT	Test mode. <i>Must keep low in normal mode.</i> Default: pull-down	
GPS_AVSS_RF	RF ground	RF ground pins	
GPS_JDO	2.8V, LVTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial output for UART 2 Default: pull-up Default: 8mA driving Strap pin tcxo_sw_sel 1'b0: AVDD_TCXO_SW output 1.8V 1'b1: AVDD_TCXO_SW output 2.8V	

Pin name	Type	Description	Power domain
GPS_EINT0	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	External interrupt 0 Default: pull-down Default: 8mA driving	
GPS_EINT1	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	External interrupt 1 Default: pull-down Default: 8mA driving	
GPS_AVSS_RF	RF ground	RF ground pins	
GPS_JMS	SMPS	SMPS GND pin	
GPS_AVDD18_RXFE	2.8V, LVTTTL I/O PPU, PPD, SMT 4mA, 8mA, 12mA, 16mA PDR	Serial output for UART 1 Default: pull-up Default: 8mA driving	
Analog baseband			
AU_HPR	AIO	Audio head phone output (R channel)	AVDD28_ABB
AU_HPL	AIO	Audio head phone output (L channel)	AVDD28_ABB
AU_HSP	AIO	Voice handset output (positive)	AVDD28_ABB
AU_HSN	AIO	Voice handset output (negative)	AVDD28_ABB
AU_VIN0_P	AIO	Microphone 0 input (positive)	AVDD28_ABB
AU_VIN0_N	AIO	Microphone 0 input (negative)	AVDD28_ABB
AU_VIN1_P	AIO	Microphone 1 input (positive)	AVDD28_ABB
AU_VIN1_N	AIO	Microphone 1 input (negative)	AVDD28_ABB
AUX_IN4	AIO	Auxiliary ADC input	AVDD28_ABB
SPK_OUTP	AIO	Speaker positive output	VBAT_SPK
SPK_OUTN	AIO	Speaker negative output	VBAT_SPK

Pin name	Type	Description	Power domain
APC	AIO	Automatic power control DAC output	AVDD28_ABB
AU_MICBIAS0	AIO	Microphone bias source 0	AVDD28_ABB
ACCDDET	AIO	Accessory detection	AVDD28_ABB
Power management unit			
VA	AIO	LDO output for ABB - VA	VBAT_ANALOG
VCAMA	AIO	LDO output for sensor – VCAMA	VBAT_VA
VIBR	AIO	LDO output for vibrator - VIBR	VBAT_DIGITAL
VIO18	AIO	LDO output for 1.8V power - VIO18	VBAT_DIGITAL
VIO28	AIO	LDO output for 2.8V power - VIO28	VBAT_DIGITAL
VMC	AIO	LDO output for memory card - VMC	VBAT_DIGITAL
VSF	AIO	LDO output - VSF	VBAT_DIGITAL
VRF	AIO	LDO output for GSMRF - VRF	VBAT_VA
VRTC	AIO	LDO output for RTC - VRTC	VBAT_DIGITAL
VSIM1	AIO	LDO output for 1 st SIM - VSIM	VBAT_DIGITAL
VSIM2	AIO	LDO output for 2 nd SIM - VSIM2	VBAT_DIGITAL
VUSB	AIO	LDO output for USB - VUSB	VBAT_DIGITAL
VCORE	AIO	LDO output for core circuit - Vcore	VBAT_DIGITAL
VREF	AIO	Band gap reference	BATSNS
VCDT	AIO	Charger-In level sense pin	BATSNS
DRV	AIO	IDAC current output open-drain pin	BATSNS
BATON	AIO	Battery Pack, NTC connected pin	BATSNS
ISENSE	AIO	Top node of current sensing 0.2ohm Rsense resistor	BATSNS
CHRLDO	AIO	2.8V shunt-regulator output	BATSNS
BATDET	AIO	Battery detection pin	BATSNS
ISINK0	AIO	Backlight driver channel 0	VBAT_VA
KPLED	AIO	Keypad led driver	VBAT_VA
TESTMODE	AIO	Test mode	BATSNS
PWRKEY	AIO	PWR key	BATSNS
AVDD25_V2P5	AIO	Reference voltage for ABT	-
Analog power			
AVDD15_BTRF	P	BTRF power input	-
VBAT_DIGITAL	P	Digital LDOs used battery voltage input	-
VBAT_VA	P	Analog LDOs used battery voltage input	-
AVDD_SPK	P	Input for loud speaker driver	-
BATSNS	P	Battery node of battery pack	-
AVDD15_BTRF	P	BTRF power input	-
Analog ground			
AVSS28_ABB	G	ABB 2.8V ground	-
AVSS_BT	G	BT ground	-
AVSS_2G	G	2G RF ground	-
AVSS_FM	G	FM ground	-

Pin name	Type	Description	Power domain
AVSS44_PMU	G	PMU ground	-
AVSS44_ALDO	G	ALDO ground	-
AVSS44_DLDO	G	DLDO ground	-
AVSS_SPK	G	SPK ground	-
AGND	G	GND for VREF	-
AVSS44_BOOST	G	Audio boost GND	-
Digital power			
VDDK	P	Core power	-
Digital ground			
GND	G	Ground	-

Table 4. Acronym for state of pins

Abbreviation	Description
I	Input
LO	Low output
HO	High output
LO	Low output
PU	Pull-up
PD	Pull-down
-	No PU/PD
0~N	Aux. function number
X	Delicate function pin

Table 5. State of pins

Name	Reset			Output drivability	Termination when not used	IO type
	State ¹	Aux ²	PU/PD ³			
System						
RESETB	HO	1	-	DIOH3/DIOL3	No need	IO Type 3
SRCLKENAI	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
EINT	I	0	PD	DIOH1/DIOL1	No need	IO Type 1

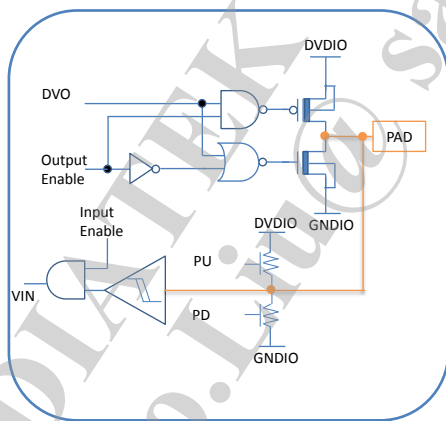
¹ The column "State" of "Reset" shows the pin state during reset. (Input, High Output, Low Output, etc)

² The column "Aux" for "Reset" means the default aux function number, shown in the table "Pin Multiplexing, Capability and Settings".

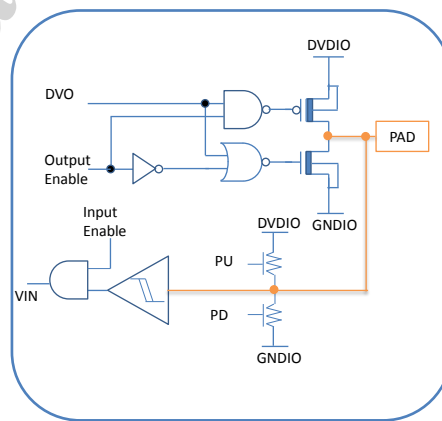
³ The column "PU/PD" for "Reset" means if there is internal pull-up or pull-down when the pin is input in the reset state.

Name	Reset			Output drivability	Termination when not used	IO type
	State ¹	Aux ²	PU/PD ³			
GPIO_0	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
GPIO_1	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
GPIO_2	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
GPIO_3	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
GPIO_4	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
GPIO_5	LO	0	PD	DIOH2/DIOL2	No need	IO Type 2
GPIO_6	LO	0	PD	DIOH2/DIOL2	No need	IO Type 2
GPIO_7	LO	0	PD	DIOH2/DIOL2	No need	IO Type 2
GPIO_8	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
GPIO_9	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
GPIO_10	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
GPIO_11	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
RF control circuitry						
BPI_BUS0	LO	1	PD	DIOH2/DIOL2	No need	IO Type 2
BPI_BUS1	I	1	PD	DIOH2/DIOL2	No need	IO Type 2
BPI_BUS2	I	1	PD	DIOH2/DIOL2	No need	IO Type 2
UART interface						
URXD1	I	1	PU	DIOH3/DIOL3	No need	IO Type 3
UTXD1	HO	1	PU	DIOH2/DIOL2	No need	IO Type 2
Keypad Interface						
KCOL0	I	0	PU	DIOH4/DIOL4	No need	IO Type 4
KCOL1	I	0	PU	DIOH4/DIOL4	No need	IO Type 4
KCOL2	I	0	PU	DIOH4/DIOL4	No need	IO Type 4
KCOL3	I	0	PU	DIOH4/DIOL4	No need	IO Type 4
KCOL4	I	0	PU	DIOH4/DIOL4	No need	IO Type 4
KROW0	I	0	PD	DIOH5/DIOL5	No need	IO Type 5
KROW1	I	0	PD	DIOH5/DIOL5	No need	IO Type 5
KROW2	I	0	PD	DIOH6/DIOL6	No need	IO Type 6
KROW3	I	0	PD	DIOH6/DIOL6	No need	IO Type 6
KROW4	I	0	PD	DIOH6/DIOL6	No need	IO Type 6
Camera interface						
CMRST	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
CMPDN	HO	0	-	DIOH3/DIOL3	No need	IO Type 3
CMCSD0	I	0	PU	DIOH3/DIOL3	No need	IO Type 3
CMCSD1	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
CMMCLK	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
CMCSK	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
MS/SD card interface						
MCDA0	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCDA1	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCDA2	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCDA3	I	0	PD	DIOH3/DIOL3	No need	IO Type 3

Name	Reset			Output drivability	Termination when not used	IO type
	State ¹	Aux ²	PU/PD ³			
MCKK	I	0	PU	DIOH3/DIOL3	No need	IO Type 3
MCCM0	I	0	PU	DIOH3/DIOL3	No need	IO Type 3
SIM card interface						
SIM1_SIO	I	1	PD	DIOH6/DIOL6	No need	IO Type 6
SIM1_SRST	I	1	PD	DIOH6/DIOL6	No need	IO Type 6
SIM1_SCLK	I	1	PD	DIOH6/DIOL6	No need	IO Type 6
SIM2_SIO	I	1	PD	DIOH6/DIOL6	No need	IO Type 6
SIM2_SRST	I	1	PD	DIOH6/DIOL6	No need	IO Type 6
SIM2_SCLK	I	1	PD	DIOH6/DIOL6	No need	IO Type 6
I2C interface						
SCL28	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
SDA28	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
LCD interface						
LSCE_B	HO	1	-	DIOH3/DIOL3	No need	IO Type 3
LSCK	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
LSDA	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
LSA0	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
LPTE	I	0	PD	DIOH3/DIOL3	No need	IO Type 3



IO type1



IO type2

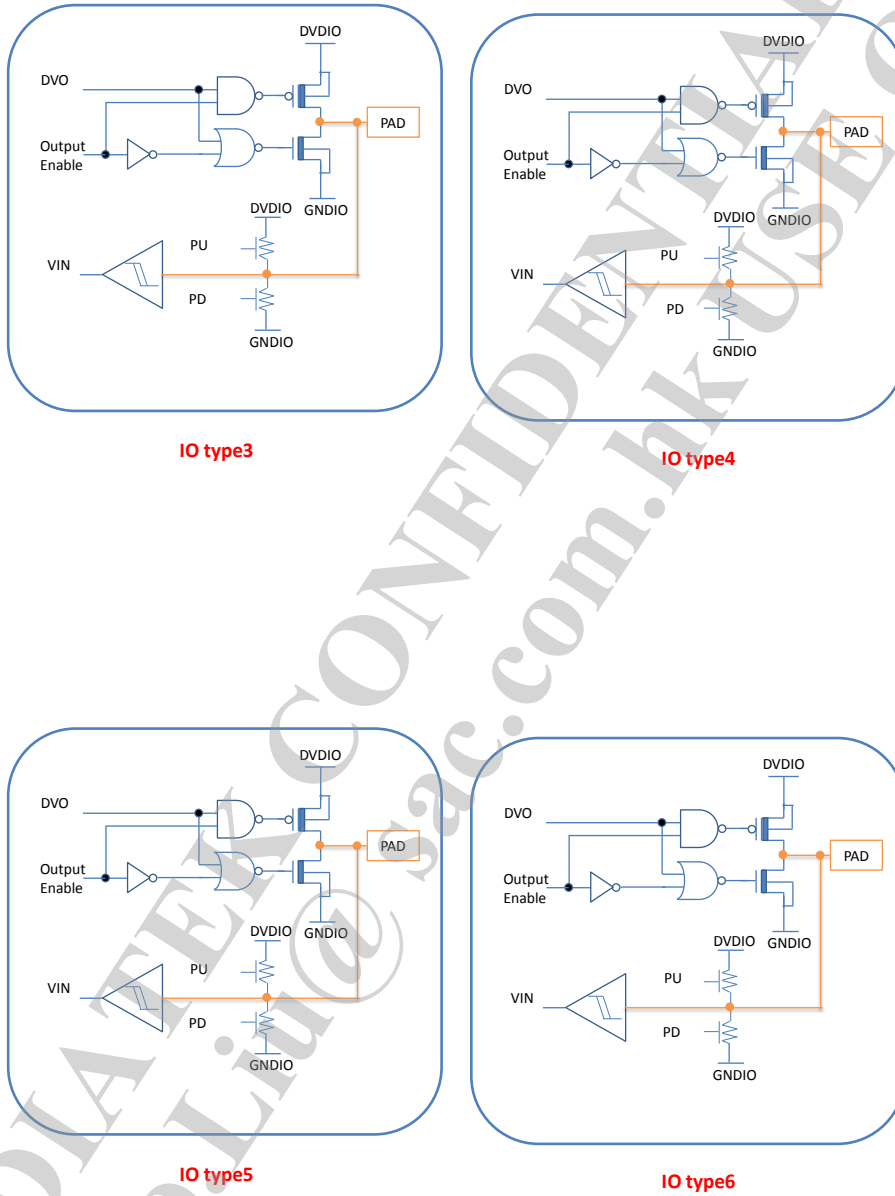


Figure 4. IO types in state of pins

2.1.4 Pin Multiplexing, Capability and Settings

Table 6. Acronym for pull-up and pull-down types

Abbreviation	Description
PU	Pull-up, not controllable
PD	Pull-down, not controllable
CU	Pull-up, controllable
CD	Pull-down, controllable
X	Cannot pull-up or pull-down

Table 7. Capability of PU/PD, driving and Schmitt trigger

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
GPIO_0	0	GPIO0	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT0	I	CU, CD	4, 8, 12, 16mA	0
	2	XP	AIO	-	4, 8, 12, 16mA	0
	3	U3RXD	I	CU, CD	4, 8, 12, 16mA	0
	4	CMCSD2	I	CU, CD	4, 8, 12, 16mA	0
	5	CMCSK	I	CU, CD	4, 8, 12, 16mA	0
	6	EDIDO	O	CU, CD	4, 8, 12, 16mA	0
	7	JTDI	I	PU	4, 8, 12, 16mA	0
	8	BTJTDI	I	CU, CD	4, 8, 12, 16mA	0
	9	FMJTDI	I	CU, CD	4, 8, 12, 16mA	0
GPIO_1	0	GPIO1	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT1	I	CU, CD	4, 8, 12, 16mA	0
	2	XM	AIO	-	4, 8, 12, 16mA	0
	3	U3TXD	O	CU, CD	4, 8, 12, 16mA	0
	4	U1CTS	I	CU, CD	4, 8, 12, 16mA	0
	5	CMMCLK	O	CU, CD	4, 8, 12, 16mA	0
	6	EDIDI	I	CU, CD	4, 8, 12, 16mA	0
	7	JTMS	I	PU	4, 8, 12, 16mA	0
	8	BTJTMS	I	CU, CD	4, 8, 12, 16mA	0
	9	FMJTMS	I	CU, CD	4, 8, 12, 16mA	0
GPIO_2	0	GPIO2	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT2	O	CU, CD	4, 8, 12, 16mA	0
	2	YP	AIO	-	4, 8, 12, 16mA	0
	3	GPSFSYNC	O	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/CU/CD	Driving	SMT
	4	PWM0	O	CU, CD	4, 8, 12, 16mA	0
	5	CMCSD0	I	CU, CD	4, 8, 12, 16mA	0
	6	EDIWS	O	CU, CD	4, 8, 12, 16mA	0
	7	JTRST_B	I	PD	4, 8, 12, 16mA	0
	8	BTJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
	9	FMJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
GPIO_3	0	GPIO3	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCINS	I	CU, CD	4, 8, 12, 16mA	0
	2	YM	AIO	-	4, 8, 12, 16mA	0
	4	PWM1	O	CU, CD	4, 8, 12, 16mA	0
	5	CMCSD1	I	CU, CD	4, 8, 12, 16mA	0
	6	EDICK	O	CU, CD	4, 8, 12, 16mA	0
	7	JTDO	O	CU, CD	4, 8, 12, 16mA	0
	8	BTJTDO	O	CU, CD	4, 8, 12, 16mA	0
	9	FMJTDO	O	CU, CD	4, 8, 12, 16mA	0
GPIO_4	0	GPIO4	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT3	I	CU, CD	4, 8, 12, 16mA	0
	4	U1RTS	O	CU, CD	4, 8, 12, 16mA	0
GPIO_5	0	GPIO5	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT4	I	CU, CD	4, 8, 12, 16mA	0
	3	BPI_BUS3	O	CU, CD	4, 8, 12, 16mA	0
GPIO_6	0	GPIO6	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT5	I	CU, CD	4, 8, 12, 16mA	0
	2	MCINS	I	CU, CD	4, 8, 12, 16mA	0
	3	BPI_BUS4	O	CU, CD	4, 8, 12, 16mA	0
GPIO_7	0	GPIO7	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT6	I	CU, CD	4, 8, 12, 16mA	0
	3	BPI_BUS5	O	CU, CD	4, 8, 12, 16mA	0
GPIO_8	0	GPIO8	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT7	I	CU, CD	4, 8, 12, 16mA	0
	2	SCL	IO	CU, CD	4, 8, 12, 16mA	0
GPIO_9	0	GPIO9	IO	CU, CD	4, 8, 12, 16mA	0
	1	EINT8	I	CU, CD	4, 8, 12, 16mA	0
	2	SDA	IO	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/CU/CD	Driving	SMT
URXD1	0	GPIO10	IO	CU, CD	4, 8, 12, 16mA	0
	1	U1RXD	I	PU	4, 8, 12, 16mA	0
	2	CMRST	O	CU, CD	4, 8, 12, 16mA	0
	3	EINT9	I	CU, CD	4, 8, 12, 16mA	0
	4	MCINS	I	CU, CD	4, 8, 12, 16mA	0
UTXD1	0	GPIO11	IO	CU, CD	4, 8, 12, 16mA	0
	1	U1TXD	O	CU, CD	4, 8, 12, 16mA	0
	2	CMPDN	O	CU, CD	4, 8, 12, 16mA	0
	3	EINT10	I	CU, CD	4, 8, 12, 16mA	0
KCOL4	0	GPIO12	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL4	IO	-	4, 8, 12, 16mA	0
	2	U2RXD	I	CU, CD	4, 8, 12, 16mA	0
	3	EDIDI	I	CU, CD	4, 8, 12, 16mA	0
	4	FMJTDI	I	CU, CD	4, 8, 12, 16mA	0
	5	JTDI	I	PU	4, 8, 12, 16mA	0
	6	BTJTDI	I	CU, CD	4, 8, 12, 16mA	0
KCOL3	0	GPIO13	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL3	IO	-	4, 8, 12, 16mA	0
	2	EINT11	I	CU, CD	4, 8, 12, 16mA	0
	3	PWM0	O	CU, CD	4, 8, 12, 16mA	0
	4	FMJTMS	I	CU, CD	4, 8, 12, 16mA	0
	5	JTMS	I	PU	4, 8, 12, 16mA	0
	6	BTJTMS	I	CU, CD	4, 8, 12, 16mA	0
KCOL2	0	GPIO14	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL2	IO	-	4, 8, 12, 16mA	0
	2	EINT12	I	CU, CD	4, 8, 12, 16mA	0
	3	U1RTS	I	CU, CD	4, 8, 12, 16mA	0
KCOL1	0	GPIO15	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL1	IO	-	4, 8, 12, 16mA	0
	2	GPSFSYNC	O	CU, CD	4, 8, 12, 16mA	0
	3	U1CTS	I	CU, CD	4, 8, 12, 16mA	0
	4	FMJTCK	I	CU, CD	4, 8, 12, 16mA	0
	5	JTCK	I	PU	4, 8, 12, 16mA	0
	6	BTJTCK	I	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/CU/CD	Driving	SMT
KCOL0	0	GPIO16	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL0	IO	-	4, 8, 12, 16mA	0
KROW4	0	GPIO17	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW4	IO	-	4, 8, 12, 16mA	0
	2	U2TXD	O	CU, CD	4, 8, 12, 16mA	0
	3	EDICK	O	CU, CD	4, 8, 12, 16mA	0
KROW3	0	GPIO18	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW3	IO	-	4, 8, 12, 16mA	0
	2	EINT13	I	CU, CD	4, 8, 12, 16mA	0
	3	CLK00	O	CU, CD	4, 8, 12, 16mA	0
	4	FMJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
	5	JTRST_B	I	PD	4, 8, 12, 16mA	0
KROW2	0	GPIO19	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW2	IO	-	4, 8, 12, 16mA	0
	2	PWM1	O	CU, CD	4, 8, 12, 16mA	0
	3	EDIWS	O	CU, CD	4, 8, 12, 16mA	0
	4	FMJTDO	O	CU, CD	4, 8, 12, 16mA	0
	5	JTDO	O	CU, CD	4, 8, 12, 16mA	0
KROW1	0	GPIO20	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW1	IO	-	4, 8, 12, 16mA	0
	2	EINT14	I	CU, CD	4, 8, 12, 16mA	0
	3	EDIDO	O	CU, CD	4, 8, 12, 16mA	0
	4	BTPRI	IO	CU, CD	4, 8, 12, 16mA	0
	5	JTRCK	O	CU, CD	4, 8, 12, 16mA	0
KROW0	0	GPIO21	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW0	IO	-	4, 8, 12, 16mA	0
	5	MCINS	I	CU, CD	4, 8, 12, 16mA	0
	6	BTDBGIN	I	CU, CD	4, 8, 12, 16mA	0
BPI_BUS2	0	GPIO22	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPI_BUS2	O	CU, CD	4, 8, 12, 16mA	0
BPI_BUS1	0	GPIO23	IO	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/CU/CD	Driving	SMT
	1	BPI_BUS1	O	CU, CD	4, 8, 12, 16mA	0
BPI_BUS0	0	GPIO24	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPI_BUS0	IO	CU, CD	4, 8, 12, 16mA	0
CMRST	0	GPIO25	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMRST	O	CU, CD	4, 8, 12, 16mA	0
	2	TESTMODE_D	O	CU, CD	4, 8, 12, 16mA	0
	3	CLKO1	O	CU, CD	4, 8, 12, 16mA	0
	4	EINT15	I	CU, CD	4, 8, 12, 16mA	0
	5	FMJTDI	I	CU, CD	4, 8, 12, 16mA	0
	6	JTDI	I	PU	4, 8, 12, 16mA	0
CMPDN	0	GPIO26	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMPDN	O	CU, CD	4, 8, 12, 16mA	0
	2	LSCK1	O	CU, CD	4, 8, 12, 16mA	0
	3	DAICLK	O	CU, CD	4, 8, 12, 16mA	0
	4	SPICS	IO	CU, CD	4, 8, 12, 16mA	0
	5	FMJTMS	I	CU, CD	4, 8, 12, 16mA	0
	6	JTMS	I	PU	4, 8, 12, 16mA	0
CMCSD0	0	GPIO27	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMCSD0	I	CU, CD	4, 8, 12, 16mA	0
	2	LSCE_B1	O	CU, CD	4, 8, 12, 16mA	0
	3	DAIPCMIN	I	CU, CD	4, 8, 12, 16mA	0
	4	SPISCK	IO	CU, CD	4, 8, 12, 16mA	0
	5	FMJTCK	I	CU, CD	4, 8, 12, 16mA	0
	6	JTCK	I	PU	4, 8, 12, 16mA	0
	8	MC2CM0	O	-	4, 8, 12, 16mA	0
CMCSD1	0	GPIO28	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMCSD1	I	CU, CD	4, 8, 12, 16mA	0
	2	LSDA1	IO	CU, CD	4, 8, 12, 16mA	0
	3	DAIPCMOUT	O	CU, CD	4, 8, 12, 16mA	0
	4	SPIMOSI	IO	CU, CD	4, 8, 12, 16mA	0
	5	FMJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
	6	JTRST_B	I	PD	4, 8, 12, 16mA	0
	8	MC2CK	O	-	4, 8, 12, 16mA	0
CMMCLK	0	GPIO29	IO	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
	1	CMMCLK	O	CU, CD	4, 8, 12, 16mA	0
	2	LSA0DA1	O	CU, CD	4, 8, 12, 16mA	0
	3	DAISYNC	O	CU, CD	4, 8, 12, 16mA	0
	4	SPIMISO	IO	CU, CD	4, 8, 12, 16mA	0
	5	FMJTDO	O	CU, CD	4, 8, 12, 16mA	0
	6	JTDO	O	CU, CD	4, 8, 12, 16mA	0
	8	MC2DA0	IO	-	4, 8, 12, 16mA	0
CMCSK	0	GPIO30	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMCSK	I	CU, CD	4, 8, 12, 16mA	0
	2	LPTE	I	CU, CD	4, 8, 12, 16mA	0
	3	CMCSD2	I	CU, CD	4, 8, 12, 16mA	0
	4	EINT16	I	CU, CD	4, 8, 12, 16mA	0
	6	JTRCK	O	CU, CD	4, 8, 12, 16mA	0
MCKK	0	GPIO31	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCKK	O	-	4, 8, 12, 16mA	0
	4	U2RXD	I	CU, CD	4, 8, 12, 16mA	0
MCCM0	0	GPIO32	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCCM0	IO	-	4, 8, 12, 16mA	0
	4	U2TXD	O	CU, CD	4, 8, 12, 16mA	0
MCDA0	0	GPIO33	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA0	IO	-	4, 8, 12, 16mA	0
	4	DAISYNC	O	CU, CD	4, 8, 12, 16mA	0
MCDA1	0	GPIO34	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA1	IO	-	4, 8, 12, 16mA	0
	2	EINT17	I	CU, CD	4, 8, 12, 16mA	0
	4	DAIPCMIN	I	CU, CD	4, 8, 12, 16mA	0
MCDA2	0	GPIO35	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA2	IO	-	4, 8, 12, 16mA	0
	2	EINT18	I	CU, CD	4, 8, 12, 16mA	0
	4	DAICK	O	CU, CD	4, 8, 12, 16mA	0
MCDA3	0	GPIO36	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA3	IO	-	4, 8, 12, 16mA	0
	2	EINT19	I	CU, CD	4, 8, 12, 16mA	0
	3	CLKO2	O	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/CU/CD	Driving	SMT
	4	DAIPCMOUT	O	CU, CD	4, 8, 12, 16mA	0
SIM1_SIO	0	GPIO37	IO	CU, CD	2, 4, 6, 8mA	0
	1	SIM1_SIO	IO	-	2, 4, 6, 8mA	0
SIM1_SRST	0	GPIO38	IO	CU, CD	2, 4, 6, 8mA	0
	1	SIM1_SRST	IO	-	2, 4, 6, 8mA	0
SIM1_SCLK	0	GPIO39	IO	CU, CD	2, 4, 6, 8mA	0
	1	SIM1_SCLK	IO	-	2, 4, 6, 8mA	0
SIM2_SIO	0	GPIO40	IO	CU, CD	2, 4, 6, 8mA	0
	1	SIM2_SIO	IO	-	2, 4, 6, 8mA	0
	3	U2RTS	O	CU, CD	2, 4, 6, 8mA	0
SIM2_SRST	0	GPIO41	IO	CU, CD	2, 4, 6, 8mA	0
	1	SIM2_SRST	IO	-	2, 4, 6, 8mA	0
	2	CLKO3	O	CU, CD	2, 4, 6, 8mA	0
	3	U2CTS	I	CU, CD	2, 4, 6, 8mA	0
	4	SCL18	IO	CU, CD	4, 8, 12, 16mA	0
SIM2_SCLK	0	GPIO42	IO	CU, CD	2, 4, 6, 8mA	0
	1	SIM2_SCLK	IO	-	2, 4, 6, 8mA	0
	2	LSCE1_B1	O	CU, CD	2, 4, 6, 8mA	0
	4	SDA18	IO	CU, CD	4, 8, 12, 16mA	0
SCL	0	GPIO43	IO	CU, CD	4, 8, 12, 16mA	0
	1	SCL	IO	CU, CD	4, 8, 12, 16mA	0
SDA	0	GPIO44	IO	CU, CD	4, 8, 12, 16mA	0
	1	SDA	IO	CU, CD	4, 8, 12, 16mA	0
TESTMODE_D	0	GPIO45	IO	CU, CD	4, 8, 12, 16mA	0
	1	TESTMODE_D	O	CU, CD	4, 8, 12, 16mA	0
	3	CMRST	O	CU, CD	4, 8, 12, 16mA	0
LSCE_B0	0	GPIO46	IO	CU, CD	4, 8, 12, 16mA	0
	1	LSCE_B0	O	CU, CD	4, 8, 12, 16mA	0
	2	EINT20	I	CU, CD	4, 8, 12, 16mA	0
	3	CMCSD0	I	CU, CD	4, 8, 12, 16mA	0
	4	CLKO4	O	CU, CD	4, 8, 12, 16mA	0
LSCK0	0	GPIO47	IO	CU, CD	4, 8, 12, 16mA	0
	1	LSCK0	O	CU, CD	4, 8, 12, 16mA	0
	3	CMPDN	O	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
LSDA0	0	GPIO48	IO	CU, CD	4, 8, 12, 16mA	0
	1	LSDA0	IO	-	4, 8, 12, 16mA	0
	2	EINT21	I	CU, CD	4, 8, 12, 16mA	0
	3	CMCSD1	I	CU, CD	4, 8, 12, 16mA	0
LSA0	4	WIFITOBT	I	CU, CD	4, 8, 12, 16mA	0
	0	GPIO49	IO	CU, CD	4, 8, 12, 16mA	0
	1	LSA0DA0	O	-	4, 8, 12, 16mA	0
	2	LSCE1_B0	O	CU, CD	4, 8, 12, 16mA	0
LPTE	3	CMMCLK	O	CU, CD	4, 8, 12, 16mA	0
	0	GPIO50	IO	CU, CD	4, 8, 12, 16mA	0
	1	LPTE	I	CU, CD	4, 8, 12, 16mA	0
	2	EINT22	I	CU, CD	4, 8, 12, 16mA	0
	3	CMCSK	I	CU, CD	4, 8, 12, 16mA	0
	4	CMCSD2	I	CU, CD	4, 8, 12, 16mA	0
	6	MCINS	I	CU, CD	4, 8, 12, 16mA	0
RESETB	9	CLKO5	O	CU, CD	4, 8, 12, 16mA	0
	0	GPIO51	IO	CU, CD	4, 8, 12, 16mA	0
EINT	1	RESETB	IO	CU, CD	4, 8, 12, 16mA	0
	0	AGPI52	I	CU, CD	8mA	0
SRCLKENAI	2	EINT23	I	CU, CD	8mA	0
	0	AGPI53	I	CU, CD	8mA	0
	1	SRCLKENAI	I	CU, CD	8mA	0
GPIO_10	2	EINT24	I	-	8mA	0
	0	AGPIO54	IO	CU, CD	8mA	0
GPIO_11	0	AGPIO55	IO	CU, CD	8mA	0

2.2 Electrical Characteristics

2.2.1 Absolute Maximum Ratings

Table 8. Absolute maximum ratings for power supply

Symbol or pin name	Description	Min.	Max.	Unit
VBAT_DIGITAL	Digital used battery voltage input	-0.3	+4.4	V
VBAT_VA	Analog used battery voltage input	-0.3	+4.4	V

Symbol or pin name	Description	Min.	Max.	Unit
AVDD_SPK	VBAT input for loud speaker driver	-0.3	+5.5	V
VDDK	1.3v core power	-0.3	+1.43	V

Table 9. Absolute maximum ratings for voltage input

Symbol or pin name	Description	Min.	Max.	Unit
VIN1	Digital input voltage for IO Type 1	-0.3	3.08	V
VIN2	Digital input voltage for IO Type 2	-0.3	3.08	V
VIN3	Digital input voltage for IO Type 3	-0.3	3.63	V
VIN4	Digital input voltage for IO Type 4	-0.3	3.08	V
VIN5	Digital input voltage for IO Type 5	-0.3	3.08	V
VIN6	Digital input voltage for IO Type 6	-0.3	3.08	V
VIN7	Digital input voltage for IO Type 7	-0.3	3.63	V

Table 10. Absolute maximum ratings for storage temperature

Symbol or pin name	Description	Min.	Max.	Unit
Tstg	Storage temperature	-55	125	°C

2.2.2 Recommended Operating Conditions

Table 11. Recommended operating conditions for power supply

Symbol or pin name	Description	Min.	Typ.	Max.	Unit
VBAT_DIGITAL	Digital used battery voltage input	3.4	3.8	4.2	V
VBAT_VA	Analog used battery voltage input	3.4	3.8	4.2	V
AVDD_SPK	VBAT input for loud speaker driver	3.4	3.8	4.2	V
VDDK	1.2v core power	1.17	1.3	1.43	V

Table 12. Recommended operating conditions for voltage input

Symbol or pin name	Description	Min.	Typ.	Max.	Unit
VIN1	Digital input voltage for IO Type 1	-0.3	-	DVDIO+0.3	V
VIN2	Digital input voltage for IO Type 2	-0.3	-	DVDIO+0.3	V
VIN3	Digital input voltage for IO Type 3	-0.3	-	DVDIO+0.3	V
VIN4	Digital input voltage for IO Type 4	-0.3	-	DVDIO+0.3	V
VIN5	Digital input voltage for IO Type 5	-0.3	-	DVDIO+0.3	V
VIN6	Digital input voltage for IO Type 6	-0.3	-	DVDIO+0.3	V
VIN7	Digital input voltage for IO Type 7	-0.3	-	DVDIO+0.3	V

Table 13. Recommended operating conditions for operating temperature

Symbol or pin name	Description	Min.	Typ	Max.	Unit
Tc	Operating temperature	-20	-	85	°C

2.2.3 Electrical Characteristics under Recommended Operating Conditions

Table 14. Electrical characteristics

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
DIIH1	Digital high input current for IO Type 1	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	6.1	-	82.5	
DIIL1	Digital low input current for IO Type 1	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN1 < 0.7	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN1 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN1 < 0.7	-12.5	-	22.5	
DIOH1	Digital high output current for IO Type 1	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
DIOL1	Digital low output current for IO Type 1	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
DRPU1	Digital I/O pull-up resistance for IO Type 1	DVDIO = 2.8V	40	85	190	kΩ
DRPD1	Digital I/O pull-down resistance for IO Type 1	DVDIO = 2.8V	40	85	190	kΩ
DVOH1	Digital output high voltage for IO Type 1	DVDIO = 2.8V	2.38			V
DVOL1	Digital output low voltage for IO Type 1	DVDIO = 2.8V			0.42	V
DIIH2	Digital high input current for IO Type 2	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	-22.5	-	12.5	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		PD enabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	6.1	-	82.5	μA
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN1 < 2.1	-5	-	5	
		PU enabled, DVDIO = 1.8V, 1.35 < VIN1 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN1 < 2.1	-0.8	-	35	
DIIL2	Digital low input current for IO Type 2	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN2 < 0.7	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN2 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN2 < 0.7	-12.5	-	22.5	
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN1 < 0.45	-5	-	5	μA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN1 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN1 < 0.45	-9.3	-	11.4	
DIOH2	Digital high output current for IO Type 2	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL2	Digital low output current for IO Type 2	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU2	Digital I/O pull-up resistance for IO Type 2	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPD2	Digital I/O pull-down resistance for IO Type 2	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DVOH2	Digital output high voltage for IO Type 2	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL2	Digital output low voltage for IO Type 2	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V

Symbol	Description	Condition	Min.	Typ.	Max.	Unit		
DIIH3	Digital high input current for IO Type 3	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN3 < 3.1	-5	-	5	μA		
		PU enabled, DVDIO = 2.8V, 2.1 < VIN3 < 3.1	-22.5	-	12.5			
		PD enabled, DVDIO = 2.8V, 2.1 < VIN3 < 3.1	6.1	-	82.5			
		DIIH3	Digital high input current for IO Type 3	PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN3 < 2.1	-5	-	5	μA
				PU enabled, DVDIO = 1.8V, 1.35 < VIN3 < 2.1	-11.4	-	9.3	
				PD enabled, DVDIO = 1.8V, 1.35 < VIN3 < 2.1	-0.8	-	35	
DIIL3	Digital low input current for IO Type 3	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN3 < 0.7	-5	-	5	μA		
		PU enabled, DVDIO = 2.8V, -0.3 < VIN3 < 0.7	-82.5	-	-6.1			
		PD enabled, DVDIO = 2.8V, -0.3 < VIN3 < 0.7	-12.5	-	22.5			
		DIIL3	Digital low input current for IO Type 3	PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN3 < 0.45	-5	-	5	μA
				PU enabled, DVDIO = 1.8V, -0.3 < VIN3 < 0.45	-35	-	0.8	
				PD enabled, DVDIO = 1.8V, -0.3 < VIN3 < 0.45	-9.3	-	11.4	
DIOH3	Digital high output current for IO Type 3	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA		
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA		
DIOL3	Digital low output current for IO Type 3	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA		
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA		
DRPU3	Digital I/O pull-up resistance for IO Type 3	DVDIO = 2.8V	10	47	100	kΩ		
		DVDIO = 1.8V	10	47	100	kΩ		
DRPD3	Digital I/O pull-down	DVDIO = 1.8V	10	47	100	kΩ		

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	resistance for IO Type 3	DVDIO = 2.8V	10	47	100	kΩ
DVOH3	Digital output high voltage for IO Type 3	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL3	Digital output low voltage for IO Type 3	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V
DIIH4	Digital high input current for IO Type 4	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN4 < 3.1	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN4 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN4 < 3.1	6.1	-	82.5	
DIIL4	Digital low input current for IO Type 4	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN4 < 0.7	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN4 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN4 < 0.7	-12.5	-	22.5	
DIOH4	Digital high output current for IO Type 4	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
DIOL4	Digital low output current for IO Type 4	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
DRPU4	Digital I/O pull-up resistance for IO Type 4 (GPIO mode)	DVDIO = 2.8V	15	36	55	kΩ
DRPD4	Digital I/O pull-down resistance for IO Type 4 (GPIO mode)	DVDIO = 2.8V	15	36	55	kΩ
DRPU4 1200K	Digital I/O pull-up resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	1200	-	-	kΩ
DRPD4 1200K	Digital I/O pull-down resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	1200	-	-	kΩ
DVOH4	Digital output high voltage for IO Type 4	DVDIO = 2.8V	2.38			V
DVOL4	Digital output low voltage for IO Type 4	DVDIO = 2.8V			0.42	V
DIIH5	Digital high input current for IO Type 5	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN5 < 3.1	-5	-	5	μA

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		PU enabled, DVDIO = 2.8V, 2.1 < VIN5 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN5 < 3.1	6.1	-	82.5	
DIIL5	Digital low input current for IO Type 5	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN5 < 0.7	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN5 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN5 < 0.7	-12.5	-	22.5	
DIOH5	Digital high output current for IO Type 5	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
DIOL5	Digital low output current for IO Type 5	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
DRPU5	Digital I/O pull-up resistance for IO Type 5 (GPIO mode)	DVDIO = 2.8V	15	36	55	kΩ
DRPD5	Digital I/O pull-down resistance for IO Type 5 (GPIO mode)	DVDIO = 2.8V	15	36	55	kΩ
DRPU5 1K	Digital I/O pull-up resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	1	-	-	kΩ
DRPD5 1K	Digital I/O pull-down resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	1	-	-	kΩ
DVOH5	Digital output high voltage for IO Type 5	DVDIO = 2.8V	2.38			V
DVOL5	Digital output low voltage for IO Type 5	DVDIO = 2.8V			0.42	V
DIIH6	Digital high input current for IO Type 6	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN6 < 3.1	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN6 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN6 < 3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN6 < 2.1	-5	-	5	μA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN6 < 2.1	-11.4	-	9.3	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit		
		PD enabled, DVDIO = 1.8V, 1.35 < VIN6 < 2.1	-0.8	-	35			
DIIL6	Digital low input current for IO Type 6	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN6 < 0.7	-5	-	5	μA		
		PU enabled, DVDIO = 2.8V, -0.3 < VIN6 < 0.7	-82.5	-	-6.1			
		PD enabled, DVDIO = 2.8V, -0.3 < VIN6 < 0.7	-12.5	-	22.5			
				PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN6 < 0.45	-5	-	5	μA
				PU enabled, DVDIO = 1.8V, -0.3 < VIN6 < 0.45	-35	-	0.8	
				PD enabled, DVDIO = 1.8V, -0.3 < VIN6 < 0.45	-9.3	-	11.4	
DIOH6	Digital high output current for IO Type 6	DVOH > 2.38V, DVDIO = 2.8V	-8	-	-	mA		
		DVOH > 1.53V, DVDIO = 1.8V	-6	-	-	mA		
DIOL6	Digital low output current for IO Type 6	DVOL < 0.42V, DVDIO = 2.8V	-	-	8	mA		
		DVOL < 0.27V, DVDIO = 1.8V	-	-	6	mA		
DRPU6	Digital I/O pull-up resistance for IO Type 6	DVDIO = 2.8V	40	85	190	kΩ		
		DVDIO = 1.8V	70	150	320	kΩ		
DRPD6	Digital I/O pull-down resistance for IO Type 6	DVDIO = 2.8V	40	85	190	kΩ		
		DVDIO = 1.8V	70	150	320	kΩ		
DVOH6	Digital output high voltage for IO Type 6	DVDIO = 2.8V	2.38			V		
		DVDIO = 1.8V	1.53			V		
DVOL6	Digital output low voltage for IO Type 6	DVDIO = 2.8V			0.42	V		
		DVDIO = 1.8V			0.27	V		
DIIH7	Digital high input current for IO Type 7	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN7 < 3.1	-5	-	5	μA		
		PU enabled, DVDIO = 2.8V, 2.1 < VIN7 < 3.1	-22.5	-	12.5			
		PD enabled, DVDIO = 2.8V, 2.1 < VIN7 < 3.1	6.1	-	82.5			

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN7 < 2.1	-5	-	5	μA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN7 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN7 < 2.1	-0.8	-	35	
DIIL7	Digital low input current for IO Type 7	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN7 < 0.7	-5	-	5	μA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN7 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN7 < 0.7	-12.5	-	22.5	
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN7 < 0.45	-5	-	5	μA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN7 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN7 < 0.45	-5	-	5	
DIOH7	Digital high output current for IO Type 7	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL7	Digital low output current for IO Type 7	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU7	Digital I/O pull-up resistance for IO Type 7	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPD7	Digital I/O pull-down resistance for IO Type 7	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DVOH7	Digital output high voltage for IO Type 7	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL7	Digital output low voltage for IO Type 7	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V

2.3 System Configuration

2.3.1 Strapping Resistors

Table 15. Strapping table

Pin name	Description	Trapping condition
LSA0	Pull-up with 10K resistor(Default internal pull-down with 47K resistor)	Power-on reset
BPI_BUS1	Pull-up with 10K resistor (Default internal pull-down with 75K resistor)	Power-on reset
BPI_BUS2	Pull-up with 10K resistor (Default internal pull-down with 75K resistor)	Power-on reset

2.3.2 Mode Selection

Table 16. Mode selection of chip

Pin name	Description
EXT_CLK_SEL	GND: Uses DCXO as 26M clock source VRF: Uses external clock as 26M clock source
LSA0	GND: Uses 1.8V serial flash device DVDD18_EMI: Uses 3.3V serial flash device
KCOL0	GND: Boots ROM to enter USB download mode DVDD28: Normal boot-up mode
{BPI_BUS1,BPI_BUS2}	{GND, GND}: No JTAG {GND, DVDD28}: JTAG at keypad pins {DVDD28, GND}: JTAG at GPIO pins {DVDD28, DVDD28}: JTAG at camera pins

2.4 Power-on Sequence and Protection Logic

MT2503D provides 32K crystal removal feature. The XOSC32_ENB state tells if MT2503D provides this feature or not. VRF will be turned on at the same time with VRTC when XOSC32_ENB = 1. The power-on/off sequence controlled by “Control” and “Reset Generator” is shown as the figure below.

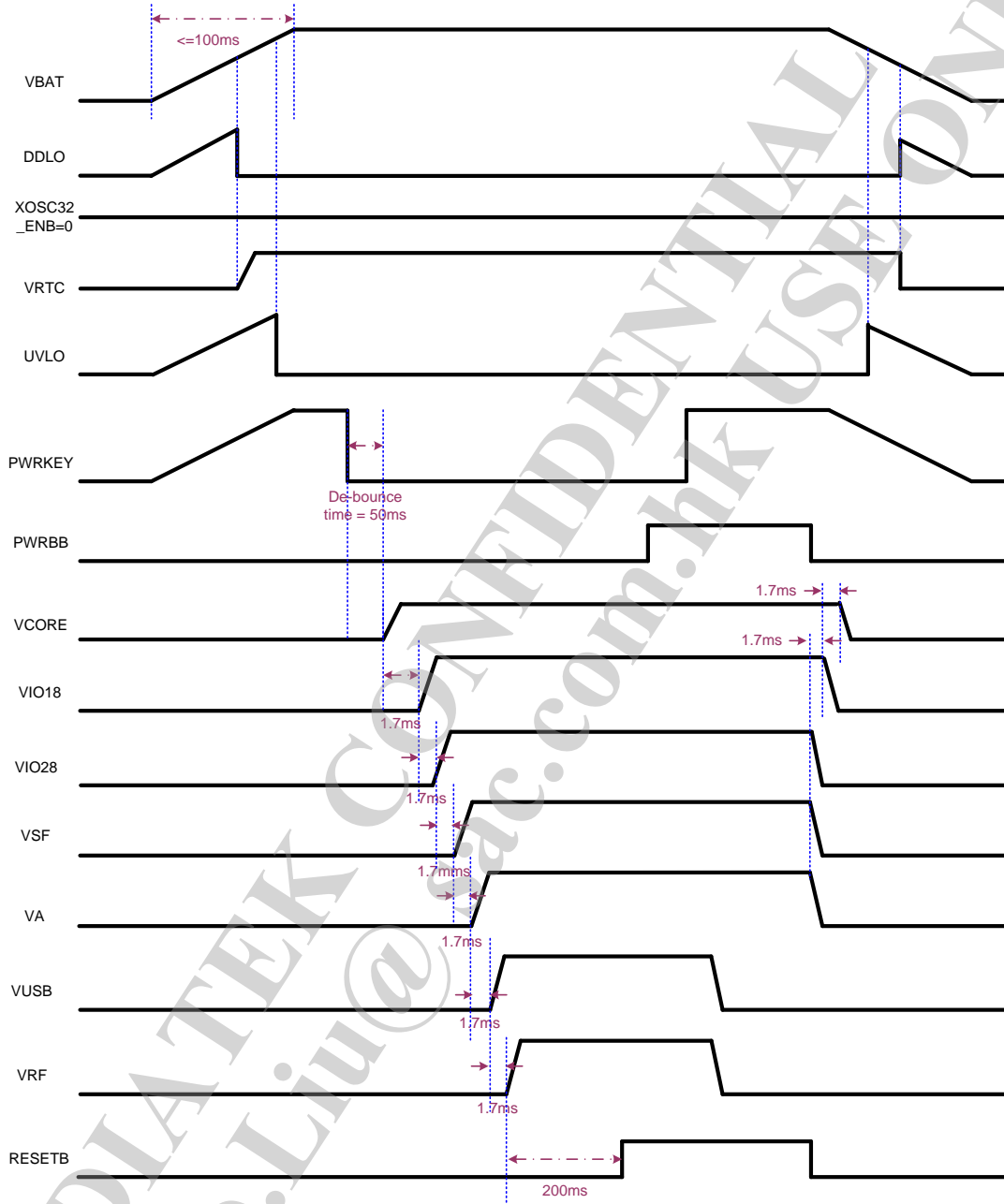


Figure 5. Power-on/off control sequence by pressing PWRKEY and XOSC32_ENB = 0

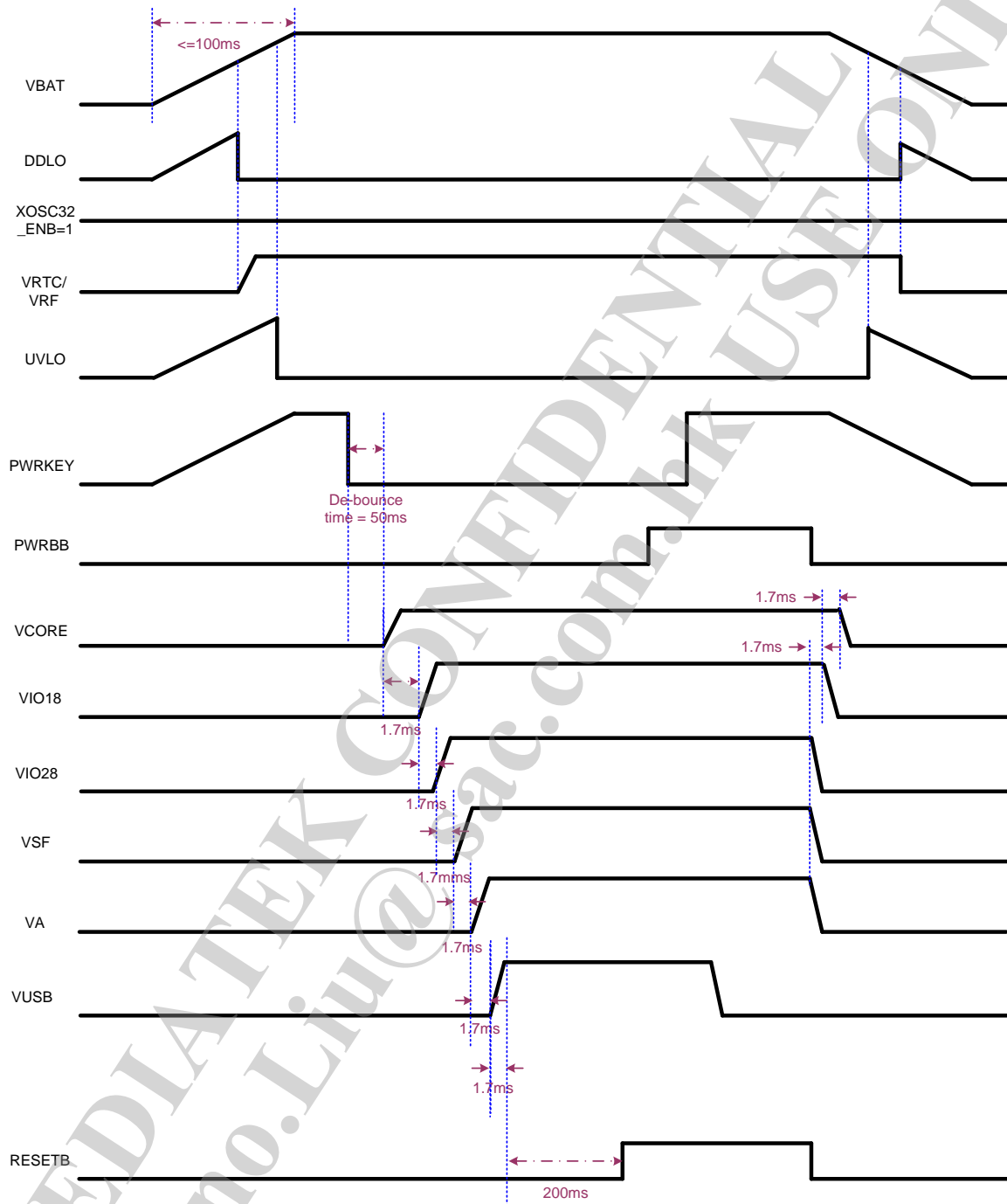


Figure 6. Power-on/off control sequence by pressing PWRKEY and XOSC32_ENB = 1

Note that each of the above figures only shows one power-on/off condition when XOSC32_ENB = 0 or XOSC32_ENB = 1. MT2503D handles the power-on and off of the handset. The following three methods can switch on the handset (when leaving UVLO): XOSC32_ENB = 0

1. Push PWRKEY (Pull the PWRKEY pin to the low level.)

Pulling PWRKEY low is the typical way to turn on the handset. The turn-on sequence is VCORE → VIO18 → VIO28 → VSF → VA → VUSB → VRF.

The supplies for the baseband are ready, and the system reset ends at the moment when the above LDOs are fully turned on to ensure correct timing and function. After that, the baseband will send the PWRBB signal back to the PMU for acknowledgement. To successfully power on the handset, PWRKEY should be kept low until PMU receives PWRBB from the baseband.

2. RTC module generates PWRBB to wake up the system.

If the RTC module is scheduled to wake up the handset at a certain time, the PWRBB signal will be directly sent to the PMU. In this case, PWRBB will become high at specific moment and allow the PMU to be powered on as the sequence described above. This is called the RTC alarm.

3. Valid charger plug-in (CHRIN voltage is within the valid range.)

The charger plug-in will also turn on the handset if the charger is valid (no OVP takes place). However, if the battery voltage is too low to power on the handset (UVLO state), the system will not be turned on by any of the three methods. In this case, the charger will charge the battery first and the handset will be powered on automatically as long as the battery voltage is high enough.

Under-voltage lockout (UVLO)

The UVLO state in the PMU prevents startup if the initial voltage of the main battery is below the 3.2V threshold. It ensures that the handset is powered on with the battery in good condition. The UVLO function is performed by a hysteretic comparator which ensures a smooth power-on sequence. In addition, when the battery voltage is getting lower, it will enter the UVLO state, and the PMU will be turned off by itself, except for VRTC LDO, to prevent further discharging. Once the PMU enters the UVLO state, it will draw low quiescent current. The RTC LDO will still be working until the DDLO disables it.

Deep discharge lockout (DDLO)

The PMU will enter the deep discharge lockout (DDLO) state when the battery voltage drops below 2.5V. In this state, the VRTC LDO will be shut down. Otherwise, it will draw very low quiescent current to prevent further discharging or damage to the cells.

Reset

The PMU contains a reset control circuit which takes effect at both power-up and power-down. The RESETB pin is held low in the beginning of power-up and returns to high after the pre-determined delay time. The delay time is controlled by a large counter which uses the clock from internal ring-oscillator. At power-off, the RESETB pin will return to low immediately without any delay.

Over-temperature protection

If the die temperature of PMU exceeds 150°C, the PMU will automatically disable all the LDOs except for VRTC. Once the over-temperature state is resolved, a new power-on sequence will be required to enable the LDOs.

2.5 Analog Baseband

To communicate with analog blocks, a common control interface for all analog blocks is implemented. In addition, there are some dedicated interfaces for data transfer. The common control interface translates the APB bus write and read cycle for specific addresses related to analog front-end control. During the writing or reading of any of these control registers, there is a latency associated with the transfer of data to or from the analog front-end. Dedicated data interface of each analog block is implemented in the corresponding digital block. An analog block includes the following analog functions for the complete GSM/GPRS baseband signal processing:

1. RF control: DAC for automatic power control (APC) is included, and its output is provided to external RF power amplifier respectively.
2. Auxiliary ADC: Provides an ADC for the battery and other auxiliary analog functions monitoring
3. Audio mixed-signal block: Provides complete analog voice signal processing including microphone amplification, A/D conversion, D/A conversion, earphone driver, etc. Dedicated stereo D/A conversion and amplification for audio signals are also included.
4. Clock generation: Includes a clock squarer for shaping the system clock, and PLL providing clock signals to DSP, MCU and USB unit

2.5.1 APC-DAC

2.5.1.1 Block Description

APC-DAC is a 10-bit DAC with output buffer aiming at automatic power control. See the tables below for its analog pin assignment and functional specifications. It is an event-driven scheme for power saving purpose.

2.5.1.2 Functional Specifications

Table 17. APC-DAC specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
N	Resolution		10		Bit
FS	Sampling rate			1.0833	MSPS

Symbol	Parameter	Min.	Typ.	Max.	Unit
SINAD	Signal to noise and distortion ratio (10-kHz sine with 1.0V swing & 100-kHz BW)	47			dB
	99% settling time (full swing on maximal capacitance)			5	μS
	Output swing	0		AVDD	V
	Output capacitance		200	2200	pF
	Output resistance	0.47	10		KΩ
DNL	Differential nonlinearity for code 20 to 970		± 1		LSB
INL	Integral nonlinearity for code 20 to 970		± 1		LSB
DVDD	Digital power supply	1.1	1.2	1.3	V
AVDD	Analog power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	°C
	Current consumption				
	Power-up		400		μA
	Power-down		1		μA

2.5.2 Auxiliary ADC

2.5.2.1 Block Description

The auxiliary ADC includes the following functional blocks:

1. Analog multiplexer: Selects signal from one of the seven auxiliary input pins. Real-world messages to be monitored, e.g. temperature, should be transferred to the voltage domain.
2. 10-bit A/D converter: Converts the multiplexed input signal to 10-bit digital data.

Channel	Application	Input range [V]
0	BATSNS	3.2 ~ 4.2
1	ISENSE	3.2 ~ 4.2
2	VCDT	Decided by application circuit
3	BATON	0 ~ AVDD28
4	AUXIN4	0 ~ AVDD28
others	Internal use	N/A

2.5.2.2 Functional Specifications

The functional specifications of the auxiliary ADC are listed in the following table.

Table 18. Functional specifications of auxiliary ADC

Symbol	Parameter	Min.	Typ.	Max.	Unit
--------	-----------	------	------	------	------

Symbol	Parameter	Min.	Typ.	Max.	Unit
N	Resolution		10		Bit
FC	Clock rate		1.08		MHz
FS	Sampling rate @ N-Bit		1.08/(N+1)		MSPS
	Input swing	0		AVDD	V
CIN	Input capacitance			50	fF
	Unselected channel			4	pF
RIN	Input resistance				MΩ
	Unselected channel	400			MΩ
	Selected channel	1			MΩ
	Clock latency		N+1		1/FC
DNL	Differential nonlinearity		± 1		LSB
INL	Integral nonlinearity		± 1		LSB
OE	Offset error		± 10		mV
FSE	Full swing error		± 10		mV
SINAD	Signal to noise and distortion ratio (10-kHz full swing input & 1.0833-MHz clock rate)		50		dB
DVDD	Digital power supply	1.1	1.2	1.3	V
AVDD	Analog power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	°C
	Current consumption				
	Power-up		280		μA
	Power-down		1		μA

2.5.3 Audio Mixed-Signal Blocks

2.5.3.1 Block Description

Audio mixed-signal blocks (AMB) integrate complete voice uplink/downlink and audio playback functions. As shown in the figure below, it includes three parts. The first consists of stereo audio DACs and audio amplifiers for audio playback. The second part is the voice downlink path, including voice-band DACs (left channel audio DAC) and voice amplifier, which produces voice signals to earphones or other auxiliary output devices. Moreover, a ClassK amplifier is embedded to support continuous >1W output power with an on-chip boost. The last part is the voice uplink path, which is the interface between the microphone (or other auxiliary input device) input and MT2503D DSP. A set of bias voltage is provided for the external electric microphone.

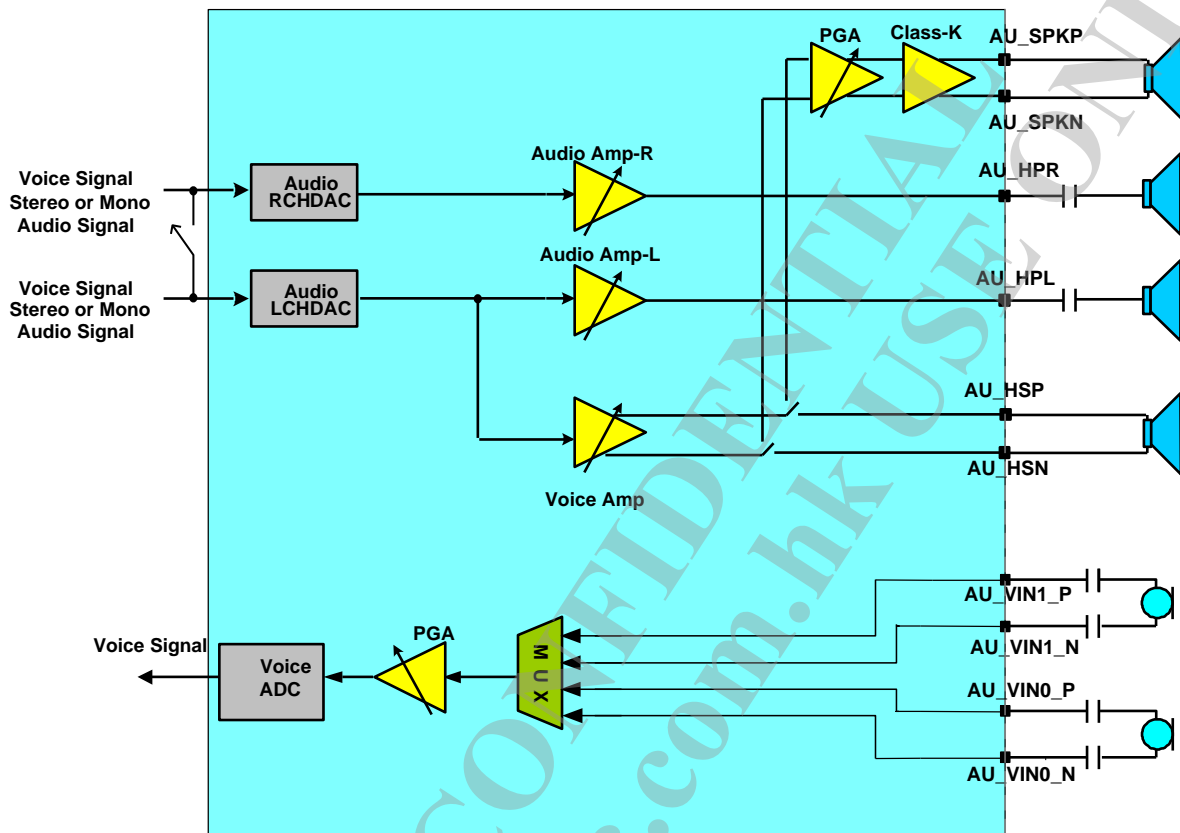


Figure 7. Block diagram of audio mixed-signal blocks

2.5.3.2 Functional Specifications

See the table below for the functional specifications of voice-band uplink/downlink blocks.

Table 19. Functional specifications of analog voice blocks

Symbol	Parameter	Min.	Typ.	Max.	Unit
FS	Sampling rate		6,500		kHz
DVDD	Digital power supply	1.1	1.2	1.3	V
AVDD	Analog power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	°C
VMIC	Microphone biasing voltage		1.9	2.2	V
IMIC	Current draw from microphone bias pins			2	mA
Uplink path⁴					
IDC	Current consumption for one channel		1.5		mA
SINAD	Signal to noise and distortion ratio Input level: -40 dbm0	29			dB

⁴ For uplink-path, not all gain settings of VUPG meet the specifications listed in the table, especially for several the lowest gains. The minimum gain that meets the specifications is to be determined.

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Input level: 0 dbm0		69		dB
RIN	Input impedance (differential)	13	20	27	K Ω
ICN	Idle channel noise			-67	dBm0
Downlink path					
IDC	Current consumption		4		mA
SINAD	Signal to noise and distortion ratio	29			dB
	Input level: -40 dBm0			69	
	Input level: 0 dBm0				
RLOAD	Output resistor load (differential)	16	32		Ω
CLOAD	Output capacitor load			250	pF
ICN	Idle channel noise of transmit path			-64	dBPa
XT	Crosstalk level on transmit path			-66	dBm0

See the table below for the functional specifications of audio blocks.

Table 20. Functional specifications of analog audio blocks

Symbol	Parameter	Min.	Typ.	Max.	Unit
FCK	Clock frequency		6.5		MHz
Fs	Sampling rate	32	44.1	48	kHz
AVDD	Power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	$^{\circ}$ C
IDC	Current consumption		4		mA
PSNR	Peak signal to noise ratio		88		dB
DR	Dynamic range		88		dB
VOUT	Output swing for 0dBFS input level @ -1dB headphone gain		0.707		Vrms
VOUT _{MAX}	Maximum output swing		2.0		Vpp
THD	Total harmonic distortion 10mW at 64 Ω load			-70	dB
RLOAD	Output resistor load (single-ended)	64			Ω
CLOAD	Output capacitor load			250	pF
XT	L-R channel cross talk	70			dB

2.6 Power Management Unit Blocks

The power management unit (PMU) manages the power supply of the entire chip, such as baseband, processor, memory, SIM cards, camera, vibrator, etc. The digital part of PMU is integrated into the analog part (see the figure below). PMU includes the following analog functions for signal processing:

- LDO: Regulates battery voltage to lower voltage level
- BOOST: Boosts battery voltage to target voltage for Class-AB audio amplifier
- Keypad LED driver (KPLED) and current sink (ISINK) switches: Sink current for keypad LED and LCM module

- Start-up (STRUP): Generates power-on/off control sequence of start-up circuits
- Pulse charger (PCHR): Controls battery charging

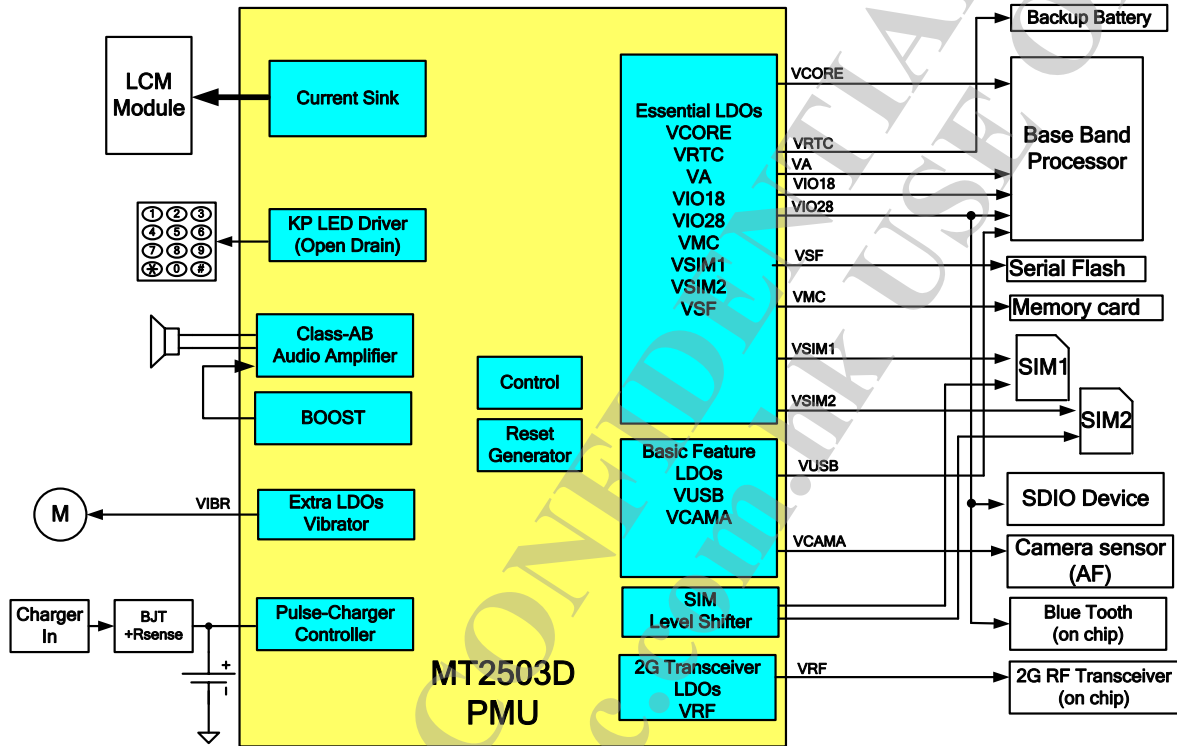


Figure 8. PMU system block diagram

2.6.1 LDO

PMU integrates 13 general low dropout regulators (LDO) optimized for their given functions by balancing the quiescent current, dropout voltage, line/load regulation, ripple rejection and output noise.

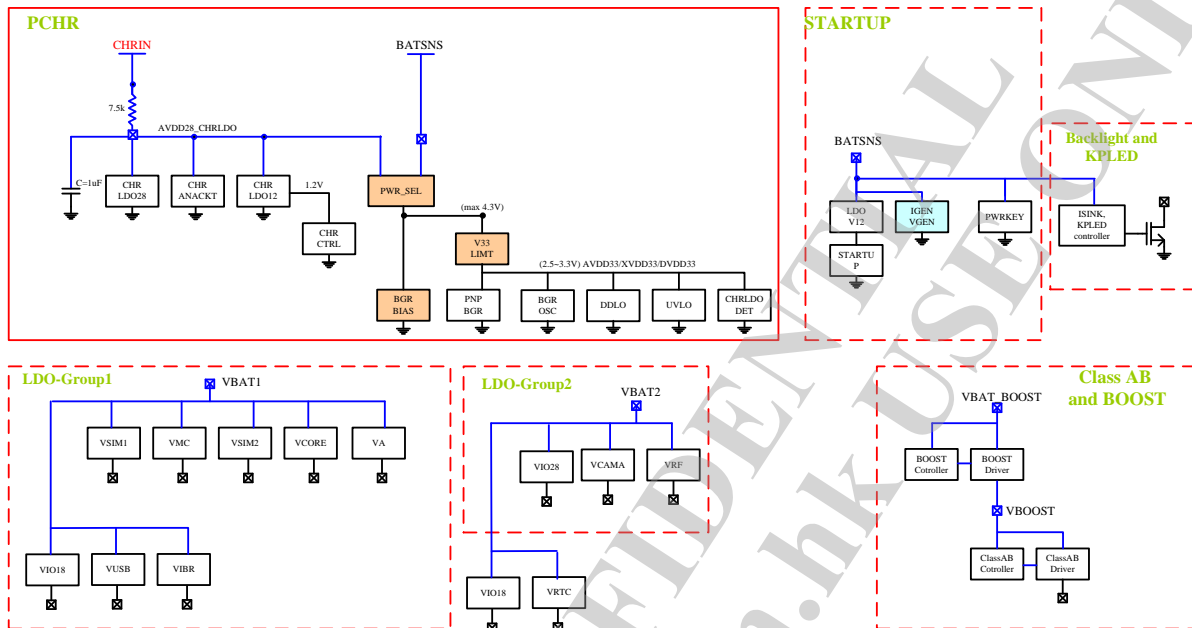


Figure 9. Power domain

2.6.1.1 LDO

A low-dropout regulator (LDO) is capable of maintaining its specified output voltage over a wide range of load current and input voltage, down to a very small difference between input and output voltages.

There are several features in the design of LDO, including discharge control, soft start and current limit. Before LDO is enabled, the output pin of LDO should be discharged first to avoid voltage accumulation on the capacitance. The soft-start limits inrush current and controls output-voltage rising time during the power-up. The current limit is the current protection to limit the LDO's output current and power dissipation.

There are three types of LDOs in PMU of MT2503D PMU. The analog LDO is optimized for low-frequency ripple rejection in order to reject the ripples coming from the burst of RF power amplifier. The digital IO LDO is a linear regulator optimized for very low quiescent current. The single-step RTC LDO is a linear regulator that can charge up a capacitor-type backup coin cell, which also supplies the RTC module even at the absence of the main battery. The single-step LDO features reverse current protection and is optimized for ultra-low quiescent current while sustaining the RTC function as long as possible.

2.6.1.1.1 Block Description

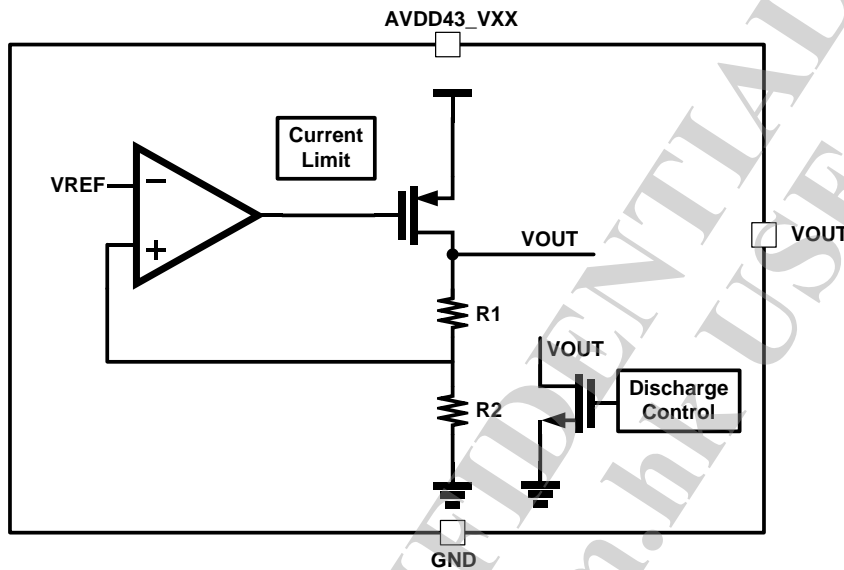


Figure 10. LDO block diagram

2.6.1.1.2 LDO Types

Table 21. LDO types and brief specifications

Type	LDO name	Vout (Volt)	I _{max} (mA)	Description
ALDO	VRF	2.8	150	RF chip and 26MHz reference clock
ALDO	VA	2.8	150	Analog baseband
ALDO	VCAMA	2.8	70	Camera sensor
DLDO	VIO28	2.8	100	Digital IO and Blue tooth
DLDO	VSIM1	1.8/3.0	30	SIM card
DLDO	VSIM2	1.8/3.0	30	SIM card
DLDO	VUSB	3.3	50	USB
DLDO	VIO18	1.8	100	Digital IO
DLDO	VCORE	0.75~1.35	150	Digital baseband
DLDO	VIBR	1.8/2.8/3.0	100	Vibrator
DLDO	VMC	1.8/2.8/3.0/3.3	100	Memory card
DLDO	VSF	1.86/2.8/3.0/3.3	50	Serial flash
RTCLDO	VRTC	2.8/3.3	2	Real-time clock

2.6.1.1.3 Functional Specifications
Table 22. Analog LDO specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Load capacitor			1		μF
	Current limit		1.2*I _{max}		5*I _{max}	mA
	V _{out}	Includes load regulation, regulation, and temperature coefficient	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Transient response	Slew: 15mA/us	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Temperature coefficient				100	ppm/C
	PSRR	I _{out} < 0.5*I _{max} 10 < f < 3 kHz	65			dB
		I _{out} < 0.5*I _{max} 3K < f < 30 kHz	45			dB
	Output noise	With A-weighted filter			90	uV _{rms}
	Quiescent current	I _{out} = 0		55		μA
	Turn-on overshoot	I _{out} = 0			Max. (+10%, +0.1V)	V
	Turn-on settling time	I _{out} = 0			240	μsec

Table 23. Digital LDO specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Load capacitor			1 ⁵		μF
	Current limit		1.2*I _{max}		5*I _{max}	mA
	V _{out}	Includes load regulation, regulation, and temperature coefficient	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Transient response	Slew: 15mA/us	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Temperature coefficient				100	ppm/C
	Quiescent current	I _{out} = 0		30		μA

⁵ VCORE loading capacitor typical value is 2.2uF. Other LDOs are 1uF.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Turn-on overshoot	I _{out} = 0			Max. (+10%, +0.1V)	V
	Turn-on settling time	I _{out} = 0			240	μs

Table 24. RTC LDO specification

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Load capacitor			1		μF
	V _{out}	Includes load regulation, line regulation, and temperature coefficient	2	2.8	3	V
	Temperature coefficient				100	ppm/C
	Quiescent current	I _{out} = 0		15		μA

2.6.2 BOOST

2.6.2.1 Functional Specifications

Table 25. RTC LDO specification.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	C _{in} C _{out}			2.2μF 4.7μF		μF
	L	R _{dcr,max} <80mOhm		0.68		μH
	V _{out}			5.3		V
	Ripple	V _{in} =3.4V/3.8V/4.2V, C _{in} =2.2μF & C _{out} =4.7μF, L= 0.68μH (R _{dcr,max} <80mOhm) 650mA , switching Freq 2MHz			100	mV
	Switching frequency			2		MHz
	Quiescent current	I _{out} = 0		4	6	mA

2.6.3 ISINK and KPLED Switches

One built-in open-drain output switch drives the keypad LED (KPLED) in the handset. The switch is controlled by the baseband with enabling registers. The switch of keypad LED can sink as much as

60mA current, and the output is high impedance when disabled. The value of the sink current decides the brightness of the LED.

The current controlled open drain drivers are also implemented to drive the LCM backlight module, and provides current from 4mA to 96mA.

2.6.3.1 Block Description

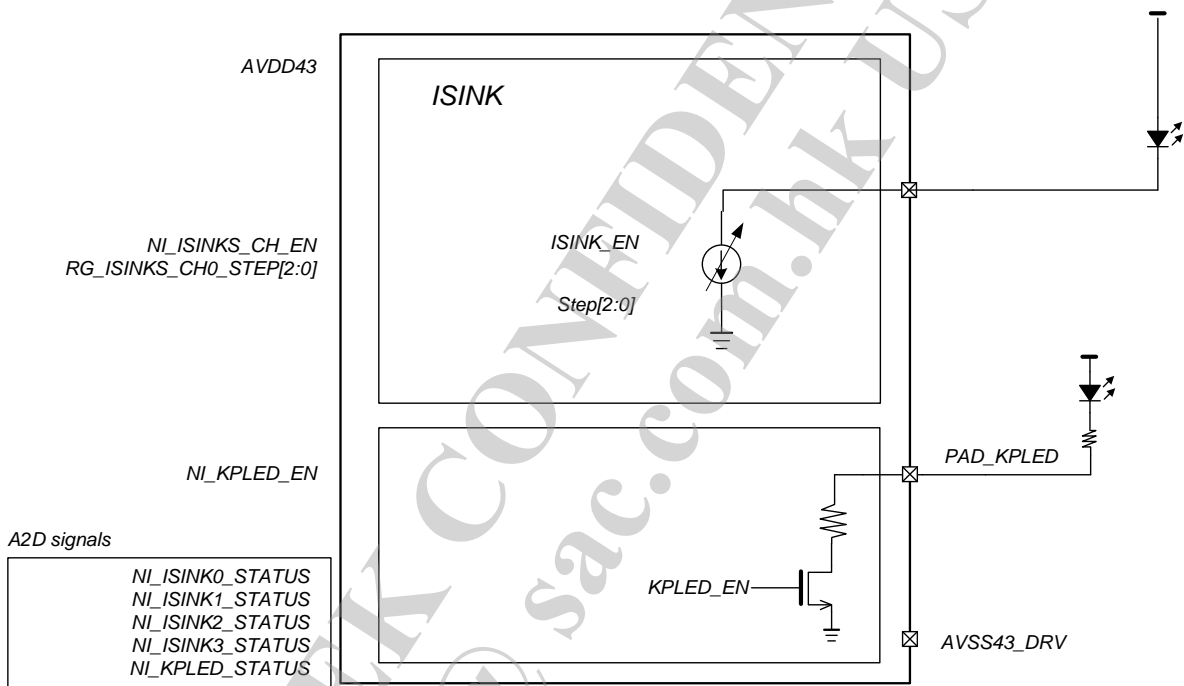


Figure 11. ISINKs and KPLED switches bock diagram.

2.6.3.2 Functional Specifications

Table 26. ISINKs and KPLED Switches Specification.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Sink current of keypad LED driver	Von > 0.5V, 100% dimming duty	60			mA
	1 ch Sink current of ISINK without current double option	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 000		4		mA
	1 ch Sink current of ISINK without current double option	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 001		8		mA
	1 ch Sink current of	Von > 0.15V, 100%		12		mA

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	ISINK without current double option	dimming duty, ISINKS_CHx_STEP = 010				
	1 ch Sink current of ISINK without current double option	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 011		16		mA
	1 ch Sink current of ISINK without current double option	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 100		20		mA
	1 ch Sink current of ISINK without current double option	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 101		24		mA
	Current mismatch	Von > 0.15V, 100% dimming duty	-5		5	%

2.6.4 STRUP

PMU handles the power-on and off of the handset. If the battery voltage is neither in the UVLO state (VBAT ≥ 3.4V) nor in the thermal condition, there are three methods to power on the handset system: pulling PWRKEY low (the user pushes PWRKEY), pulling PWRBB high (baseband BB_WakeUp) or valid charger plug-in.

According to different battery voltage (VBAT) and phone states, control signals and regulators will have different responses.

2.6.5 PCHR

The charger controller senses the charger input voltage from either a standard AC-DC adaptor or an USB charger. When the charger input voltage is within a pre-determined range, the charging process will be activated. This detector can resist higher input voltage than other parts of the PMU.

2.6.5.1 Block Description

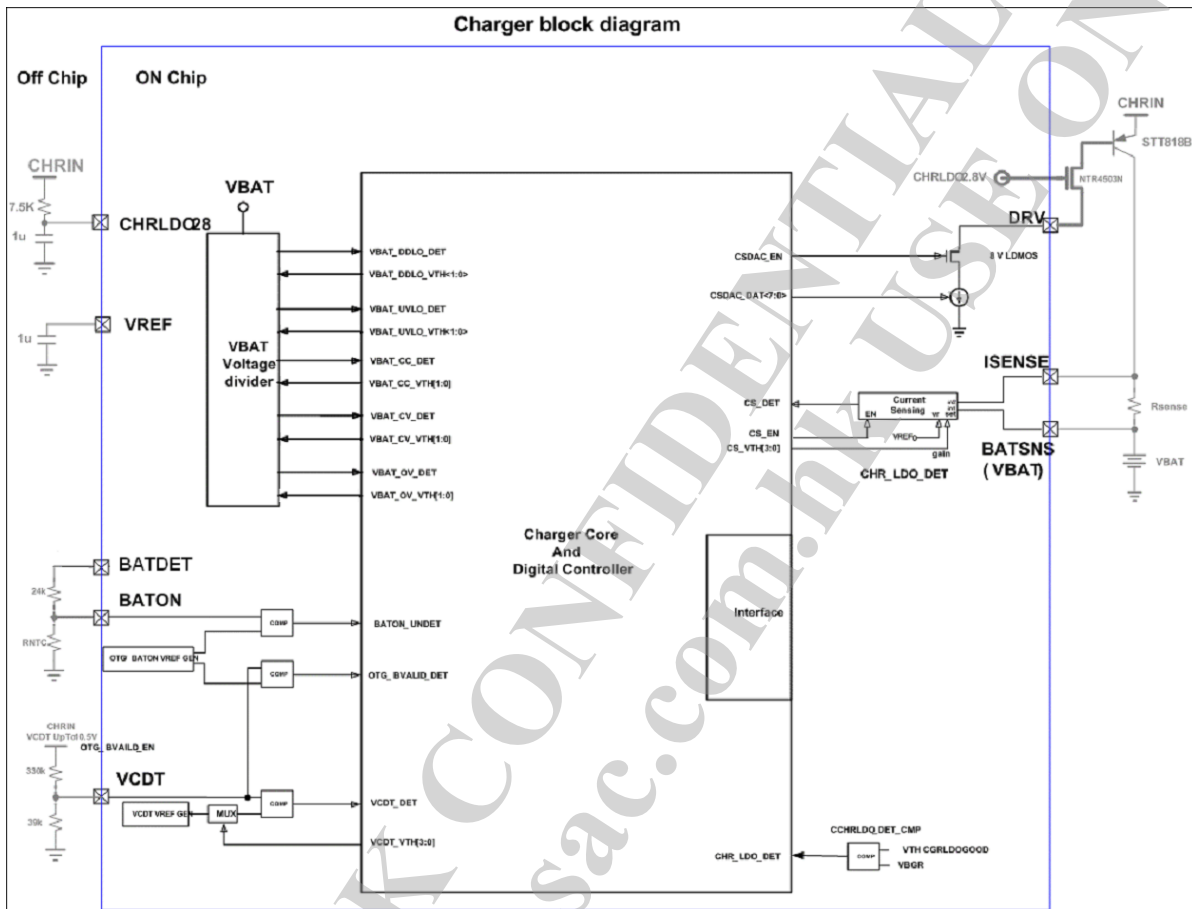


Figure 12. PCHR block diagram.

2.6.5.1.1 Charger Detection

Whenever an invalid charging source is detected ($> 7.0\text{ V}$), the charger detector stops the charging process immediately to avoid burning out the chip or even the phone. In addition, if the charger-in level is not high enough ($< 4.3\text{V}$), the charger will also be disabled to avoid improper charging behavior.

2.6.5.1.2 Charging Control

When the charger is active, the charger controller manages the charging phase according to the battery status. During the charging period, the battery voltage is constantly monitored. The battery charger supports pre-charge mode ($\text{VBAT} < 3.2\text{V}$, PMU power-off state), CC mode (constant current mode or fast charging mode at the range $3.2\text{V} < \text{VBAT} < 4.2\text{V}$) and CV mode (constant voltage mode) to optimize the charging procedure for Li-ion battery. The charging states diagram is shown in the figure below.

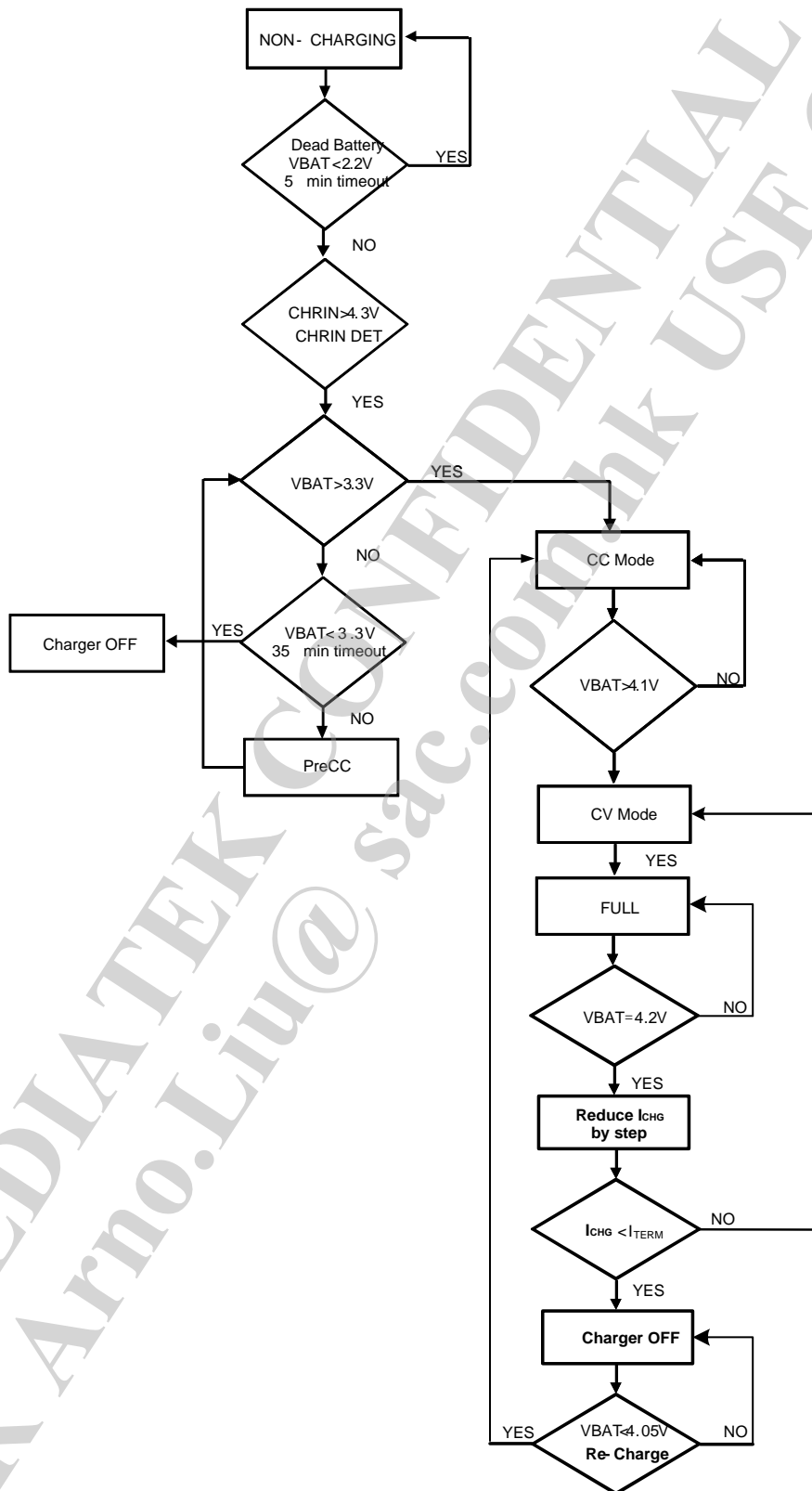


Figure 13. Charging states diagram

Pre-charge mode

When the battery voltage is in the UVLO state, the charger operates in the pre-charge mode. There are two steps in this mode. When the battery voltage is deeply discharged below 2.2V, PRECC1 trickle charging current will be applied to the battery.

The PRECC1 trickle charging current is about 550ms pulse 70mA current when VBAT is under 2.2V.

When the battery voltage exceeds 2.2V, called the PRECC2 stage, the closed-loop pre-charge is enabled. The voltage drop across the external RSENSE is kept around 40mV (AC charger) or 14mV (USB host). The closed-loop pre-charge current can be calculated:

$$I_{\text{PRECC2.ACadapter}} = \frac{V_{\text{SENSE}}}{R_{\text{sense}}} = \frac{40\text{mV}}{R_{\text{sense}}}$$

$$I_{\text{PRECC2.USBHOST}} = \frac{V_{\text{SENSE}}}{R_{\text{sense}}} = \frac{14\text{mV}}{R_{\text{sense}}}$$

Constant current mode

As the battery is charged up and over 3.4V, it can switch to the CC mode. (CHR_EN should be high) In the CC mode, several charging currents can be set by programming registers or the external RSENSE resistor. The charging current can be determined by CS_VTH/RSENSE, where CS_VTH is programmed by registers. For example, if RSENSE is selected as 0.2ohm, the CC mode charging current can be set from 70 to 800mA. It can accommodate the battery charger to various charger inputs with different current capabilities.

Constant voltage mode and over-voltage protection (OV)

While the battery voltage reaches about 4.2V, a constant voltage is used for charging. This is called the full-voltage charging mode or constant-voltage charging mode in correspondence to a linear charger. While the battery voltage actually reaches 4.2V, the charging current is gradually decreased step-by-step, the end-of-charging process starts. It may prolong the charging and detecting period for acquiring optimized full charging volume. The charging process is completed once the current reaches zero automatically and this mechanism is optimized for different battery

BC1.1 Dead-Battery Support of China Standard

MT2503D supports dead-battery condition from China standard (called BC1.1). The specification protects dead-battery charging by timer and trickle current. Once the battery voltage is below 2.2V, a period (TUNIT) of trickle current (IUNIT) will be applied to the battery.

If the battery voltage is still below 2.2V after applying trickle current, the charger will be disabled. On the other hand, if the battery voltage is raised to above 2.2V, the charger will enter the PRECC2 stage, and the charging current will be 70mA or 200mA depending on the type of charging port.

Under the condition of battery voltage from 2.2V to 3.3V, the charger will charge the battery with the PRECC2 current.

A dedicated 5mins (T1) timer will be timed out and disable the charger if the battery voltage is always below 2.7V under charging. Another 35mins (T2) timer will also be timed out and disable the charger if the battery voltage is always kept between 2.7V and 3.3V under charging.

The trickle current (IUNIT) and two dedicated timers protect the charging action if the battery is dead.

2.6.5.2 Functional Specifications

Table 27. Charger detection specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Charger detect-on range		4.3		7	V

Table 28. Pre-charge specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	IUNIT with 500ms pulse	VBAT < 2.2V	20	48	100	mA
	Pre-charging current	VBAT < 2.2V (500ms pulse)	20	48	100	mA
		VBAT ≥ 2.2V (USB host)	7/R _{sense}	14/R _{sense}	20/R _{sense}	mA
		VBAT ≥ 2.2V (AC adapter < 7V)	30/R _{sense}	40/R _{sense}	50/R _{sense}	mA
		VBAT ≥ 2.2V (AC adapter > 7V)	7/R _{sense}	14/R _{sense}	20/R _{sense}	mA
	Pre-charging off threshold	CHR_EN = L		3.3		V
	Pre-charging off hysteresis			0.4		V

Table 29. Constant current specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	CC mode charging current (CS_VTH)	CS_VTH [3:0] = 0000		320/R _{sense}		mA
		CS_VTH [3:0] = 0001		300/R _{sense}		mA
		CS_VTH [3:0] = 0010		280/R _{sense}		mA
		CS_VTH [3:0] = 0011		260/R _{sense}		mA
		CS_VTH [3:0] = 0100		240/R _{sense}		mA
		CS_VTH [3:0] = 0101		220/R _{sense}		mA
		CS_VTH [3:0] = 0110		200/R _{sense}		mA

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		CS_VTH [3:0] = 0111		180/R _{sense}		mA
		CS_VTH [3:0] = 1000		160/R _{sense}		
		CS_VTH [3:0] = 1001		140/R _{sense}		
		CS_VTH [3:0] = 1010		130/R _{sense}		
		CS_VTH [3:0] = 1011		110/R _{sense}		
		CS_VTH [3:0] = 1100		90/R _{sense}		
		CS_VTH [3:0] = 1101		60/R _{sense}		
		CS_VTH [3:0] = 1110		40/R _{sense}		
		CS_VTH [3:0] = 1111		14/R _{sense}		
	Current sensing resistor	RSENSE		0.2		ohm

Table 30. Constant voltage and over-voltage protection specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Charging complete threshold		4.15	4.2	4.25	V
	Battery over-voltage protection threshold (OV)			4.3		V

Table 31. BC1.1 specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
IUNIT	BC1.1 trickle current	VBAT < 2.2V		50	100	mA
IPRECC2 (USB host)	PRECC2 current	2.2 < VBAT < 3.3V		70	100	mA
IPRECC2 (AC adapter)	PRECC2 current	2.2 < VBAT < 3.3V		200	250	mA
T1	5 minute dedicated timer	2.2 < VBAT < 2.7V		5	6.5	min.
T2	35 minute dedicated timer	2.7 < VBAT < 3.3V		35	38.5	min.
TUNIT	BC1.1 trickle current period			550	770	ms.

2.7 GSM/GPRS RF

2.7.1 General Description

2G RFSYS which is built in MT2503D SOC is a highly integrated RF transceiver for multi-band GSM and GPRS cellular systems.

The features include:

Receiver

- Single-end saw-less Rx
- Quadrature RF mixer
- Fully integrated channel filter
- High dynamic range ADC
- 12dB PGA gain with 6dB gain step

Transmitter

- High accurate transmitter modulator for GSM/GPRS application
- Built-in calibration of SX loop filter and loop gain

Frequency synthesizer

- Programmable fractional-N synthesizer
- Integrated wide range RFVCO
- Integrated loop filter
- Fast settling time suitable for multi-slot GSM/GPRS applications

Digitally-Controlled Crystal Oscillator (DCXO)

- Two-pin 26 MHz crystal oscillator
- On-chip programmable capacitor array for coarse-tuning
- On-chip programmable capacitor array for fine-tuning

- Supports 32K XTAL-less operation

2.7.2 Functional Block Diagram

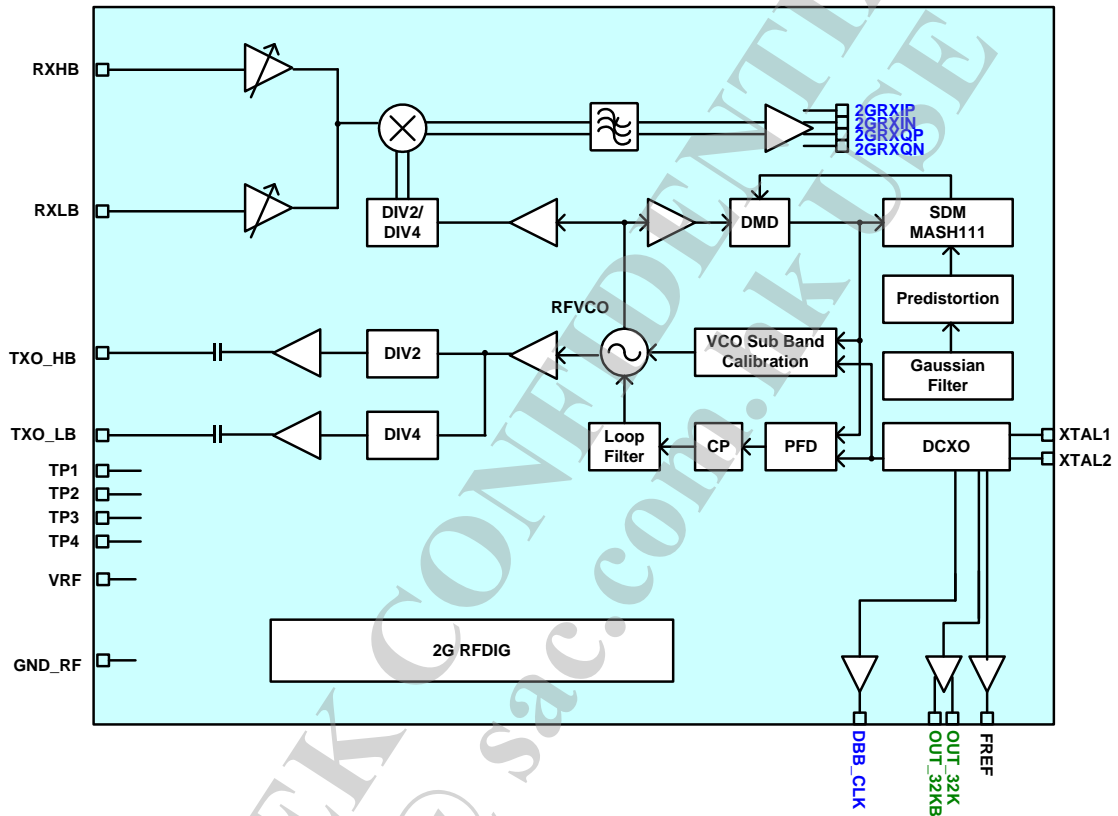


Figure 14. Diagram of MT2503D 2G RFSYS

2.7.3 Electrical Characteristics

Table 32. DC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

RFSYS mode	VRF	AVDD28_2GAPE	RFSYS total	Unit
BCM_Deep sleep (DCXO is off)	17	1	18	uA
BCM_Sleep (DCXO is on)	1.2	0.26	1.5	mA
Low power mode	60	1	61	uA
Full power mode	1.2	0.26	1.5	mA
RX (GSM850/EGSM)	62	5	67	mA
RX (DCS/PCS)	66	5	71	mA
TX (GSM850/EGSM)	41	2	43	mA
TX (DCS/PCS)	36	2	38	mA

Table 33. Rx AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
Input frequency	F _{RX}	GSM850		869		894	MHz
		GSM900		925		960	MHz
		DCS1800		1,805		1,880	MHz
		PCS1900		1,930		1,990	MHz
Voltage gain 1	G ₁	GSM850	LNA = High gain	52 ¹	55		dB
		GSM900	PGA = High gain	52 ²	55		dB
		DCS1800	LNA = High gain	52 ³	55		dB
		PCS1900	PGA = High gain	52 ⁴	55		dB
Voltage gain 2	G ₂	GSM850	LNA = Middle gain		46		dB
		GSM900	PGA = High gain		46		dB
		DCS1800	LNA = Middle gain		45		dB
		PCS1900	PGA = High gain		45		dB
Voltage gain 3	G ₃	GSM850	LNA = Low gain		26		dB
		GSM900	PGA = High gain		26		dB
		DCS1800	LNA = Low gain		26		dB
		PCS1900	PGA = High gain		26		dB
Noise figure at 25°C	NF ₂₅	GSM850	G ₁		3	5 ¹	dB
		GSM900			3	5 ²	dB
		DCS1800			3	5 ³	dB
		PCS1900			3	5 ⁴	dB
Noise figure at 85°C	NF ₈₅	GSM850	G ₁		4.5		dB
		GSM900			4.5		dB
		DCS1800			4.5		dB
		PCS1900			4.5		dB
2 nd -order input intercept point	IIP2	GSM850	G ₂		31 ¹	43	dBm
		GSM900			31 ²	43	dBm
		DCS1800			31 ³	43	dBm
		PCS1900			31 ⁴	43	dBm
3 rd -order input intercept point	IIP3	GSM850	G ₂		-14 ¹	-3	dBm
		GSM900			-14 ²	-3	dBm
		DCS1800			-14 ³	-3	dBm
		PCS1900			-14 ⁴	-3	dBm
3 rd -order input intercept point @ -20°C	IIP3-20	GSM850	G ₂		-5		dBm
		GSM900			-5		dBm
		DCS1800			-5		dBm
		PCS1900			-5		dBm
Receiver S/N with 3MHz blocker	SN _{3M}	GSM850	G ₂	8 ¹	12		dB
		GSM900	Blocker = -23dBm	8 ²	12		dB
		DCS1800	G ₂	8 ³	12		dB
		PCS1900	Blocker = -26dBm	8 ⁴	12		dB

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
Receiver S/N with OBB	SN _{OOB}	GSM850	G2	6 ⁵	8		dB
		GSM900	Blocker = 2dBm, offset +/-20MHz	6 ⁵	8		dB
		DCS1800	G2	6 ⁵	8		dB
		PCS1900	Blocker = -10/2dBm, offset +/-80/-100MHz	6 ⁵	8		dB
Image rejection ratio	IRR	ALL	G2	32 ^{1,2,3,4}	40		dB
Receiver channel response attenuation		ALL	@3MHz offset		20		dB
			@6MHz offset		35		dB
Receiver filtering 3-dB bandwidth		ALL	For all gain settings		900		kHz
PGA gain linearity		ALL	INL		0.2	1 ⁵	dBΩ
			DNL		0.1	0.5 ⁵	dBΩ
PGA gain step		ALL			6		dBΩ
PGA dynamic range		ALL			12		dBΩ
I/Q common-mode output voltage		ALL	G1	1.1 ⁵	1.2	1.3 ⁵	V
Output static dc offset		ALL	G1		100	200	mV

Table 34. Tx GMSK AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
Frequency	F _{TX}	GSM850		824		849	MHz
		GSM900		880		915	MHz
		DCS1800		1,710		1,785	MHz
		PCS1900		1,850		1,910	MHz
RMS phase error	PE _{rms}	GSM850 GSM900			1.5	2.5 ^{1,2}	degree
		DCS1800 PCS1900			1.5	2.5 ^{3,4}	degree
Output modulation spectrum	ORFS	GSM850 GSM900	400kHz offset (RBW = 30kHz bandwidth)		-66	-64 ^{1,2}	dBc
		DCS1800 PCS1900			-66	-64 ^{3,4}	dBc
		GSM850 GSM900	1.8MHz offset (RBW = 30kHz bandwidth)			-75 ⁵	dBc
		DCS1800 PCS1900				-75 ⁵	dBc
Tx noise in Rx band		GSM850	20MHz offset		-165	-163 ⁵	dBc/Hz
			35MHz offset		-166	-164 ⁵	dBc/Hz
		GSM900	20MHz offset		-165	-163 ⁵	dBc/Hz
			35MHz offset		-166	-164 ⁵	dBc/Hz
		DCS1800	20MHz offset		-160	-156 ⁵	dBc/Hz
		PCS1900	20MHz offset		-160	-156 ⁵	dBc/Hz

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
Output power level	P _{out}	GSM850 GSM900	PA driver amplifier R _{load} = 50Ω	1 ^{1,2}	3	6 ^{1,2}	dBm
		DCS1800 PCS1900		1 ^{3,4}	3	6 ^{3,4}	dBm
Output 3 rd harmonics		ALL	PA driver amplifier		-10		dBc

Table 35. SX AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Test condition	Min.	Typ.	Max.	Unit
Frequency range	F _{range}		3,296		3,980	MHz
Reference frequency	F _{ref}			26		MHz
Frequency step resolution	F _{res}			3		Hz
Phase noise	PN _{10k}	@ 10kHz offset		-83		dBc/Hz
	PN _{400k}	@ 400kHz offset		-114		dBc/Hz
	PN _{3M}	@ 3MHz offset		-136		dBc/Hz
Lock time of Rx burst	T _{lock_rx}	Frequency error < ± 0.1ppm		150	200 ⁵	us
Lock time of Tx burst	T _{lock_tx}	Frequency error < ± 0.1ppm		200	300 ⁵	us
Pushing figure		With internal RFVCO LDO		400		kHz/V

Table 36. DCXO AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Test condition	Min.	Typ.	Max.	Unit
Operating frequency	F _{ref}			26		MHz
Crystal C load	C _L			7.5		pF
Crystal tuning sensitivity	T _s		27.5	32.3		ppm/pF
Static range	SR	CDAC from 0 to 255	± 22	± 50		ppm
Dynamic range	DR	CAFC from 0 to 8191	36	50		ppm
AFC tuning step	F _{res-AFC}			0.008		ppm/DAC
AFC settling time	T _{AFC}	CAFC from 0 to 8191 CAFC from 8191 to 0 Frequency error < 0.1ppm		100	200 ⁵	us
Start-up time	T _{DCXO}	Frequency error < 1ppm Amplitude > 90 %			4 ⁵	ms
Pushing figure				0.2		ppm/V
Fref buffer output level	V _{Fref}	Max. loading = 19pF	0.8 ⁵			V _{p-p}
Fref buffer output phase noise		10kHz offset Jitter noise		-135		dBc/Hz

^{1,2}: Tested at E-GSM Tx channel 0 and GSM850 Rx channel 190.

^{3,4}: Tested at PCS Tx channel 601 and DCS Rx channel 636.

⁵: Not subject to production test – verified by characterization and design.

2.8 Bluetooth

2.8.1 Block Description

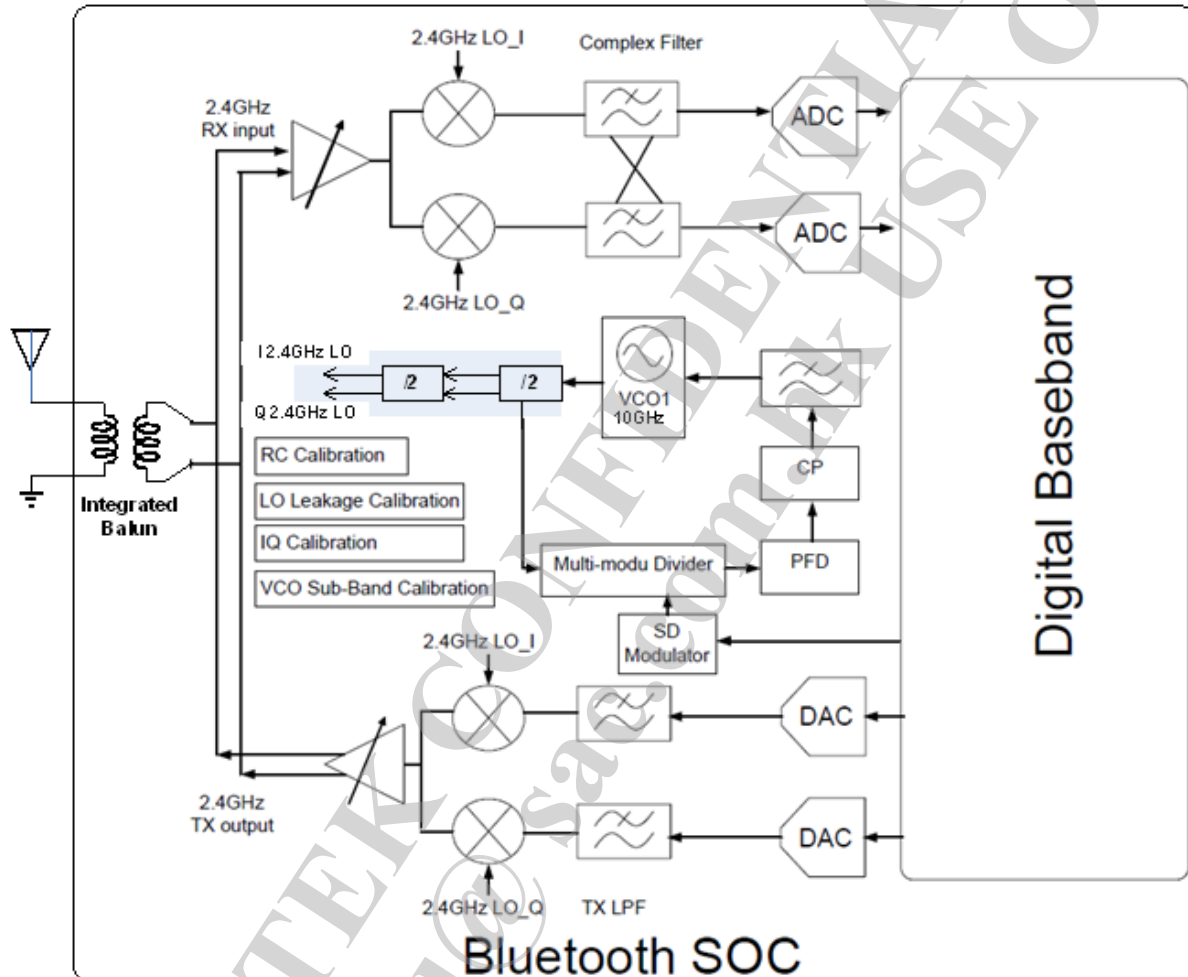


Figure 15. System diagram of Bluetooth RF transceiver

The Bluetooth RF subsystem contains a fully integrated transceiver.

For TX path, the baseband transmit data are digitally modulated in the baseband processor then up-converted to 2.4GHz RF channels through DA converter, filter, IQ up-converter and power amplifier. The power amplifier is capable of transmitting 7.5dBm power for class-1 operation.

For RX path, MT2503D is a low IF receiver architecture. An image-reject mixer down-converts the RF signal to the IF with LO from the synthesizer, which supports different clock frequencies as the reference clock. The mixer output is then converted to digital signal and down-converted to baseband for demodulation. A fast AGC enables effective discovery of device within dynamic range of the receiver.

MT2503D features self calibration schemes to compensate the process and temperature variation to maintain high performance. Those calibrations are performed automatically right after system boot-up.

2.8.2 Functional Specifications

2.8.2.1 Basic Data Rate – Receiver Specifications

Table 37. Basic data rate – receiver specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Receiver sensitivity	BER < 0.1%	-	-95	-	dBm
	Max. detectable input power	BER < 0.1%	-	0	-	dBm
	C/I co-channel selectivity	BER < 0.1%	-	4	-	dB
	C/I 1 MHz adj. channel selectivity	BER < 0.1%	-	-12	-	dB
	C/I 2 MHz adj. channel selectivity	BER < 0.1%	-	-42.5	-	dB
	C/I ≥ 3 MHz adj. channel selectivity	BER < 0.1%	-	-46	-	dB
	C/I image channel selectivity	BER < 0.1%	-	-24	-	dB
	C/I image 1 MHz adj. channel selectivity	BER < 0.1%	-	-45	-	dB
	Out-of-band blocking	30 to 2,000 MHz	-	-4	-	dBm
		2,000 to 2,399 MHz	-	-18	-	dBm
		2,498 to 3,000 MHz	-	-18	-	dBm
		3,000 MHz to 12.75 GHz	-	1	-	dBm
	Intermodulation		-	-22	-	dBm

2.8.2.2 Basic Data Rate – Transmitter Specification

Table 38. Basic data rate – transmitter specification

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Maximum transmit power		-	7.5	-	dBm
	Gain step		-	4	-	dB
	Δf1avg (00001111)		140	158	175	kHz
	Δf2max (10101010)		115	130	-	kHz
	Δf1avg/Δf2avg		0.8	0.9	-	kHz
	Initial carrier frequency drift		-75	5	75	kHz
	Frequency drift	DH1	-25	9	25	kHz
		DH3	-40	10	40	kHz

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		DH5	-40	10	40	kHz
	Max. drift rate		-	100	400	Hz/μs
	BW _{20dB} of Tx output spectrum		-	920	1,000	kHz
	In-band spurious emission	±2 MHz offset	-	-38	-	dBm
		±3 MHz offset	-	-43	-	dBm
		> ±3 MHz offset	-	-43	-	dBm
	Out-of-band spurious emission	30 MHz to 1 GHz	-	-36	-	dBm
		1 to 12.75 GHz	-	-30	-	dBm
		1.8 to 1.9 GHz	-	-47	-	dBm
		5.15 to 5.3 GHz	-	-47	-	dBm

2.8.2.3 Enhanced Data Rate – Receiver Specifications

Table 39. Enhanced data rate – receiver specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Receiver sensitivity	π/4 DQPSK, BER < 0.01%	-	-95	-	dBm
		8PSK, BER < 0.01%	-	-88	-	dBm
	Max. detectable input power	π/4 DQPSK, BER < 0.01%	-	-4.5	-	dBm
		8PSK, BER < 0.01%	-	-4.5	-	dBm
	C/I co-channel selectivity	π/4 DQPSK, BER < 0.01%	-	8	-	dB
		8PSK, BER < 0.01%	-	14.5	-	dB
	C/I 1MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-13	-	dB
		8PSK, BER < 0.01%	-	-7	-	dB
	C/I 2MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-42	-	dB
		8PSK, BER < 0.01%	-	-41.5	-	dB
	C/I ≥ 3MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-48	-	dB
		8PSK, BER < 0.01%	-	-44.5	-	dB
	C/I image channel selectivity	π/4 DQPSK, BER < 0.01%	-	-30	-	dB
		8PSK, BER < 0.01%	-	-23	-	dB
	C/I image 1 MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-47.5	-	dB
		8PSK, BER < 0.01%	-	-44.5	-	dB

2.8.2.4 Enhanced Data Rate – Transmitter Specifications

Table 40. Enhanced data rate – transmitter specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
--------	-------------	-----------	------	------	------	------

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Max. transmit power	$\pi/4$ DQPSK	-	4.5	-	dBm
		8PSK	-	4.5	-	dBm
	Relative transmit power	$\pi/4$ DQPSK	-	-1.7	-	dB
		8PSK	-	-1.7	-	dB
	Freq. stability ω_0	$\pi/4$ DQPSK	-	1.5	-	kHz
		8PSK	-	1.5	-	kHz
	Freq. stability ω_1	$\pi/4$ DQPSK	-	3	-	kHz
		8PSK	-	3	-	kHz
	$ \omega_0 + \omega_1 $	$\pi/4$ DQPSK	-	2.8	-	kHz
		8PSK	-	2.8	-	kHz
	RMS DEVM	$\pi/4$ DQPSK	-	7	-	%
		8PSK	-	6	-	%
	99% DEVM	$\pi/4$ DQPSK	-	11	-	%
		8PSK	-	11	-	%
	Peak DEVM	$\pi/4$ DQPSK	-	18	-	%
		8PSK	-	18	-	%
	In-band emission spurious	$\pi/4$ DQPSK, ± 1 MHz offset	-	-28	-	dBm
		8PSK, ± 1 MHz offset	-	-28	-	dBm
		$\pi/4$ DQPSK, ± 2 MHz offset	-	-25	-	dBm
		8PSK, ± 2 MHz offset	-	-25	-	dBm
		$\pi/4$ DQPSK, ± 3 MHz offset	-	-40.5	-	dBm
		8PSK, ± 3 MHz offset	-	-40.5	-	dBm

Note: To meet the specifications, use a front-end band-pass filter.

2.9 FM RF

2.9.1 Block Description

The connection between internal modules, as well as external interfaces, are as shown in Figure 16. The FM receiver section incorporates the complete receiving path with wide tuning range. The FM baseband signal processor incorporates the digital demodulator and audio processing function which provides superior audio quality.

FM contains completely integrated FM audio receiver functions (RDS/RBDS may also be supported depending on the model number). The integrated receiver enables superior sensitivity, ACI performance and FM audio performances with minimum external BOM.

The FM subsystem supports either high performance stereo analog line out or digital audio output (I2S).

For models supporting RDS/RBDS, large dedicated internal data buffers are allocated to reduce the frequency of the interrupt to the host, so that the receiving host can enter low power states efficiently.

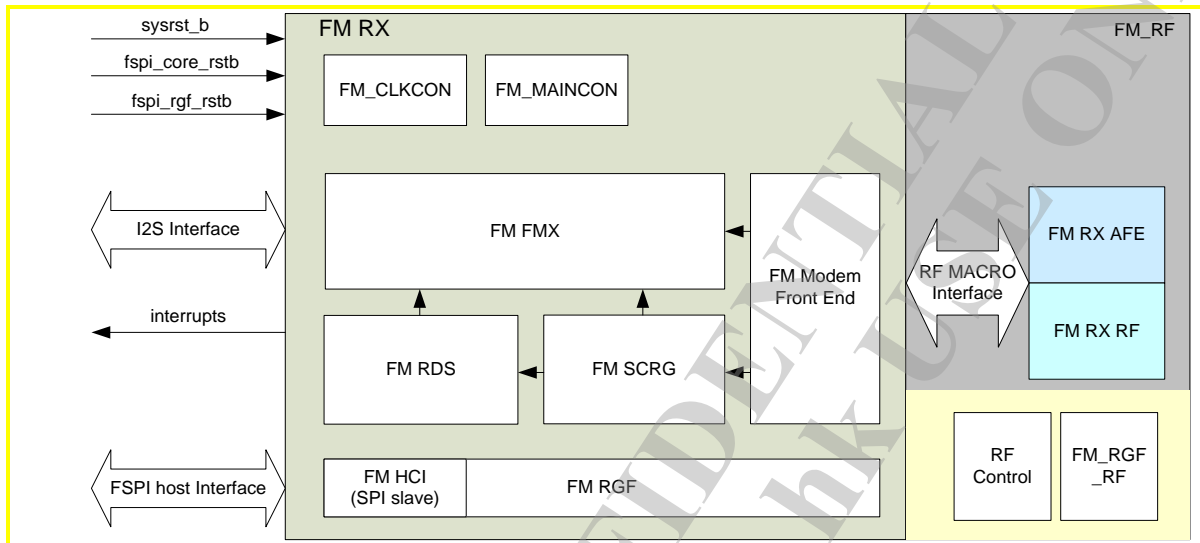


Figure 16. Block diagram of hardware top-level architecture

2.9.2 Functional Specifications

Table 41. FM receiver DC characteristics (TA=25°C, VDD=2.8V unless otherwise stated)

Operating mode	Current consumption	Unit
Idle	5	µA
FM receiver	12	mA

Unless otherwise stated, all receiver characteristics are applicable to both long and short antenna ports when operated under the recommended operating conditions. Typical specifications are for channel 98.7MHz, default register settings and under recommended operating conditions. The minimum and maximum specifications are for extreme operating voltage and temperature conditions, unless otherwise stated.

Table 42. FM receiver AC characteristics

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Input frequency range		65		108	MHz
	Sensitivity (long antenna) ^{1,3}	SINAD = 26dB, unmatched		3		dBµVemf
		SINAD = 26dB, matched		2		dBµVemf
	RDS sensitivity (long antenna)	Δf=2kHz, BLER < 5%, unmatched		18		dBµVemf
	Sensitivity (short antenna) ^{1,3}	SINAD = 26dB, unmatched		3		dBµVemf
		RDS sensitivity (short antenna)	Δf = 2kHz, BLER < 5%, unmatched		18	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	LNA input resistance ⁴	Antenna port		2.4k		Ohm
	LNA input capacitance ⁴	Antenna port		8		pF
	AM suppression ^{1,4}	M = 0.3		58		dB
	Adjacent channel selectivity ^{1,4}	±200kHz		53		dB
	Alternate channel selectivity ^{1,4}	±400kHz		65		dB
	Spurious response rejection ⁴	In-band		55		dB
	Maximum input level			117		dB μ Vemf
	Audio mono (S+N+D)/(N+D) ^{1,3,4}			60		dB
	Audio stereo (S+N+D)/(N+D) ^{2,3,4}			52		dB
	Audio stereo separation ⁴	$\Delta f = 75\text{kHz}$		45		dB
	Audio output load resistance	Single-ended at AFR/AFL outputs		10k		Ohm
	Audio output load capacitance	Single-ended at AFR/AFL outputs		12.5		pF
	Audio output voltage ^{1,4}	At AFR/AFL outputs		80		mVrms
	Audio output THD ^{1,4}			0.05	0.1	%
	Audio output frequency range	3dB corner frequency	30		15k	Hz

¹ $\Delta f = 22.5\text{kHz}$, $f_m = 1\text{kHz}$, $50\mu\text{s}$ de-emphasis, mono, L = R

² $\Delta f = 22.5\text{kHz}$, $f_m = 1\text{kHz}$, $50\mu\text{s}$ de-emphasis, stereo

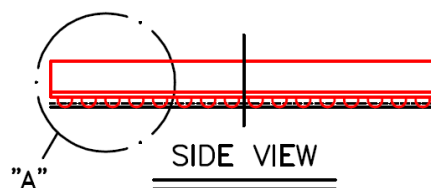
³ A-weighting, BW = 300 Hz to 15 kHz

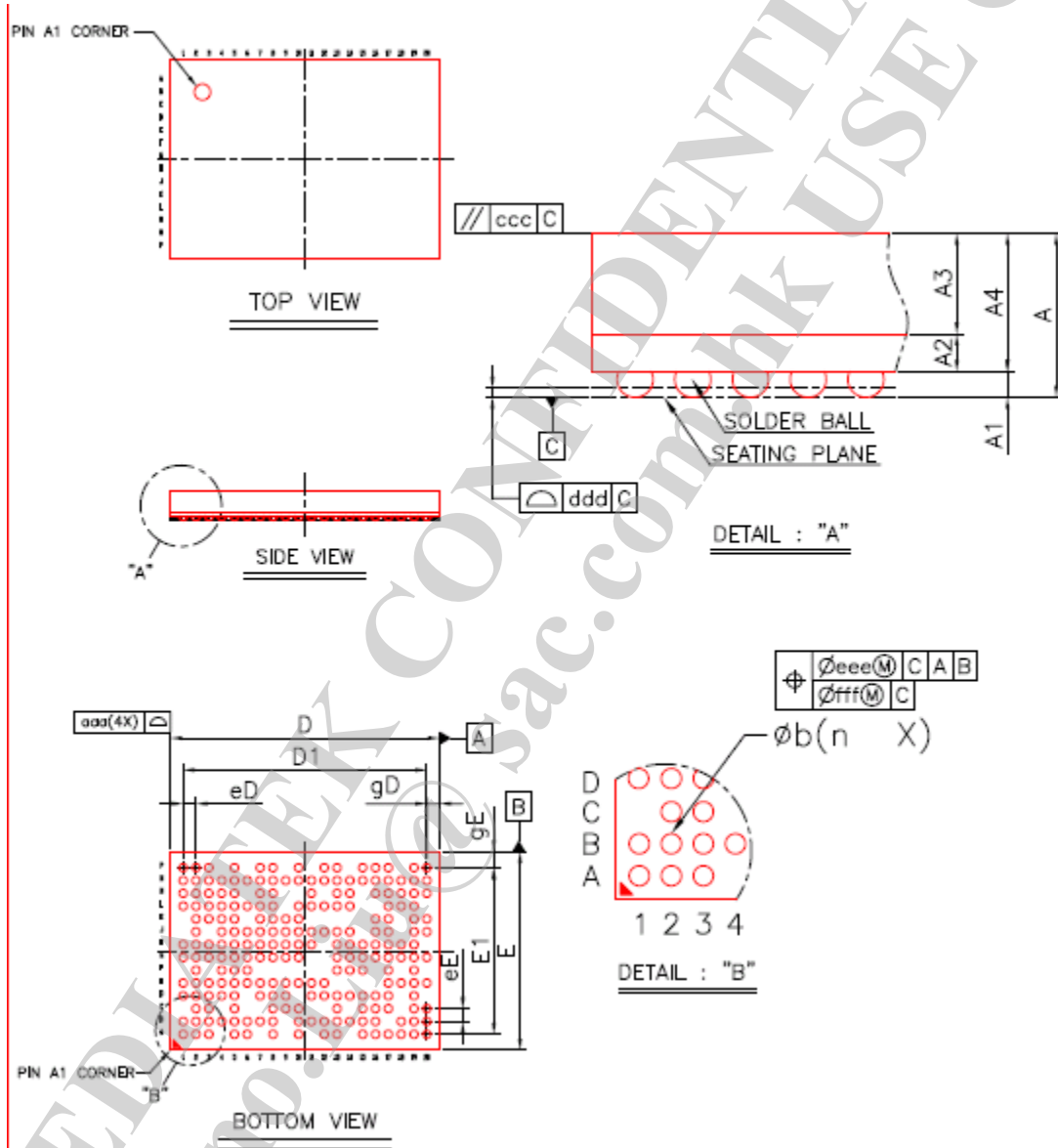
⁴ $V_{in} = 60\text{dB}\mu\text{Vemf}$

⁵ Reference clock accuracy assumes ideal FM source. If the input FM source has less frequency error, it is recommended to use a reference clock of accuracy within $\pm 100\text{ppm}$ so as not to affect the channel scan quality.

2.10 Package Information

2.10.1 Package Outlines





Item	Symbol	Common Dimensions			
		MIN.	NOM.	MAX.	
Package Type		VFBGA			
Body Size	X	D	8.30	8.40	8.50
	Y	E	6.10	6.20	6.30
Ball Pitch	X	eD	0.40		
	Y	eE	0.40		
Mold Thickness	A3	0.65 Ref.			
Substrate Thickness	A2	0.13 Ref.			
Substrate+Mold Thickness	A4	0.73	0.78	0.83	
Total Thickness	A	—	—	1.00	
Ball Diameter		0.25			
Ball Stand Off	A1	0.110	0.150	0.190	
Ball Width	b	0.220	0.270	0.320	
Package Edge Tolerance	aaa	0.05			
Mold Flatness	ccc	0.10			
Coplanarity	ddd	0.08			
Ball Offset (Package)	eee	0.15			
Ball Offset (Ball)	fff	0.05			
Ball Count	n	215			
Edge Ball Center to Center	X	D1	7.60		
	Y	E1	5.20		
Edge Ball Center to Package Edge	X	gD	0.40		
	Y	gE	0.50		

Figure 17. Outlines and dimension of VFBGA 8.4mm*6.2mm, 215-ball, 0.4 mm pitch package

2.10.2 Thermal Operating Specifications

Symbol	Description	Value	Unit	Notes
	Thermal resistance from device junction to package case	48	C/W	
	Maximum package temperature	65	Deg C	
	Maximum power dissipation	1.28	W	

2.10.3 Lead-free Packaging

MT2503D is provided in a lead-free package and meets RoHS requirements

2.11 Ordering Information

2.11.1 Top Marking Definition

<p>MEDIATEK ARM MT2503DV DDDD - ##### LLLLLL</p>	<p>MTXXXXXX Product No. DDDD: Date Code #####: Subcontractor Code LLLLLL: Die Lot No.</p>
---	---

Figure 18. Mass production top marking of MT2503D

Part number	Package	Description
MT2503DV/B	VFPGA	8.4mm*6.2mm, 215-ball, 0.4 mm pitch package, non-security version

3 Micro-Controller Unit Peripherals

3.1 Pulse-Width Modulation Outputs (2 Channel)

3.1.1 General Description

2 generic pulse-width modulators are implemented to generate pulse sequences with programmable frequency and duty cycles for LCD backlight. As long as the internal counter value is bigger than or equal to the threshold value, the duration of the PWM output signal is LOW. The waveform is shown in Figure 19.

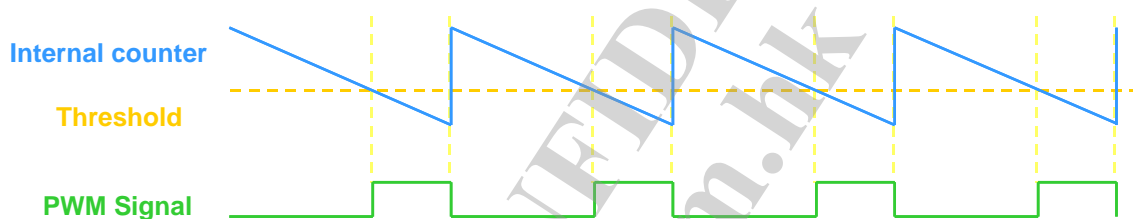


Figure 19. PWM waveform

The frequency and volume of the PWM output signal are determined by PWM1_COUNT, PWM1_THRES and PWM1_CON. The POWERDOWN (pdn1_pwm) signal is applied to power-down the PWM_1ch module. When PWM_1ch is deactivated (pwm1_pdn=1), the output is in the LOW state.

The output PWM frequency is determined by:

$$\frac{CLK}{CLOCK_DIV \times (PWM_COUNT + 1)}$$

CLK = 13000000 when CLKSEL = 0, CLK = 32000 when CLKSEL = 1

CLOCK_DIV = 1, when CLK[1:0] = 00b

CLOCK_DIV = 2, when CLK[1:0] = 01b

CLOCK_DIV = 4, when CLK[1:0] = 10b

CLOCK_DIV = 8, when CLK[1:0] = 11b

The output PWM duty cycle is determined by: $\frac{PWM_THRES}{PWM_COUNT + 1}$

Note: PWM_THRES should be less than the PWM_COUNT. If this condition is not satisfied, the output pulse of the PWM will always be HIGH.

Figure 20 shows the PWM waveform with the indicated register values.

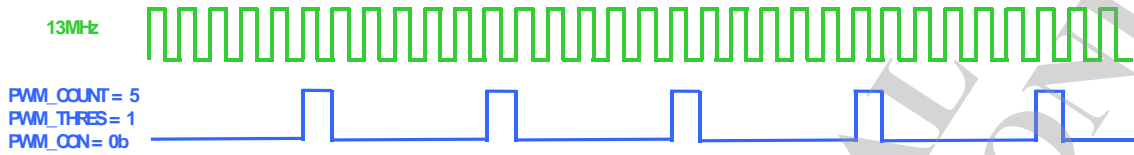


Figure 20. PWM waveform with register values

3.1.2 Register Definition

Module name: Pulse Width Modulation base address: (+A00E000h)

Address	Name	Width	Register function
A00E0000	PWM1_CTRL_ADDR	16	PWM1 control register
A00E0004	PWM1_COUNT_ADDR	16	PWM1 max counter value register
A00E0008	PWM1_THRESH_ADDR	16	PWM1 threshold value register

A00E0000 **PWM1_CTRL_ADDR** **PWM1 Control Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PWM1_CLK_SEL	PWM1_CLK_DIV	
Type														RW	RW	
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2	PWM1_CLK_SEL	CLK_SEL	Selects source clock frequency of PWM1 0: CLK = 13MHz 1: CLK = 32kHz
1:0	PWM1_CLK_DIV	CLK_DIV	Selects clock prescaler scale of PWM1 2'b00: f = fclk 2'b01: f = fclk/2 2'b10: f = fclk/4 2'b11: f = fclk/8

A00E0004 **PWM1_COUNT_ADDR** **PWM1 Max. Counter Value Register** **1FFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM1_COUNT															
Type	RW															
Reset				1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
12:0	PWM1_COUNT	PWM1_COUNT	PWM1 maximum counter value This value is the initial value of the internal counter. Regardless of the operation mode, if PWM1_COUNT is written while the internal counter is counting backwards, the new initial value will not take effect until the internal counter counts down to 0, i.e. a complete period.

A00E0008 **PWM1_THRESH_ADDR** **PWM1 Threshold Value Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM1_THRES															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
12:0	PWM1_THRES	PWM1_THRES	PWM1 threshold value When the internal counter value is bigger than or equal to PWM1_THRES, the PWM1 output signal will be "0". When the internal counter is less than PWM1_THRES, the PWM1 output signal will be "1".

Module name: Pulse Width Modulation base address: (+A0280000h)

Address	Name	Width	Register function
A0280000	<u>PMW4_CTRL_ADDR</u>	16	PMW4 control register
A0280004	<u>PMW4_COUNT_ADDR</u>	16	PMW4 max counter value register
A0280008	<u>PMW4_THRESH_ADDR</u>	16	PMW4 threshold value register

A0280000 **PMW4_CTRL_ADDR** **PMW4 Control Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PMW4_CLK_SEL	PMW4_CLK_DIV	
Type														RW	RW	
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2	PMW4_CLK_SEL	CLK_SEL	Selects source clock frequency of PMW4 0: CLK = 13MHz 1: CLK = 32kHz
1:0	PMW4_CLK_DIV	CLK_DIV	Selects clock prescaler scale of PMW4 2'b00: f = fclk 2'b01: f = fclk/2 2'b10: f = fclk/4 2'b11: f = fclk/8

A0280004 **PMW4_COUNT_ADDR** **PMW4 Max. Counter Value Register** **1FFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PMW4_COUNT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
12:0	PMW4_COUNT	PMW4_COUNT	PMW4 maximum counter value This value is the initial value of the internal counter. Regardless of the operation mode, if PMW4_COUNT is written while the internal counter is counting backwards, the new initial value will not take effect until the internal counter counts down to 0, i.e. a complete period.

A0280008 PMW4_THRESH_ADDR PMW4 Threshold Value Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				PMW4_THRES												
Type				RW												
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
12:0	PMW4_THRES	PMW4_THRES	PMW4 threshold value When the internal counter value is bigger than or equal to PMW4_THRES, the PMW4 output signal will be "0". When the internal counter is less than PMW4_THRES, the PMW4 output signal will be "1".

Module name: PWM_2CH base address: (+A0740000h)

Address	Name	Width	Register Function
A074000C	PWM2_CTRL	16	PWM2 control register Selects CLK SRC and prescaler scale.
A0740014	PWM2_THRES	16	PWM2 threshold value register Controls the duty of waveform
A0740018	PWM3_CTRL	16	PWM3 control register Select CLK SRC and prescaler scale.
A074001C	PWM3_COUNT	16	PWM3 max counter value register Configures internal counter's max. value
A0740020	PWM3_THRES	16	PWM3 threshold value register

A074000C PWM2_CTRL PWM2 Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PWM2_CLK_SEL		
Type														RW		
Reset														1		

Bit(s)	Name	Description
2	PWM2_CLK_SEL	Selects PWM2 CLK 0: CLK = 13M 1: CLK = 32k CLK

A0740014 PWM2_THRES PWM2 Threshold Value Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name																	PWM2_THRES
Type																	RW
Reset																	0 0

Bit(s)	Name	Description
		PWM2 threshold value
1:0	PWM2_THRES	0: Duty = 0% 1: Duty = 50% 2: Duty = 100%

A0740018 PWM3_CTRL PWM3 Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													PWM3_ALWAYS_HIGH	PWM3_CLK_SEL	PWM3_CLK_DIV	
Type													RW	RW	RW	
Reset													0	0	0 0	

Bit(s)	Name	Description
3	PWM3_ALWAYS_HIGH	When pwm3_thresh is set to be bigger than pwm3_width, which means the PWM output is always high, the driver should set this register to 1. It is specially used by ISINK. 0: Duty! 1: Duty = 100%
2	PWM3_CLK_SEL	Selects PWM3 CLK 0: CLK = 13M 1: CLK = 32k CLK
1:0	PWM3_CLK_DIV	PWM3 CLK division 2'b0: f = fclk 2'b1: f = fclk/2 2'b2: f = fclk/4 2'b3: f = fclk/8

A074001C PWM3_COUNT PWM3 Max Counter Value Register 1FFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM3_COUNT															
Type	RW															
Reset				1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
		PWM3 maximum counter value
12:0	PWM3_COUNT	This value is the initial value of the internal counter. Regardless of the operation mode, if PWM3_COUNT is written while the internal counter is counting backwards, the new initial value will not take effect until the internal

Bit(s)	Name	Description
		counter counts down to 0, i.e. a complete period.

A0740020 PWM3_THRES PWM3 Threshold Value Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PWM3_THRES															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
12:0	PWM3_THRES	PWM3 threshold value When the internal counter value is bigger than or equal to PWM3_THRES, the PWM3 output signal will be 0. When the internal counter is less than PWM3_THRES, the PWM3 output signal will be 1.

3.2 SIM Interface

MT2503D contains two dedicated smart card interfaces to allow the MCU to access two SIM cards. Each interface can operate via 5 terminals. See Figure 21, SIMVCC, SIMSEL, SIMRST, SIMCLK and SIMDATA are for one SIM interface, and SIM2VCC, SIM2SEL, SIM2RST, SIM2CLK and SIM2DATA are for the other SIM interface.

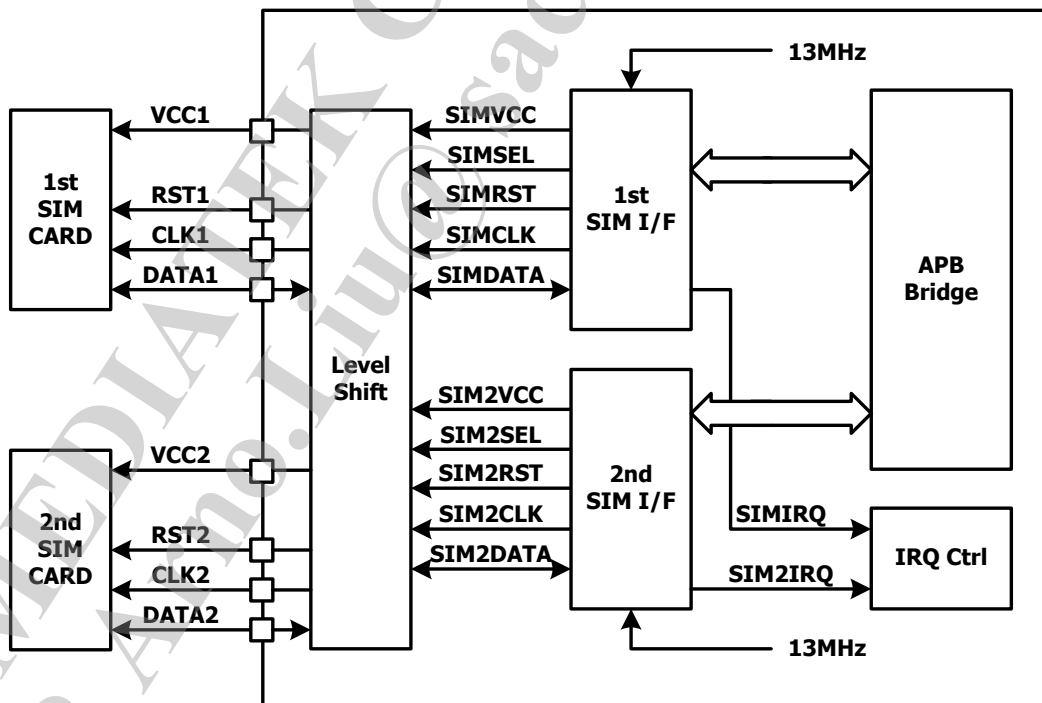


Figure 21. Block diagram of SIM interface

The functions of the two SIM interfaces are identical; therefore, only the first SIM interface will be described in this document. SIMVCC is used to control the external voltage supply to the SIM card, and SIMSEL determines the regulated smart card supply voltage. SIMRST is used as the SIM card reset signal. Besides, SIMDATA and SIMCLK are used for data exchange.

The SIM interface is a half duplex asynchronous communication port, and its data format is composed of ten consecutive bits: a start bit in state “low”, eight information bits and a tenth bit used for parity checking. The data format can be divided into two modes as follows:

- Direct convention mode (ODD = SDIR = SINV = 0)

SB D0 D1 D2 D3 D4 D5 D6 D7 PB

SB: Start bit (in state “low”)

Dx: Data byte (LSB is the first and logic level ONE is in state “high”)

PB: Even parity check bit

- Inverse convention mode (ODD = SDIR = SINV = 1)

SB N7 N6 N5 N4 N3 N2 N1 N0 PB

SB: Start bit (in state “low”)

Nx: Data byte (MSB is the first and logic level ONE is in state “low”)

PB: Odd parity check bit

If the receiver obtains a wrong parity bit, it will respond by pulling the SIMDATA “low” to inform the transmitter, and the transmitter will retransmit the character.

If the receiver is an SIM card, the error response will start 0.5 bit after the PB and may last for 1 ~ 2-bit period. If the receiver is an SIM interface, the error response will start 0.5 bit after the PB and last for 1.5-bit period.

If the SIM interface is a transmitter, it will take total 14 bits guard period wherever the error response appears. If the receiver shows the error response, the SIM interface will retransmit the previous character again, or it will transmit the next character.

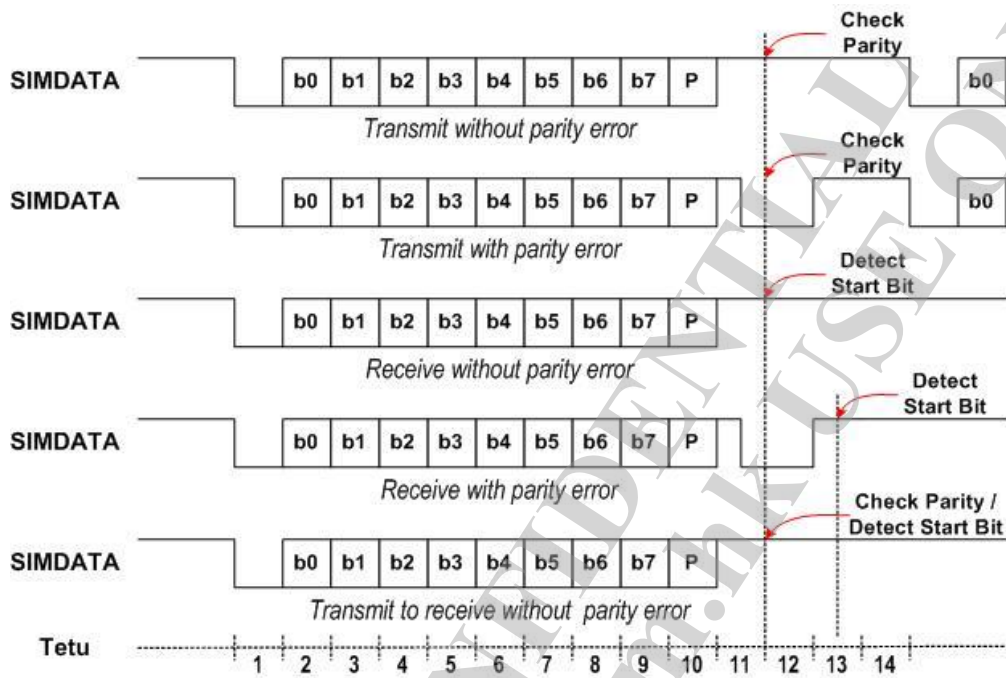


Figure 22. Timing diagram of SIM interface

3.2.1 Register Definition

When the MCU controls two SIM card interfaces, all registers will be duplicated to two copies but with different base address. n = "0" is for the 1st SIM card interface; n=1 is for the 2nd SIM card interface. For example, address SIMIF0+0000h is mapped to the SIMIF0_SIM_CTRL register while address SIMIF1+0000h is mapped to the SIMIF1_SIM_CTRL register.

3.2.1.1 Register Overview

MCU register address (hex)	Acronym	Description
1st SIM card interface		
SIMIF0+0000h	SIMIF0_SIM_CTRL	Control register
SIMIF0+0004h	SIMIF0_SIM_CONF	Configuration register
SIMIF0+0008h	SIMIF0_SIM_BRR	Baudrate register
SIMIF0+0010h	SIMIF0_SIM_IRQEN	Interrupt enabling register
SIMIF0+0014h	SIMIF0_SIM_STS	Status register
SIMIF0+0018h	SIMIF0_SIM_CLR_STA	SIM clear status
SIMIF0+0020h	SIMIF0_SIM_RETRY	Retry limit register
SIMIF0+0024h	SIMIF0_SIM_TIDE	FIFO tide mark register
SIMIF0+0030h	SIMIF0_SIM_DATA	TX/RX data register

MCU register address (hex)	Acronym	Description
SIMIF0+0034h	SIMIF0_SIM_COUNT	FIFO count register
SIMIF0+0040h	SIMIF0_SIM_ETIME	Activation time register
SIMIF0+0044h	SIMIF0_SIM_DTIME	Deactivation time register
SIMIF0+0048h	SIMIF0_SIM_TOUT	Character to character waiting time register
SIMIF0+004Ch	SIMIF0_SIM_GTIME	Block to block guard time register
SIMIF0+0050h	SIMIF0_SIM_ETIME	Block to error signal time register
SIMIF0+0054h	SIMIF0_SIM_EXT_TIME	Extend data I/O state switch time register
SIMIF0+0058h	SIMIF0_SIM_CGTIME	Character to character guard time register
SIMIF0+0060h	SIMIF0_SIM_INS	Command header register : INS
SIMIF0+0064h	SIMIF0_SIM_IMP3	Command header register : P3
SIMIF0+0068h	SIMIF0_SIM_SW1	Procedure byte register : SW1
SIMIF0+006Ch	SIMIF0_SIM_SW2	Procedure byte register : SW2
SIMIF0+0070h	SIMIF0_SIM_ATRSTA	ATR state register
SIMIF0+0074h	SIMIF0_SIM_STATUS	Protocol state register
SIMIF0+0080h	SIMIF0_SIM_DMADATA	TX/RX data register for DMA
SIMIF0+0090h	SIMIF0_SIM_DBG	Debug register
SIMIF0+0094h	SIMIF0_SIM_DBGDATA	FIFO data debug register
SIMIF0+00A0h	SIMIF0_SIM_SCLK	SCLK PAD control register
SIMIF0+00A4h	SIMIF0_SIM_SRST	SRST PAD control register
SIMIF0+00A8h	SIMIF0_SIM_SIO	SIO PAD control register
SIMIF0+00ACh	SIMIF0_SIM_MON	PAD monitor register
SIMIF0+00B0h	SIMIF0_SIM_SEL	Testing output select
2nd SIM card interface		
SIMIF1+0000h	SIMIF1_SIM_CTRL	Control register
SIMIF1+0004h	SIMIF1_SIM_CONF	Configuration register
SIMIF1+0008h	SIMIF1_SIM_BRR	Baudrate register
SIMIF1+0010h	SIMIF1_SIM_IRQEN	Interrupt enabling register
SIMIF1+0014h	SIMIF1_SIM_STS	Status register
SIMIF1+0018h	SIMIF1_SIM_CLR_STA	Sim clear status
SIMIF1+0020h	SIMIF1_SIM_RETRY	Retry limit register
SIMIF1+0024h	SIMIF1_SIM_TIDE	FIFO tide mark register
SIMIF1+0030h	SIMIF1_SIM_DATA	TX/RX data register
SIMIF1+0034h	SIMIF1_SIM_COUNT	FIFO count register
SIMIF1+0040h	SIMIF1_SIM_ETIME	Activation time register
SIMIF1+0044h	SIMIF1_SIM_DTIME	Deactivation time register
SIMIF1+0048h	SIMIF1_SIM_TOUT	Character to character waiting time register
SIMIF1+004Ch	SIMIF1_SIM_GTIME	Block to block guard time register
SIMIF1+0050h	SIMIF1_SIM_ETIME	Block to error signal time register
SIMIF1+0054h	SIMIF1_SIM_EXT_TIME	Extend data I/O state switch time register
SIMIF1+0058h	SIMIF1_SIM_CGTIME	Character to character guard time register
SIMIF1+0060h	SIMIF1_SIM_INS	Command header register : INS

MCU register address (hex)	Acronym	Description
SIMIF1+0064h	SIMIF1_SIM_IMP3	Command header register : P3
SIMIF1+0068h	SIMIF1_SIM_SW1	Procedure byte register : SW1
SIMIF1+006Ch	SIMIF1_SIM_SW2	Procedure byte register : SW2
SIMIF1+0070h	SIMIF1_SIM_ATRSTA	ATR state register
SIMIF1+0074h	SIMIF1_SIM_STATUS	Protocol state register
SIMIF1+0080h	SIMIF1_SIM_DMADATA	TX/RX data register for DMA
SIMIF1+0090h	SIMIF1_SIM_DBGD	Debug register
SIMIF1+0094h	SIMIF1_SIM_DBGDATA	FIFO data debug register
SIMIF1+00A0h	SIMIF1_SIM_SCLK	SCLK PAD control register
SIMIF1+00A4h	SIMIF1_SIM_SRST	SRST PAD control register
SIMIF1+00A8h	SIMIF1_SIM_SIO	SIO PAD control register
SIMIF1+00ACh	SIMIF1_SIM_MON	PAD monitor register
SIMIF1+00B0h	SIMIF1_SIM_SEL	Testing output select

3.2.1.2 Register Description

SIMn+0000h SIM Module Control Register										SIMIFN_SIM_CTRL						
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										VCCCTRL	VCCLV	RSTCTRL	RSTLV	WRST	CSTOP	SIMON
Type										R/W	R/W	R/W	R/W	W	R/W	R/W
Reset										0	0	0	0	0	0	0

- SIMON** Controls SIM card power-up/power-down
0 1-to-0 change will start the card deactivation sequence.
1 0-to-1 change will start the card activation sequence.
- CSTOP** Enables clock stop mode. Together with CPOL in the SIM_CONF register, it determines the polarity of SIMCLK in this mode.
0 Enable SIMCLK output
1 Disable SIMCLK output
- WRST** Controls SIM card warm reset
- RSTLV** Controls SIMRST parking level in SIMRST direct control mode
- RSTCTRL** Enables SIMRST direct control mode
- VCCLV** Controls SIMVCC parking level in SIMVCC direct control mode
- VCCCTRL** Enables SIMVCC direct control mode

SIMn+0004h SIM Module Configuration Register										SIMIFN_SIM_CONF						
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			T1TX2RXEN	TXRDIS	RXRDIS	HFEN	T0EN	T1EN	TOUT	SIMSE L	ODD	SDIR	SINV	CPOL	TXACK	RXACK
Type			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

RXACK	Handshaking control of SIM card reception error 0 Disable character receipt handshaking 1 Enable character receipt handshaking
TXACK	Handshaking control of SIM card transmission error 0 Disable character transmission handshaking 1 Enable character transmission handshaking
CPOL	SIMCLK polarity control in clock stop mode 0 Make SIMCLK stop in "low" level 1 Make SIMCLK stop in "high" level
SINV	Data inversion mode 0 Does not invert the transmitted and received data; data logic ONE is in "high" state 1 Invert the transmitted and received data; data logic ONE is in "low" state
SDIR	Direction of data transfer 0 LSB is transmitted and received first. 1 MSB is transmitted and received first.
ODD	Selecting odd or even parity 0 Even parity 1 Odd parity
SIMSEL	Selects SIM card supply voltage (also configure SIMSEL in PMU register) 0 SIMSEL pin is set to "low" level, 1.8V 1 SIMSEL pin is set to "high" level, 3V
TOUT	Controls SIM work waiting time counter 0 Disable time-out counter 1 Enable time-out counter
T1EN	Controls T = 1 protocol controller 0 Disable T = 1 protocol controller 1 Enable T = 1 protocol controller
T0EN	Controls T = 0 protocol controller 0 Disable T = 0 protocol controller 1 Enable T = 0 protocol controller
HFEN	Controls hardware flow 0 Disable hardware flow control 1 Enable hardware flow control
RXRDIS	Disables RX DMA request 0 Enable RX DMA request (default) RXRDIS must be set to 0 for protocol T = 1 1 Disable RX DMA request During TX transmission and not protocol T = 1, the recommended setting of RXRDIS is 1
TXRDIS	Disables TX DMA request disable 0 Enable TX DMA request (default) TXRDIS must be set to 0 for protocol T = 1 1 Disable TX DMA request

During RX transmission and not protocol T = 1, the recommended setting of TXRDIS is 1.

- T1TX2RXEN** Enables DMA type auto switch for protocol T = 1 (this function is not supported in MT6260)
- 0** Disable DMA type auto switch function
If the current block is TX transmission and the next block is also TX transmission, disabling this bit is recommended
 - 1** Enable DMA type auto switch function
If the current block is TX transmission and the next block is RX transmission, enabling this bit is recommended to improve transmission quality

SIMn +0008h SIM Baudrate Register											SIMIFN_SIM_BRR						
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name										ETU[8:0]						SIMCLK[1:0]	
Type										R/W						R/W	
Reset										372d						01	

- SIMCLK** Sets up SIMCLK frequency
- 00** Reserved
 - 01** 13/4 MHz
 - 10** 13/8 MHz
 - 11** 13/12 MHz

ETU Determines duration of elementary time unit in SIMCLK unit
The minimum valid setting of ETU is 8

SIMn +0010h SIM Interrupt Enable Register											SIMIFN_SIM_IRQEN					
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					UDRU N	EDCE RR	T1EN D	RXER R	TOEN D	SIMO FF	ATRER R	TXER R	TOU T	OVRU N	RXTID E	TXTID E
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0	0	0	0	0	0	0	0	0	0	0	0

- For all the bits
- 0** Disable interrupt
 - 1** Enable interrupt

SIMn +0014h SIM Module Status Register											SIMIFN_SIM_STS					
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					UDRU N	EDCE RR	T1EN D	RXER R	TOEN D	SIMO FF	ATRER R	TXER R	TOU T	OVRU N	RXTID E	TXTID E
Type					R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R	R
Reset				-	-	-	-	-	-	-	-	-	-	-	-	-

- TXTIDE** The interrupt occurs when the number of transmitted data in the FIFO is less than the transmitted tide.
- RXTIDE** The interrupt occurs when the number of received data in the FIFO is less than the received tide.
- OVRUN** Receiving FIFO overflow interrupt occurs.
- TOUT** Between characters time-out interrupt occurs.
- TXERR** Character transmission error interrupt occurs.
- ATRERR** ATR start time-out interrupt occurs.
- SIMOFF** Card deactivation completed interrupt occurs.
- T0END** Data transfer handled by T = 0 controller completed interrupt occurs.
- RXERR** Character reception error interrupt occurs.
- T1END** Data transfer handled by T = 1 controller completed interrupt occurs.
- EDCERR** T = 1 controller CRC error occurs.
- UDRUN** FIFO underflow interrupt occurs (still reading FIFO when FIFO is empty).

SIMn +0018h SIM Clear Status Register

SIMIFN_SIM_CLR_ST

A

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																CLR_STA
Type																RO
Reset																0

CLA_STA 1: Clear SIMIF. Do not write to SIMIF; 0: SIMIF clear finished or not in clear status, you can write data to SIMIF.

SIMn +0020h SIM Retry Limit Register

SIMIFN_SIM_RETRY

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						TXRETRY[2:0]								RXRETRY[2:0]		
Type						R/W								R/W		
Reset						3h								3h		

RXRETRY Specifies maximum number of receive retries allowed when parity error occurs.

TXRETRY Specifies maximum number of transmit retries allowed when parity error occurs.

SIMn +0024h SIM FIFO Tide Mark Register

SIMIFN_SIM_TIDE

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						TXTIDE[3:0]								RXTIDE[3:0]		
Type						R/W								R/W		
Reset						0h								0h		

RXTIDE Trigger point of RXTIDE interrupt

TXTIDE Trigger point of TXTIDE interrupt

SIMn +0030h Data Register Used As Tx/Rx Data Register SIMIFN_SIM_DATA

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												DATA[7:0]				
Type												R/W				
Reset												-				

DATA Eight data digits, corresponding to the character being read or written

SIMn +0034h SIM FIFO Count Register SIMIFN_SIM_COUNT

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												COUNT[4:0]				
Type												R/W				
Reset												0h				

COUNT Number of characters in the SIM FIFO when read and flushes when written.

SIMn +0040h SIM Activation Time Register SIMIFN_SIM_ETIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ETIME[9:0]				
Type												R/W				
Reset												2BEh				

ETIME Defines the duration, in 64 SIM clock cycles, of the time taken for each of the three stages of the card activation process, from SIMON transiting to “high” to turning on VCC, from turning on VCC to pull data “high” and then from pulling data “high” to turning on CLK.

SIMn +0044h SIM Deactivation Time Register SIMIFN_SIM_DTIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												DTIME[5:0]				
Type												R/W				
Reset												Fh				

DTIME Defines the duration, in 64 13 MHz clock cycles, of the time taken for each of the three stages of the card deactivation sequence, from pulling RST “low” to turning off CLK, from turning off CLK to pulling data “low”, from pulling data “low” to turning off VCC.

SIMn +0048h Character to Character Waiting Time Register SIMIFN_SIM_TOUT

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												WTIME[21:0]				

Type																	R/W
Reset																	260h
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	WTIME[21:0]																
Type	R/W																
Reset	260h																

WTIME Maximum interval between the leading edge of two consecutive characters in 16 ETU units

SIMn +004Ch Block to Block Guard Time Register **SIMIFN_SIM_GTIME**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	GTIME[3:0]																
Type	R/W																
Reset	10d																

GTIME Minimum interval between the leading edge of two consecutive characters sent in opposite directions in ETU unit

SIMn +0050h Block to Error Signal Time Register **SIMN_SIM_ETIME**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	ETIME[5:0]																
Type	R/W																
Reset	15d																

ETIME Defines the interval, in 1/16 ETU unit, between the end of the transmitted parity bit and the time to check the parity error signal sent from SIM card.

SIMn +0054h Active High Period Control Register **SIMIFN_SIM_EXT_TIM**
E

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	EXT_TIME[3:0]																
Type	R/W																
Reset	1d																

EXT_TIME Defines the interval, in 1/16 ETU unit, between the end of the transmitted parity bit and the time to switch SIO to input mode. This value should be smaller than ETIME.

SIMn +0058h Character to Character Guard Time Register **SIMIFN_SIM_CGTIME**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	CGTIME[7:0]																
Type	R/W																
Reset	2h																

CGTIME Defines the minimum interval between the leading edges of two consecutive characters in ETU unit.

In the same transmission direction, the minimum interval is (12 + CGTIME) ETU. In opposite transmission direction, the minimum interval is (12 + CGTIME + GTIME) ETU.

SIMn +0060h SIM Command Header Register: INS **SIMIFN_SIM_INS**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								INSD				SIMINS[7:0]				
Type								R/W				R/W				
Reset								0h				0h				

SIMINS This field should be identical to the INS instruction code. When writing to this register, the T = 0 controller will be activated and data transfer initiated.

INSD Instruction direction

- 0 T = 0 controller receives data from the SIM card.
- 1 T = 0 controller sends data to the SIM card.

SIMn +0064h SIM Command Header Register: P3 **SIMIFN_SIM_IMP3
(ICC_LEN)**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								SIMP3 [8]				SIMP3[7:0]				
Type								R				R/W				
Reset								0h				0h				

SIMP3 This field should be identical to the P3 instruction code. It should be written prior to the SIM_INS register. When the data transfer is being conducted, this field will show the number of the remaining data to be sent or to be received.

SIMn +0068h SIM Procedure Byte Register: SW1 **SIMIFN_SIM_SW1
(ICC_LEN)**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												SIMSW1[7:0]				
Type												R				
Reset												0h				

SIMSW1 This field holds the last received procedure byte for debugging. When the T0END interrupt occurs, it will keep the SW1 procedure byte.

SIMn +006Ch SIM Procedure Byte Register: SW2 **SIMIFN_SIM_SW2
(ICC_EDC)**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												SIMSW2[7:0]				

Type																			R
Reset																			0h

SIMSW2 This field holds the SW2 procedure byte

SIMn +0070h SIM ATR State Register **SIMIFN_SIM_ATRST**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	A
Name								AL	IR								OFF
Type								R	R								R
Reset								0h	0h								1h

The SIM card is initially turned off. After configuring SIMON of SIMn_SIM_CTRL and ATR procedure, SIMn_SIM_ATRSTA will set IR or AL to 1 to indicate the card's feature.

- OFF** Indicates On/Off of the SIM card
- IR** SIM card is IR (internal reset) card
- AL** SIM card is AL (active low reset) card

SIMn +0074h SIM Protocol State Register **SIMIFN_SIM_STATUS**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name											ALL	ONE					IDLE
Type											R	R					R
Reset											0h	0h					1h

T0 or T1 protocol of the SIM card is initially turned off. When T0 or T1 protocol is turned on, SIMn_SIM_T0STA will transit between ONE or ALL according to the procedure byte of the SIM card.

- IDLE** SIM card's T0 or T1 protocol is active or idle.
- ONE** SIM card will send the next byte
- ALL** SIM card will send all the remaining bytes.

SIMn +0080h Data Register Used As Tx/Rx Data Register **SIMIFN_SIM_DMADATA**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name												DATA[7:0]					
Type												R/W					
Reset												-					

DATA Eight data digits, corresponding to the character being read or written

SIMn +0090h SIM Module Debug Register **SIMIFN_SIM_DBG**

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	--

Name					DBG7	DBG6	DBG5	DBG4									DBG3[4:0]	
Type					R	R	R	R									R	
Reset					0h	0h	0h	0h									0h	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name					DBG2[4:0]													DBG1[4:0]
Type					R													R
Reset					0h													0h

- DBG1** Debugging register 1
- DBG2** Debugging register 2
- DBG3** Debugging register 3
- DBG4** Debugging register 4
- DBG5** Debugging register 5
- DBG6** Debugging register 6
- DBG7** Debugging register 7

SIMn +0094h SIM FIFO Data Debug Register **SIMIFN_SIM_DBGDATA**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DBGRPTR[3:0]								DBGDATA[7:0]			
Type					R								R			
Reset					0h								-			

- DBGDATA** FIFO data debugging register
There is no impact on data transmission when this register is read.
- DBGRPTR** FIFO read pointer related to DBGDATA
Automatically increases by 1 after this register is read.

SIMn +00A0h SIM SCLK PAD Control Register **SIMIFN_SIM_SCLK**

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															DEBU G	ACD_ FUN C
Type															R/W	R/W
Reset															0h	0h
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IES_ CTRL	IES_ L V			TDSEL[1:0]		RDSEL[1:0]		R1	R0	PUPD	SMT	E4	E2	SR[1:0]	
Type	R/W	R/W			R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0h	0h			0h		0h		0h	0h	0h	1h	0h	1h	3h	

- SR** Output slew rate control
High asserted. SR = 1, slower slew. SR = 0, no slew rate control.
For SIM card mode, SR[1:0] = [1 1] is the recommended setting to eliminate overshooting/undershooting. For non-SIM card mode, SR[1:0] = [0 0] is set for best speed.
- E2** TX driving strength control
For SIM card mode, E2 = [1] is the recommended setting for SCLK/SRST/SIO. (SIO/SRST can use [0])

- E4** TX driving strength control
For SIM card mode, E4 = [0] is the recommended setting for SCLK/SRST/SIO.
- SMT** RX input buffer schmit trigger hysteresis control enable
High asserted. SMT = 1, schmit trigger enable.
For SIM card mode, SMT = [1] is the recommended setting.
- PUPD** Weak pull-up/pull-down control
0 Pull-up
1 Pull-down
- R0** Weak pull-up/pull-down resistance select
Check the table in register "R1".
- R1** Weak pull-up/pull-down resistance select
Check the following table.

E	PUPD	R1	R0	R Value
0	0	0	0	High - Z
0	0	0	1	PU - 20k
0	0	1	0	PU - 5k
0	0	1	1	PU - 4k
0	1	0	0	High - Z
0	1	0	1	PD - 75k
0	1	1	0	PD - 75k
0	1	1	1	PD - 37.5k
1	x	x	x	High - Z

- RDSEL** Selects RX duty
RDSEL[0]: Input buffer duty high when asserted. (high pulse width adjustment)
RDSEL[1]: Input buffer duty low when asserted. (low pulse width adjustment)
For SIM card mode, RDSEL = [0 0] is the recommended setting.
- TDSEL** Selects TX duty
TDSEL[0]: Output level shifter duty high when asserted. (high pulse width adjustment)
TDSEL[1]: Output level shifter duty low when asserted. (low pulse width adjustment)
For SIM card mode, TDSEL = [0 0] is the recommended setting.
- IES_LV** Controls IES (RX input buffer enable) parking level in IES direct control mode
High asserted. Datapath: From IO to O. IES = 0, O = 0.
In quiescent mode, IES = 0 is suggested for power saving.
- IES_CTRL** Enables IES direct control mode
- ACD_FUNC** ACD function mode for analog designer
- DEBUG** Output PAD related signals for monitoring
0 Disable
1 Enable

SIMn +00A4h SIM SRST PAD Control Register

SIMIFN_SIM_SRST

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IES_CTRL	IES_LV			TDSEL[1:0]		RDSEL[1:0]		R1	R0	PUPD	SMT	E4	E2	SR[1:0]	
Type	R/W	R/W			R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0h	0h			0h		0h		0h	0h	0h	1h	0h	1h	3h	

- SR** Output slew rate control
High asserted. SR = 1, slower slew. SR = 0, no slew rate control.
For SIM card mode, SR[1:0] = [1 1] is the recommended setting to eliminate overshooting/undershooting. For non-SIM card mode, SR[1:0] = [0 0] is set for best speed.
- E2** TX driving strength control
For SIM card mode, E2 = [1] is the recommended setting for SCLK/SRST/SIO. (SIO/SRST can use [0])
- E4** TX driving strength control
For SIM card mode, E4 = [0] is the recommended setting for SCLK/SRST/SIO.
- SMT** RX input buffer schmit trigger hysteresis control enable
High asserted. SMT = 1, schmit trigger enable.
For SIM card mode, SMT = [1] is the recommended setting.
- PUPD** Weak pull-up/pull-down control
0 Pull-up
1 Pull-down
- R0** Weak pull-up/pull-down resistance select
Check the table in register "R1".
- R1** Weak pull-up/pull-down resistance select
Check the following table.

E	PUPD	R1	R0	R Value
0	0	0	0	High – Z
0	0	0	1	PU – 20k
0	0	1	0	PU – 5k
0	0	1	1	PU – 4k
0	1	0	0	High – Z
0	1	0	1	PD – 75k
0	1	1	0	PD – 75k
0	1	1	1	PD – 37.5k
1	x	x	x	High - Z

- RDSEL** Selects RX duty
RDSEL[0]: Input buffer duty high when asserted. (high pulse width adjustment)
RDSEL[1]: Input buffer duty low when asserted. (low pulse width adjustment)
For SIM card mode, RDSEL = [0 0] is the recommended setting.
- TDSEL** Selects TX duty
TDSEL[0]: Output level shifter duty high when asserted. (high pulse width adjustment)
TDSEL[1]: Output level shifter duty low when asserted. (low pulse width adjustment)

For SIM card mode, TDSEL = [0 0] is the recommended setting.

IES_LV Controls IES (RX input buffer enable) parking level in IES direct control mode
High asserted. Datapath: From IO to O. IES = 0, O = 0.

In quiescent mode, IES = 0 is suggested for power saving.

IES_CTRL Enables IES direct control mode

SIMn +00A8h SIM SIO PAD Control Register

SIMIFN_SIM_SIO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IES_CTRL	IES_LV			TDSEL[1:0]		RDSEL[1:0]		R1	R0	PUPD	SMT	E4	E2	SR[1:0]	
Type	R/W	R/W			R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0h	0h			0h		0h		1h	0h	0h	1h	0h	1h		3h

SR Output slew rate control

High asserted. SR = 1, slower slew. SR = 0, no slew rate control.

For SIM card mode, SR[1:0] = [1 1] is the recommended setting to eliminate overshooting/undershooting. For non-SIM card mode, SR[1:0] = [0 0] is set for best speed.

E2 TX driving strength control

For SIM card mode, E2 = [1] is the recommended setting for SCLK/SRST/SIO. (SIO/SRST can use [0])

E4 TX driving strength control

For SIM card mode, E4 = [0] is the recommended setting for SCLK/SRST/SIO.

SMT RX input buffer schmit trigger hysteresis control enable

High asserted. SMT = 1, schmit trigger enable.

For SIM card mode, SMT = [1] is the recommended setting.

PUPD Weak pull-up/pull-down control

0 Pull-up

1 Pull-down

R0 Weak pull-up/pull-down resistance select

Check the table in register "R1".

R1 Weak pull-up/pull-down resistance select

Check the following table.

For SIO, [R1 R0] = [1 0] is the recommended setting for 5k weak pull-up. In 4 SIM application and SIO is connected to external SIM switch, please disable pull-up resistance. ([R1 R0] = [0 0])

E	PUPD	R1	R0	R Value
0	0	0	0	High – Z
0	0	0	1	PU – 20k
0	0	1	0	PU – 5k
0	0	1	1	PU – 4k
0	1	0	0	High – Z
0	1	0	1	PD – 75k
0	1	1	0	PD – 75k

E	PUPD	R1	R0	R Value
0	1	1	1	PD – 37.5k
1	x	x	x	High - Z

- RDSEL** Selects RX duty
 RDSEL[0]: Input buffer duty high when asserted. (high pulse width adjustment)
 RDSEL[1]: Input buffer duty low when asserted. (low pulse width adjustment)
 For SIM card mode, RDSEL = [0 0] is the recommended setting.
- TDSEL** Selects TX duty
 TDSEL[0]: Output level shifter duty high when asserted. (high pulse width adjustment)
 TDSEL[1]: Output level shifter duty low when asserted. (low pulse width adjustment)
 For SIM card mode, TDSEL = [0 0] is the recommended setting.
- IES_LV** Controlling IES (RX input buffer enable) parking level in IES direct control mode
 High asserted. Datapath: From IO to O. IES = 0, O = 0.
 In quiescent mode, IES = 0 is suggested for power saving.
- IES_CTRL** Enables IES direct control mode

SIMn +00ACh SIM Monitor Register

SIMIFN_SIM_MON

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		MON1 2	MON1 1	MON1 0		MON9	MON8	MON7		MON6	MON5	MON4		MON3	MON2	MON1
Type		R	R	R		R	R	R		R	R	R		R	R	R
Reset		0h	0h	0h		0h	0h	0h		0h	0h	0h		0h	0h	0h

- MON1** Monitor signal 1
- MON2** Monitor signal 2
- MON3** Monitor signal 3
- MON4** Monitor signal 4
- MON5** Monitor signal 5
- MON6** Monitor signal 6
- MON7** Monitor signal 7
- MON8** Monitor signal 8
- MON9** Monitor signal 9
- MON10** Monitor signal 10
- MON11** Monitor signal 11
- MON12** Monitor signal 12

SIMn+00B0h SIM Test Select

SIMIFN_SIM_SEL

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SIMSEL
Type																R/W
Reset																3'b001

SIMSEL	Selects monitor SIMRST, SIMCLK, SIMIO input signal
001	SIMRST input is monitored.
010	SIMCLK input is monitored.
100	SIMSIO input is monitored.
Others	No meaning

3.2.2 SIM Card Insertion and Removal

The detection of physical connection to the SIM card and card removal can be done by the external interrupt controller or by GPIO.

3.2.3 Card activation and Deactivation

The card activation and deactivation sequence are both controlled by H/W. The MCU initiates the activation sequence by writing "1" to bit 0 of the SIM_CTRL register, and then the interface performs the following activation sequence:

- Assert SIMRST "low"
- Set SIMVCC at "high" level and SIMDATA in the reception mode
- Enable SIMCLK clock
- De-assert SIMRST "high" (required if it belongs to active low reset SIM card)

The final step in a typical card session is contacting deactivation in case the card will be electrically damaged. The deactivation sequence is initiated by writing "0" to bit 0 of the SIM_CTRL register, and the interface will perform the following deactivation sequence:

- Assert SIMRST "low"
- Set SCIMCLK at "low" level
- Set SIMDATA at "low" level
- Set SIMVCC at "low" level

3.2.4 Answering to Reset Sequence

After the card is activated, a reset operation will result in an answer from the card consisting of the initial character TS, followed by maximum 32 characters. The initial character TS provides a bit synchronization sequence and defines the conventions to interpret data bytes in all subsequent characters.

On reception of the first character, TS, the MCU should read this character, establish the respective required convention and re-program the related registers. These processes should be completed prior to the completion of reception of the next character. Next, the remainder of the ATR sequence will be received, read via the SIM_DATA in the selected convention and interpreted by the S/W.

The timing requirement and procedures for ATR sequence are handled by H/W and shall meet the requirement of ISO 7816-3, as shown in Figure 23.

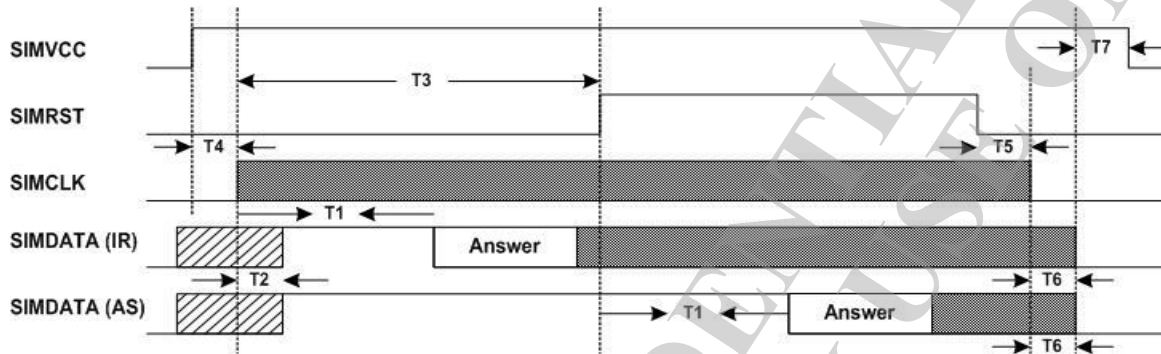


Figure 23. Answering to reset sequence

Table 43. Time-out condition for answering to reset sequence

Time	Value	Comment
T1	> 400 SIMCLK	SIMCLK start to ATR appears
T2	< 200 SIMCLK	SIMCLK start to SIMDATA in reception mode
T3	> 40,000 SIMCLK	SIMCLK start to SIMRST "high"
T4	-	SIMVCC "high" to SIMCLK start
T5	-	SIMRST "low" to SIMCLK stop
T6	-	SIMCLK stop to SIMDATA "low"
T7	-	SIMDATA "low" to SIMVCC "low"

3.2.5 SIM Data Transfer

There are two transfer modes provided, in software controlled byte by byte fashion or in a block fashion using T=0 controller and DMA controller. In both modes, the time-out counter can be enabled to monitor the elapsed time between two consecutive bytes.

3.2.5.1 Byte Transfer Mode

This mode is used during ATR and PPS procedure. In this mode, the SIM interface only ensures error free character transmission and reception.

Receiving characters

Upon detection of the start-bit sent by SIM card, the interface transforms into reception mode and the following bits are shifted into an internal register. If no parity error is detected or the character-received handshaking is disabled, the received-character will be written into the SIM FIFO and the SIM_COUNT register increased by one. Otherwise, the SIMDATA line will be held "low" at 0.5 etu after detecting the parity error for 1.5 etus, and the character will be re-received. If a character fails to be

received correctly for the RXRETRY times, the receive-handshaking will be aborted, the last-received character written into the SIM FIFO, the SIM_COUNT increased by one and the RXERR interrupt generated.

When the number of characters held in the received FIFO exceeds the level defined in the SIM_TIDE register, a RXTIDE interrupt will be generated. The number of characters held in the SIM FIFO can be determined by reading the SIM_COUNT register, and writing to this register will flush the SIM FIFO.

Sending characters

Characters that are to be sent to the card are first written into the SIM FIFO and then automatically transmitted to the card at timed intervals. If character-transmitted handshaking is enabled, the SIMDATA line will be sampled at 1 etu after the parity bit. If the card indicates that it does not receive the character correctly, the character will be re-transmitted for maximum of TXRETRY times before a TXERR interrupt is generated and the transmission is aborted. Otherwise, the succeeding byte in the SIM FIFO will be transmitted.

If a character fails to be transmitted and a TXERR interrupt is generated, the interface will need to be reset by flushing the SIM FIFO before any subsequent transmission or reception operation.

When the number of characters held in the SIM FIFO falls below the level defined in the SIM_TIDE register, a TXTIDE interrupt will be generated. The number of characters held in the SIM FIFO can be determined by reading the SIM_COUNT register, and writing to this register will flush the SIM FIFO.

3.2.5.2 Block Transfer Mode

Basically the SIM interface is designed to work in conjunction with the T=0 protocol controller and the DMA controller during non-ATR and non-PPS phase, though it is still possible for software to service the data transfer manually as in the byte transfer mode if necessary. Thus the T=0 protocol should be controlled by software.

The T=0 controller can be accessed via four registers representing the instruction header bytes INS and P3, and the procedure bytes SW1 and SW2. The registers are:

- SIM_INS, SIM_P3
- SIM_SW1, SIM_SW2

During the character transfer, SIM_P3 holds the number of characters to be sent or to be received, and SIM_SW1 holds the last received procedure byte including NULL, ACK, NACK and SW1 for debugging.

Data receiving instruction

Data receiving instructions receive data from the SIM card. See the following instantiated procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in the SIM_CONF register.
2. Program the SIM_TIDE register to 0x0000 (TXTIDE = 0, RXTIDE = 0).
3. Program the SIM_IRQEN to 0x019C (enable RXERR, TXERR, T0END, TOUT and OVRUN interrupts).
4. Write CLA, INS, P1, P2 and P3 into SIM FIFO.
5. Program the DMA controller:
 - DMA_n_MSBSRC and DMA_n_LSBSRC: Address of the SIM_DATA register
 - DMA_n_MSBDST and DMA_n_LSBDST: Memory address reserved to store the received characters
 - DMA_n_COUNT: Identical to P3 or 256 (if P3 = 0)
 - DMA_n_CON: 0x0078
6. Write P3 into the SIM_P3 register and then INS into SIM_INS register. (Data transfer is initiated now.)
7. Enable the time-out counter by setting the TOUT bit to 1 in the SIM_CONF register.
8. Start the DMA controller by writing 0x8000 into the DMA_n_START register.

Upon completion of the data receiving instruction, T0END interrupt will be generated and the time-out counter should be disabled by setting the TOUT bit to 0 in the SIM_CONF register.

If error occurs during the data transfer (RXERR, TXERR, OVRUN or TOUT interrupt is generated), the SIM card should be deactivated first and then activate the prior subsequent operations.

Data sending instruction

Data sending instructions send data to the SIM card. See the following instantiated procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in the SIM_CONF register.
2. Program the SIM_TIDE register to 0x0100 (TXTIDE = 1, RXTIDE = 0)
3. Program the SIM_IRQEN to 0x019C (enable RXERR, TXERR, T0END, TOUT and OVRUN interrupts)
4. Write CLA, INS, P1, P2 and P3 into SIM FIFO
5. Program the DMA controller:
 - DMA_n_MSBSRC and DMA_n_LSBSRC: Memory address reserved to store the transmitted characters
 - DMA_n_MSBDST and DMA_n_LSBDST: Address of the SIM_DATA register
 - DMA_n_COUNT: Identical to P3
 - DMA_n_CON: 0x0074
6. Write P3 into the SIM_P3 register and then (0x0100 | INS) into SIM_INS register. (Data transfer is initiated now.)
7. Enable the time-out counter by setting the TOUT bit to 1 in the SIM_CONF register.
8. Start the DMA controller by writing 0x8000 into the DMA_n_START register.

Upon completion of the data sending instruction, T0END interrupt will be generated and the time-out counter should be disabled by setting the TOUT bit back to 0 in the SIM_CONF register.

If error occurs during the data transfer (RXERR, TXERR, OVRUN or TOUT interrupt is generated), the SIM card should be deactivated first and then activate the prior subsequent operations.

3.3 Keypad Scanner

3.3.1 General Description

The keypad supports two types of keypads: 5*5 double keypad and 5*5 triple keypad.

The 5*5 keypad can be divided into two parts: 1) The keypad interface including 5 columns and 5 rows (see Figure 24 and Figure 25); 2) The key detection block providing key pressed, key released and de-bounce mechanisms.

Each time the key is pressed or released, i.e. something different in the 5*5 matrix, the key detection block senses the change and recognizes if a key has been pressed or released. Whenever the key status changes and is stable, a KEYPAD IRQ will be issued. The MCU can then read the key(s) pressed directly in the KP_MEM1, KP_MEM2, KP_MEM3, KP_MEM4 and KP_MEM5 registers. To ensure the key pressed information is not missed, the status register in keypad is not read-cleared by the APB read command. The status register can only be changed by the key-pressed detection FSM.

This keypad can detect one or two keys pressed simultaneously. Figure 27 shows the one key pressed condition. Figure 28(a) and Figure 28(b) illustrate the two keys pressed cases. Since the key pressed detection depends on the HIGH or LOW level of the external keypad interface, if the keys are pressed at the same time and there exists a key that is on the same column and the same row with other keys, the pressed key cannot be correctly decoded. For example, if there are three key pressed: key1 = (x1, y1), key2 = (x2, y2), and key3 = (x1, y2), then both key3 and key4 = (x2, y1) will be detected, and therefore they cannot be distinguished correctly. Hence, the keypad can detect only one or two keys pressed simultaneously at any combination. More than two keys pressed simultaneously in a specific pattern will retrieve the wrong information.

The 5*5 double keypad (Figure 24) supports a $5*5*2 = 50$ keys matrix. The 50 keys are divided into 25 sub groups and each group consists of 2 keys and a off-chip resistor. 5*5 double keypad has another limitation, which is it cannot detect two keys pressed simultaneously when the two keys are in one group, i.e. the 5*5 keypad cannot detect key0 and key1 pressed simultaneously or key15 and key16 pressed simultaneously.

The 5*5 triple keypad (Figure 25) supports a $5*5*3 = 75$ keys matrix. The 75 keys are divided into 25 sub groups and each group consists of 3 keys and two off-chip resistors. 5*5 triple keypad has another limitation, which is it cannot detect three keys pressed simultaneously when the three keys are in one group, i.e. 5*5 keypad cannot detect key0, key1 and key2 pressed simultaneously or key15, key16 and key17 pressed simultaneously.

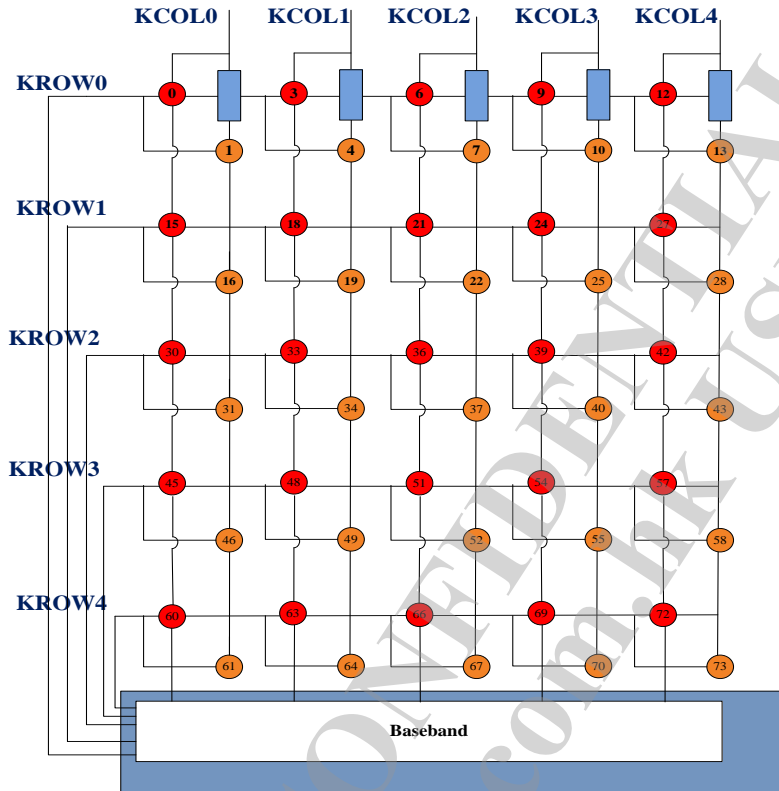


Figure 24. 5x5 double keypad matrix (50 keys)

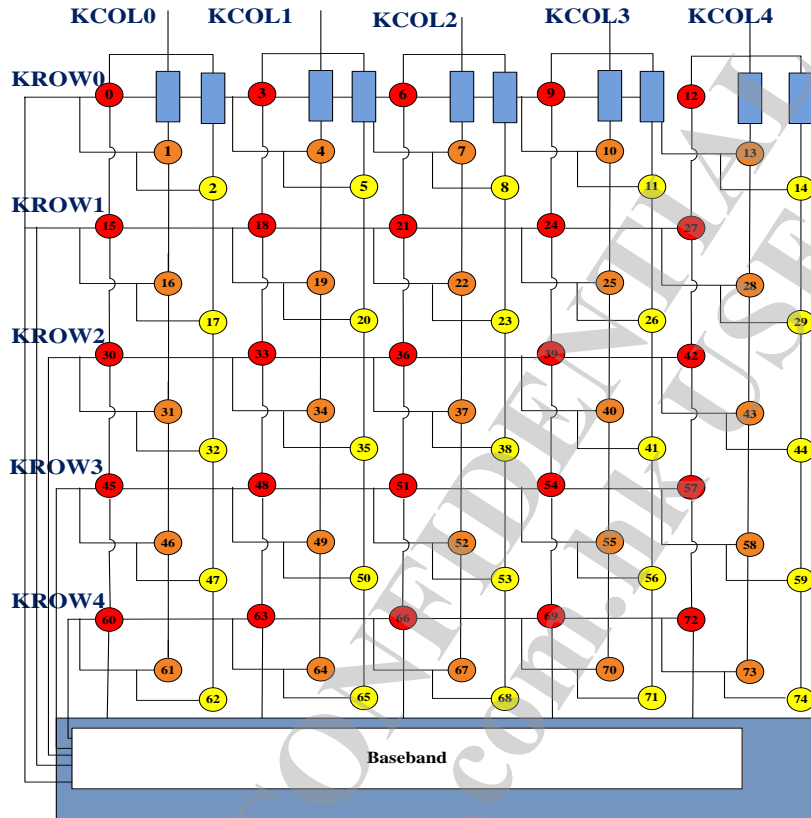


Figure 25. 5x5 triple keypad matrix (75 keys)

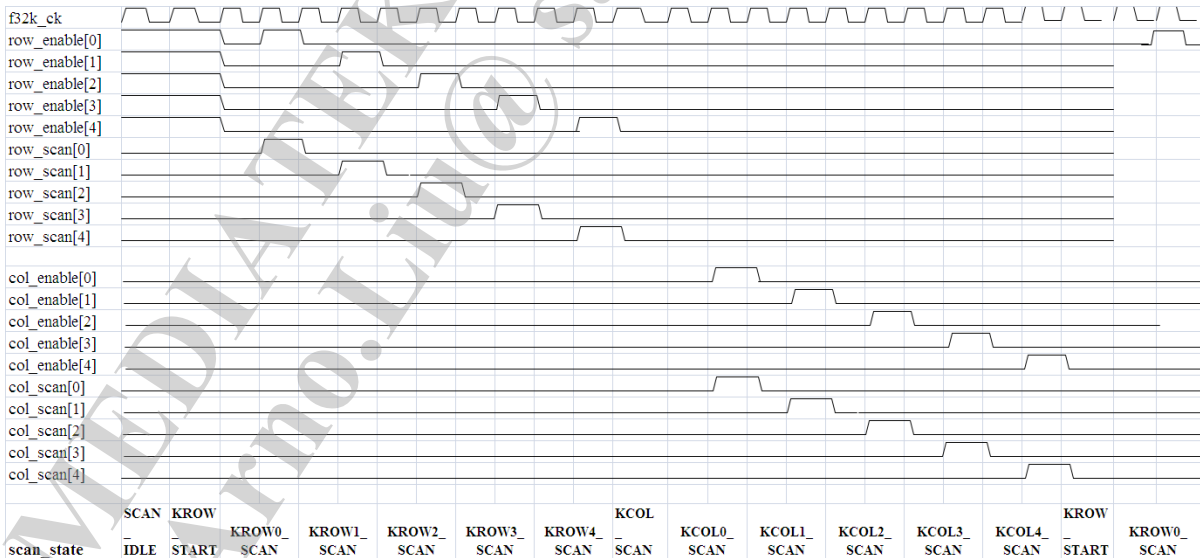


Figure 26. 5*5 double keypad scan waveform

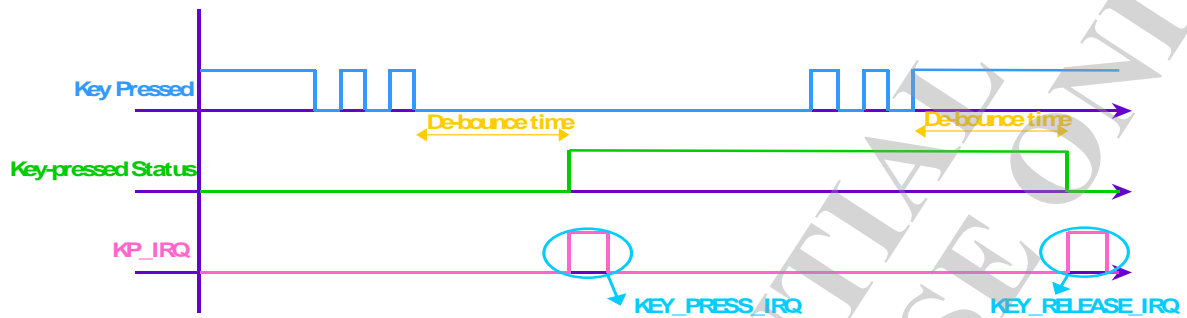


Figure 27. One key pressed with de-bounce mechanism denoted

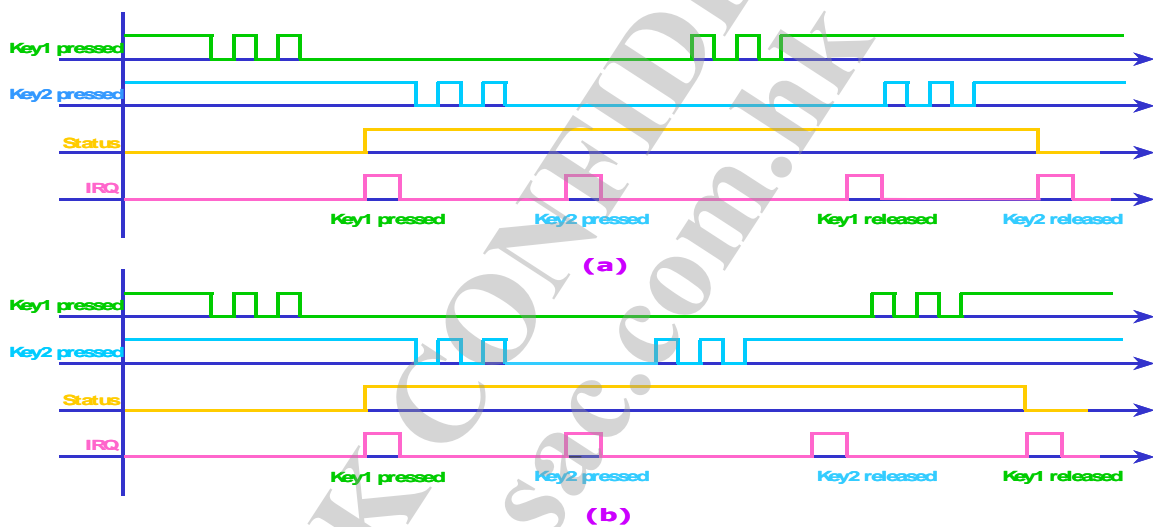


Figure 28. (a) Two keys pressed, case 1; (b) Two keys pressed, case 2

3.3.2 Register Definitions

Module name: KP Base address: (+A00D0000)

Address	Name	Width	Register Function
A00D0000	KP_STA	16	Keypad Status
A00D0004	KP_MEM1	16	Keypad Scanning Output Register Shows the key-pressed status of key 0 (LSB) ~ key 15. Refer to Table 1 and Table 2.
A00D0008	KP_MEM2	16	Keypad Scanning Output Register Shows the key-pressed status of key 16 (LSB) ~ key 31. Refer to Table 1 and Table 2.
A00D000C	KP_MEM3	16	Keypad Scanning Output Register Shows the key-pressed status of key 32 (LSB) ~ key 47. Refer to Table 1 and Table 2.
A00D0010	KP_MEM4	16	Keypad Scanning Output Register Shows the key-pressed status of key 48 (LSB) ~ key 63. Refer to Table 1 and Table 2.

Address	Name	Width	Register Function
A00D0014	KP_MEM5	16	Keypad Scanning Output Register Shows the key-pressed status of key 64 (LSB) ~ key 77. Refer to Table 1 and Table 2.
A00D0018	KP_DEBOUNCE	16	De-bounce Period Setting Defines the waiting period before key pressing or release events are considered stable. If the de-bounce setting is too small, the keypad will be too sensitive and detect too many unexpected key presses. The suitable de-bounce time setting must be adjusted according to the user's habit.
A00D001C	KP_SCAN_TIMING	16	Keypad Scan Timing Adjustment Register Sets up the keypad scan timing. <i>Note: ROW_SCAN_DIV > ROW_INTERVAL_DIV and COL_SCAN_DIV > COL_INTERVAL_DIV. ROW_INTERVAL_DIV/COL_INTERVAL_DIV are used to lower the power consumption for it decreases the actual scan number during the de-bounce time.</i>
A00D0020	KP_SEL	16	Keypad Selection Register Selects: 1: Use the double keypad or triple keypad 2: Which cols and rows are used
A00D0024	KP_EN	16	Keypad Enable Register Enables/Disables keypad.

A00D0000 [KP_STA](#) Keypad Status **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																STA
Type																RO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	STA	STA	Indicates the keypad status The register is not cleared by the read operation. 0: No key pressed 1: Key pressed

A00D0004 [KP_MEM1](#) Keypad Scanning Output Register **FFFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY15	KEY14	KEY13	KEY12	KEY11	KEY10	KEY9	KEY8	KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview: Shows the key-pressed status of key 0 (LSB) ~ key 15. Refer to Table 1 and Table 2.

Bit(s)	Mnemonic	Name	Description
15	KEY15	KEY15	
14	KEY14	KEY14	

Bit(s)	Mnemonic	Name	Description
13	KEY13	KEY13	
12	KEY12	KEY12	
11	KEY11	KEY11	
10	KEY10	KEY10	
9	KEY9	KEY9	
8	KEY8	KEY8	
7	KEY7	KEY7	
6	KEY6	KEY6	
5	KEY5	KEY5	
4	KEY4	KEY4	
3	KEY3	KEY3	
2	KEY2	KEY2	
1	KEY1	KEY1	
0	KEY0	KEY0	

A00D0008 **KP_MEM2** **Keypad Scanning Output Register** **FFFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY31	KEY30	KEY29	KEY28	KEY27	KEY26	KEY25	KEY24	KEY23	KEY22	KEY21	KEY20	KEY19	KEY18	KEY17	KEY16
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview: Shows the key-pressed status of key 16 (LSB) ~ key 31. Refer to Table 1 and Table 2.

Bit(s)	Mnemonic	Name	Description
15	KEY31	KEY31	
14	KEY30	KEY30	
13	KEY29	KEY29	
12	KEY28	KEY28	
11	KEY27	KEY27	
10	KEY26	KEY26	
9	KEY25	KEY25	
8	KEY24	KEY24	
7	KEY23	KEY23	
6	KEY22	KEY22	
5	KEY21	KEY21	
4	KEY20	KEY20	
3	KEY19	KEY19	
2	KEY18	KEY18	
1	KEY17	KEY17	
0	KEY16	KEY16	

A00D000C **KP_MEM3** **Keypad Scanning Output Register** **FFFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY47	KEY46	KEY45	KEY44	KEY43	KEY42	KEY41	KEY40	KEY39	KEY38	KEY37	KEY36	KEY35	KEY34	KEY33	KEY32
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview: Shows the key-pressed status of key 32 (LSB) ~ key 47. Refer to Table 1 and Table 2.

Bit(s)	Mnemonic	Name	Description
15	KEY47	KEY47	
14	KEY46	KEY46	
13	KEY45	KEY45	
12	KEY44	KEY44	
11	KEY43	KEY43	
10	KEY42	KEY42	
9	KEY41	KEY41	
8	KEY40	KEY40	
7	KEY39	KEY39	
6	KEY38	KEY38	
5	KEY37	KEY37	
4	KEY36	KEY36	
3	KEY35	KEY35	
2	KEY34	KEY34	
1	KEY33	KEY33	
0	KEY32	KEY32	

A00D0010 **KP_MEM4** **Keypad Scanning Output Register** **FFFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY63	KEY62	KEY61	KEY60	KEY59	KEY58	KEY57	KEY56	KEY55	KEY54	KEY53	KEY52	KEY51	KEY50	KEY49	KEY48
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview: Shows the key-pressed status of key 48 (LSB) ~ key 63. Refer to Table 1 and Table 2.

Bit(s)	Mnemonic	Name	Description
15	KEY63	KEY63	
14	KEY62	KEY62	
13	KEY61	KEY61	
12	KEY60	KEY60	
11	KEY59	KEY59	
10	KEY58	KEY58	
9	KEY57	KEY57	
8	KEY56	KEY56	
7	KEY55	KEY55	

Bit(s)	Mnemonic	Name	Description
6	KEY54	KEY54	
5	KEY53	KEY53	
4	KEY52	KEY52	
3	KEY51	KEY51	
2	KEY50	KEY50	
1	KEY49	KEY49	
0	KEY48	KEY48	

A00D0014 **KP_MEM5** **Keypad Scanning Output Register** **07FF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						KEY74	KEY73	KEY72	KEY71	KEY70	KEY69	KEY68	KEY67	KEY66	KEY65	KEY64
Type						RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset						1	1	1	1	1	1	1	1	1	1	1

Overview: Shows the key-pressed status of key 64 (LSB) ~ key 77. Refer to Table 1 and Table 2.

Bit(s)	Mnemonic	Name	Description
10	KEY74	KEY74	
9	KEY73	KEY73	
8	KEY72	KEY72	
7	KEY71	KEY71	
6	KEY70	KEY70	
5	KEY69	KEY69	
4	KEY68	KEY68	
3	KEY67	KEY67	
2	KEY66	KEY66	
1	KEY65	KEY65	
0	KEY64	KEY64	

The five registers list the status of 75 keys on the keypad. For 5*5 keypad, KP_MEM1~4 registers list the status of 75 keys on the keypad. When the MCU receives KEYPAD IRQ, both two registers must be read. If any key is pressed, the relative bit will be set to 0.

In order to work normally, the corresponding pull-up/down setting must be programmed correctly. If some keys can be used because their COL or ROW is used as GPIO, these corresponding enabling bit should be set.

KEYS Status list of the 75 keys.

A00D0018 **KP_DEBOUNCE** **De-bounce Period Setting** **0400**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Name	DEBOUNCE															
Type	RW															
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Defines the waiting period before key pressing or release events are considered stable. If the de-bounce setting is too small, the keypad will be too sensitive and detect too many unexpected key presses. The suitable de-bounce time setting must be adjusted according to the user's habit.

Bit(s)	Mnemonic	Name	Description
13:0	DEBOUNCE	DEBOUNCE	De-bounce time = KP_DEBOUNCE/32ms.

A00D001C [KP_SCAN_TIMING](#) Keypad Scan Timing Adjustment Register 0011

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COL_INTERVAL_DIV				ROW_INTERVAL_DIV				COL_SCAN_DIV				ROW_SCAN_DIV			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Overview: Sets up the keypad scan timing. Note: ROW_SCAN_DIV > ROW_INTERVAL_DIV and COL_SCAN_DIV > COL_INTERVAL_DIV. ROW_INTERVAL_DIV/COL_INTERVAL_DIV are used to lower the power consumption for it decreases the actual scan number during the de-bounce time.

Bit(s)	Mnemonic	Name	Description
15:12	COL_INTERVAL_DIV	COL_INTERVAL_DIV	Sets up the COL SCAN interval cycle, i.e. cycles between two scans Default 0 means there is 1 cycle between two high scan pulses.
11:8	ROW_INTERVAL_DIV	ROW_INTERVAL_DIV	Sets up the ROW SCAN interval cycle, i.e. cycles between two scans Default 0 means there is 1 cycle between two high scan pulses.
7:4	COL_SCAN_DIV	COL_SCAN_DIV	Sets up the COL SCAN cycle which includes COL_INTERVAL_DIV and the high pulse period Default 1 means there are 2 cycles for each scan, including 1 cycle high pulse and 1 cycle interval.
3:0	ROW_SCAN_DIV	ROW_SCAN_DIV	Sets up the ROW SCAN cycle which includes ROW_INTERVAL_DIV and the high pulse period Default 1 means there are 2 cycles for each scan, including 1 cycle high pulse and 1 cycle interval.

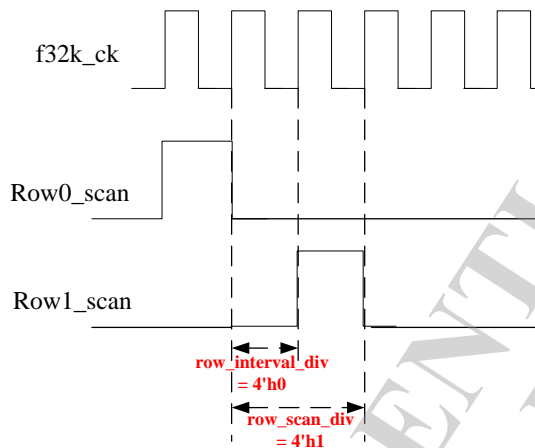


Figure 29. kp timing register

A00D0020 **KP_SEL** Keypad Selection Register FFC0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	KP1_COL_SEL					SAMPLE_DELAY										KP_SEL	
Type	RW					RW										RW	
Reset	1	1	1	1	1							0	0	0	0	0	0

Overview: Selects 1: Use the double keypad or triple keypad; 2: Which cols and rows are used.

Bit(s)	Mnemonic	Name	Description
15:11	KP1_COL_SEL	KP1_COL_SEL	Selects to use which col 0: Disable corresponding column 1: Enable corresponding column
5:1	SAMPLE_DELAY	SAMPLE_DELAY	Sets up delay cycles to sample col 0: No delay n: n*31.25ns delay to sample col
0	KP_SEL	KP_SEL	Selects to use double keypad or triple keypad 0: Use triple keypad 1: Use double keypad

A00D0024 **KP_EN** Keypad Enable Register 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																KP_EN
Type																RW
Reset																1

Overview: Enables/Disables keypad.

Bit(s)	Mnemonic	Name	Description
0	KP_EN	KP_EN	0: Disable keypad (Both double and triple keypads will not work.)

Bit(s)	Mnemonic	Name	Description
			1: Enable keypad (Only either of double or triple keypads can work.)

Table 44. 5*5 double KEY's order number in COL/ROW matrix

	COL0	COL1	COL2	COL3	COL4
ROW4	60/61	63/64	66/67	69/70	72/73
ROW3	45/46	48/49	51/52	54/55	57/58
ROW2	30/31	33/34	36/37	39/40	42/43
ROW1	15/16	18/19	21/22	24/25	27/28
ROW0	0/1	3/4	6/7	9/10	12/13

Table 45. 5*5 triple KEY's order number in COL/ROW matrix

	COL0	COL1	COL2	COL3	COL4
ROW4	60/61/62	63/64/65	66/67/68	69/70/71	72/73/74
ROW3	45/46/47	48/49/50	51/52/53	54/55/56	57/58/59
ROW2	30/31/32	33/34/35	36/37/38	39/40/41	42/43/44
ROW1	15/16/17	18/19/20	21/22/23	24/25/26	27/28/29
ROW0	0/1/2	3/4/5	6/7/8	9/10/11	12/13/14

3.4 General Purpose Inputs/Outputs

3.4.1 General Description

MT2503D offers 56 general purpose I/O pins. By setting up the control registers, the MCU software can control the direction, the output value, and read the input values on these pins. These GPIOs and GPOs are multiplexed with other functions to reduce the pin count. In addition, all GPO pins are removed. To facilitate application use, the software can configure which clock to send outside the chip. There are 6 clock-out ports embedded in 56 GPIO pins, and each clock-out can be programmed to output appropriate clock source. Besides, when 2 GPIO function for the same peripheral IP, the smaller GPIO serial numbers have higher priority than larger numbers.

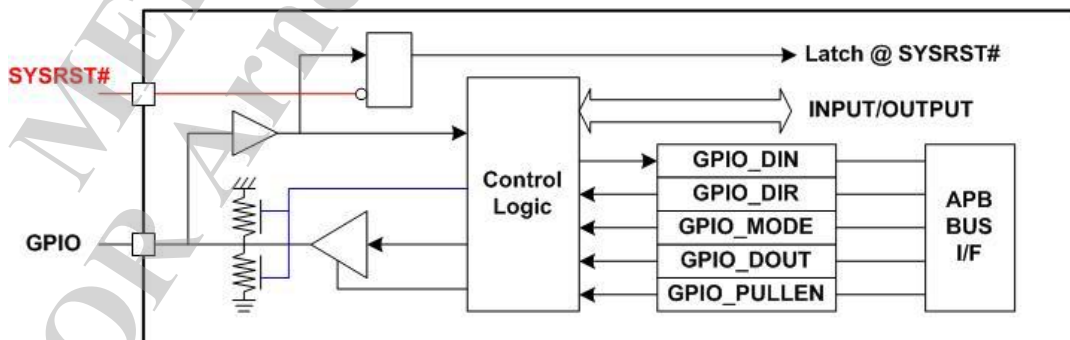


Figure 30. GPIO block diagram

3.4.2 Register Definitions

Module name: gpio_reg Base address: (+A0020000h)

Address	Name	Width	Register Function
A0020000	<u>GPIO_DIR0</u>	32	GPIO Direction Control Configures GPIO direction
A0020004	<u>GPIO_DIR0_SET</u>	32	GPIO Direction Control For bitwise access of GPIO_DIR0
A0020008	<u>GPIO_DIR0_CLR</u>	32	GPIO Direction Control For bitwise access of GPIO_DIR0
A0020010	<u>GPIO_DIR1</u>	32	GPIO Direction Control Configures GPIO direction
A0020014	<u>GPIO_DIR1_SET</u>	32	GPIO Direction Control For bitwise access of GPIO_DIR1
A0020018	<u>GPIO_DIR1_CLR</u>	32	GPIO Direction Control For bitwise access of GPIO_DIR1
A0020100	<u>GPIO_PULLEN0</u>	32	GPIO Pull-up/down Enable Control Configures GPIO pull enabling
A0020104	<u>GPIO_PULLEN0_SET</u>	32	GPIO Pull-up/down Enable Control For bitwise access of GPIO_PULLEN0
A0020108	<u>GPIO_PULLEN0_CLR</u>	32	GPIO Pull-up/down Enable Control For bitwise access of GPIO_PULLEN0
A0020110	<u>GPIO_PULLEN1</u>	32	GPIO Pull-up/down Enable Control Configures GPIO pull enabling
A0020114	<u>GPIO_PULLEN1_SET</u>	32	GPIO Pull-up/down Enable Control For bitwise access of GPIO_PULLEN1
A0020118	<u>GPIO_PULLEN1_CLR</u>	32	GPIO Pull-up/down Enable Control For bitwise access of GPIO_PULLEN1
A0020200	<u>GPIO_DINV0</u>	32	GPIO Data Inversion Control Configures GPIO inversion enabling
A0020204	<u>GPIO_DINV0_SET</u>	32	GPIO Data Inversion Control For bitwise access of GPIO_DINV0
A0020208	<u>GPIO_DINV0_CLR</u>	32	GPIO Data Inversion Control For bitwise access of GPIO_DINV0
A0020210	<u>GPIO_DINV1</u>	32	GPIO Data Inversion Control Configures GPIO inversion enabling
A0020214	<u>GPIO_DINV1_SET</u>	32	GPIO Data Inversion Control For bitwise access of GPIO_DINV1
A0020218	<u>GPIO_DINV1_CLR</u>	32	GPIO Data Inversion Control For bitwise access of GPIO_DINV1
A0020300	<u>GPIO_DOUT0</u>	32	GPIO Output Data Control Configures GPIO output value
A0020304	<u>GPIO_DOUT0_SET</u>	32	GPIO Output Data Control For bitwise access of GPIO_DIR0
A0020308	<u>GPIO_DOUT0_CLR</u>	32	GPIO Output Data Control For bitwise access of GPIO_DIR0
A0020310	<u>GPIO_DOUT1</u>	32	GPIO Output Data Control Configures GPIO output value

Address	Name	Width	Register Function
A0020314	<u>GPIO DOUT1 SE</u> <u>I</u>	32	GPIO Output Data Control For bitwise access of GPIO_DIR1
A0020318	<u>GPIO DOUT1 CL</u> <u>R</u>	32	GPIO Output Data Control For bitwise access of GPIO_DIR1
A0020400	<u>GPIO DIN0</u>	32	GPIO Input Data Value Reads GPIO input value
A0020410	<u>GPIO DIN1</u>	32	GPIO Input Data Value Reads GPIO input value
A0020500	<u>GPIO PULLSEL0</u>	32	GPIO Pullsel Control Configures GPIO PUPD selection
A0020504	<u>GPIO PULLSEL0</u> <u>SET</u>	32	GPIO Pullsel Control For bitwise access of GPIO_PULLSEL0
A0020508	<u>GPIO PULLSEL0</u> <u>CLR</u>	32	GPIO Pullsel Control For bitwise access of GPIO_PULLSEL0
A0020510	<u>GPIO PULLSEL1</u>	32	GPIO Pullsel Control Configures GPIO PUPD selection
A0020514	<u>GPIO PULLSEL1</u> <u>SET</u>	32	GPIO Pullsel Control For bitwise access of GPIO_PULLSEL1
A0020518	<u>GPIO PULLSEL1</u> <u>CLR</u>	32	GPIO Pullsel Control For bitwise access of GPIO_PULLSEL1
A0020600	<u>GPIO SMT0</u>	32	GPIO SMT Control Configures GPIO Schmit trigger control
A0020604	<u>GPIO SMT0 SET</u>	32	GPIO SMT Control For bitwise access of GPIO_SMT0
A0020608	<u>GPIO SMT0 CLR</u>	32	GPIO SMT Control For bitwise access of GPIO_SMT0
A0020610	<u>GPIO SMT1</u>	32	GPIO SMT Control Configures GPIO Schmit trigger control
A0020614	<u>GPIO SMT1 SET</u>	32	GPIO SMT Control For bitwise access of GPIO_SMT1
A0020618	<u>GPIO SMT1 CLR</u>	32	GPIO SMT Control For bitwise access of GPIO_SMT1
A0020700	<u>GPIO SR0</u>	32	GPIO SR Control Configures GPIO slew rate control
A0020704	<u>GPIO SR0 SET</u>	32	GPIO SR Control For bitwise access of GPIO_SR0
A0020708	<u>GPIO SR0 CLR</u>	32	GPIO SR Control For bitwise access of GPIO_SR0
A0020710	<u>GPIO SR1</u>	32	GPIO SR Control Configures GPIO slew rate control
A0020714	<u>GPIO SR1 SET</u>	32	GPIO SR Control For bitwise access of GPIO_SR1
A0020718	<u>GPIO SR1 CLR</u>	32	GPIO SR Control For bitwise access of GPIO_SR1
A0020720	<u>GPIO SIM SR</u>	32	GPIO SIM SR Control Configures GPIO slew rate control for SIM IO
A0020724	<u>GPIO SIM SR S</u> <u>ET</u>	32	GPIO SIM SR Control For bitwise access of GPIO_SIM_SR
A0020728	<u>GPIO SIM SR C</u> <u>LR</u>	32	GPIO SIM SR Control For bitwise access of GPIO_SIM_SR
A0020800	<u>GPIO DRV0</u>	32	GPIO DRV Control

Address	Name	Width	Register Function
			Configures GPIO driving control
A0020804	<u>GPIO_DRV0_SET</u>	32	GPIO DRV Control For bitwise access of GPIO_DRV0
A0020808	<u>GPIO_DRV0_CLR</u>	32	GPIO DRV Control For bitwise access of GPIO_DRV0
A0020810	<u>GPIO_DRV1</u>	32	GPIO DRV Control Configures GPIO driving control
A0020814	<u>GPIO_DRV1_SET</u>	32	GPIO DRV Control For bitwise access of GPIO_DRV1
A0020818	<u>GPIO_DRV1_CLR</u>	32	GPIO DRV Control For bitwise access of GPIO_DRV1
A0020900	<u>GPIO_IES0</u>	32	GPIO IES Control Configures GPIO input enabling control
A0020904	<u>GPIO_IES0_SET</u>	32	GPIO IES Control For bitwise access of GPIO_IES0
A0020908	<u>GPIO_IES0_CLR</u>	32	GPIO IES Control For bitwise access of GPIO_IES0
A0020910	<u>GPIO_IES1</u>	32	GPIO IES Control Configures GPIO input enabling control
A0020914	<u>GPIO_IES1_SET</u>	32	GPIO IES Control For bitwise access of GPIO_IES1
A0020918	<u>GPIO_IES1_CLR</u>	32	GPIO IES Control For bitwise access of GPIO_IES1
A0020A00	<u>GPIO_PUPD0</u>	32	GPIO PUPD Control Configures GPIO PUPD control
A0020A04	<u>GPIO_PUPD0_SE</u> <u>I</u>	32	GPIO PUPD Control For bitwise access of GPIO_PUPD0
A0020A08	<u>GPIO_PUPD0_CL</u> <u>R</u>	32	GPIO PUPD Control For bitwise access of GPIO_PUPD0
A0020A10	<u>GPIO_PUPD1</u>	32	GPIO PUPD Control Configures GPIO PUPD control
A0020A14	<u>GPIO_PUPD1_SE</u> <u>I</u>	32	GPIO PUPD Control For bitwise access of GPIO_PUPD1
A0020A18	<u>GPIO_PUPD1_CL</u> <u>R</u>	32	GPIO PUPD Control For bitwise access of GPIO_PUPD1
A0020B00	<u>GPIO_RESEN0_0</u>	32	GPIO R0 Control Configures GPIO R0 control
A0020B04	<u>GPIO_RESEN0_0</u> <u>_SET</u>	32	GPIO R0 Control For bitwise access of GPIO_RESEN0_0
A0020B08	<u>GPIO_RESEN0_0</u> <u>_CLR</u>	32	GPIO R0 Control For bitwise access of GPIO_RESEN0_0
A0020B10	<u>GPIO_RESEN0_1</u>	32	GPIO R0 Control Configures GPIO R0 control
A0020B14	<u>GPIO_RESEN0_1</u> <u>_SET</u>	32	GPIO R0 Control For bitwise access of GPIO_RESEN0_1
A0020B18	<u>GPIO_RESEN0_1</u> <u>_CLR</u>	32	GPIO R0 Control For bitwise access of GPIO_RESEN0_1
A0020B20	<u>GPIO_RESEN1_0</u>	32	GPIO R1 Control Configures GPIO R1 control
A0020B24	<u>GPIO_RESEN1_0</u> <u>_SET</u>	32	GPIO R1 Control For bitwise access of GPIO_RESEN1_0

Address	Name	Width	Register Function
A0020B28	<u>GPIO_RESEN1_0</u> <u>CLR</u>	32	GPIO R1 Control For bitwise access of GPIO_RESEN1_0
A0020B30	<u>GPIO_RESEN1_1</u>	32	GPIO R1 Control Configures GPIO R1 control
A0020B34	<u>GPIO_RESEN1_1</u> <u>SET</u>	32	GPIO R1 Control For bitwise access of GPIO_RESEN1_1
A0020B38	<u>GPIO_RESEN1_1</u> <u>CLR</u>	32	GPIO R1 Control For bitwise access of GPIO_RESEN1_1
A0020C00	<u>GPIO_MODE0</u>	32	GPIO Mode Control Configures GPIO aux. mode
A0020C04	<u>GPIO_MODE0_S</u> <u>ET</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE0
A0020C08	<u>GPIO_MODE0_C</u> <u>LR</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE0
A0020C10	<u>GPIO_MODE1</u>	32	GPIO Mode Control Configures GPIO aux. mode
A0020C14	<u>GPIO_MODE1_S</u> <u>ET</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE1
A0020C18	<u>GPIO_MODE1_C</u> <u>LR</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE1
A0020C20	<u>GPIO_MODE2</u>	32	GPIO Mode Control Configures GPIO aux. mode
A0020C24	<u>GPIO_MODE2_S</u> <u>ET</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE2
A0020C28	<u>GPIO_MODE2_C</u> <u>LR</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE2
A0020C30	<u>GPIO_MODE3</u>	32	GPIO Mode Control Configures GPIO aux. mode
A0020C34	<u>GPIO_MODE3_S</u> <u>ET</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE3
A0020C38	<u>GPIO_MODE3_C</u> <u>LR</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE3
A0020C40	<u>GPIO_MODE4</u>	32	GPIO Mode Control Configures GPIO aux. mode
A0020C44	<u>GPIO_MODE4_S</u> <u>ET</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE4
A0020C48	<u>GPIO_MODE4_C</u> <u>LR</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE4
A0020C50	<u>GPIO_MODE5</u>	32	GPIO Mode Control Configures GPIO aux. mode
A0020C54	<u>GPIO_MODE5_S</u> <u>ET</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE5
A0020C58	<u>GPIO_MODE5_C</u> <u>LR</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE5
A0020C60	<u>GPIO_MODE6</u>	32	GPIO Mode Control Configures GPIO aux. mode
A0020C64	<u>GPIO_MODE6_S</u> <u>ET</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE6
A0020C68	<u>GPIO_MODE6_C</u> <u>LR</u>	32	GPIO Mode Control For bitwise access of GPIO_MODE6
A0020D10	<u>GPIO_TDSEL</u>	32	GPIO TDSEL Control

Address	Name	Width	Register Function
			GPIO TX duty control register
A0020D14	GPIO_TDSEL_SE I	32	GPIO TDSEL Control For bitwise access of GPIO_TDSEL
A0020D18	GPIO_TDSEL_CL R	32	GPIO TDSEL Control For bitwise access of GPIO_TDSEL
A0020E00	CLK_OUT0	32	CLK Out Selection Control CLK OUT0 Setting
A0020E10	CLK_OUT1	32	CLK Out Selection Control CLK OUT1 Setting
A0020E20	CLK_OUT2	32	CLK Out Selection Control CLK OUT2 Setting
A0020E30	CLK_OUT3	32	CLK Out Selection Control CLK OUT3 Setting
A0020E40	CLK_OUT4	32	CLK Out Selection Control CLK OUT4 Setting
A0020E50	CLK_OUT5	32	CLK Out Selection Control CLK OUT5 Setting

A0020000 GPIO_DIR0 GPIO Direction Control 040008E0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne	GPIO3	GPIO3	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO1	GPIO1	GPIO1	GPIO1
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0

Overview: Configures GPIO direction

Bit(s)	Mnemonic	Name	Description
31	GPIO31	GPIO31_DIR	GPIO31 direction control 0: GPIO as input 1: GPIO as output
30	GPIO30	GPIO30_DIR	GPIO30 direction control 0: GPIO as input 1: GPIO as output
29	GPIO29	GPIO29_DIR	GPIO29 direction control 0: GPIO as input 1: GPIO as output
28	GPIO28	GPIO28_DIR	GPIO28 direction control 0: GPIO as input 1: GPIO as output
27	GPIO27	GPIO27_DIR	GPIO27 direction control 0: GPIO as input 1: GPIO as output
26	GPIO26	GPIO26_DIR	GPIO26 direction control 0: GPIO as input 1: GPIO as output

Bit(s)	Mnemonic	Name	Description
25	GPIO25	GPIO25_DIR	1: GPIO as output GPIO25 direction control 0: GPIO as output 1: GPIO as output
24	GPIO24	GPIO24_DIR	0: GPIO as output 1: GPIO as output GPIO24 direction control 0: GPIO as output 1: GPIO as output
23	GPIO23	GPIO23_DIR	0: GPIO as output 1: GPIO as output GPIO23 direction control 0: GPIO as output 1: GPIO as output
22	GPIO22	GPIO22_DIR	0: GPIO as output 1: GPIO as output GPIO22 direction control 0: GPIO as output 1: GPIO as output
21	GPIO21	GPIO21_DIR	0: GPIO as output 1: GPIO as output GPIO21 direction control 0: GPIO as output 1: GPIO as output
20	GPIO20	GPIO20_DIR	0: GPIO as output 1: GPIO as output GPIO20 direction control 0: GPIO as output 1: GPIO as output
19	GPIO19	GPIO19_DIR	0: GPIO as output 1: GPIO as output GPIO19 direction control 0: GPIO as output 1: GPIO as output
18	GPIO18	GPIO18_DIR	0: GPIO as output 1: GPIO as output GPIO18 direction control 0: GPIO as output 1: GPIO as output
17	GPIO17	GPIO17_DIR	0: GPIO as output 1: GPIO as output GPIO17 direction control 0: GPIO as output 1: GPIO as output
16	GPIO16	GPIO16_DIR	0: GPIO as output 1: GPIO as output GPIO16 direction control 0: GPIO as output 1: GPIO as output
15	GPIO15	GPIO15_DIR	0: GPIO as output 1: GPIO as output GPIO15 direction control 0: GPIO as output 1: GPIO as output
14	GPIO14	GPIO14_DIR	0: GPIO as output 1: GPIO as output GPIO14 direction control 0: GPIO as output 1: GPIO as output
13	GPIO13	GPIO13_DIR	0: GPIO as output 1: GPIO as output GPIO13 direction control 0: GPIO as output 1: GPIO as output
12	GPIO12	GPIO12_DIR	0: GPIO as output 1: GPIO as output GPIO12 direction control 0: GPIO as output 1: GPIO as output
11	GPIO11	GPIO11_DIR	0: GPIO as output 1: GPIO as output GPIO11 direction control 0: GPIO as output 1: GPIO as output
10	GPIO10	GPIO10_DIR	0: GPIO as output 1: GPIO as output GPIO10 direction control 0: GPIO as output 1: GPIO as output
9	GPIO9	GPIO9_DIR	0: GPIO as output 1: GPIO as output GPIO9 direction control 0: GPIO as output 1: GPIO as output

Bit(s)	Mnemonic	Name	Description
8	GPIO8	GPIO8_DIR	GPIO8 direction control 0: GPIO as input 1: GPIO as output
7	GPIO7	GPIO7_DIR	GPIO7 direction control 0: GPIO as input 1: GPIO as output
6	GPIO6	GPIO6_DIR	GPIO6 direction control 0: GPIO as input 1: GPIO as output
5	GPIO5	GPIO5_DIR	GPIO5 direction control 0: GPIO as input 1: GPIO as output
4	GPIO4	GPIO4_DIR	GPIO4 direction control 0: GPIO as input 1: GPIO as output
3	GPIO3	GPIO3_DIR	GPIO3 direction control 0: GPIO as input 1: GPIO as output
2	GPIO2	GPIO2_DIR	GPIO2 direction control 0: GPIO as input 1: GPIO as output
1	GPIO1	GPIO1_DIR	GPIO1 direction control 0: GPIO as input 1: GPIO as output
0	GPIO0	GPIO0_DIR	GPIO0 direction control 0: GPIO as input 1: GPIO as output

A0020004 GPIO DIR0 SET GPIO Direction Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR0

Bit(s)	Mnemonic	Name	Description
31	GPIO31	GPIO31_DIR	Bitwise SET operation of GPIO31 direction 0: Keep 1: SET bits
30	GPIO30	GPIO30_DIR	Bitwise SET operation of GPIO30 direction 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
29	GPIO29	GPIO29_DIR	Bitwise SET operation of GPIO29 direction 0: 1: SET bits	Keep
28	GPIO28	GPIO28_DIR	Bitwise SET operation of GPIO28 direction 0: 1: SET bits	Keep
27	GPIO27	GPIO27_DIR	Bitwise SET operation of GPIO27 direction 0: 1: SET bits	Keep
26	GPIO26	GPIO26_DIR	Bitwise SET operation of GPIO26 direction 0: 1: SET bits	Keep
25	GPIO25	GPIO25_DIR	Bitwise SET operation of GPIO25 direction 0: 1: SET bits	Keep
24	GPIO24	GPIO24_DIR	Bitwise SET operation of GPIO24 direction 0: 1: SET bits	Keep
23	GPIO23	GPIO23_DIR	Bitwise SET operation of GPIO23 direction 0: 1: SET bits	Keep
22	GPIO22	GPIO22_DIR	Bitwise SET operation of GPIO22 direction 0: 1: SET bits	Keep
21	GPIO21	GPIO21_DIR	Bitwise SET operation of GPIO21 direction 0: 1: SET bits	Keep
20	GPIO20	GPIO20_DIR	Bitwise SET operation of GPIO20 direction 0: 1: SET bits	Keep
19	GPIO19	GPIO19_DIR	Bitwise SET operation of GPIO19 direction 0: 1: SET bits	Keep
18	GPIO18	GPIO18_DIR	Bitwise SET operation of GPIO18 direction 0: 1: SET bits	Keep
17	GPIO17	GPIO17_DIR	Bitwise SET operation of GPIO17 direction 0: 1: SET bits	Keep
16	GPIO16	GPIO16_DIR	Bitwise SET operation of GPIO16 direction 0: 1: SET bits	Keep
15	GPIO15	GPIO15_DIR	Bitwise SET operation of GPIO15 direction 0: 1: SET bits	Keep
14	GPIO14	GPIO14_DIR	Bitwise SET operation of GPIO14 direction 0: 1: SET bits	Keep
13	GPIO13	GPIO13_DIR	Bitwise SET operation of GPIO13 direction 0: 1: SET bits	Keep
12	GPIO12	GPIO12_DIR	Bitwise SET operation of GPIO12 direction	

Bit(s)	Mnemonic	Name	Description	
			0: 1: SET bits	Keep
11	GPIO11	GPIO11_DIR	Bitwise SET operation of GPIO11 direction 0: 1: SET bits	Keep
10	GPIO10	GPIO10_DIR	Bitwise SET operation of GPIO10 direction 0: 1: SET bits	Keep
9	GPIO9	GPIO9_DIR	Bitwise SET operation of GPIO9 direction 0: 1: SET bits	Keep
8	GPIO8	GPIO8_DIR	Bitwise SET operation of GPIO8 direction 0: 1: SET bits	Keep
7	GPIO7	GPIO7_DIR	Bitwise SET operation of GPIO7 direction 0: 1: SET bits	Keep
6	GPIO6	GPIO6_DIR	Bitwise SET operation of GPIO6 direction 0: 1: SET bits	Keep
5	GPIO5	GPIO5_DIR	Bitwise SET operation of GPIO5 direction 0: 1: SET bits	Keep
4	GPIO4	GPIO4_DIR	Bitwise SET operation of GPIO4 direction 0: 1: SET bits	Keep
3	GPIO3	GPIO3_DIR	Bitwise SET operation of GPIO3 direction 0: 1: SET bits	Keep
2	GPIO2	GPIO2_DIR	Bitwise SET operation of GPIO2 direction 0: 1: SET bits	Keep
1	GPIO1	GPIO1_DIR	Bitwise SET operation of GPIO1 direction 0: 1: SET bits	Keep
0	GPIO0	GPIO0_DIR	Bitwise SET operation of GPIO0 direction 0: 1: SET bits	Keep

A0020008 GPIO_DIR0_CLR GPIO Direction Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne	GPIO3	GPIO3	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO1	GPIO1	GPIO1	GPIO1
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO0	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_DIR	Bitwise CLR operation of GPIO31 direction 0: 1: CLR bits	Keep
30	GPIO30	GPIO30_DIR	Bitwise CLR operation of GPIO30 direction 0: 1: CLR bits	Keep
29	GPIO29	GPIO29_DIR	Bitwise CLR operation of GPIO29 direction 0: 1: CLR bits	Keep
28	GPIO28	GPIO28_DIR	Bitwise CLR operation of GPIO28 direction 0: 1: CLR bits	Keep
27	GPIO27	GPIO27_DIR	Bitwise CLR operation of GPIO27 direction 0: 1: CLR bits	Keep
26	GPIO26	GPIO26_DIR	Bitwise CLR operation of GPIO26 direction 0: 1: CLR bits	Keep
25	GPIO25	GPIO25_DIR	Bitwise CLR operation of GPIO25 direction 0: 1: CLR bits	Keep
24	GPIO24	GPIO24_DIR	Bitwise CLR operation of GPIO24 direction 0: 1: CLR bits	Keep
23	GPIO23	GPIO23_DIR	Bitwise CLR operation of GPIO23 direction 0: 1: CLR bits	Keep
22	GPIO22	GPIO22_DIR	Bitwise CLR operation of GPIO22 direction 0: 1: CLR bits	Keep
21	GPIO21	GPIO21_DIR	Bitwise CLR operation of GPIO21 direction 0: 1: CLR bits	Keep
20	GPIO20	GPIO20_DIR	Bitwise CLR operation of GPIO20 direction 0: 1: CLR bits	Keep
19	GPIO19	GPIO19_DIR	Bitwise CLR operation of GPIO19 direction 0: 1: CLR bits	Keep
18	GPIO18	GPIO18_DIR	Bitwise CLR operation of GPIO18 direction 0: 1: CLR bits	Keep
17	GPIO17	GPIO17_DIR	Bitwise CLR operation of GPIO17 direction 0: 1: CLR bits	Keep
16	GPIO16	GPIO16_DIR	Bitwise CLR operation of GPIO16 direction 0: 1: CLR bits	Keep
15	GPIO15	GPIO15_DIR	Bitwise CLR operation of GPIO15 direction 0:	Keep

Bit(s)	Mnemonic	Name	Description	
14	GPIO14	GPIO14_DIR	Bitwise CLR operation of GPIO14 direction 1: CLR bits 0:	Keep
13	GPIO13	GPIO13_DIR	Bitwise CLR operation of GPIO13 direction 1: CLR bits 0:	Keep
12	GPIO12	GPIO12_DIR	Bitwise CLR operation of GPIO12 direction 1: CLR bits 0:	Keep
11	GPIO11	GPIO11_DIR	Bitwise CLR operation of GPIO11 direction 1: CLR bits 0:	Keep
10	GPIO10	GPIO10_DIR	Bitwise CLR operation of GPIO10 direction 1: CLR bits 0:	Keep
9	GPIO9	GPIO9_DIR	Bitwise CLR operation of GPIO9 direction 1: CLR bits 0:	Keep
8	GPIO8	GPIO8_DIR	Bitwise CLR operation of GPIO8 direction 1: CLR bits 0:	Keep
7	GPIO7	GPIO7_DIR	Bitwise CLR operation of GPIO7 direction 1: CLR bits 0:	Keep
6	GPIO6	GPIO6_DIR	Bitwise CLR operation of GPIO6 direction 1: CLR bits 0:	Keep
5	GPIO5	GPIO5_DIR	Bitwise CLR operation of GPIO5 direction 1: CLR bits 0:	Keep
4	GPIO4	GPIO4_DIR	Bitwise CLR operation of GPIO4 direction 1: CLR bits 0:	Keep
3	GPIO3	GPIO3_DIR	Bitwise CLR operation of GPIO3 direction 1: CLR bits 0:	Keep
2	GPIO2	GPIO2_DIR	Bitwise CLR operation of GPIO2 direction 1: CLR bits 0:	Keep
1	GPIO1	GPIO1_DIR	Bitwise CLR operation of GPIO1 direction 1: CLR bits 0:	Keep
0	GPIO0	GPIO0_DIR	Bitwise CLR operation of GPIO0 direction 1: CLR bits 0:	Keep

A0020010	GPIO_DIR1	GPIO Direction Control	00004000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5	GPIO5			GPIO5	GPIO5	GPIO4	GPIO4

Type									5	4			1	0	9	8
Reset									RW	RW			RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO direction

Bit(s)	Mnemonic	Name	Description
23	GPIO55	GPIO55_DIR	GPIO55 direction control 0: GPIO as input 1: GPIO as output
22	GPIO54	GPIO54_DIR	GPIO54 direction control 0: GPIO as input 1: GPIO as output
19	GPIO51	GPIO51_DIR	GPIO51 direction control 0: GPIO as input 1: GPIO as output
18	GPIO50	GPIO50_DIR	GPIO50 direction control 0: GPIO as input 1: GPIO as output
17	GPIO49	GPIO49_DIR	GPIO49 direction control 0: GPIO as input 1: GPIO as output
16	GPIO48	GPIO48_DIR	GPIO48 direction control 0: GPIO as input 1: GPIO as output
15	GPIO47	GPIO47_DIR	GPIO47 direction control 0: GPIO as input 1: GPIO as output
14	GPIO46	GPIO46_DIR	GPIO46 direction control 0: GPIO as input 1: GPIO as output
13	GPIO45	GPIO45_DIR	GPIO45 direction control 0: GPIO as input 1: GPIO as output
12	GPIO44	GPIO44_DIR	GPIO44 direction control 0: GPIO as input 1: GPIO as output
11	GPIO43	GPIO43_DIR	GPIO43 direction control 0: GPIO as input 1: GPIO as output
10	GPIO42	GPIO42_DIR	GPIO42 direction control 0: GPIO as input 1: GPIO as output
9	GPIO41	GPIO41_DIR	GPIO41 direction control 0: GPIO as input 1: GPIO as output
8	GPIO40	GPIO40_DIR	GPIO40 direction control 0: GPIO as input 1: GPIO as output

Bit(s)	Mnemonic	Name	Description
7	GPIO39	GPIO39_DIR	GPIO39 direction control 0: GPIO as output 1: GPIO as output
6	GPIO38	GPIO38_DIR	GPIO38 direction control 0: GPIO as output 1: GPIO as output
5	GPIO37	GPIO37_DIR	GPIO37 direction control 0: GPIO as output 1: GPIO as output
4	GPIO36	GPIO36_DIR	GPIO36 direction control 0: GPIO as output 1: GPIO as output
3	GPIO35	GPIO35_DIR	GPIO35 direction control 0: GPIO as output 1: GPIO as output
2	GPIO34	GPIO34_DIR	GPIO34 direction control 0: GPIO as output 1: GPIO as output
1	GPIO33	GPIO33_DIR	GPIO33 direction control 0: GPIO as output 1: GPIO as output
0	GPIO32	GPIO32_DIR	GPIO32 direction control 0: GPIO as output 1: GPIO as output

A0020014 GPIO DIR1 SE GPIO Direction Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5	GPIO5			GPIO5	GPIO5	GPIO4	GPIO4
Type									WO	WO			WO	WO	WO	WO
Reset									0	0			0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR1

Bit(s)	Mnemonic	Name	Description
23	GPIO55	GPIO55_DIR	Bitwise SET operation of GPIO55 direction 0: Keep 1: SET bits
22	GPIO54	GPIO54_DIR	Bitwise SET operation of GPIO54 direction 0: Keep 1: SET bits
19	GPIO51	GPIO51_DIR	Bitwise SET operation of GPIO51 direction 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
18	GPIO50	GPIO50_DIR	Bitwise SET operation of GPIO50 direction 0: 1: SET bits	Keep
17	GPIO49	GPIO49_DIR	Bitwise SET operation of GPIO49 direction 0: 1: SET bits	Keep
16	GPIO48	GPIO48_DIR	Bitwise SET operation of GPIO48 direction 0: 1: SET bits	Keep
15	GPIO47	GPIO47_DIR	Bitwise SET operation of GPIO47 direction 0: 1: SET bits	Keep
14	GPIO46	GPIO46_DIR	Bitwise SET operation of GPIO46 direction 0: 1: SET bits	Keep
13	GPIO45	GPIO45_DIR	Bitwise SET operation of GPIO45 direction 0: 1: SET bits	Keep
12	GPIO44	GPIO44_DIR	Bitwise SET operation of GPIO44 direction 0: 1: SET bits	Keep
11	GPIO43	GPIO43_DIR	Bitwise SET operation of GPIO43 direction 0: 1: SET bits	Keep
10	GPIO42	GPIO42_DIR	Bitwise SET operation of GPIO42 direction 0: 1: SET bits	Keep
9	GPIO41	GPIO41_DIR	Bitwise SET operation of GPIO41 direction 0: 1: SET bits	Keep
8	GPIO40	GPIO40_DIR	Bitwise SET operation of GPIO40 direction 0: 1: SET bits	Keep
7	GPIO39	GPIO39_DIR	Bitwise SET operation of GPIO39 direction 0: 1: SET bits	Keep
6	GPIO38	GPIO38_DIR	Bitwise SET operation of GPIO38 direction 0: 1: SET bits	Keep
5	GPIO37	GPIO37_DIR	Bitwise SET operation of GPIO37 direction 0: 1: SET bits	Keep
4	GPIO36	GPIO36_DIR	Bitwise SET operation of GPIO36 direction 0: 1: SET bits	Keep
3	GPIO35	GPIO35_DIR	Bitwise SET operation of GPIO35 direction 0: 1: SET bits	Keep
2	GPIO34	GPIO34_DIR	Bitwise SET operation of GPIO34 direction 0: 1: SET bits	Keep
1	GPIO33	GPIO33_DIR	Bitwise SET operation of GPIO33 direction	

Bit(s)	Mnemonic	Name	Description
0	GPIO32	GPIO32_DIR	0: Keep 1: SET bits Bitwise SET operation of GPIO32 direction 0: Keep 1: SET bits

A0020018 GPIO_DIR1_CLR GPIO Direction Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO55	GPIO54			GPIO51	GPIO50	GPIO49	GPIO48
Type									WO	WO			WO	WO	WO	WO
Reset									0	0			0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR1

Bit(s)	Mnemonic	Name	Description
23	GPIO55	GPIO55_DIR	Bitwise CLR operation of GPIO55 direction 0: Keep 1: CLR bits
22	GPIO54	GPIO54_DIR	Bitwise CLR operation of GPIO54 direction 0: Keep 1: CLR bits
19	GPIO51	GPIO51_DIR	Bitwise CLR operation of GPIO51 direction 0: Keep 1: CLR bits
18	GPIO50	GPIO50_DIR	Bitwise CLR operation of GPIO50 direction 0: Keep 1: CLR bits
17	GPIO49	GPIO49_DIR	Bitwise CLR operation of GPIO49 direction 0: Keep 1: CLR bits
16	GPIO48	GPIO48_DIR	Bitwise CLR operation of GPIO48 direction 0: Keep 1: CLR bits
15	GPIO47	GPIO47_DIR	Bitwise CLR operation of GPIO47 direction 0: Keep 1: CLR bits
14	GPIO46	GPIO46_DIR	Bitwise CLR operation of GPIO46 direction 0: Keep 1: CLR bits
13	GPIO45	GPIO45_DIR	Bitwise CLR operation of GPIO45 direction 0: Keep 1: CLR bits
12	GPIO44	GPIO44_DIR	Bitwise CLR operation of GPIO44 direction 0: Keep

Bit(s)	Mnemonic	Name	Description	
11	GPIO43	GPIO43_DIR	1: CLR bits Bitwise CLR operation of GPIO43 direction 0: 1: CLR bits	Keep
10	GPIO42	GPIO42_DIR	1: CLR bits Bitwise CLR operation of GPIO42 direction 0: 1: CLR bits	Keep
9	GPIO41	GPIO41_DIR	1: CLR bits Bitwise CLR operation of GPIO41 direction 0: 1: CLR bits	Keep
8	GPIO40	GPIO40_DIR	1: CLR bits Bitwise CLR operation of GPIO40 direction 0: 1: CLR bits	Keep
7	GPIO39	GPIO39_DIR	1: CLR bits Bitwise CLR operation of GPIO39 direction 0: 1: CLR bits	Keep
6	GPIO38	GPIO38_DIR	1: CLR bits Bitwise CLR operation of GPIO38 direction 0: 1: CLR bits	Keep
5	GPIO37	GPIO37_DIR	1: CLR bits Bitwise CLR operation of GPIO37 direction 0: 1: CLR bits	Keep
4	GPIO36	GPIO36_DIR	1: CLR bits Bitwise CLR operation of GPIO36 direction 0: 1: CLR bits	Keep
3	GPIO35	GPIO35_DIR	1: CLR bits Bitwise CLR operation of GPIO35 direction 0: 1: CLR bits	Keep
2	GPIO34	GPIO34_DIR	1: CLR bits Bitwise CLR operation of GPIO34 direction 0: 1: CLR bits	Keep
1	GPIO33	GPIO33_DIR	1: CLR bits Bitwise CLR operation of GPIO33 direction 0: 1: CLR bits	Keep
0	GPIO32	GPIO32_DIR	1: CLR bits Bitwise CLR operation of GPIO32 direction 0: 1: CLR bits	Keep

A0020100 GPIO_PULLEN0 GPIO Pull-up/down Enable Control 43C00BFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO30					GPIO25	GPIO24	GPIO23	GPIO22						
Type		RW					RW	RW	RW	RW						
Reset		1					1	1	1	1						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO11		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset					1		1	1	1	1	1	1	1	1	1	1

Overview: Configures GPIO pull enabling

Bit(s)	Mnemonic	Name	Description
30	GPIO30	GPIO30_PULLEN	GPIO30 PULLEN 0: Disable 1: Enable
25	GPIO25	GPIO25_PULLEN	GPIO25 PULLEN 0: Disable 1: Enable
24	GPIO24	GPIO24_PULLEN	GPIO24 PULLEN 0: Disable 1: Enable
23	GPIO23	GPIO23_PULLEN	GPIO23 PULLEN 0: Disable 1: Enable
22	GPIO22	GPIO22_PULLEN	GPIO22 PULLEN 0: Disable 1: Enable
11	GPIO11	GPIO11_PULLEN	GPIO11 PULLEN 0: Disable 1: Enable
9	GPIO9	GPIO9_PULLEN	GPIO9 PULLEN 0: Disable 1: Enable
8	GPIO8	GPIO8_PULLEN	GPIO8 PULLEN 0: Disable 1: Enable
7	GPIO7	GPIO7_PULLEN	GPIO7 PULLEN 0: Disable 1: Enable
6	GPIO6	GPIO6_PULLEN	GPIO6 PULLEN 0: Disable 1: Enable
5	GPIO5	GPIO5_PULLEN	GPIO5 PULLEN 0: Disable 1: Enable
4	GPIO4	GPIO4_PULLEN	GPIO4 PULLEN 0: Disable 1: Enable
3	GPIO3	GPIO3_PULLEN	GPIO3 PULLEN 0: Disable 1: Enable
2	GPIO2	GPIO2_PULLEN	GPIO2 PULLEN 0: Disable 1: Enable
1	GPIO1	GPIO1_PULLEN	GPIO1 PULLEN 0: Disable 1: Enable
0	GPIO0	GPIO0_PULLEN	GPIO0 PULLEN 0: Disable 1: Enable

A0020104 **GPIO_PULLEN0 SET** GPIO Pull-up/down Enable Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO30					GPIO25	GPIO24	GPIO23	GPIO22						
Type		WO					WO	WO	WO	WO						
Reset		0					0	0	0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO11		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					WO		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset					0		0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_PULLEN0

Bit(s)	Mnemonic	Name	Description	
30	GPIO30	GPIO30_PULLEN	Bitwise SET operation of GPIO30_PULLEN_SET 0: 1: SET bits	Keep
25	GPIO25	GPIO25_PULLEN	Bitwise SET operation of GPIO25_PULLEN_SET 0: 1: SET bits	Keep
24	GPIO24	GPIO24_PULLEN	Bitwise SET operation of GPIO24_PULLEN_SET 0: 1: SET bits	Keep
23	GPIO23	GPIO23_PULLEN	Bitwise SET operation of GPIO23_PULLEN_SET 0: 1: SET bits	Keep
22	GPIO22	GPIO22_PULLEN	Bitwise SET operation of GPIO22_PULLEN_SET 0: 1: SET bits	Keep
11	GPIO11	GPIO11_PULLEN	Bitwise SET operation of GPIO11_PULLEN_SET 0: 1: SET bits	Keep
9	GPIO9	GPIO9_PULLEN	Bitwise SET operation of GPIO9_PULLEN_SET 0: 1: SET bits	Keep
8	GPIO8	GPIO8_PULLEN	Bitwise SET operation of GPIO8_PULLEN_SET 0: 1: SET bits	Keep
7	GPIO7	GPIO7_PULLEN	Bitwise SET operation of GPIO7_PULLEN_SET 0: 1: SET bits	Keep
6	GPIO6	GPIO6_PULLEN	Bitwise SET operation of GPIO6_PULLEN_SET 0: 1: SET bits	Keep
5	GPIO5	GPIO5_PULLEN	Bitwise SET operation of GPIO5_PULLEN_SET 0: 1: SET bits	Keep
4	GPIO4	GPIO4_PULLEN	Bitwise SET operation of GPIO4_PULLEN_SET 0: 1: SET bits	Keep
3	GPIO3	GPIO3_PULLEN	Bitwise SET operation of GPIO3_PULLEN_SET	

Bit(s)	Mnemonic	Name	Description	
			0: 1: SET bits	Keep
2	GPIO2	GPIO2_PULLEN	Bitwise SET operation of GPIO2 PULLEN_SET	
			0: 1: SET bits	Keep
1	GPIO1	GPIO1_PULLEN	Bitwise SET operation of GPIO1 PULLEN_SET	
			0: 1: SET bits	Keep
0	GPIO0	GPIO0_PULLEN	Bitwise SET operation of GPIO0 PULLEN_SET	
			0: 1: SET bits	Keep

A0020108 **GPIO_PULLEN0 CLR** GPIO Pull-up/down Enable Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO30					GPIO25	GPIO24	GPIO23	GPIO22						
Type		WO					WO	WO	WO	WO						
Reset		0					0	0	0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO11		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					WO		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset					0		0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_PULLEN0

Bit(s)	Mnemonic	Name	Description	
30	GPIO30	GPIO30_PULLEN	Bitwise CLR operation of GPIO30 PULLEN_CLR	
			0: 1: CLR bits	Keep
25	GPIO25	GPIO25_PULLEN	Bitwise CLR operation of GPIO25 PULLEN_CLR	
			0: 1: CLR bits	Keep
24	GPIO24	GPIO24_PULLEN	Bitwise CLR operation of GPIO24 PULLEN_CLR	
			0: 1: CLR bits	Keep
23	GPIO23	GPIO23_PULLEN	Bitwise CLR operation of GPIO23 PULLEN_CLR	
			0: 1: CLR bits	Keep
22	GPIO22	GPIO22_PULLEN	Bitwise CLR operation of GPIO22 PULLEN_CLR	
			0: 1: CLR bits	Keep
11	GPIO11	GPIO11_PULLEN	Bitwise CLR operation of GPIO11 PULLEN_CLR	
			0: 1: CLR bits	Keep
9	GPIO9	GPIO9_PULLEN	Bitwise CLR operation of GPIO9 PULLEN_CLR	
			0: 1: CLR bits	Keep
8	GPIO8	GPIO8_PULLEN	Bitwise CLR operation of GPIO8 PULLEN_CLR	
			0:	Keep

Bit(s)	Mnemonic	Name	Description	
7	GPIO7	GPIO7_PULLEN	Bitwise CLR operation of GPIO7 PULLEN_CLR 1: CLR bits 0: 1: CLR bits	Keep
6	GPIO6	GPIO6_PULLEN	Bitwise CLR operation of GPIO6 PULLEN_CLR 0: 1: CLR bits	Keep
5	GPIO5	GPIO5_PULLEN	Bitwise CLR operation of GPIO5 PULLEN_CLR 0: 1: CLR bits	Keep
4	GPIO4	GPIO4_PULLEN	Bitwise CLR operation of GPIO4 PULLEN_CLR 0: 1: CLR bits	Keep
3	GPIO3	GPIO3_PULLEN	Bitwise CLR operation of GPIO3 PULLEN_CLR 0: 1: CLR bits	Keep
2	GPIO2	GPIO2_PULLEN	Bitwise CLR operation of GPIO2 PULLEN_CLR 0: 1: CLR bits	Keep
1	GPIO1	GPIO1_PULLEN	Bitwise CLR operation of GPIO1 PULLEN_CLR 0: 1: CLR bits	Keep
0	GPIO0	GPIO0_PULLEN	Bitwise CLR operation of GPIO0 PULLEN_CLR 0: 1: CLR bits	Keep

A0020110 GPIO_PULLEN1 GPIO Pull-up/down Enable Control 00F01800

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5 5	GPIO5 4	GPIO5 3	GPIO5 2				
Type									RW	RW	RW	RW				
Reset									1	1	1	1				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO4 4	GPIO4 3											
Type				RW	RW											
Reset				1	1											

Overview: Configures GPIO pull enabling

Bit(s)	Mnemonic	Name	Description	
23	GPIO55	GPIO55_PULLEN	GPIO55 PULLEN 0: 1: Enable	Disable
22	GPIO54	GPIO54_PULLEN	GPIO54 PULLEN 0: 1: Enable	Disable
21	GPIO53	GPIO53_PULLEN	GPIO53 PULLEN 0: 1: Enable	Disable

Bit(s)	Mnemonic	Name	Description
20	GPIO52	GPIO52_PULLEN	GPIO52 PULLEN 0: Disable 1: Enable
12	GPIO44	GPIO44_PULLEN	GPIO44 PULLEN 0: Disable 1: Enable
11	GPIO43	GPIO43_PULLEN	GPIO43 PULLEN 0: Disable 1: Enable

A0020114 **GPIO_PULLEN1** **GPIO Pull-up/down Enable Control** **00000000**
SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO55	GPIO54	GPIO53	GPIO52				
Type									WO	WO	WO	WO				
Reset									0	0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO44	GPIO43											
Type				WO	WO											
Reset				0	0											

Overview: For bitwise access of GPIO_PULLEN1

Bit(s)	Mnemonic	Name	Description
23	GPIO55	GPIO55_PULLEN	Bitwise SET operation of GPIO51 PULLEN_SET 0: Keep 1: SET bits
22	GPIO54	GPIO54_PULLEN	Bitwise SET operation of GPIO44 PULLEN_SET 0: Keep 1: SET bits
21	GPIO53	GPIO53_PULLEN	Bitwise SET operation of GPIO43 PULLEN_SET 0: Keep 1: SET bits
20	GPIO52	GPIO52_PULLEN	Bitwise SET operation of GPIO30 PULLEN_SET 0: Keep 1: SET bits
12	GPIO44	GPIO44_PULLEN	Bitwise SET operation of GPIO44 PULLEN_SET 0: Keep 1: SET bits
11	GPIO43	GPIO43_PULLEN	Bitwise SET operation of GPIO43 PULLEN_SET 0: Keep 1: SET bits

A0020118 **GPIO_PULLEN1** **GPIO Pull-up/down Enable Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Name										GPIO5	GPIO5	GPIO5	GPIO5				
Type										WO	WO	WO	WO				
Reset										0	0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name				GPIO4	GPIO4												
Type				WO	WO												
Reset				0	0												

Overview: For bitwise access of GPIO_PULLEN1

Bit(s)	Mnemonic	Name	Description	
23	GPIO55	GPIO55_PULLEN	Bitwise CLR operation of GPIO51 PULLEN_CLR 0: 1: CLR bits	Keep
22	GPIO54	GPIO54_PULLEN	Bitwise CLR operation of GPIO44 PULLEN_CLR 0: 1: CLR bits	Keep
21	GPIO53	GPIO53_PULLEN	Bitwise CLR operation of GPIO43 PULLEN_CLR 0: 1: CLR bits	Keep
20	GPIO52	GPIO52_PULLEN	Bitwise CLR operation of GPIO30 PULLEN_CLR 0: 1: CLR bits	Keep
12	GPIO44	GPIO44_PULLEN	Bitwise CLR operation of GPIO44 PULLEN_CLR 0: 1: CLR bits	Keep
11	GPIO43	GPIO43_PULLEN	Bitwise CLR operation of GPIO43 PULLEN_CLR 0: 1: CLR bits	Keep

A0020200 **GPIO_DINV0** **GPIO Data Inversion Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INV31	INV30	INV29	INV28	INV27	INV26	INV25	INV24	INV23	INV22	INV21	INV20	INV19	INV18	INV17	INV16
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV15	INV14	INV13	INV12	INV11	INV10	INV9	INV8	INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO inversion enabling

Bit(s)	Mnemonic	Name	Description	
31	INV31	GPIO31_DINV	GPIO31 inversion control 0: Keep input value 1: Invert input value	
30	INV30	GPIO30_DINV	GPIO30 inversion control 0: Keep input value 1: Invert input value	
29	INV29	GPIO29_DINV	GPIO29 inversion control	

Bit(s)	Mnemonic	Name	Description		
			0: Keep 1: Invert input value	input	value
28	INV28	GPIO28_DINV	GPIO28 inversion control 0: Keep 1: Invert input value	input	value
27	INV27	GPIO27_DINV	GPIO27 inversion control 0: Keep 1: Invert input value	input	value
26	INV26	GPIO26_DINV	GPIO26 inversion control 0: Keep 1: Invert input value	input	value
25	INV25	GPIO25_DINV	GPIO25 inversion control 0: Keep 1: Invert input value	input	value
24	INV24	GPIO24_DINV	GPIO24 inversion control 0: Keep 1: Invert input value	input	value
23	INV23	GPIO23_DINV	GPIO23 inversion control 0: Keep 1: Invert input value	input	value
22	INV22	GPIO22_DINV	GPIO22 inversion control 0: Keep 1: Invert input value	input	value
21	INV21	GPIO21_DINV	GPIO21 inversion control 0: Keep 1: Invert input value	input	value
20	INV20	GPIO20_DINV	GPIO20 inversion control 0: Keep 1: Invert input value	input	value
19	INV19	GPIO19_DINV	GPIO19 inversion control 0: Keep 1: Invert input value	input	value
18	INV18	GPIO18_DINV	GPIO18 inversion control 0: Keep 1: Invert input value	input	value
17	INV17	GPIO17_DINV	GPIO17 inversion control 0: Keep 1: Invert input value	input	value
16	INV16	GPIO16_DINV	GPIO16 inversion control 0: Keep 1: Invert input value	input	value
15	INV15	GPIO15_DINV	GPIO15 inversion control 0: Keep 1: Invert input value	input	value
14	INV14	GPIO14_DINV	GPIO14 inversion control 0: Keep 1: Invert input value	input	value
13	INV13	GPIO13_DINV	GPIO13 inversion control 0: Keep 1: Invert input value	input	value
12	INV12	GPIO12_DINV	GPIO12 inversion control 0: Keep	input	value

Bit(s)	Mnemonic	Name	Description
11	INV11	GPIO11_DINV	1: Invert input value GPIO11 inversion control 0: Keep input value 1: Invert input value
10	INV10	GPIO10_DINV	1: Invert input value GPIO10 inversion control 0: Keep input value 1: Invert input value
9	INV9	GPIO9_DINV	1: Invert input value GPIO9 inversion control 0: Keep input value 1: Invert input value
8	INV8	GPIO8_DINV	1: Invert input value GPIO8 inversion control 0: Keep input value 1: Invert input value
7	INV7	GPIO7_DINV	1: Invert input value GPIO7 inversion control 0: Keep input value 1: Invert input value
6	INV6	GPIO6_DINV	1: Invert input value GPIO6 inversion control 0: Keep input value 1: Invert input value
5	INV5	GPIO5_DINV	1: Invert input value GPIO5 inversion control 0: Keep input value 1: Invert input value
4	INV4	GPIO4_DINV	1: Invert input value GPIO4 inversion control 0: Keep input value 1: Invert input value
3	INV3	GPIO3_DINV	1: Invert input value GPIO3 inversion control 0: Keep input value 1: Invert input value
2	INV2	GPIO2_DINV	1: Invert input value GPIO2 inversion control 0: Keep input value 1: Invert input value
1	INV1	GPIO1_DINV	1: Invert input value GPIO1 inversion control 0: Keep input value 1: Invert input value
0	INV0	GPIO0_DINV	1: Invert input value GPIO0 inversion control 0: Keep input value 1: Invert input value

A0020204 GPIO_DINV0_S GPIO Data Inversion Control 00000000
ET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INV31	INV30	INV29	INV28	INV27	INV26	INV25	INV24	INV23	INV22	INV21	INV20	INV19	INV18	INV17	INV16
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV15	INV14	INV13	INV12	INV11	INV10	INV9	INV8	INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DINV0

Bit(s)	Mnemonic	Name	Description	
31	INV31	GPIO31_DINV	Bitwise SET operation of GPIO31 inversion control 0: 1: SET bits	Keep
30	INV30	GPIO30_DINV	Bitwise SET operation of GPIO30 inversion control 0: 1: SET bits	Keep
29	INV29	GPIO29_DINV	Bitwise SET operation of GPIO29 inversion control 0: 1: SET bits	Keep
28	INV28	GPIO28_DINV	Bitwise SET operation of GPIO28 inversion control 0: 1: SET bits	Keep
27	INV27	GPIO27_DINV	Bitwise SET operation of GPIO27 inversion control 0: 1: SET bits	Keep
26	INV26	GPIO26_DINV	Bitwise SET operation of GPIO26 inversion control 0: 1: SET bits	Keep
25	INV25	GPIO25_DINV	Bitwise SET operation of GPIO25 inversion control 0: 1: SET bits	Keep
24	INV24	GPIO24_DINV	Bitwise SET operation of GPIO24 inversion control 0: 1: SET bits	Keep
23	INV23	GPIO23_DINV	Bitwise SET operation of GPIO23 inversion control 0: 1: SET bits	Keep
22	INV22	GPIO22_DINV	Bitwise SET operation of GPIO22 inversion control 0: 1: SET bits	Keep
21	INV21	GPIO21_DINV	Bitwise SET operation of GPIO21 inversion control 0: 1: SET bits	Keep
20	INV20	GPIO20_DINV	Bitwise SET operation of GPIO20 inversion control 0: 1: SET bits	Keep
19	INV19	GPIO19_DINV	Bitwise SET operation of GPIO19 inversion control 0: 1: SET bits	Keep
18	INV18	GPIO18_DINV	Bitwise SET operation of GPIO18 inversion control 0: 1: SET bits	Keep
17	INV17	GPIO17_DINV	Bitwise SET operation of GPIO17 inversion control 0: 1: SET bits	Keep
16	INV16	GPIO16_DINV	Bitwise SET operation of GPIO16 inversion control 0: 1: SET bits	Keep
15	INV15	GPIO15_DINV	Bitwise SET operation of GPIO15 inversion control 0: 1: SET bits	Keep
14	INV14	GPIO14_DINV	Bitwise SET operation of GPIO14 inversion control	

Bit(s)	Mnemonic	Name	Description	
			0: 1: SET bits	Keep
13	INV13	GPIO13_DINV	Bitwise SET operation of GPIO13 inversion control 0: 1: SET bits	Keep
12	INV12	GPIO12_DINV	Bitwise SET operation of GPIO12 inversion control 0: 1: SET bits	Keep
11	INV11	GPIO11_DINV	Bitwise SET operation of GPIO11 inversion control 0: 1: SET bits	Keep
10	INV10	GPIO10_DINV	Bitwise SET operation of GPIO10 inversion control 0: 1: SET bits	Keep
9	INV9	GPIO9_DINV	Bitwise SET operation of GPIO9 inversion control 0: 1: SET bits	Keep
8	INV8	GPIO8_DINV	Bitwise SET operation of GPIO8 inversion control 0: 1: SET bits	Keep
7	INV7	GPIO7_DINV	Bitwise SET operation of GPIO7 inversion control 0: 1: SET bits	Keep
6	INV6	GPIO6_DINV	Bitwise SET operation of GPIO6 inversion control 0: 1: SET bits	Keep
5	INV5	GPIO5_DINV	Bitwise SET operation of GPIO5 inversion control 0: 1: SET bits	Keep
4	INV4	GPIO4_DINV	Bitwise SET operation of GPIO4 inversion control 0: 1: SET bits	Keep
3	INV3	GPIO3_DINV	Bitwise SET operation of GPIO3 inversion control 0: 1: SET bits	Keep
2	INV2	GPIO2_DINV	Bitwise SET operation of GPIO2 inversion control 0: 1: SET bits	Keep
1	INV1	GPIO1_DINV	Bitwise SET operation of GPIO1 inversion control 0: 1: SET bits	Keep
0	INV0	GPIO0_DINV	Bitwise SET operation of GPIO0 inversion control 0: 1: SET bits	Keep

A0020208 **GPIO_DINV0_C** **GPIO Data Inversion Control** **00000000**
LR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INV31	INV30	INV29	INV28	INV27	INV26	INV25	INV24	INV23	INV22	INV21	INV20	INV19	INV18	INV17	INV16
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV15	INV14	INV13	INV12	INV11	INV10	INV9	INV8	INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DINV0

Bit(s)	Mnemonic	Name	Description	
31	INV31	GPIO31_DINV	Bitwise CLR operation of GPIO31 inversion control 0: 1: CLR bits	Keep
30	INV30	GPIO30_DINV	Bitwise CLR operation of GPIO30 inversion control 0: 1: CLR bits	Keep
29	INV29	GPIO29_DINV	Bitwise CLR operation of GPIO29 inversion control 0: 1: CLR bits	Keep
28	INV28	GPIO28_DINV	Bitwise CLR operation of GPIO28 inversion control 0: 1: CLR bits	Keep
27	INV27	GPIO27_DINV	Bitwise CLR operation of GPIO27 inversion control 0: 1: CLR bits	Keep
26	INV26	GPIO26_DINV	Bitwise CLR operation of GPIO26 inversion control 0: 1: CLR bits	Keep
25	INV25	GPIO25_DINV	Bitwise CLR operation of GPIO25 inversion control 0: 1: CLR bits	Keep
24	INV24	GPIO24_DINV	Bitwise CLR operation of GPIO24 inversion control 0: 1: CLR bits	Keep
23	INV23	GPIO23_DINV	Bitwise CLR operation of GPIO23 inversion control 0: 1: CLR bits	Keep
22	INV22	GPIO22_DINV	Bitwise CLR operation of GPIO22 inversion control 0: 1: CLR bits	Keep
21	INV21	GPIO21_DINV	Bitwise CLR operation of GPIO21 inversion control 0: 1: CLR bits	Keep
20	INV20	GPIO20_DINV	Bitwise CLR operation of GPIO20 inversion control 0: 1: CLR bits	Keep
19	INV19	GPIO19_DINV	Bitwise CLR operation of GPIO19 inversion control 0: 1: CLR bits	Keep
18	INV18	GPIO18_DINV	Bitwise CLR operation of GPIO18 inversion control 0: 1: CLR bits	Keep
17	INV17	GPIO17_DINV	Bitwise CLR operation of GPIO17 inversion control 0:	Keep

Bit(s)	Mnemonic	Name	Description	
16	INV16	GPIO16_DINV	Bitwise CLR operation of GPIO16 inversion control 1: CLR bits 0: 1: CLR bits	Keep
15	INV15	GPIO15_DINV	Bitwise CLR operation of GPIO15 inversion control 0: 1: CLR bits	Keep
14	INV14	GPIO14_DINV	Bitwise CLR operation of GPIO14 inversion control 0: 1: CLR bits	Keep
13	INV13	GPIO13_DINV	Bitwise CLR operation of GPIO13 inversion control 0: 1: CLR bits	Keep
12	INV12	GPIO12_DINV	Bitwise CLR operation of GPIO12 inversion control 0: 1: CLR bits	Keep
11	INV11	GPIO11_DINV	Bitwise CLR operation of GPIO11 inversion control 0: 1: CLR bits	Keep
10	INV10	GPIO10_DINV	Bitwise CLR operation of GPIO10 inversion control 0: 1: CLR bits	Keep
9	INV9	GPIO9_DINV	Bitwise CLR operation of GPIO9 inversion control 0: 1: CLR bits	Keep
8	INV8	GPIO8_DINV	Bitwise CLR operation of GPIO8 inversion control 0: 1: CLR bits	Keep
7	INV7	GPIO7_DINV	Bitwise CLR operation of GPIO7 inversion control 0: 1: CLR bits	Keep
6	INV6	GPIO6_DINV	Bitwise CLR operation of GPIO6 inversion control 0: 1: CLR bits	Keep
5	INV5	GPIO5_DINV	Bitwise CLR operation of GPIO5 inversion control 0: 1: CLR bits	Keep
4	INV4	GPIO4_DINV	Bitwise CLR operation of GPIO4 inversion control 0: 1: CLR bits	Keep
3	INV3	GPIO3_DINV	Bitwise CLR operation of GPIO3 inversion control 0: 1: CLR bits	Keep
2	INV2	GPIO2_DINV	Bitwise CLR operation of GPIO2 inversion control 0: 1: CLR bits	Keep
1	INV1	GPIO1_DINV	Bitwise CLR operation of GPIO1 inversion control 0: 1: CLR bits	Keep
0	INV0	GPIO0_DINV	Bitwise CLR operation of GPIO0 inversion control 0: 1: CLR bits	Keep

A0020210 GPIO_DINV1 GPIO Data Inversion Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									INV55	INV54	INV53	INV52	INV51	INV50	INV49	INV48
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV47	INV46	INV45	INV44	INV43	INV42	INV41	INV40	INV39	INV38	INV37	INV36	INV35	INV34	INV33	INV32
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO inversion enabling

Bit(s)	Mnemonic	Name	Description
23	INV55	GPIO55_DINV	GPIO55 inversion control 0: Keep input value 1: Invert input value
22	INV54	GPIO54_DINV	GPIO54 inversion control 0: Keep input value 1: Invert input value
21	INV53	GPIO53_DINV	GPIO53 inversion control 0: Keep input value 1: Invert input value
20	INV52	GPIO52_DINV	GPIO52 inversion control 0: Keep input value 1: Invert input value
19	INV51	GPIO51_DINV	GPIO51 inversion control 0: Keep input value 1: Invert input value
18	INV50	GPIO50_DINV	GPIO50 inversion control 0: Keep input value 1: Invert input value
17	INV49	GPIO49_DINV	GPIO49 inversion control 0: Keep input value 1: Invert input value
16	INV48	GPIO48_DINV	GPIO48 inversion control 0: Keep input value 1: Invert input value
15	INV47	GPIO47_DINV	GPIO47 inversion control 0: Keep input value 1: Invert input value
14	INV46	GPIO46_DINV	GPIO46 inversion control 0: Keep input value 1: Invert input value
13	INV45	GPIO45_DINV	GPIO45 inversion control 0: Keep input value 1: Invert input value
12	INV44	GPIO44_DINV	GPIO44 inversion control 0: Keep input value 1: Invert input value
11	INV43	GPIO43_DINV	GPIO43 inversion control

Bit(s)	Mnemonic	Name	Description
			0: Keep input value 1: Invert input value
10	INV42	GPIO42_DINV	GPIO42 inversion control 0: Keep input value 1: Invert input value
9	INV41	GPIO41_DINV	GPIO41 inversion control 0: Keep input value 1: Invert input value
8	INV40	GPIO40_DINV	GPIO40 inversion control 0: Keep input value 1: Invert input value
7	INV39	GPIO39_DINV	GPIO39 inversion control 0: Keep input value 1: Invert input value
6	INV38	GPIO38_DINV	GPIO38 inversion control 0: Keep input value 1: Invert input value
5	INV37	GPIO37_DINV	GPIO37 inversion control 0: Keep input value 1: Invert input value
4	INV36	GPIO36_DINV	GPIO36 inversion control 0: Keep input value 1: Invert input value
3	INV35	GPIO35_DINV	GPIO35 inversion control 0: Keep input value 1: Invert input value
2	INV34	GPIO34_DINV	GPIO34 inversion control 0: Keep input value 1: Invert input value
1	INV33	GPIO33_DINV	GPIO33 inversion control 0: Keep input value 1: Invert input value
0	INV32	GPIO32_DINV	GPIO32 inversion control 0: Keep input value 1: Invert input value

A0020214 **GPIO_DINV1_S** **GPIO Data Inversion Control** **00000000**
ET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									INV55	INV54	INV53	INV52	INV51	INV50	INV49	INV48
Type									WO	WO	WO	WO	WO	WO	WO	WO
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV47	INV46	INV45	INV44	INV43	INV42	INV41	INV40	INV39	INV38	INV37	INV36	INV35	INV34	INV33	INV32
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DINV1

Bit(s)	Mnemonic	Name	Description
--------	----------	------	-------------

Bit(s)	Mnemonic	Name	Description	
23	INV55	GPIO55_DINV	Bitwise SET operation of GPIO55 inversion control 0: 1: SET bits	Keep
22	INV54	GPIO54_DINV	Bitwise SET operation of GPIO54 inversion control 0: 1: SET bits	Keep
21	INV53	GPIO53_DINV	Bitwise SET operation of GPIO53 inversion control 0: 1: SET bits	Keep
20	INV52	GPIO52_DINV	Bitwise SET operation of GPIO52 inversion control 0: 1: SET bits	Keep
19	INV51	GPIO51_DINV	Bitwise SET operation of GPIO51 inversion control 0: 1: SET bits	Keep
18	INV50	GPIO50_DINV	Bitwise SET operation of GPIO50 inversion control 0: 1: SET bits	Keep
17	INV49	GPIO49_DINV	Bitwise SET operation of GPIO49 inversion control 0: 1: SET bits	Keep
16	INV48	GPIO48_DINV	Bitwise SET operation of GPIO48 inversion control 0: 1: SET bits	Keep
15	INV47	GPIO47_DINV	Bitwise SET operation of GPIO47 inversion control 0: 1: SET bits	Keep
14	INV46	GPIO46_DINV	Bitwise SET operation of GPIO46 inversion control 0: 1: SET bits	Keep
13	INV45	GPIO45_DINV	Bitwise SET operation of GPIO45 inversion control 0: 1: SET bits	Keep
12	INV44	GPIO44_DINV	Bitwise SET operation of GPIO44 inversion control 0: 1: SET bits	Keep
11	INV43	GPIO43_DINV	Bitwise SET operation of GPIO43 inversion control 0: 1: SET bits	Keep
10	INV42	GPIO42_DINV	Bitwise SET operation of GPIO42 inversion control 0: 1: SET bits	Keep
9	INV41	GPIO41_DINV	Bitwise SET operation of GPIO41 inversion control 0: 1: SET bits	Keep
8	INV40	GPIO40_DINV	Bitwise SET operation of GPIO40 inversion control 0: 1: SET bits	Keep
7	INV39	GPIO39_DINV	Bitwise SET operation of GPIO39 inversion control 0: 1: SET bits	Keep
6	INV38	GPIO38_DINV	Bitwise SET operation of GPIO38 inversion control	

Bit(s)	Mnemonic	Name	Description	
			0: 1: SET bits	Keep
5	INV37	GPIO37_DINV	Bitwise SET operation of GPIO37 inversion control 0: 1: SET bits	Keep
4	INV36	GPIO36_DINV	Bitwise SET operation of GPIO36 inversion control 0: 1: SET bits	Keep
3	INV35	GPIO35_DINV	Bitwise SET operation of GPIO35 inversion control 0: 1: SET bits	Keep
2	INV34	GPIO34_DINV	Bitwise SET operation of GPIO34 inversion control 0: 1: SET bits	Keep
1	INV33	GPIO33_DINV	Bitwise SET operation of GPIO33 inversion control 0: 1: SET bits	Keep
0	INV32	GPIO32_DINV	Bitwise SET operation of GPIO32 inversion control 0: 1: SET bits	Keep

A0020218 GPIO_DINV1_C LR **GPIO Data Inversion Control** 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									INV55	INV54	INV53	INV52	INV51	INV50	INV49	INV48
Type									WO	WO	WO	WO	WO	WO	WO	WO
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV47	INV46	INV45	INV44	INV43	INV42	INV41	INV40	INV39	INV38	INV37	INV36	INV35	INV34	INV33	INV32
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DINV1

Bit(s)	Mnemonic	Name	Description	
23	INV55	GPIO55_DINV	Bitwise CLR operation of GPIO55 inversion control 0: 1: CLR bits	Keep
22	INV54	GPIO54_DINV	Bitwise CLR operation of GPIO54 inversion control 0: 1: CLR bits	Keep
21	INV53	GPIO53_DINV	Bitwise CLR operation of GPIO53 inversion control 0: 1: CLR bits	Keep
20	INV52	GPIO52_DINV	Bitwise CLR operation of GPIO52 inversion control 0: 1: CLR bits	Keep
19	INV51	GPIO51_DINV	Bitwise CLR operation of GPIO51 inversion control 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
18	INV50	GPIO50_DINV	Bitwise CLR operation of GPIO50 inversion control 0: 1: CLR bits	Keep
17	INV49	GPIO49_DINV	Bitwise CLR operation of GPIO49 inversion control 0: 1: CLR bits	Keep
16	INV48	GPIO48_DINV	Bitwise CLR operation of GPIO48 inversion control 0: 1: CLR bits	Keep
15	INV47	GPIO47_DINV	Bitwise CLR operation of GPIO47 inversion control 0: 1: CLR bits	Keep
14	INV46	GPIO46_DINV	Bitwise CLR operation of GPIO46 inversion control 0: 1: CLR bits	Keep
13	INV45	GPIO45_DINV	Bitwise CLR operation of GPIO45 inversion control 0: 1: CLR bits	Keep
12	INV44	GPIO44_DINV	Bitwise CLR operation of GPIO44 inversion control 0: 1: CLR bits	Keep
11	INV43	GPIO43_DINV	Bitwise CLR operation of GPIO43 inversion control 0: 1: CLR bits	Keep
10	INV42	GPIO42_DINV	Bitwise CLR operation of GPIO42 inversion control 0: 1: CLR bits	Keep
9	INV41	GPIO41_DINV	Bitwise CLR operation of GPIO41 inversion control 0: 1: CLR bits	Keep
8	INV40	GPIO40_DINV	Bitwise CLR operation of GPIO40 inversion control 0: 1: CLR bits	Keep
7	INV39	GPIO39_DINV	Bitwise CLR operation of GPIO39 inversion control 0: 1: CLR bits	Keep
6	INV38	GPIO38_DINV	Bitwise CLR operation of GPIO38 inversion control 0: 1: CLR bits	Keep
5	INV37	GPIO37_DINV	Bitwise CLR operation of GPIO37 inversion control 0: 1: CLR bits	Keep
4	INV36	GPIO36_DINV	Bitwise CLR operation of GPIO36 inversion control 0: 1: CLR bits	Keep
3	INV35	GPIO35_DINV	Bitwise CLR operation of GPIO35 inversion control 0: 1: CLR bits	Keep
2	INV34	GPIO34_DINV	Bitwise CLR operation of GPIO34 inversion control 0: 1: CLR bits	Keep
1	INV33	GPIO33_DINV	Bitwise CLR operation of GPIO33 inversion control	

Bit(s)	Mnemonic	Name	Description
0	INV32	GPIO32_DINV	0: Keep 1: CLR bits Bitwise CLR operation of GPIO32 inversion control 0: Keep 1: CLR bits

A0020300 **GPIO_DOUT0** **GPIO Output Data Control** **04000800**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO output value

Bit(s)	Mnemonic	Name	Description
31	GPIO31	GPIO31_OUT	GPIO31 data output value 0: GPIO output LO 1: GPIO output HI
30	GPIO30	GPIO30_OUT	GPIO30 data output value 0: GPIO output LO 1: GPIO output HI
29	GPIO29	GPIO29_OUT	GPIO29 data output value 0: GPIO output LO 1: GPIO output HI
28	GPIO28	GPIO28_OUT	GPIO28 data output value 0: GPIO output LO 1: GPIO output HI
27	GPIO27	GPIO27_OUT	GPIO27 data output value 0: GPIO output LO 1: GPIO output HI
26	GPIO26	GPIO26_OUT	GPIO26 data output value 0: GPIO output LO 1: GPIO output HI
25	GPIO25	GPIO25_OUT	GPIO25 data output value 0: GPIO output LO 1: GPIO output HI
24	GPIO24	GPIO24_OUT	GPIO24 data output value 0: GPIO output LO 1: GPIO output HI
23	GPIO23	GPIO23_OUT	GPIO23 data output value 0: GPIO output LO 1: GPIO output HI
22	GPIO22	GPIO22_OUT	GPIO22 data output value 0: GPIO output LO 1: GPIO output HI

Bit(s)	Mnemonic	Name	Description	
21	GPIO21	GPIO21_OUT	GPIO21 data output value 0: GPIO output 1: GPIO output HI	LO
20	GPIO20	GPIO20_OUT	GPIO20 data output value 0: GPIO output 1: GPIO output HI	LO
19	GPIO19	GPIO19_OUT	GPIO19 data output value 0: GPIO output 1: GPIO output HI	LO
18	GPIO18	GPIO18_OUT	GPIO18 data output value 0: GPIO output 1: GPIO output HI	LO
17	GPIO17	GPIO17_OUT	GPIO17 data output value 0: GPIO output 1: GPIO output HI	LO
16	GPIO16	GPIO16_OUT	GPIO16 data output value 0: GPIO output 1: GPIO output HI	LO
15	GPIO15	GPIO15_OUT	GPIO15 data output value 0: GPIO output 1: GPIO output HI	LO
14	GPIO14	GPIO14_OUT	GPIO14 data output value 0: GPIO output 1: GPIO output HI	LO
13	GPIO13	GPIO13_OUT	GPIO13 data output value 0: GPIO output 1: GPIO output HI	LO
12	GPIO12	GPIO12_OUT	GPIO12 data output value 0: GPIO output 1: GPIO output HI	LO
11	GPIO11	GPIO11_OUT	GPIO11 data output value 0: GPIO output 1: GPIO output HI	LO
10	GPIO10	GPIO10_OUT	GPIO10 data output value 0: GPIO output 1: GPIO output HI	LO
9	GPIO9	GPIO9_OUT	GPIO9 data output value 0: GPIO output 1: GPIO output HI	LO
8	GPIO8	GPIO8_OUT	GPIO8 data output value 0: GPIO output 1: GPIO output HI	LO
7	GPIO7	GPIO7_OUT	GPIO7 data output value 0: GPIO output 1: GPIO output HI	LO
6	GPIO6	GPIO6_OUT	GPIO6 data output value 0: GPIO output 1: GPIO output HI	LO
5	GPIO5	GPIO5_OUT	GPIO5 data output value 0: GPIO output 1: GPIO output HI	LO
4	GPIO4	GPIO4_OUT	GPIO4 data output value	

Bit(s)	Mnemonic	Name	Description	
3	GPIO3	GPIO3_OUT	GPIO3 data output value 0: GPIO output HI 1: GPIO output HI	output LO
2	GPIO2	GPIO2_OUT	GPIO2 data output value 0: GPIO output HI 1: GPIO output HI	output LO
1	GPIO1	GPIO1_OUT	GPIO1 data output value 0: GPIO output HI 1: GPIO output HI	output LO
0	GPIO0	GPIO0_OUT	GPIO0 data output value 0: GPIO output HI 1: GPIO output HI	output LO

A0020304 GPIO_DOUT0_SET GPIO Output Data Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO3_1	GPIO3_0	GPIO2_9	GPIO2_8	GPIO2_7	GPIO2_6	GPIO2_5	GPIO2_4	GPIO2_3	GPIO2_2	GPIO2_1	GPIO2_0	GPIO1_9	GPIO1_8	GPIO1_7	GPIO1_6
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO1_5	GPIO1_4	GPIO1_3	GPIO1_2	GPIO1_1	GPIO1_0	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_OUT	Bitwise SET operation of GPIO31 data output value 0: 1: SET bits	Keep
30	GPIO30	GPIO30_OUT	Bitwise SET operation of GPIO30 data output value 0: 1: SET bits	Keep
29	GPIO29	GPIO29_OUT	Bitwise SET operation of GPIO29 data output value 0: 1: SET bits	Keep
28	GPIO28	GPIO28_OUT	Bitwise SET operation of GPIO28 data output value 0: 1: SET bits	Keep
27	GPIO27	GPIO27_OUT	Bitwise SET operation of GPIO27 data output value 0: 1: SET bits	Keep
26	GPIO26	GPIO26_OUT	Bitwise SET operation of GPIO26 data output value 0: 1: SET bits	Keep
25	GPIO25	GPIO25_OUT	Bitwise SET operation of GPIO25 data output value 0:	Keep

Bit(s)	Mnemonic	Name	Description	
24	GPIO24	GPIO24_OUT	1: SET bits Bitwise SET operation of GPIO24 data output value 0:	Keep
23	GPIO23	GPIO23_OUT	1: SET bits Bitwise SET operation of GPIO23 data output value 0:	Keep
22	GPIO22	GPIO22_OUT	1: SET bits Bitwise SET operation of GPIO22 data output value 0:	Keep
21	GPIO21	GPIO21_OUT	1: SET bits Bitwise SET operation of GPIO21 data output value 0:	Keep
20	GPIO20	GPIO20_OUT	1: SET bits Bitwise SET operation of GPIO20 data output value 0:	Keep
19	GPIO19	GPIO19_OUT	1: SET bits Bitwise SET operation of GPIO19 data output value 0:	Keep
18	GPIO18	GPIO18_OUT	1: SET bits Bitwise SET operation of GPIO18 data output value 0:	Keep
17	GPIO17	GPIO17_OUT	1: SET bits Bitwise SET operation of GPIO17 data output value 0:	Keep
16	GPIO16	GPIO16_OUT	1: SET bits Bitwise SET operation of GPIO16 data output value 0:	Keep
15	GPIO15	GPIO15_OUT	1: SET bits Bitwise SET operation of GPIO15 data output value 0:	Keep
14	GPIO14	GPIO14_OUT	1: SET bits Bitwise SET operation of GPIO14 data output value 0:	Keep
13	GPIO13	GPIO13_OUT	1: SET bits Bitwise SET operation of GPIO13 data output value 0:	Keep
12	GPIO12	GPIO12_OUT	1: SET bits Bitwise SET operation of GPIO12 data output value 0:	Keep
11	GPIO11	GPIO11_OUT	1: SET bits Bitwise SET operation of GPIO11 data output value 0:	Keep
10	GPIO10	GPIO10_OUT	1: SET bits Bitwise SET operation of GPIO10 data output value 0:	Keep
9	GPIO9	GPIO9_OUT	1: SET bits Bitwise SET operation of GPIO9 data output value 0:	Keep
8	GPIO8	GPIO8_OUT	1: SET bits Bitwise SET operation of GPIO8 data output value 0:	Keep

Bit(s)	Mnemonic	Name	Description	
7	GPIO7	GPIO7_OUT	Bitwise SET operation of GPIO7 data output value 0: 1: SET bits	Keep
6	GPIO6	GPIO6_OUT	Bitwise SET operation of GPIO6 data output value 0: 1: SET bits	Keep
5	GPIO5	GPIO5_OUT	Bitwise SET operation of GPIO5 data output value 0: 1: SET bits	Keep
4	GPIO4	GPIO4_OUT	Bitwise SET operation of GPIO4 data output value 0: 1: SET bits	Keep
3	GPIO3	GPIO3_OUT	Bitwise SET operation of GPIO3 data output value 0: 1: SET bits	Keep
2	GPIO2	GPIO2_OUT	Bitwise SET operation of GPIO2 data output value 0: 1: SET bits	Keep
1	GPIO1	GPIO1_OUT	Bitwise SET operation of GPIO1 data output value 0: 1: SET bits	Keep
0	GPIO0	GPIO0_OUT	Bitwise SET operation of GPIO0 data output value 0: 1: SET bits	Keep

A0020308 **GPIO DOUT0 CLR** **GPIO Output Data Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_OUT	Bitwise CLR operation of GPIO31 data output value 0: 1: CLR bits	Keep
30	GPIO30	GPIO30_OUT	Bitwise CLR operation of GPIO30 data output value 0: 1: CLR bits	Keep
29	GPIO29	GPIO29_OUT	Bitwise CLR operation of GPIO29 data output value 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
28	GPIO28	GPIO28_OUT	Bitwise CLR operation of GPIO28 data output value 0: 1: CLR bits	Keep
27	GPIO27	GPIO27_OUT	Bitwise CLR operation of GPIO27 data output value 0: 1: CLR bits	Keep
26	GPIO26	GPIO26_OUT	Bitwise CLR operation of GPIO26 data output value 0: 1: CLR bits	Keep
25	GPIO25	GPIO25_OUT	Bitwise CLR operation of GPIO25 data output value 0: 1: CLR bits	Keep
24	GPIO24	GPIO24_OUT	Bitwise CLR operation of GPIO24 data output value 0: 1: CLR bits	Keep
23	GPIO23	GPIO23_OUT	Bitwise CLR operation of GPIO23 data output value 0: 1: CLR bits	Keep
22	GPIO22	GPIO22_OUT	Bitwise CLR operation of GPIO22 data output value 0: 1: CLR bits	Keep
21	GPIO21	GPIO21_OUT	Bitwise CLR operation of GPIO21 data output value 0: 1: CLR bits	Keep
20	GPIO20	GPIO20_OUT	Bitwise CLR operation of GPIO20 data output value 0: 1: CLR bits	Keep
19	GPIO19	GPIO19_OUT	Bitwise CLR operation of GPIO19 data output value 0: 1: CLR bits	Keep
18	GPIO18	GPIO18_OUT	Bitwise CLR operation of GPIO18 data output value 0: 1: CLR bits	Keep
17	GPIO17	GPIO17_OUT	Bitwise CLR operation of GPIO17 data output value 0: 1: CLR bits	Keep
16	GPIO16	GPIO16_OUT	Bitwise CLR operation of GPIO16 data output value 0: 1: CLR bits	Keep
15	GPIO15	GPIO15_OUT	Bitwise CLR operation of GPIO15 data output value 0: 1: CLR bits	Keep
14	GPIO14	GPIO14_OUT	Bitwise CLR operation of GPIO14 data output value 0: 1: CLR bits	Keep
13	GPIO13	GPIO13_OUT	Bitwise CLR operation of GPIO13 data output value 0: 1: CLR bits	Keep
12	GPIO12	GPIO12_OUT	Bitwise CLR operation of GPIO12 data output value 0: 1: CLR bits	Keep
11	GPIO11	GPIO11_OUT	Bitwise CLR operation of GPIO11 data output value	

Bit(s)	Mnemonic	Name	Description	
			0: 1: CLR bits	Keep
10	GPIO10	GPIO10_OUT	Bitwise CLR operation of GPIO10 data output value 0: 1: CLR bits	Keep
9	GPIO9	GPIO9_OUT	Bitwise CLR operation of GPIO9 data output value 0: 1: CLR bits	Keep
8	GPIO8	GPIO8_OUT	Bitwise CLR operation of GPIO8 data output value 0: 1: CLR bits	Keep
7	GPIO7	GPIO7_OUT	Bitwise CLR operation of GPIO7 data output value 0: 1: CLR bits	Keep
6	GPIO6	GPIO6_OUT	Bitwise CLR operation of GPIO6 data output value 0: 1: CLR bits	Keep
5	GPIO5	GPIO5_OUT	Bitwise CLR operation of GPIO5 data output value 0: 1: CLR bits	Keep
4	GPIO4	GPIO4_OUT	Bitwise CLR operation of GPIO4 data output value 0: 1: CLR bits	Keep
3	GPIO3	GPIO3_OUT	Bitwise CLR operation of GPIO3 data output value 0: 1: CLR bits	Keep
2	GPIO2	GPIO2_OUT	Bitwise CLR operation of GPIO2 data output value 0: 1: CLR bits	Keep
1	GPIO1	GPIO1_OUT	Bitwise CLR operation of GPIO1 data output value 0: 1: CLR bits	Keep
0	GPIO0	GPIO0_OUT	Bitwise CLR operation of GPIO0 data output value 0: 1: CLR bits	Keep

A0020310 **GPIO_DOUT1** **GPIO Output Data Control** **00004000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5	GPIO5			GPIO5	GPIO5	GPIO4	GPIO4
Type									RW	RW			RW	RW	RW	RW
Reset									0	0			0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO output value

Bit(s)	Mnemonic	Name	Description
--------	----------	------	-------------

Bit(s)	Mnemonic	Name	Description	
23	GPIO55	GPIO55_OUT	GPIO55 data output value 0: GPIO output 1: GPIO output HI	LO
22	GPIO54	GPIO54_OUT	GPIO54 data output value 0: GPIO output 1: GPIO output HI	LO
19	GPIO51	GPIO51_OUT	GPIO51 data output value 0: GPIO output 1: GPIO output HI	LO
18	GPIO50	GPIO50_OUT	GPIO50 data output value 0: GPIO output 1: GPIO output HI	LO
17	GPIO49	GPIO49_OUT	GPIO49 data output value 0: GPIO output 1: GPIO output HI	LO
16	GPIO48	GPIO48_OUT	GPIO48 data output value 0: GPIO output 1: GPIO output HI	LO
15	GPIO47	GPIO47_OUT	GPIO47 data output value 0: GPIO output 1: GPIO output HI	LO
14	GPIO46	GPIO46_OUT	GPIO46 data output value 0: GPIO output 1: GPIO output HI	LO
13	GPIO45	GPIO45_OUT	GPIO45 data output value 0: GPIO output 1: GPIO output HI	LO
12	GPIO44	GPIO44_OUT	GPIO44 data output value 0: GPIO output 1: GPIO output HI	LO
11	GPIO43	GPIO43_OUT	GPIO43 data output value 0: GPIO output 1: GPIO output HI	LO
10	GPIO42	GPIO42_OUT	GPIO42 data output value 0: GPIO output 1: GPIO output HI	LO
9	GPIO41	GPIO41_OUT	GPIO41 data output value 0: GPIO output 1: GPIO output HI	LO
8	GPIO40	GPIO40_OUT	GPIO40 data output value 0: GPIO output 1: GPIO output HI	LO
7	GPIO39	GPIO39_OUT	GPIO39 data output value 0: GPIO output 1: GPIO output HI	LO
6	GPIO38	GPIO38_OUT	GPIO38 data output value 0: GPIO output 1: GPIO output HI	LO
5	GPIO37	GPIO37_OUT	GPIO37 data output value 0: GPIO output 1: GPIO output HI	LO
4	GPIO36	GPIO36_OUT	GPIO36 data output value	

Bit(s)	Mnemonic	Name	Description
3	GPIO35	GPIO35_OUT	GPIO output 0: GPIO output HI 1: GPIO output HI GPIO35 data output value
2	GPIO34	GPIO34_OUT	GPIO output 0: GPIO output HI 1: GPIO output HI GPIO34 data output value
1	GPIO33	GPIO33_OUT	GPIO output 0: GPIO output HI 1: GPIO output HI GPIO33 data output value
0	GPIO32	GPIO32_OUT	GPIO output 0: GPIO output HI 1: GPIO output HI GPIO32 data output value

A0020314 **GPIO DOUT1 SET** GPIO Output Data Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO55	GPIO54			GPIO51	GPIO50	GPIO49	GPIO48
Type									WO	WO			WO	WO	WO	WO
Reset									0	0			0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR1

Bit(s)	Mnemonic	Name	Description
23	GPIO55	GPIO55_OUT	Bitwise SET operation of GPIO55 data output value 0: Keep 1: SET bits
22	GPIO54	GPIO54_OUT	Bitwise SET operation of GPIO54 data output value 0: Keep 1: SET bits
19	GPIO51	GPIO51_OUT	Bitwise SET operation of GPIO51 data output value 0: Keep 1: SET bits
18	GPIO50	GPIO50_OUT	Bitwise SET operation of GPIO50 data output value 0: Keep 1: SET bits
17	GPIO49	GPIO49_OUT	Bitwise SET operation of GPIO49 data output value 0: Keep 1: SET bits
16	GPIO48	GPIO48_OUT	Bitwise SET operation of GPIO48 data output value 0: Keep 1: SET bits
15	GPIO47	GPIO47_OUT	Bitwise SET operation of GPIO47 data output value 0: Keep

Bit(s)	Mnemonic	Name	Description	
14	GPIO46	GPIO46_OUT	1: SET bits Bitwise SET operation of GPIO46 data output value 0:	Keep
13	GPIO45	GPIO45_OUT	1: SET bits Bitwise SET operation of GPIO45 data output value 0:	Keep
12	GPIO44	GPIO44_OUT	1: SET bits Bitwise SET operation of GPIO44 data output value 0:	Keep
11	GPIO43	GPIO43_OUT	1: SET bits Bitwise SET operation of GPIO43 data output value 0:	Keep
10	GPIO42	GPIO42_OUT	1: SET bits Bitwise SET operation of GPIO42 data output value 0:	Keep
9	GPIO41	GPIO41_OUT	1: SET bits Bitwise SET operation of GPIO41 data output value 0:	Keep
8	GPIO40	GPIO40_OUT	1: SET bits Bitwise SET operation of GPIO40 data output value 0:	Keep
7	GPIO39	GPIO39_OUT	1: SET bits Bitwise SET operation of GPIO39 data output value 0:	Keep
6	GPIO38	GPIO38_OUT	1: SET bits Bitwise SET operation of GPIO38 data output value 0:	Keep
5	GPIO37	GPIO37_OUT	1: SET bits Bitwise SET operation of GPIO37 data output value 0:	Keep
4	GPIO36	GPIO36_OUT	1: SET bits Bitwise SET operation of GPIO36 data output value 0:	Keep
3	GPIO35	GPIO35_OUT	1: SET bits Bitwise SET operation of GPIO35 data output value 0:	Keep
2	GPIO34	GPIO34_OUT	1: SET bits Bitwise SET operation of GPIO34 data output value 0:	Keep
1	GPIO33	GPIO33_OUT	1: SET bits Bitwise SET operation of GPIO33 data output value 0:	Keep
0	GPIO32	GPIO32_OUT	1: SET bits Bitwise SET operation of GPIO32 data output value 0:	Keep

A0020318 GPIO_DOUT1_C **GPIO Output Data Control** **00000000**
LR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Name										GPIO5	GPIO5			GPIO5	GPIO5	GPIO4	GPIO4
Type										WO	WO			WO	WO	WO	WO
Reset										0	0			0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DIR1

Bit(s)	Mnemonic	Name	Description	
23	GPIO55	GPIO55_OUT	Bitwise CLR operation of GPIO55 data output value 0: 1: CLR bits	Keep
22	GPIO54	GPIO54_OUT	Bitwise CLR operation of GPIO54 data output value 0: 1: CLR bits	Keep
19	GPIO51	GPIO51_OUT	Bitwise CLR operation of GPIO51 data output value 0: 1: CLR bits	Keep
18	GPIO50	GPIO50_OUT	Bitwise CLR operation of GPIO50 data output value 0: 1: CLR bits	Keep
17	GPIO49	GPIO49_OUT	Bitwise CLR operation of GPIO49 data output value 0: 1: CLR bits	Keep
16	GPIO48	GPIO48_OUT	Bitwise CLR operation of GPIO48 data output value 0: 1: CLR bits	Keep
15	GPIO47	GPIO47_OUT	Bitwise CLR operation of GPIO47 data output value 0: 1: CLR bits	Keep
14	GPIO46	GPIO46_OUT	Bitwise CLR operation of GPIO46 data output value 0: 1: CLR bits	Keep
13	GPIO45	GPIO45_OUT	Bitwise CLR operation of GPIO45 data output value 0: 1: CLR bits	Keep
12	GPIO44	GPIO44_OUT	Bitwise CLR operation of GPIO44 data output value 0: 1: CLR bits	Keep
11	GPIO43	GPIO43_OUT	Bitwise CLR operation of GPIO43 data output value 0: 1: CLR bits	Keep
10	GPIO42	GPIO42_OUT	Bitwise CLR operation of GPIO42 data output value 0: 1: CLR bits	Keep
9	GPIO41	GPIO41_OUT	Bitwise CLR operation of GPIO41 data output value 0: 1: CLR bits	Keep
8	GPIO40	GPIO40_OUT	Bitwise CLR operation of GPIO40 data output value 0:	Keep

Bit(s)	Mnemonic	Name	Description	
7	GPIO39	GPIO39_OUT	1: CLR bits Bitwise CLR operation of GPIO39 data output value 0: 1: CLR bits	Keep
6	GPIO38	GPIO38_OUT	1: CLR bits Bitwise CLR operation of GPIO38 data output value 0: 1: CLR bits	Keep
5	GPIO37	GPIO37_OUT	1: CLR bits Bitwise CLR operation of GPIO37 data output value 0: 1: CLR bits	Keep
4	GPIO36	GPIO36_OUT	1: CLR bits Bitwise CLR operation of GPIO36 data output value 0: 1: CLR bits	Keep
3	GPIO35	GPIO35_OUT	1: CLR bits Bitwise CLR operation of GPIO35 data output value 0: 1: CLR bits	Keep
2	GPIO34	GPIO34_OUT	1: CLR bits Bitwise CLR operation of GPIO34 data output value 0: 1: CLR bits	Keep
1	GPIO33	GPIO33_OUT	1: CLR bits Bitwise CLR operation of GPIO33 data output value 0: 1: CLR bits	Keep
0	GPIO32	GPIO32_OUT	1: CLR bits Bitwise CLR operation of GPIO32 data output value 0: 1: CLR bits	Keep

A0020400 **GPIO_DIN0** **GPIO Input Data Value** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Reads GPIO input value

Bit(s)	Mnemonic	Name	Description
31	GPIO31	GPIO31_DIN	GPIO31 data input value
30	GPIO30	GPIO30_DIN	GPIO30 data input value
29	GPIO29	GPIO29_DIN	GPIO29 data input value
28	GPIO28	GPIO28_DIN	GPIO28 data input value
27	GPIO27	GPIO27_DIN	GPIO27 data input value
26	GPIO26	GPIO26_DIN	GPIO26 data input value
25	GPIO25	GPIO25_DIN	GPIO25 data input value
24	GPIO24	GPIO24_DIN	GPIO24 data input value

Bit(s)	Mnemonic	Name	Description
23	GPIO23	GPIO23_DIN	GPIO23 data input value
22	GPIO22	GPIO22_DIN	GPIO22 data input value
21	GPIO21	GPIO21_DIN	GPIO21 data input value
20	GPIO20	GPIO20_DIN	GPIO20 data input value
19	GPIO19	GPIO19_DIN	GPIO19 data input value
18	GPIO18	GPIO18_DIN	GPIO18 data input value
17	GPIO17	GPIO17_DIN	GPIO17 data input value
16	GPIO16	GPIO16_DIN	GPIO16 data input value
15	GPIO15	GPIO15_DIN	GPIO15 data input value
14	GPIO14	GPIO14_DIN	GPIO14 data input value
13	GPIO13	GPIO13_DIN	GPIO13 data input value
12	GPIO12	GPIO12_DIN	GPIO12 data input value
11	GPIO11	GPIO11_DIN	GPIO11 data input value
10	GPIO10	GPIO10_DIN	GPIO10 data input value
9	GPIO9	GPIO9_DIN	GPIO9 data input value
8	GPIO8	GPIO8_DIN	GPIO8 data input value
7	GPIO7	GPIO7_DIN	GPIO7 data input value
6	GPIO6	GPIO6_DIN	GPIO6 data input value
5	GPIO5	GPIO5_DIN	GPIO5 data input value
4	GPIO4	GPIO4_DIN	GPIO4 data input value
3	GPIO3	GPIO3_DIN	GPIO3 data input value
2	GPIO2	GPIO2_DIN	GPIO2 data input value
1	GPIO1	GPIO1_DIN	GPIO1 data input value
0	GPIO0	GPIO0_DIN	GPIO0 data input value

A0020410 **GPIO_DIN1** **GPIO Input Data Value** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO4	GPIO4
Type									RO	RO	RO	RO	RO	RO	RO	RO
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Reads GPIO input value

Bit(s)	Mnemonic	Name	Description
23	GPIO55	GPIO55_DIN	GPIO55 data input value
22	GPIO54	GPIO54_DIN	GPIO54 data input value
21	GPIO53	GPIO53_DIN	GPIO53 data input value
20	GPIO52	GPIO52_DIN	GPIO52 data input value
19	GPIO51	GPIO51_DIN	GPIO51 data input value
18	GPIO50	GPIO50_DIN	GPIO50 data input value

Bit(s)	Mnemonic	Name	Description
17	GPIO49	GPIO49_DIN	GPIO49 data input value
16	GPIO48	GPIO48_DIN	GPIO48 data input value
15	GPIO47	GPIO47_DIN	GPIO47 data input value
14	GPIO46	GPIO46_DIN	GPIO46 data input value
13	GPIO45	GPIO45_DIN	GPIO45 data input value
12	GPIO44	GPIO44_DIN	GPIO44 data input value
11	GPIO43	GPIO43_DIN	GPIO43 data input value
10	GPIO42	GPIO42_DIN	GPIO42 data input value
9	GPIO41	GPIO41_DIN	GPIO41 data input value
8	GPIO40	GPIO40_DIN	GPIO40 data input value
7	GPIO39	GPIO39_DIN	GPIO39 data input value
6	GPIO38	GPIO38_DIN	GPIO38 data input value
5	GPIO37	GPIO37_DIN	GPIO37 data input value
4	GPIO36	GPIO36_DIN	GPIO36 data input value
3	GPIO35	GPIO35_DIN	GPIO35 data input value
2	GPIO34	GPIO34_DIN	GPIO34 data input value
1	GPIO33	GPIO33_DIN	GPIO33 data input value
0	GPIO32	GPIO32_DIN	GPIO32 data input value

A0020500 **GPIO_PULLSEL** **GPIO Pullsel Control** **00000000**
0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO30					GPIO25	GPIO24	GPIO23	GPIO22						
Type		RW					RW	RW	RW	RW						
Reset		0					0	0	0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO11		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset					0		0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO PUPD selection

Bit(s)	Mnemonic	Name	Description
30	GPIO30	GPIO30_PULLSEL	GPIO30 PULLSEL 0: Pull 1: Pull up
25	GPIO25	GPIO25_PULLSEL	GPIO25 PULLSEL 0: Pull 1: Pull up
24	GPIO24	GPIO24_PULLSEL	GPIO24 PULLSEL 0: Pull 1: Pull up
23	GPIO23	GPIO23_PULLSEL	GPIO23 PULLSEL 0: Pull 1: Pull up

Bit(s)	Mnemonic	Name	Description
22	GPIO22	GPIO22_PULLSEL	GPIO22 PULLSEL 0: Pull down 1: Pull up
11	GPIO11	GPIO11_PULLSEL	GPIO11 PULLSEL 0: Pull down 1: Pull up
9	GPIO9	GPIO9_PULLSEL	GPIO9 PULLSEL 0: Pull down 1: Pull up
8	GPIO8	GPIO8_PULLSEL	GPIO8 PULLSEL 0: Pull down 1: Pull up
7	GPIO7	GPIO7_PULLSEL	GPIO7 PULLSEL 0: Pull down 1: Pull up
6	GPIO6	GPIO6_PULLSEL	GPIO6 PULLSEL 0: Pull down 1: Pull up
5	GPIO5	GPIO5_PULLSEL	GPIO5 PULLSEL 0: Pull down 1: Pull up
4	GPIO4	GPIO4_PULLSEL	GPIO4 PULLSEL 0: Pull down 1: Pull up
3	GPIO3	GPIO3_PULLSEL	GPIO3 PULLSEL 0: Pull down 1: Pull up
2	GPIO2	GPIO2_PULLSEL	GPIO2 PULLSEL 0: Pull down 1: Pull up
1	GPIO1	GPIO1_PULLSEL	GPIO1 PULLSEL 0: Pull down 1: Pull up
0	GPIO0	GPIO0_PULLSEL	GPIO0 PULLSEL 0: Pull down 1: Pull up

A0020504 **GPIO PULLSEL** **GPIO Pullsel Control** **00000000**
0_SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO3					GPIO2	GPIO2	GPIO2	GPIO2						
Type		WO					WO	WO	WO	WO						
Reset		0					0	0	0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO1		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					WO		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset					0		0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_PULLSEL0

Bit(s)	Mnemonic	Name	Description	
30	GPIO30	GPIO30_PULLSEL	Bitwise SET operation of GPIO30 PULLSEL_SET 0: 1: SET bits	Keep
25	GPIO25	GPIO25_PULLSEL	Bitwise SET operation of GPIO25 PULLSEL_SET 0: 1: SET bits	Keep
24	GPIO24	GPIO24_PULLSEL	Bitwise SET operation of GPIO24 PULLSEL_SET 0: 1: SET bits	Keep
23	GPIO23	GPIO23_PULLSEL	Bitwise SET operation of GPIO23 PULLSEL_SET 0: 1: SET bits	Keep
22	GPIO22	GPIO22_PULLSEL	Bitwise SET operation of GPIO22 PULLSEL_SET 0: 1: SET bits	Keep
11	GPIO11	GPIO11_PULLSEL	Bitwise SET operation of GPIO11 PULLSEL_SET 0: 1: SET bits	Keep
9	GPIO9	GPIO9_PULLSEL	Bitwise SET operation of GPIO9 PULLSEL_SET 0: 1: SET bits	Keep
8	GPIO8	GPIO8_PULLSEL	Bitwise SET operation of GPIO8 PULLSEL_SET 0: 1: SET bits	Keep
7	GPIO7	GPIO7_PULLSEL	Bitwise SET operation of GPIO7 PULLSEL_SET 0: 1: SET bits	Keep
6	GPIO6	GPIO6_PULLSEL	Bitwise SET operation of GPIO6 PULLSEL_SET 0: 1: SET bits	Keep
5	GPIO5	GPIO5_PULLSEL	Bitwise SET operation of GPIO5 PULLSEL_SET 0: 1: SET bits	Keep
4	GPIO4	GPIO4_PULLSEL	Bitwise SET operation of GPIO4 PULLSEL_SET 0: 1: SET bits	Keep
3	GPIO3	GPIO3_PULLSEL	Bitwise SET operation of GPIO3 PULLSEL_SET 0: 1: SET bits	Keep
2	GPIO2	GPIO2_PULLSEL	Bitwise SET operation of GPIO2 PULLSEL_SET 0: 1: SET bits	Keep
1	GPIO1	GPIO1_PULLSEL	Bitwise SET operation of GPIO1 PULLSEL_SET 0: 1: SET bits	Keep
0	GPIO0	GPIO0_PULLSEL	Bitwise SET operation of GPIO0 PULLSEL_SET 0: 1: SET bits	Keep

A0020508 **GPIO_PULLSEL_0_CLR** GPIO Pullsel Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO30					GPIO25	GPIO24	GPIO23	GPIO22						
Type		WO					WO	WO	WO	WO						
Reset		0					0	0	0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO11		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					WO		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset					0		0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_PULLSEL0

Bit(s)	Mnemonic	Name	Description	
30	GPIO30	GPIO30_PULLSEL	Bitwise CKR operation of GPIO30 PULLSEL_CLR 0: 1: CLR bits	Keep
25	GPIO25	GPIO25_PULLSEL	Bitwise CKR operation of GPIO25 PULLSEL_CLR 0: 1: CLR bits	Keep
24	GPIO24	GPIO24_PULLSEL	Bitwise CKR operation of GPIO24 PULLSEL_CLR 0: 1: CLR bits	Keep
23	GPIO23	GPIO23_PULLSEL	Bitwise CKR operation of GPIO23 PULLSEL_CLR 0: 1: CLR bits	Keep
22	GPIO22	GPIO22_PULLSEL	Bitwise CKR operation of GPIO22 PULLSEL_CLR 0: 1: CLR bits	Keep
11	GPIO11	GPIO11_PULLSEL	Bitwise CKR operation of GPIO11 PULLSEL_CLR 0: 1: CLR bits	Keep
9	GPIO9	GPIO9_PULLSEL	Bitwise CKR operation of GPIO9 PULLSEL_CLR 0: 1: CLR bits	Keep
8	GPIO8	GPIO8_PULLSEL	Bitwise CKR operation of GPIO8 PULLSEL_CLR 0: 1: CLR bits	Keep
7	GPIO7	GPIO7_PULLSEL	Bitwise CKR operation of GPIO7 PULLSEL_CLR 0: 1: CLR bits	Keep
6	GPIO6	GPIO6_PULLSEL	Bitwise CKR operation of GPIO6 PULLSEL_CLR 0: 1: CLR bits	Keep
5	GPIO5	GPIO5_PULLSEL	Bitwise CKR operation of GPIO5 PULLSEL_CLR 0: 1: CLR bits	Keep
4	GPIO4	GPIO4_PULLSEL	Bitwise CKR operation of GPIO4 PULLSEL_CLR 0: 1: CLR bits	Keep
3	GPIO3	GPIO3_PULLSEL	Bitwise CKR operation of GPIO3 PULLSEL_CLR	

Bit(s)	Mnemonic	Name	Description	
			0: 1: CLR bits	Keep
2	GPIO2	GPIO2_PULLSEL	Bitwise CKR operation of GPIO2 PULLSEL_CLR	
			0: 1: CLR bits	Keep
1	GPIO1	GPIO1_PULLSEL	Bitwise CKR operation of GPIO1 PULLSEL_CLR	
			0: 1: CLR bits	Keep
0	GPIO0	GPIO0_PULLSEL	Bitwise CKR operation of GPIO0 PULLSEL_CLR	
			0: 1: CLR bits	Keep

A0020510 **GPIO_PULLSEL** **GPIO Pullsel Control** **00000000**
1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5 5	GPIO5 4	GPIO5 3	GPIO5 2				
Type									RW	RW	RW	RW				
Reset									0	0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO4 4	GPIO4 3											
Type				RW	RW											
Reset				0	0											

Overview: Configures GPIO PUPD selection

Bit(s)	Mnemonic	Name	Description	
23	GPIO55	GPIO55_PULLSEL	GPIO55 PULLSEL	
			0: 1: Pull up	Pull down
22	GPIO54	GPIO54_PULLSEL	GPIO54 PULLSEL	
			0: 1: Pull up	Pull down
21	GPIO53	GPIO53_PULLSEL	GPIO53 PULLSEL	
			0: 1: Pull up	Pull down
20	GPIO52	GPIO52_PULLSEL	GPIO52 PULLSEL	
			0: 1: Pull up	Pull down
12	GPIO44	GPIO44_PULLSEL	GPIO44 PULLSEL	
			0: 1: Pull up	Pull down
11	GPIO43	GPIO43_PULLSEL	GPIO43 PULLSEL	
			0: 1: Pull up	Pull down

A0020514 **GPIO_PULLSEL** **GPIO Pullsel Control** **00000000**

1 SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5 5	GPIO5 4	GPIO5 3	GPIO5 2				
Type									WO	WO	WO	WO				
Reset									0	0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO4 4	GPIO4 3											
Type				WO	WO											
Reset				0	0											

Overview: For bitwise access of GPIO_PULLSEL1

Bit(s)	Mnemonic	Name	Description	
23	GPIO55	GPIO55_PULLSEL	Bitwise SET operation of GPIO55 PULLSEL_SET 0: 1: SET bits	Keep
22	GPIO54	GPIO54_PULLSEL	Bitwise SET operation of GPIO54 PULLSEL_SET 0: 1: SET bits	Keep
21	GPIO53	GPIO53_PULLSEL	Bitwise SET operation of GPIO53 PULLSEL_SET 0: 1: SET bits	Keep
20	GPIO52	GPIO52_PULLSEL	Bitwise SET operation of GPIO52 PULLSEL_SET 0: 1: SET bits	Keep
12	GPIO44	GPIO44_PULLSEL	Bitwise SET operation of GPIO44 PULLSEL_SET 0: 1: SET bits	Keep
11	GPIO43	GPIO43_PULLSEL	Bitwise SET operation of GPIO43 PULLSEL_SET 0: 1: SET bits	Keep

A0020518 **GPIO_PULLSEL** **GPIO Pullsel Control** **00000000**
1_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									GPIO5 5	GPIO5 4	GPIO5 3	GPIO5 2				
Type									WO	WO	WO	WO				
Reset									0	0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO4 4	GPIO4 3											
Type				WO	WO											
Reset				0	0											

Overview: For bitwise access of GPIO_PULLSEL1

Bit(s)	Mnemonic	Name	Description	
23	GPIO55	GPIO55_PULLSEL	Bitwise CKR operation of GPIO55 PULLSEL_CLR 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
22	GPIO54	GPIO54_PULLSEL	Bitwise CKR operation of GPIO54 PULLSEL_CLR 0: 1: CLR bits	Keep
21	GPIO53	GPIO53_PULLSEL	Bitwise CKR operation of GPIO53 PULLSEL_CLR 0: 1: CLR bits	Keep
20	GPIO52	GPIO52_PULLSEL	Bitwise CKR operation of GPIO52 PULLSEL_CLR 0: 1: CLR bits	Keep
12	GPIO44	GPIO44_PULLSEL	Bitwise CKR operation of GPIO44 PULLSEL_CLR 0: 1: CLR bits	Keep
11	GPIO43	GPIO43_PULLSEL	Bitwise CKR operation of GPIO43 PULLSEL_CLR 0: 1: CLR bits	Keep

A0020600 GPIO_SMT0 GPIO SMT Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO Schmit trigger control

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_SMT	SMT for GPIO31 0: 1: Enable	Disable
30	GPIO30	GPIO30_SMT	SMT for GPIO30 0: 1: Enable	Disable
29	GPIO29	GPIO29_SMT	SMT for GPIO29 0: 1: Enable	Disable
28	GPIO28	GPIO28_SMT	SMT for GPIO28 0: 1: Enable	Disable
27	GPIO27	GPIO27_SMT	SMT for GPIO27 0: 1: Enable	Disable
26	GPIO26	GPIO26_SMT	SMT for GPIO26 0: 1: Enable	Disable
25	GPIO25	GPIO25_SMT	SMT for GPIO25 0:	Disable

Bit(s)	Mnemonic	Name	Description
24	GPIO24	GPIO24_SMT	1: Enable SMT for GPIO24 0: Disable
23	GPIO23	GPIO23_SMT	1: Enable SMT for GPIO23 0: Disable
22	GPIO22	GPIO22_SMT	1: Enable SMT for GPIO22 0: Disable
21	GPIO21	GPIO21_SMT	1: Enable SMT for GPIO21 0: Disable
20	GPIO20	GPIO20_SMT	1: Enable SMT for GPIO20 0: Disable
19	GPIO19	GPIO19_SMT	1: Enable SMT for GPIO19 0: Disable
18	GPIO18	GPIO18_SMT	1: Enable SMT for GPIO18 0: Disable
17	GPIO17	GPIO17_SMT	1: Enable SMT for GPIO17 0: Disable
16	GPIO16	GPIO16_SMT	1: Enable SMT for GPIO16 0: Disable
15	GPIO15	GPIO15_SMT	1: Enable SMT for GPIO15 0: Disable
14	GPIO14	GPIO14_SMT	1: Enable SMT for GPIO14 0: Disable
13	GPIO13	GPIO13_SMT	1: Enable SMT for GPIO13 0: Disable
12	GPIO12	GPIO12_SMT	1: Enable SMT for GPIO12 0: Disable
11	GPIO11	GPIO11_SMT	1: Enable SMT for GPIO11 0: Disable
10	GPIO10	GPIO10_SMT	1: Enable SMT for GPIO10 0: Disable
9	GPIO9	GPIO9_SMT	1: Enable SMT for GPIO9 0: Disable
8	GPIO8	GPIO8_SMT	1: Enable SMT for GPIO8 0: Disable

Bit(s)	Mnemonic	Name	Description
7	GPIO7	GPIO7_SMT	SMT for GPIO7 0: Disable 1: Enable
6	GPIO6	GPIO6_SMT	SMT for GPIO6 0: Disable 1: Enable
5	GPIO5	GPIO5_SMT	SMT for GPIO5 0: Disable 1: Enable
4	GPIO4	GPIO4_SMT	SMT for GPIO4 0: Disable 1: Enable
3	GPIO3	GPIO3_SMT	SMT for GPIO3 0: Disable 1: Enable
2	GPIO2	GPIO2_SMT	SMT for GPIO2 0: Disable 1: Enable
1	GPIO1	GPIO1_SMT	SMT for GPIO1 0: Disable 1: Enable
0	GPIO0	GPIO0_SMT	SMT for GPIO0 0: Disable 1: Enable

A0020604 **GPIO SMT0 SE** **GPIO SMT Control** **00000000**
I

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_SMT0

Bit(s)	Mnemonic	Name	Description
31	GPIO31	GPIO31_SMT	Bitwise SET operation of GPIO31 SMT 0: Keep 1: SET bits
30	GPIO30	GPIO30_SMT	Bitwise SET operation of GPIO30 SMT 0: Keep 1: SET bits
29	GPIO29	GPIO29_SMT	Bitwise SET operation of GPIO29 SMT 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
28	GPIO28	GPIO28_SMT	Bitwise SET operation of GPIO28 SMT 0: 1: SET bits	Keep
27	GPIO27	GPIO27_SMT	Bitwise SET operation of GPIO27 SMT 0: 1: SET bits	Keep
26	GPIO26	GPIO26_SMT	Bitwise SET operation of GPIO26 SMT 0: 1: SET bits	Keep
25	GPIO25	GPIO25_SMT	Bitwise SET operation of GPIO25 SMT 0: 1: SET bits	Keep
24	GPIO24	GPIO24_SMT	Bitwise SET operation of GPIO24 SMT 0: 1: SET bits	Keep
23	GPIO23	GPIO23_SMT	Bitwise SET operation of GPIO23 SMT 0: 1: SET bits	Keep
22	GPIO22	GPIO22_SMT	Bitwise SET operation of GPIO22 SMT 0: 1: SET bits	Keep
21	GPIO21	GPIO21_SMT	Bitwise SET operation of GPIO21 SMT 0: 1: SET bits	Keep
20	GPIO20	GPIO20_SMT	Bitwise SET operation of GPIO20 SMT 0: 1: SET bits	Keep
19	GPIO19	GPIO19_SMT	Bitwise SET operation of GPIO19 SMT 0: 1: SET bits	Keep
18	GPIO18	GPIO18_SMT	Bitwise SET operation of GPIO18 SMT 0: 1: SET bits	Keep
17	GPIO17	GPIO17_SMT	Bitwise SET operation of GPIO17 SMT 0: 1: SET bits	Keep
16	GPIO16	GPIO16_SMT	Bitwise SET operation of GPIO16 SMT 0: 1: SET bits	Keep
15	GPIO15	GPIO15_SMT	Bitwise SET operation of GPIO15 SMT 0: 1: SET bits	Keep
14	GPIO14	GPIO14_SMT	Bitwise SET operation of GPIO14 SMT 0: 1: SET bits	Keep
13	GPIO13	GPIO13_SMT	Bitwise SET operation of GPIO13 SMT 0: 1: SET bits	Keep
12	GPIO12	GPIO12_SMT	Bitwise SET operation of GPIO12 SMT 0: 1: SET bits	Keep
11	GPIO11	GPIO11_SMT	Bitwise SET operation of GPIO11 SMT	

Bit(s)	Mnemonic	Name	Description	
			0: 1: SET bits	Keep
10	GPIO10	GPIO10_SMT	Bitwise SET operation of GPIO10 SMT	
			0: 1: SET bits	Keep
9	GPIO9	GPIO9_SMT	Bitwise SET operation of GPIO9 SMT	
			0: 1: SET bits	Keep
8	GPIO8	GPIO8_SMT	Bitwise SET operation of GPIO8 SMT	
			0: 1: SET bits	Keep
7	GPIO7	GPIO7_SMT	Bitwise SET operation of GPIO7 SMT	
			0: 1: SET bits	Keep
6	GPIO6	GPIO6_SMT	Bitwise SET operation of GPIO6 SMT	
			0: 1: SET bits	Keep
5	GPIO5	GPIO5_SMT	Bitwise SET operation of GPIO5 SMT	
			0: 1: SET bits	Keep
4	GPIO4	GPIO4_SMT	Bitwise SET operation of GPIO4 SMT	
			0: 1: SET bits	Keep
3	GPIO3	GPIO3_SMT	Bitwise SET operation of GPIO3 SMT	
			0: 1: SET bits	Keep
2	GPIO2	GPIO2_SMT	Bitwise SET operation of GPIO2 SMT	
			0: 1: SET bits	Keep
1	GPIO1	GPIO1_SMT	Bitwise SET operation of GPIO1 SMT	
			0: 1: SET bits	Keep
0	GPIO0	GPIO0_SMT	Bitwise SET operation of GPIO0 SMT	
			0: 1: SET bits	Keep

A0020608 **GPIO_SMT0_CLR** GPIO SMT Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO3_1	GPIO3_0	GPIO2_9	GPIO2_8	GPIO2_7	GPIO2_6	GPIO2_5	GPIO2_4	GPIO2_3	GPIO2_2	GPIO2_1	GPIO2_0	GPIO1_9	GPIO1_8	GPIO1_7	GPIO1_6
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO1_5	GPIO1_4	GPIO1_3	GPIO1_2	GPIO1_1	GPIO1_0	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_SMT0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_SMT	Bitwise CLR operation of GPIO31 SMT 0: 1: CLR bits	Keep
30	GPIO30	GPIO30_SMT	Bitwise CLR operation of GPIO30 SMT 0: 1: CLR bits	Keep
29	GPIO29	GPIO29_SMT	Bitwise CLR operation of GPIO29 SMT 0: 1: CLR bits	Keep
28	GPIO28	GPIO28_SMT	Bitwise CLR operation of GPIO28 SMT 0: 1: CLR bits	Keep
27	GPIO27	GPIO27_SMT	Bitwise CLR operation of GPIO27 SMT 0: 1: CLR bits	Keep
26	GPIO26	GPIO26_SMT	Bitwise CLR operation of GPIO26 SMT 0: 1: CLR bits	Keep
25	GPIO25	GPIO25_SMT	Bitwise CLR operation of GPIO25 SMT 0: 1: CLR bits	Keep
24	GPIO24	GPIO24_SMT	Bitwise CLR operation of GPIO24 SMT 0: 1: CLR bits	Keep
23	GPIO23	GPIO23_SMT	Bitwise CLR operation of GPIO23 SMT 0: 1: CLR bits	Keep
22	GPIO22	GPIO22_SMT	Bitwise CLR operation of GPIO22 SMT 0: 1: CLR bits	Keep
21	GPIO21	GPIO21_SMT	Bitwise CLR operation of GPIO21 SMT 0: 1: CLR bits	Keep
20	GPIO20	GPIO20_SMT	Bitwise CLR operation of GPIO20 SMT 0: 1: CLR bits	Keep
19	GPIO19	GPIO19_SMT	Bitwise CLR operation of GPIO19 SMT 0: 1: CLR bits	Keep
18	GPIO18	GPIO18_SMT	Bitwise CLR operation of GPIO18 SMT 0: 1: CLR bits	Keep
17	GPIO17	GPIO17_SMT	Bitwise CLR operation of GPIO17 SMT 0: 1: CLR bits	Keep
16	GPIO16	GPIO16_SMT	Bitwise CLR operation of GPIO16 SMT 0: 1: CLR bits	Keep
15	GPIO15	GPIO15_SMT	Bitwise CLR operation of GPIO15 SMT 0: 1: CLR bits	Keep
14	GPIO14	GPIO14_SMT	Bitwise CLR operation of GPIO14 SMT	

Bit(s)	Mnemonic	Name	Description	
			0: 1: CLR bits	Keep
13	GPIO13	GPIO13_SMT	Bitwise CLR operation of GPIO13 SMT	
			0: 1: CLR bits	Keep
12	GPIO12	GPIO12_SMT	Bitwise CLR operation of GPIO12 SMT	
			0: 1: CLR bits	Keep
11	GPIO11	GPIO11_SMT	Bitwise CLR operation of GPIO11 SMT	
			0: 1: CLR bits	Keep
10	GPIO10	GPIO10_SMT	Bitwise CLR operation of GPIO10 SMT	
			0: 1: CLR bits	Keep
9	GPIO9	GPIO9_SMT	Bitwise CLR operation of GPIO9 SMT	
			0: 1: CLR bits	Keep
8	GPIO8	GPIO8_SMT	Bitwise CLR operation of GPIO8 SMT	
			0: 1: CLR bits	Keep
7	GPIO7	GPIO7_SMT	Bitwise CLR operation of GPIO7 SMT	
			0: 1: CLR bits	Keep
6	GPIO6	GPIO6_SMT	Bitwise CLR operation of GPIO6 SMT	
			0: 1: CLR bits	Keep
5	GPIO5	GPIO5_SMT	Bitwise CLR operation of GPIO5 SMT	
			0: 1: CLR bits	Keep
4	GPIO4	GPIO4_SMT	Bitwise CLR operation of GPIO4 SMT	
			0: 1: CLR bits	Keep
3	GPIO3	GPIO3_SMT	Bitwise CLR operation of GPIO3 SMT	
			0: 1: CLR bits	Keep
2	GPIO2	GPIO2_SMT	Bitwise CLR operation of GPIO2 SMT	
			0: 1: CLR bits	Keep
1	GPIO1	GPIO1_SMT	Bitwise CLR operation of GPIO1 SMT	
			0: 1: CLR bits	Keep
0	GPIO0	GPIO0_SMT	Bitwise CLR operation of GPIO0 SMT	
			0: 1: CLR bits	Keep

A0020610 GPIO SMT1 GPIO SMT Control												00000000				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO5 1	GPIO5 0	GPIO4 9	GPIO4 8
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO4_7	GPIO4_6	GPIO4_5	GPIO4_4	GPIO4_3	GPIO4_2	GPIO4_1	GPIO4_0	GPIO3_9	GPIO3_8	GPIO3_7	GPIO3_6	GPIO3_5	GPIO3_4	GPIO3_3	GPIO3_2
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO Schmit trigger control

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_SMT	SMT for GPIO51 0: 1: Enable	Disable
18	GPIO50	GPIO50_SMT	SMT for GPIO50 0: 1: Enable	Disable
17	GPIO49	GPIO49_SMT	SMT for GPIO49 0: 1: Enable	Disable
16	GPIO48	GPIO48_SMT	SMT for GPIO48 0: 1: Enable	Disable
15	GPIO47	GPIO47_SMT	SMT for GPIO47 0: 1: Enable	Disable
14	GPIO46	GPIO46_SMT	SMT for GPIO46 0: 1: Enable	Disable
13	GPIO45	GPIO45_SMT	SMT for GPIO45 0: 1: Enable	Disable
12	GPIO44	GPIO44_SMT	SMT for GPIO44 0: 1: Enable	Disable
11	GPIO43	GPIO43_SMT	SMT for GPIO43 0: 1: Enable	Disable
10	GPIO42	GPIO42_SMT	SMT for GPIO42 0: 1: Enable	Disable
9	GPIO41	GPIO41_SMT	SMT for GPIO41 0: 1: Enable	Disable
8	GPIO40	GPIO40_SMT	SMT for GPIO40 0: 1: Enable	Disable
7	GPIO39	GPIO39_SMT	SMT for GPIO39 0: 1: Enable	Disable
6	GPIO38	GPIO38_SMT	SMT for GPIO38 0: 1: Enable	Disable
5	GPIO37	GPIO37_SMT	SMT for GPIO37 0:	Disable

Bit(s)	Mnemonic	Name	Description
4	GPIO36	GPIO36_SMT	SMT for GPIO36 1: Enable 0: Disable
3	GPIO35	GPIO35_SMT	SMT for GPIO35 1: Enable 0: Disable
2	GPIO34	GPIO34_SMT	SMT for GPIO34 1: Enable 0: Disable
1	GPIO33	GPIO33_SMT	SMT for GPIO33 1: Enable 0: Disable
0	GPIO32	GPIO32_SMT	SMT for GPIO32 1: Enable 0: Disable

A0020614 GPIO_SMT1_SE **GPIO SMT Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_SMT1

Bit(s)	Mnemonic	Name	Description
19	GPIO51	GPIO51_SMT	Bitwise SET operation of GPIO51 SMT 0: Keep 1: SET bits
18	GPIO50	GPIO50_SMT	Bitwise SET operation of GPIO50 SMT 0: Keep 1: SET bits
17	GPIO49	GPIO49_SMT	Bitwise SET operation of GPIO49 SMT 0: Keep 1: SET bits
16	GPIO48	GPIO48_SMT	Bitwise SET operation of GPIO48 SMT 0: Keep 1: SET bits
15	GPIO47	GPIO47_SMT	Bitwise SET operation of GPIO47 SMT 0: Keep 1: SET bits
14	GPIO46	GPIO46_SMT	Bitwise SET operation of GPIO46 SMT 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
13	GPIO45	GPIO45_SMT	Bitwise SET operation of GPIO45 SMT 0: 1: SET bits	Keep
12	GPIO44	GPIO44_SMT	Bitwise SET operation of GPIO44 SMT 0: 1: SET bits	Keep
11	GPIO43	GPIO43_SMT	Bitwise SET operation of GPIO43 SMT 0: 1: SET bits	Keep
10	GPIO42	GPIO42_SMT	Bitwise SET operation of GPIO42 SMT 0: 1: SET bits	Keep
9	GPIO41	GPIO41_SMT	Bitwise SET operation of GPIO41 SMT 0: 1: SET bits	Keep
8	GPIO40	GPIO40_SMT	Bitwise SET operation of GPIO40 SMT 0: 1: SET bits	Keep
7	GPIO39	GPIO39_SMT	Bitwise SET operation of GPIO39 SMT 0: 1: SET bits	Keep
6	GPIO38	GPIO38_SMT	Bitwise SET operation of GPIO38 SMT 0: 1: SET bits	Keep
5	GPIO37	GPIO37_SMT	Bitwise SET operation of GPIO37 SMT 0: 1: SET bits	Keep
4	GPIO36	GPIO36_SMT	Bitwise SET operation of GPIO36 SMT 0: 1: SET bits	Keep
3	GPIO35	GPIO35_SMT	Bitwise SET operation of GPIO35 SMT 0: 1: SET bits	Keep
2	GPIO34	GPIO34_SMT	Bitwise SET operation of GPIO34 SMT 0: 1: SET bits	Keep
1	GPIO33	GPIO33_SMT	Bitwise SET operation of GPIO33 SMT 0: 1: SET bits	Keep
0	GPIO32	GPIO32_SMT	Bitwise SET operation of GPIO32 SMT 0: 1: SET bits	Keep

A0020618 **GPIO SMT1 CL** **GPIO SMT Control** **00000000**
R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO5	GPIO5	GPIO4	GPIO4
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	GPIO4 7	GPIO4 6	GPIO4 5	GPIO4 4	GPIO4 3	GPIO4 2	GPIO4 1	GPIO4 0	GPIO3 9	GPIO3 8	GPIO3 7	GPIO3 6	GPIO3 5	GPIO3 4	GPIO3 3	GPIO3 2
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_SMT1

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_SMT	Bitwise CLR operation of GPIO51 SMT 0: 1: CLR bits	Keep
18	GPIO50	GPIO50_SMT	Bitwise CLR operation of GPIO50 SMT 0: 1: CLR bits	Keep
17	GPIO49	GPIO49_SMT	Bitwise CLR operation of GPIO49 SMT 0: 1: CLR bits	Keep
16	GPIO48	GPIO48_SMT	Bitwise CLR operation of GPIO48 SMT 0: 1: CLR bits	Keep
15	GPIO47	GPIO47_SMT	Bitwise CLR operation of GPIO47 SMT 0: 1: CLR bits	Keep
14	GPIO46	GPIO46_SMT	Bitwise CLR operation of GPIO46 SMT 0: 1: CLR bits	Keep
13	GPIO45	GPIO45_SMT	Bitwise CLR operation of GPIO45 SMT 0: 1: CLR bits	Keep
12	GPIO44	GPIO44_SMT	Bitwise CLR operation of GPIO44 SMT 0: 1: CLR bits	Keep
11	GPIO43	GPIO43_SMT	Bitwise CLR operation of GPIO43 SMT 0: 1: CLR bits	Keep
10	GPIO42	GPIO42_SMT	Bitwise CLR operation of GPIO42 SMT 0: 1: CLR bits	Keep
9	GPIO41	GPIO41_SMT	Bitwise CLR operation of GPIO41 SMT 0: 1: CLR bits	Keep
8	GPIO40	GPIO40_SMT	Bitwise CLR operation of GPIO40 SMT 0: 1: CLR bits	Keep
7	GPIO39	GPIO39_SMT	Bitwise CLR operation of GPIO39 SMT 0: 1: CLR bits	Keep
6	GPIO38	GPIO38_SMT	Bitwise CLR operation of GPIO38 SMT 0: 1: CLR bits	Keep
5	GPIO37	GPIO37_SMT	Bitwise CLR operation of GPIO37 SMT 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
4	GPIO36	GPIO36_SMT	Bitwise CLR operation of GPIO36 SMT 0: 1: CLR bits	Keep
3	GPIO35	GPIO35_SMT	Bitwise CLR operation of GPIO35 SMT 0: 1: CLR bits	Keep
2	GPIO34	GPIO34_SMT	Bitwise CLR operation of GPIO34 SMT 0: 1: CLR bits	Keep
1	GPIO33	GPIO33_SMT	Bitwise CLR operation of GPIO33 SMT 0: 1: CLR bits	Keep
0	GPIO32	GPIO32_SMT	Bitwise CLR operation of GPIO32 SMT 0: 1: CLR bits	Keep

A0020700 GPIO_SR0 GPIO SR Control FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview: Configures GPIO slew rate control

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_SR	SR for GPIO31 0: 1: Enable	Disable
30	GPIO30	GPIO30_SR	SR for GPIO30 0: 1: Enable	Disable
29	GPIO29	GPIO29_SR	SR for GPIO29 0: 1: Enable	Disable
28	GPIO28	GPIO28_SR	SR for GPIO28 0: 1: Enable	Disable
27	GPIO27	GPIO27_SR	SR for GPIO27 0: 1: Enable	Disable
26	GPIO26	GPIO26_SR	SR for GPIO26 0: 1: Enable	Disable
25	GPIO25	GPIO25_SR	SR for GPIO25 0:	Disable

Bit(s)	Mnemonic	Name	Description
24	GPIO24	GPIO24_SR	1: Enable SR for GPIO24 0: Disable
23	GPIO23	GPIO23_SR	1: Enable SR for GPIO23 0: Disable
22	GPIO22	GPIO22_SR	1: Enable SR for GPIO22 0: Disable
21	GPIO21	GPIO21_SR	1: Enable SR for GPIO21 0: Disable
20	GPIO20	GPIO20_SR	1: Enable SR for GPIO20 0: Disable
19	GPIO19	GPIO19_SR	1: Enable SR for GPIO19 0: Disable
18	GPIO18	GPIO18_SR	1: Enable SR for GPIO18 0: Disable
17	GPIO17	GPIO17_SR	1: Enable SR for GPIO17 0: Disable
16	GPIO16	GPIO16_SR	1: Enable SR for GPIO16 0: Disable
15	GPIO15	GPIO15_SR	1: Enable SR for GPIO15 0: Disable
14	GPIO14	GPIO14_SR	1: Enable SR for GPIO14 0: Disable
13	GPIO13	GPIO13_SR	1: Enable SR for GPIO13 0: Disable
12	GPIO12	GPIO12_SR	1: Enable SR for GPIO12 0: Disable
11	GPIO11	GPIO11_SR	1: Enable SR for GPIO11 0: Disable
10	GPIO10	GPIO10_SR	1: Enable SR for GPIO10 0: Disable
9	GPIO9	GPIO9_SR	1: Enable SR for GPIO9 0: Disable
8	GPIO8	GPIO8_SR	1: Enable SR for GPIO8 0: Disable

Bit(s)	Mnemonic	Name	Description
7	GPIO7	GPIO7_SR	SR for GPIO7 0: Disable 1: Enable
6	GPIO6	GPIO6_SR	SR for GPIO6 0: Disable 1: Enable
5	GPIO5	GPIO5_SR	SR for GPIO5 0: Disable 1: Enable
4	GPIO4	GPIO4_SR	SR for GPIO4 0: Disable 1: Enable
3	GPIO3	GPIO3_SR	SR for GPIO3 0: Disable 1: Enable
2	GPIO2	GPIO2_SR	SR for GPIO2 0: Disable 1: Enable
1	GPIO1	GPIO1_SR	SR for GPIO1 0: Disable 1: Enable
0	GPIO0	GPIO0_SR	SR for GPIO0 0: Disable 1: Enable

A0020704 GPIO_SR0_SET GPIO SR Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_SR0

Bit(s)	Mnemonic	Name	Description
31	GPIO31	GPIO31_SR	Bitwise SET operation of GPIO31 SR 0: Keep 1: SET bits
30	GPIO30	GPIO30_SR	Bitwise SET operation of GPIO30 SR 0: Keep 1: SET bits
29	GPIO29	GPIO29_SR	Bitwise SET operation of GPIO29 SR 0: Keep 1: SET bits
28	GPIO28	GPIO28_SR	Bitwise SET operation of GPIO28 SR 0: Keep

Bit(s)	Mnemonic	Name	Description	
27	GPIO27	GPIO27_SR	1: SET bits Bitwise SET operation of GPIO27 SR 0: 1: SET bits	Keep
26	GPIO26	GPIO26_SR	1: SET bits Bitwise SET operation of GPIO26 SR 0: 1: SET bits	Keep
25	GPIO25	GPIO25_SR	1: SET bits Bitwise SET operation of GPIO25 SR 0: 1: SET bits	Keep
24	GPIO24	GPIO24_SR	1: SET bits Bitwise SET operation of GPIO24 SR 0: 1: SET bits	Keep
23	GPIO23	GPIO23_SR	1: SET bits Bitwise SET operation of GPIO23 SR 0: 1: SET bits	Keep
22	GPIO22	GPIO22_SR	1: SET bits Bitwise SET operation of GPIO22 SR 0: 1: SET bits	Keep
21	GPIO21	GPIO21_SR	1: SET bits Bitwise SET operation of GPIO21 SR 0: 1: SET bits	Keep
20	GPIO20	GPIO20_SR	1: SET bits Bitwise SET operation of GPIO20 SR 0: 1: SET bits	Keep
19	GPIO19	GPIO19_SR	1: SET bits Bitwise SET operation of GPIO19 SR 0: 1: SET bits	Keep
18	GPIO18	GPIO18_SR	1: SET bits Bitwise SET operation of GPIO18 SR 0: 1: SET bits	Keep
17	GPIO17	GPIO17_SR	1: SET bits Bitwise SET operation of GPIO17 SR 0: 1: SET bits	Keep
16	GPIO16	GPIO16_SR	1: SET bits Bitwise SET operation of GPIO16 SR 0: 1: SET bits	Keep
15	GPIO15	GPIO15_SR	1: SET bits Bitwise SET operation of GPIO15 SR 0: 1: SET bits	Keep
14	GPIO14	GPIO14_SR	1: SET bits Bitwise SET operation of GPIO14 SR 0: 1: SET bits	Keep
13	GPIO13	GPIO13_SR	1: SET bits Bitwise SET operation of GPIO13 SR 0: 1: SET bits	Keep
12	GPIO12	GPIO12_SR	1: SET bits Bitwise SET operation of GPIO12 SR 0: 1: SET bits	Keep
11	GPIO11	GPIO11_SR	1: SET bits Bitwise SET operation of GPIO11 SR 0: 1: SET bits	Keep

Bit(s)	Mnemonic	Name	Description	
10	GPIO10	GPIO10_SR	Bitwise SET operation of GPIO10 SR 0: 1: SET bits	Keep
9	GPIO9	GPIO9_SR	Bitwise SET operation of GPIO9 SR 0: 1: SET bits	Keep
8	GPIO8	GPIO8_SR	Bitwise SET operation of GPIO8 SR 0: 1: SET bits	Keep
7	GPIO7	GPIO7_SR	Bitwise SET operation of GPIO7 SR 0: 1: SET bits	Keep
6	GPIO6	GPIO6_SR	Bitwise SET operation of GPIO6 SR 0: 1: SET bits	Keep
5	GPIO5	GPIO5_SR	Bitwise SET operation of GPIO5 SR 0: 1: SET bits	Keep
4	GPIO4	GPIO4_SR	Bitwise SET operation of GPIO4 SR 0: 1: SET bits	Keep
3	GPIO3	GPIO3_SR	Bitwise SET operation of GPIO3 SR 0: 1: SET bits	Keep
2	GPIO2	GPIO2_SR	Bitwise SET operation of GPIO2 SR 0: 1: SET bits	Keep
1	GPIO1	GPIO1_SR	Bitwise SET operation of GPIO1 SR 0: 1: SET bits	Keep
0	GPIO0	GPIO0_SR	Bitwise SET operation of GPIO0 SR 0: 1: SET bits	Keep

A0020708 **GPIO_SR0_CLR** GPIO SR Control **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_SR0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_SR	Bitwise CLR operation of GPIO31 SR 0:	Keep

Bit(s)	Mnemonic	Name	Description	
30	GPIO30	GPIO30_SR	1: CLR bits Bitwise CLR operation of GPIO30 SR 0:	Keep
29	GPIO29	GPIO29_SR	1: CLR bits Bitwise CLR operation of GPIO29 SR 0:	Keep
28	GPIO28	GPIO28_SR	1: CLR bits Bitwise CLR operation of GPIO28 SR 0:	Keep
27	GPIO27	GPIO27_SR	1: CLR bits Bitwise CLR operation of GPIO27 SR 0:	Keep
26	GPIO26	GPIO26_SR	1: CLR bits Bitwise CLR operation of GPIO26 SR 0:	Keep
25	GPIO25	GPIO25_SR	1: CLR bits Bitwise CLR operation of GPIO25 SR 0:	Keep
24	GPIO24	GPIO24_SR	1: CLR bits Bitwise CLR operation of GPIO24 SR 0:	Keep
23	GPIO23	GPIO23_SR	1: CLR bits Bitwise CLR operation of GPIO23 SR 0:	Keep
22	GPIO22	GPIO22_SR	1: CLR bits Bitwise CLR operation of GPIO22 SR 0:	Keep
21	GPIO21	GPIO21_SR	1: CLR bits Bitwise CLR operation of GPIO21 SR 0:	Keep
20	GPIO20	GPIO20_SR	1: CLR bits Bitwise CLR operation of GPIO20 SR 0:	Keep
19	GPIO19	GPIO19_SR	1: CLR bits Bitwise CLR operation of GPIO19 SR 0:	Keep
18	GPIO18	GPIO18_SR	1: CLR bits Bitwise CLR operation of GPIO18 SR 0:	Keep
17	GPIO17	GPIO17_SR	1: CLR bits Bitwise CLR operation of GPIO17 SR 0:	Keep
16	GPIO16	GPIO16_SR	1: CLR bits Bitwise CLR operation of GPIO16 SR 0:	Keep
15	GPIO15	GPIO15_SR	1: CLR bits Bitwise CLR operation of GPIO15 SR 0:	Keep
14	GPIO14	GPIO14_SR	1: CLR bits Bitwise CLR operation of GPIO14 SR 0:	Keep

Bit(s)	Mnemonic	Name	Description	
13	GPIO13	GPIO13_SR	Bitwise CLR operation of GPIO13 SR 0: 1: CLR bits	Keep
12	GPIO12	GPIO12_SR	Bitwise CLR operation of GPIO12 SR 0: 1: CLR bits	Keep
11	GPIO11	GPIO11_SR	Bitwise CLR operation of GPIO11 SR 0: 1: CLR bits	Keep
10	GPIO10	GPIO10_SR	Bitwise CLR operation of GPIO10 SR 0: 1: CLR bits	Keep
9	GPIO9	GPIO9_SR	Bitwise CLR operation of GPIO9 SR 0: 1: CLR bits	Keep
8	GPIO8	GPIO8_SR	Bitwise CLR operation of GPIO8 SR 0: 1: CLR bits	Keep
7	GPIO7	GPIO7_SR	Bitwise CLR operation of GPIO7 SR 0: 1: CLR bits	Keep
6	GPIO6	GPIO6_SR	Bitwise CLR operation of GPIO6 SR 0: 1: CLR bits	Keep
5	GPIO5	GPIO5_SR	Bitwise CLR operation of GPIO5 SR 0: 1: CLR bits	Keep
4	GPIO4	GPIO4_SR	Bitwise CLR operation of GPIO4 SR 0: 1: CLR bits	Keep
3	GPIO3	GPIO3_SR	Bitwise CLR operation of GPIO3 SR 0: 1: CLR bits	Keep
2	GPIO2	GPIO2_SR	Bitwise CLR operation of GPIO2 SR 0: 1: CLR bits	Keep
1	GPIO1	GPIO1_SR	Bitwise CLR operation of GPIO1 SR 0: 1: CLR bits	Keep
0	GPIO0	GPIO0_SR	Bitwise CLR operation of GPIO0 SR 0: 1: CLR bits	Keep

A0020710		<u>GPIO SR1</u>				GPIO SR Control								000FF81F			
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														GPIO5	GPIO5	GPIO4	GPIO4
Type														RW	RW	RW	RW
Reset														1	1	1	1
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO4	GPIO4	GPIO4	GPIO4	GPIO4								GPIO3	GPIO3	GPIO3	GPIO3

	7	6	5	4	3							6	5	4	3	2
Type	RW	RW	RW	RW	RW							RW	RW	RW	RW	RW
Reset	1	1	1	1	1							1	1	1	1	1

Overview: Configures GPIO slew rate control

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_SR	SR for GPIO51 0: Disable 1: Enable	Disable
18	GPIO50	GPIO50_SR	SR for GPIO50 0: Disable 1: Enable	Disable
17	GPIO49	GPIO49_SR	SR for GPIO49 0: Disable 1: Enable	Disable
16	GPIO48	GPIO48_SR	SR for GPIO48 0: Disable 1: Enable	Disable
15	GPIO47	GPIO47_SR	SR for GPIO47 0: Disable 1: Enable	Disable
14	GPIO46	GPIO46_SR	SR for GPIO46 0: Disable 1: Enable	Disable
13	GPIO45	GPIO45_SR	SR for GPIO45 0: Disable 1: Enable	Disable
12	GPIO44	GPIO44_SR	SR for GPIO44 0: Disable 1: Enable	Disable
11	GPIO43	GPIO43_SR	SR for GPIO43 0: Disable 1: Enable	Disable
4	GPIO36	GPIO36_SR	SR for GPIO36 0: Disable 1: Enable	Disable
3	GPIO35	GPIO35_SR	SR for GPIO35 0: Disable 1: Enable	Disable
2	GPIO34	GPIO34_SR	SR for GPIO34 0: Disable 1: Enable	Disable
1	GPIO33	GPIO33_SR	SR for GPIO33 0: Disable 1: Enable	Disable
0	GPIO32	GPIO32_SR	SR for GPIO32 0: Disable 1: Enable	Disable

A0020714 GPIO_SR1_SET GPIO SR Control

00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43								GPIO36	GPIO35	GPIO34	GPIO33
Type	WO	WO	WO	WO	WO								WO	WO	WO	WO
Reset	0	0	0	0	0								0	0	0	0

Overview: For bitwise access of GPIO_SR1

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_SR	Bitwise SET operation of GPIO51 SR 0: 1: SET bits	Keep
18	GPIO50	GPIO50_SR	Bitwise SET operation of GPIO50 SR 0: 1: SET bits	Keep
17	GPIO49	GPIO49_SR	Bitwise SET operation of GPIO49 SR 0: 1: SET bits	Keep
16	GPIO48	GPIO48_SR	Bitwise SET operation of GPIO48 SR 0: 1: SET bits	Keep
15	GPIO47	GPIO47_SR	Bitwise SET operation of GPIO47 SR 0: 1: SET bits	Keep
14	GPIO46	GPIO46_SR	Bitwise SET operation of GPIO46 SR 0: 1: SET bits	Keep
13	GPIO45	GPIO45_SR	Bitwise SET operation of GPIO45 SR 0: 1: SET bits	Keep
12	GPIO44	GPIO44_SR	Bitwise SET operation of GPIO44 SR 0: 1: SET bits	Keep
11	GPIO43	GPIO43_SR	Bitwise SET operation of GPIO43 SR 0: 1: SET bits	Keep
4	GPIO36	GPIO36_SR	Bitwise SET operation of GPIO36 SR 0: 1: SET bits	Keep
3	GPIO35	GPIO35_SR	Bitwise SET operation of GPIO35 SR 0: 1: SET bits	Keep
2	GPIO34	GPIO34_SR	Bitwise SET operation of GPIO34 SR 0: 1: SET bits	Keep
1	GPIO33	GPIO33_SR	Bitwise SET operation of GPIO33 SR 0:	Keep

Bit(s)	Mnemonic	Name	Description
0	GPIO32	GPIO32_SR	Bitwise SET operation of GPIO32 SR 1: SET bits 0: 1: SET bits

Keep

A0020718 **GPIO_SR1_CLR** **GPIO SR Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO5	GPIO5	GPIO4	GPIO4
Type													1	0	9	8
Reset													WO	WO	WO	WO
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4								GPIO3	GPIO3	GPIO3	GPIO3
Type	7	6	5	4	3								6	5	4	3
Reset	WO	WO	WO	WO	WO								WO	WO	WO	WO
Reset	0	0	0	0	0								0	0	0	0

Overview: For bitwise access of GPIO_SR1

Bit(s)	Mnemonic	Name	Description
19	GPIO51	GPIO51_SR	Bitwise CLR operation of GPIO51 SR 0: 1: CLR bits
18	GPIO50	GPIO50_SR	Bitwise CLR operation of GPIO50 SR 0: 1: CLR bits
17	GPIO49	GPIO49_SR	Bitwise CLR operation of GPIO49 SR 0: 1: CLR bits
16	GPIO48	GPIO48_SR	Bitwise CLR operation of GPIO48 SR 0: 1: CLR bits
15	GPIO47	GPIO47_SR	Bitwise CLR operation of GPIO47 SR 0: 1: CLR bits
14	GPIO46	GPIO46_SR	Bitwise CLR operation of GPIO46 SR 0: 1: CLR bits
13	GPIO45	GPIO45_SR	Bitwise CLR operation of GPIO45 SR 0: 1: CLR bits
12	GPIO44	GPIO44_SR	Bitwise CLR operation of GPIO44 SR 0: 1: CLR bits
11	GPIO43	GPIO43_SR	Bitwise CLR operation of GPIO43 SR 0: 1: CLR bits
4	GPIO36	GPIO36_SR	Bitwise CLR operation of GPIO36 SR 0: 1: CLR bits

Keep

Keep

Keep

Keep

Keep

Keep

Keep

Keep

Keep

Keep

Bit(s)	Mnemonic	Name	Description	
3	GPIO35	GPIO35_SR	Bitwise CLR operation of GPIO35 SR 0: 1: CLR bits	Keep
2	GPIO34	GPIO34_SR	Bitwise CLR operation of GPIO34 SR 0: 1: CLR bits	Keep
1	GPIO33	GPIO33_SR	Bitwise CLR operation of GPIO33 SR 0: 1: CLR bits	Keep
0	GPIO32	GPIO32_SR	Bitwise CLR operation of GPIO32 SR 0: 1: CLR bits	Keep

A0020720 GPIO_SIM_SR GPIO SIM SR Control 003F003F

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37
Type											RW	RW	RW	RW	RW	RW
Reset											1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37
Type											RW	RW	RW	RW	RW	RW
Reset											1	1	1	1	1	1

Overview: Configures GPIO slew rate control for SIM IO

Bit(s)	Mnemonic	Name	Description	
21	GPIO42	GPIO42_SR1	SR1 control for GPIO42 0: 1: Enable	Disable
20	GPIO41	GPIO41_SR1	SR1 control for GPIO41 0: 1: Enable	Disable
19	GPIO40	GPIO40_SR1	SR1 control for GPIO40 0: 1: Enable	Disable
18	GPIO39	GPIO39_SR1	SR1 control for GPIO39 0: 1: Enable	Disable
17	GPIO38	GPIO38_SR1	SR1 control for GPIO38 0: 1: Enable	Disable
16	GPIO37	GPIO37_SR1	SR1 control for GPIO37 0: 1: Enable	Disable
5	GPIO42	GPIO42_SR0	SR0 control for GPIO42 0: 1: Enable	Disable
4	GPIO41	GPIO41_SR0	SR0 control for GPIO41 0:	Disable

Bit(s)	Mnemonic	Name	Description
3	GPIO40	GPIO40_SR0	SR0 control for GPIO40 1: Enable 0: Disable
2	GPIO39	GPIO39_SR0	SR0 control for GPIO39 1: Enable 0: Disable
1	GPIO38	GPIO38_SR0	SR0 control for GPIO38 1: Enable 0: Disable
0	GPIO37	GPIO37_SR0	SR0 control for GPIO37 1: Enable 0: Disable

A0020724 **GPIO_SIM_SR SET** GPIO SIM SR Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37
Type											WO	WO	WO	WO	WO	WO
Reset											0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37
Type											WO	WO	WO	WO	WO	WO
Reset											0	0	0	0	0	0

Overview: For bitwise access of GPIO_SIM_SR

Bit(s)	Mnemonic	Name	Description
21	GPIO42	GPIO42_SR1	Bitwise SET operation of GPIO42 SR1 control 0: Keep 1: SET bits
20	GPIO41	GPIO41_SR1	Bitwise SET operation of GPIO41 SR1 control 0: Keep 1: SET bits
19	GPIO40	GPIO40_SR1	Bitwise SET operation of GPIO40 SR1 control 0: Keep 1: SET bits
18	GPIO39	GPIO39_SR1	Bitwise SET operation of GPIO39 SR1 control 0: Keep 1: SET bits
17	GPIO38	GPIO38_SR1	Bitwise SET operation of GPIO38 SR1 control 0: Keep 1: SET bits
16	GPIO37	GPIO37_SR1	Bitwise SET operation of GPIO37 SR1 control 0: Keep 1: SET bits
5	GPIO42	GPIO42_SR0	Bitwise SET operation of GPIO42 SR0 control 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
4	GPIO41	GPIO41_SR0	Bitwise SET operation of GPIO41 SR0 control 0: 1: SET bits	Keep
3	GPIO40	GPIO40_SR0	Bitwise SET operation of GPIO40 SR0 control 0: 1: SET bits	Keep
2	GPIO39	GPIO39_SR0	Bitwise SET operation of GPIO39 SR0 control 0: 1: SET bits	Keep
1	GPIO38	GPIO38_SR0	Bitwise SET operation of GPIO38 SR0 control 0: 1: SET bits	Keep
0	GPIO37	GPIO37_SR0	Bitwise SET operation of GPIO37 SR0 control 0: 1: SET bits	Keep

A0020728 **GPIO_SIM_SR** **GPIO SIM SR Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37
Type											WO	WO	WO	WO	WO	WO
Reset											0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37
Type											WO	WO	WO	WO	WO	WO
Reset											0	0	0	0	0	0

Overview: For bitwise access of GPIO_SIM_SR

Bit(s)	Mnemonic	Name	Description	
21	GPIO42	GPIO42_SR1	Bitwise CLR operation of GPIO42 SR1 control 0: 1: CLR bits	Keep
20	GPIO41	GPIO41_SR1	Bitwise CLR operation of GPIO41 SR1 control 0: 1: CLR bits	Keep
19	GPIO40	GPIO40_SR1	Bitwise CLR operation of GPIO40 SR1 control 0: 1: CLR bits	Keep
18	GPIO39	GPIO39_SR1	Bitwise CLR operation of GPIO39 SR1 control 0: 1: CLR bits	Keep
17	GPIO38	GPIO38_SR1	Bitwise CLR operation of GPIO38 SR1 control 0: 1: CLR bits	Keep
16	GPIO37	GPIO37_SR1	Bitwise CLR operation of GPIO37 SR1 control 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
5	GPIO42	GPIO42_SR0	Bitwise CLR operation of GPIO42 SR0 control 0: 1: CLR bits	Keep
4	GPIO41	GPIO41_SR0	Bitwise CLR operation of GPIO41 SR0 control 0: 1: CLR bits	Keep
3	GPIO40	GPIO40_SR0	Bitwise CLR operation of GPIO40 SR0 control 0: 1: CLR bits	Keep
2	GPIO39	GPIO39_SR0	Bitwise CLR operation of GPIO39 SR0 control 0: 1: CLR bits	Keep
1	GPIO38	GPIO38_SR0	Bitwise CLR operation of GPIO38 SR0 control 0: 1: CLR bits	Keep
0	GPIO37	GPIO37_SR0	Bitwise CLR operation of GPIO37 SR0 control 0: 1: CLR bits	Keep

A0020800 GPIO_DRV0 GPIO DRV Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DRV0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRV0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO driving control

Bit(s)	Mnemonic	Name	Description
31:0	DRV0	GPIO_DRV0	GPIO_DRV0
			[1: 0]: GPIO_0
			[3: 2]: GPIO_1
			[5: 4]: GPIO_2
			[7: 6]: GPIO_3
			[9: 8]: GPIO_4
			[11: 10]: GPIO_5
			[13: 12]: GPIO_6
			[15: 14]: GPIO_7
			[17: 16]: GPIO_8
			[19: 18]: GPIO_9
			[21: 20]: URXD1
			[21: 20]: UTXD1
			[23: 22]: KCOL4
			[25: 24]: KCOL3
			[27: 26]: KCOL2
			[29: 28]: KCOL1
			[31: 30]: KCOL0

A0020804 GPIO_DRV0_SE GPIO DRV Control **00000000**
I

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DRV0[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRV0[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DRV0

Bit(s)	Mnemonic	Name	Description	
31:0		DRV0	Bitwise SET operation of GPIO_DRV0_SET	Keep
			0: 1: SET bits	

A0020808 GPIO_DRV0_CL GPIO DRV Control **00000000**
R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DRV0[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRV0[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_DRV0

Bit(s)	Mnemonic	Name	Description	
31:0		DRV0	Bitwise CLR operation of GPIO_DRV0_CLR	Keep
			0: 1: CLR bits	

A0020810 GPIO_DRV1 GPIO DRV Control **00C00000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DRV1[25:16]															
Type	RW															
Reset							0	0	1	1	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRV1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO driving control

Bit(s)	Mnemonic	Name	Description
--------	----------	------	-------------

Bit(s)	Mnemonic	Name	Description
25:0	DRV1	GPIO_DRV1	[1: 0]: KROW4 [3: 2]: KROW3 [5: 4]: KROW2 [7: 6]: KROW1 [9: 8]: KROW0 [11: 10]: BPI_BUS2 [11: 10]: BPI_BUS1 [11: 10]: BPI_BUS0 [13: 12]: CMRST [13: 12]: CMPDN [13: 12]: CMCS0 [13: 12]: CMCS1 [13: 12]: CMMCLK [13: 12]: CMCSK [15: 14]: MCK [15: 14]: MCKM0 [15: 14]: MCDA0 [15: 14]: MCDA1 [15: 14]: MCDA2 [15: 14]: MCDA3 [17: 16]: SIM1_SIO [17: 16]: SIM1_SRST [17: 16]: SIM1_SCLK [19: 18]: SIM2_SIO [19: 18]: SIM2_SRST [19: 18]: SIM2_SCLK [21: 20]: SCL28 [21: 20]: SDA28 [23: 22]: TESTMODE_D [23: 22]: LSCE_B [23: 22]: LSCK [23: 22]: LSDA [23: 22]: LSA0 [23: 22]: LPTE [25: 24]: RESETB

A0020814 **GPIO_DRV1_SE** **GPIO DRV Control** **00000000**
I

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DRV1[25:16]															
Type	WO															
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRV1[15:0]															
Type	WO															
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Overview: For bitwise access of GPIO_DRV1

Bit(s)	Mnemonic	Name	Description
25:0	DRV1	Bitwise SET operation of GPIO_DRV1_SET	0: Keep 1: SET bits

A0020818 GPIO_DRV1_CLR GPIO DRV Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DRV1[25:16]															
Type	WO															
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRV1[15:0]															
Type	WO															
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Overview: For bitwise access of GPIO_DRV1

Bit(s)	Mnemonic	Name	Description
25:0		DRV1	Bitwise CLR operation of GPIO_DRV1_CLR 0: 1: CLR bits

A0020900 GPIO_IES0 GPIO IES Control 43C00BFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16										
Name	GPIO30						GPIO25		GPIO24		GPIO23		GPIO22													
Type	RW						RW		RW		RW		RW													
Reset	1						1		1		1		1													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Name					GPIO11		GPIO9		GPIO8		GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
Type					RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW	
Reset					1		1		1		1		1		1		1		1		1		1		1	

Overview: Configures GPIO input enabling control

Bit(s)	Mnemonic	Name	Description
30	GPIO30	GPIO30_IES	Input buffer for GPIO30 0: Disable 1: Enable
25	GPIO25	GPIO25_IES	Input buffer for GPIO25 0: Disable 1: Enable
24	GPIO24	GPIO24_IES	Input buffer for GPIO24 0: Disable 1: Enable
23	GPIO23	GPIO23_IES	Input buffer for GPIO23 0: Disable 1: Enable
22	GPIO22	GPIO22_IES	Input buffer for GPIO22 0: Disable 1: Enable
11	GPIO11	GPIO11_IES	Input buffer for GPIO11 0: Disable

Bit(s)	Mnemonic	Name	Description
9	GPIO9	GPIO9_IES	Input buffer for GPIO9 1: Enable 0: Disable
8	GPIO8	GPIO8_IES	Input buffer for GPIO8 1: Enable 0: Disable
7	GPIO7	GPIO7_IES	Input buffer for GPIO7 1: Enable 0: Disable
6	GPIO6	GPIO6_IES	Input buffer for GPIO6 1: Enable 0: Disable
5	GPIO5	GPIO5_IES	Input buffer for GPIO5 1: Enable 0: Disable
4	GPIO4	GPIO4_IES	Input buffer for GPIO4 1: Enable 0: Disable
3	GPIO3	GPIO3_IES	Input buffer for GPIO3 1: Enable 0: Disable
2	GPIO2	GPIO2_IES	Input buffer for GPIO2 1: Enable 0: Disable
1	GPIO1	GPIO1_IES	Input buffer for GPIO1 1: Enable 0: Disable
0	GPIO0	GPIO0_IES	Input buffer for GPIO0 1: Enable 0: Disable

A0020904 **GPIO_IES0_SET** **GPIO IES Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO3 0					GPIO2 5	GPIO2 4	GPIO2 3	GPIO2 2						
Type		WO					WO	WO	WO	WO						
Reset		0					0	0	0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO1 1		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					WO		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset					0		0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_IES0

Bit(s)	Mnemonic	Name	Description
30	GPIO30	GPIO30_IES	Bitwise SET operation of GPIO30 input buffer 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
25	GPIO25	GPIO25_IES	Bitwise SET operation of GPIO25 input buffer 0: 1: SET bits	Keep
24	GPIO24	GPIO24_IES	Bitwise SET operation of GPIO24 input buffer 0: 1: SET bits	Keep
23	GPIO23	GPIO23_IES	Bitwise SET operation of GPIO23 input buffer 0: 1: SET bits	Keep
22	GPIO22	GPIO22_IES	Bitwise SET operation of GPIO22 input buffer 0: 1: SET bits	Keep
11	GPIO11	GPIO11_IES	Bitwise SET operation of GPIO11 input buffer 0: 1: SET bits	Keep
9	GPIO9	GPIO9_IES	Bitwise SET operation of GPIO9 input buffer 0: 1: SET bits	Keep
8	GPIO8	GPIO8_IES	Bitwise SET operation of GPIO8 input buffer 0: 1: SET bits	Keep
7	GPIO7	GPIO7_IES	Bitwise SET operation of GPIO7 input buffer 0: 1: SET bits	Keep
6	GPIO6	GPIO6_IES	Bitwise SET operation of GPIO6 input buffer 0: 1: SET bits	Keep
5	GPIO5	GPIO5_IES	Bitwise SET operation of GPIO5 input buffer 0: 1: SET bits	Keep
4	GPIO4	GPIO4_IES	Bitwise SET operation of GPIO4 input buffer 0: 1: SET bits	Keep
3	GPIO3	GPIO3_IES	Bitwise SET operation of GPIO3 input buffer 0: 1: SET bits	Keep
2	GPIO2	GPIO2_IES	Bitwise SET operation of GPIO2 input buffer 0: 1: SET bits	Keep
1	GPIO1	GPIO1_IES	Bitwise SET operation of GPIO1 input buffer 0: 1: SET bits	Keep
0	GPIO0	GPIO0_IES	Bitwise SET operation of GPIO0 input buffer 0: 1: SET bits	Keep

A0020908 **GPIO IES0_CL** **GPIO IES Control** **00000000**
R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO3					GPIO2	GPIO2	GPIO2	GPIO2						

		0					5	4	3	2						
Type		WO					WO	WO	WO	WO						
Reset		0					0	0	0	0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					GPIO1		GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type					WO		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset					0		0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_IES0

Bit(s)	Mnemonic	Name	Description	
30	GPIO30	GPIO30_IES	Bitwise CLR operation of GPIO30 input buffer 0: 1: CLR bits	Keep
25	GPIO25	GPIO25_IES	Bitwise CLR operation of GPIO25 input buffer 0: 1: CLR bits	Keep
24	GPIO24	GPIO24_IES	Bitwise CLR operation of GPIO24 input buffer 0: 1: CLR bits	Keep
23	GPIO23	GPIO23_IES	Bitwise CLR operation of GPIO23 input buffer 0: 1: CLR bits	Keep
22	GPIO22	GPIO22_IES	Bitwise CLR operation of GPIO22 input buffer 0: 1: CLR bits	Keep
11	GPIO11	GPIO11_IES	Bitwise CLR operation of GPIO11 input buffer 0: 1: CLR bits	Keep
9	GPIO9	GPIO9_IES	Bitwise CLR operation of GPIO9 input buffer 0: 1: CLR bits	Keep
8	GPIO8	GPIO8_IES	Bitwise CLR operation of GPIO8 input buffer 0: 1: CLR bits	Keep
7	GPIO7	GPIO7_IES	Bitwise CLR operation of GPIO7 input buffer 0: 1: CLR bits	Keep
6	GPIO6	GPIO6_IES	Bitwise CLR operation of GPIO6 input buffer 0: 1: CLR bits	Keep
5	GPIO5	GPIO5_IES	Bitwise CLR operation of GPIO5 input buffer 0: 1: CLR bits	Keep
4	GPIO4	GPIO4_IES	Bitwise CLR operation of GPIO4 input buffer 0: 1: CLR bits	Keep
3	GPIO3	GPIO3_IES	Bitwise CLR operation of GPIO3 input buffer 0: 1: CLR bits	Keep
2	GPIO2	GPIO2_IES	Bitwise CLR operation of GPIO2 input buffer 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
1	GPIO1	GPIO1_IES	Bitwise CLR operation of GPIO1 input buffer 0: 1: CLR bits	Keep
0	GPIO0	GPIO0_IES	Bitwise CLR operation of GPIO0 input buffer 0: 1: CLR bits	Keep

A0020910 **GPIO IES1** **GPIO IES Control** **00001FE0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO3	GPIO3	GPIO3					
Type				RW	RW	RW	RW	RW	RW	RW	RW					
Reset				1	1	1	1	1	1	1	1					

Overview: Configures GPIO input enabling control

Bit(s)	Mnemonic	Name	Description	
12	GPIO44	GPIO44_IES	Input buffer for GPIO44 0: 1: Enable	Disable
11	GPIO43	GPIO43_IES	Input buffer for GPIO43 0: 1: Enable	Disable
10	GPIO42	GPIO42_IES	Input buffer for GPIO42 0: 1: Enable	Disable
9	GPIO41	GPIO41_IES	Input buffer for GPIO41 0: 1: Enable	Disable
8	GPIO40	GPIO40_IES	Input buffer for GPIO40 0: 1: Enable	Disable
7	GPIO39	GPIO39_IES	Input buffer for GPIO39 0: 1: Enable	Disable
6	GPIO38	GPIO38_IES	Input buffer for GPIO38 0: 1: Enable	Disable
5	GPIO37	GPIO37_IES	Input buffer for GPIO37 0: 1: Enable	Disable

A0020914 **GPIO IES1_SET** **GPIO IES Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO4 4	GPIO4 3	GPIO4 2	GPIO4 1	GPIO4 0	GPIO3 9	GPIO3 8	GPIO3 7					
Type				WO	WO	WO	WO	WO	WO	WO	WO					
Reset				0	0	0	0	0	0	0	0					

Overview: For bitwise access of GPIO_IES1

Bit(s)	Mnemonic	Name	Description	
12	GPIO44	GPIO44_IES	Bitwise SET operation of GPIO44 input buffer 0: 1: SET bits	Keep
11	GPIO43	GPIO43_IES	Bitwise SET operation of GPIO43 input buffer 0: 1: SET bits	Keep
10	GPIO42	GPIO42_IES	Bitwise SET operation of GPIO42 input buffer 0: 1: SET bits	Keep
9	GPIO41	GPIO41_IES	Bitwise SET operation of GPIO41 input buffer 0: 1: SET bits	Keep
8	GPIO40	GPIO40_IES	Bitwise SET operation of GPIO40 input buffer 0: 1: SET bits	Keep
7	GPIO39	GPIO39_IES	Bitwise SET operation of GPIO39 input buffer 0: 1: SET bits	Keep
6	GPIO38	GPIO38_IES	Bitwise SET operation of GPIO38 input buffer 0: 1: SET bits	Keep
5	GPIO37	GPIO37_IES	Bitwise SET operation of GPIO37 input buffer 0: 1: SET bits	Keep

A0020918 GPIO_IES1_CL GPIO IES Control 00000000
R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				GPIO4 4	GPIO4 3	GPIO4 2	GPIO4 1	GPIO4 0	GPIO3 9	GPIO3 8	GPIO3 7					
Type				WO	WO	WO	WO	WO	WO	WO	WO					
Reset				0	0	0	0	0	0	0	0					

Overview: For bitwise access of GPIO_IES1

Bit(s)	Mnemonic	Name	Description
--------	----------	------	-------------

Bit(s)	Mnemonic	Name	Description	
12	GPIO44	GPIO44_IES	Bitwise CLR operation of GPIO44 input buffer 0: 1: CLR bits	Keep
11	GPIO43	GPIO43_IES	Bitwise CLR operation of GPIO43 input buffer 0: 1: CLR bits	Keep
10	GPIO42	GPIO42_IES	Bitwise CLR operation of GPIO42 input buffer 0: 1: CLR bits	Keep
9	GPIO41	GPIO41_IES	Bitwise CLR operation of GPIO41 input buffer 0: 1: CLR bits	Keep
8	GPIO40	GPIO40_IES	Bitwise CLR operation of GPIO40 input buffer 0: 1: CLR bits	Keep
7	GPIO39	GPIO39_IES	Bitwise CLR operation of GPIO39 input buffer 0: 1: CLR bits	Keep
6	GPIO38	GPIO38_IES	Bitwise CLR operation of GPIO38 input buffer 0: 1: CLR bits	Keep
5	GPIO37	GPIO37_IES	Bitwise CLR operation of GPIO37 input buffer 0: 1: CLR bits	Keep

A0020A00 **GPIO_PUPD0** **GPIO PUPD Control** **303E0000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO3 1		GPIO2 9	GPIO2 8	GPIO2 7	GPIO2 6					GPIO2 1	GPIO2 0	GPIO1 9	GPIO1 8	GPIO1 7	GPIO1 6
Type	RW		RW	RW	RW	RW					RW	RW	RW	RW	RW	RW
Reset	0		1	1	0	0					1	1	1	1	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO1 5	GPIO1 4	GPIO1 3	GPIO1 2		GPIO1 0										
Type	RW	RW	RW	RW		RW										
Reset	0	0	0	0		0										

Overview: Configures GPIO PUPD control

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_PUPD	PUPD for GPIO31 0: 1: Enable	Disable
29	GPIO29	GPIO29_PUPD	PUPD for GPIO29 0: 1: Enable	Disable
28	GPIO28	GPIO28_PUPD	PUPD for GPIO28 0: 1: Enable	Disable
27	GPIO27	GPIO27_PUPD	PUPD for GPIO27 0:	Disable

Bit(s)	Mnemonic	Name	Description
26	GPIO26	GPIO26_PUPD	1: Enable PUPD for GPIO26 0: Disable 1: Enable
21	GPIO21	GPIO21_PUPD	1: Enable PUPD for GPIO21 0: Disable 1: Enable
20	GPIO20	GPIO20_PUPD	1: Enable PUPD for GPIO20 0: Disable 1: Enable
19	GPIO19	GPIO19_PUPD	1: Enable PUPD for GPIO19 0: Disable 1: Enable
18	GPIO18	GPIO18_PUPD	1: Enable PUPD for GPIO18 0: Disable 1: Enable
17	GPIO17	GPIO17_PUPD	1: Enable PUPD for GPIO17 0: Disable 1: Enable
16	GPIO16	GPIO16_PUPD	1: Enable PUPD for GPIO16 0: Disable 1: Enable
15	GPIO15	GPIO15_PUPD	1: Enable PUPD for GPIO15 0: Disable 1: Enable
14	GPIO14	GPIO14_PUPD	1: Enable PUPD for GPIO14 0: Disable 1: Enable
13	GPIO13	GPIO13_PUPD	1: Enable PUPD for GPIO13 0: Disable 1: Enable
12	GPIO12	GPIO12_PUPD	1: Enable PUPD for GPIO12 0: Disable 1: Enable
10	GPIO10	GPIO10_PUPD	1: Enable PUPD for GPIO10 0: Disable 1: Enable

A0020A04 GPIO PUPD0 S **GPIO PUPD Control** **00000000**
ET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO3 1		GPIO2 9	GPIO2 8	GPIO2 7	GPIO2 6					GPIO2 1	GPIO2 0	GPIO1 9	GPIO1 8	GPIO1 7	GPIO1 6
Type	WO		WO	WO	WO	WO					WO	WO	WO	WO	WO	WO
Reset	0		0	0	0	0					0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO1 5	GPIO1 4	GPIO1 3	GPIO1 2		GPIO1 0										
Type	WO	WO	WO	WO		WO										
Reset	0	0	0	0		0										

Overview: For bitwise access of GPIO_PUPD0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_PUPD	Bitwise SET operation of GPIO31 PUPD 0: 1: SET bits	Keep
29	GPIO29	GPIO29_PUPD	Bitwise SET operation of GPIO29 PUPD 0: 1: SET bits	Keep
28	GPIO28	GPIO28_PUPD	Bitwise SET operation of GPIO28 PUPD 0: 1: SET bits	Keep
27	GPIO27	GPIO27_PUPD	Bitwise SET operation of GPIO27 PUPD 0: 1: SET bits	Keep
26	GPIO26	GPIO26_PUPD	Bitwise SET operation of GPIO26 PUPD 0: 1: SET bits	Keep
21	GPIO21	GPIO21_PUPD	Bitwise SET operation of GPIO21 PUPD 0: 1: SET bits	Keep
20	GPIO20	GPIO20_PUPD	Bitwise SET operation of GPIO20 PUPD 0: 1: SET bits	Keep
19	GPIO19	GPIO19_PUPD	Bitwise SET operation of GPIO19 PUPD 0: 1: SET bits	Keep
18	GPIO18	GPIO18_PUPD	Bitwise SET operation of GPIO18 PUPD 0: 1: SET bits	Keep
17	GPIO17	GPIO17_PUPD	Bitwise SET operation of GPIO17 PUPD 0: 1: SET bits	Keep
16	GPIO16	GPIO16_PUPD	Bitwise SET operation of GPIO16 PUPD 0: 1: SET bits	Keep
15	GPIO15	GPIO15_PUPD	Bitwise SET operation of GPIO15 PUPD 0: 1: SET bits	Keep
14	GPIO14	GPIO14_PUPD	Bitwise SET operation of GPIO14 PUPD 0: 1: SET bits	Keep
13	GPIO13	GPIO13_PUPD	Bitwise SET operation of GPIO13 PUPD 0: 1: SET bits	Keep
12	GPIO12	GPIO12_PUPD	Bitwise SET operation of GPIO12 PUPD 0: 1: SET bits	Keep
10	GPIO10	GPIO10_PUPD	Bitwise SET operation of GPIO10 PUPD 0: 1: SET bits	Keep

A0020A08 GPIO_PUPD0_C GPIO PUPD Control 00000000
LR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31		GPIO29	GPIO28	GPIO27	GPIO26					GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	WO		WO	WO	WO	WO					WO	WO	WO	WO	WO	WO
Reset	0		0	0	0	0					0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12		GPIO10										
Type	WO	WO	WO	WO		WO										
Reset	0	0	0	0		0										

Overview: For bitwise access of GPIO_PUPD0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_PUPD	Bitwise CLR operation of GPIO31 PUPD 0: 1: CLR bits	Keep
29	GPIO29	GPIO29_PUPD	Bitwise CLR operation of GPIO29 PUPD 0: 1: CLR bits	Keep
28	GPIO28	GPIO28_PUPD	Bitwise CLR operation of GPIO28 PUPD 0: 1: CLR bits	Keep
27	GPIO27	GPIO27_PUPD	Bitwise CLR operation of GPIO27 PUPD 0: 1: CLR bits	Keep
26	GPIO26	GPIO26_PUPD	Bitwise CLR operation of GPIO26 PUPD 0: 1: CLR bits	Keep
21	GPIO21	GPIO21_PUPD	Bitwise CLR operation of GPIO21 PUPD 0: 1: CLR bits	Keep
20	GPIO20	GPIO20_PUPD	Bitwise CLR operation of GPIO20 PUPD 0: 1: CLR bits	Keep
19	GPIO19	GPIO19_PUPD	Bitwise CLR operation of GPIO19 PUPD 0: 1: CLR bits	Keep
18	GPIO18	GPIO18_PUPD	Bitwise CLR operation of GPIO18 PUPD 0: 1: CLR bits	Keep
17	GPIO17	GPIO17_PUPD	Bitwise CLR operation of GPIO17 PUPD 0: 1: CLR bits	Keep
16	GPIO16	GPIO16_PUPD	Bitwise CLR operation of GPIO16 PUPD 0: 1: CLR bits	Keep
15	GPIO15	GPIO15_PUPD	Bitwise CLR operation of GPIO15 PUPD 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
14	GPIO14	GPIO14_PUPD	Bitwise CLR operation of GPIO14 PUPD 0: 1: CLR bits	Keep
13	GPIO13	GPIO13_PUPD	Bitwise CLR operation of GPIO13 PUPD 0: 1: CLR bits	Keep
12	GPIO12	GPIO12_PUPD	Bitwise CLR operation of GPIO12 PUPD 0: 1: CLR bits	Keep
10	GPIO10	GPIO10_PUPD	Bitwise CLR operation of GPIO10 PUPD 0: 1: CLR bits	Keep

A0020A10 GPIO_PUPD1 GPIO PUPD Control 0007A7FE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO5	GPIO5	GPIO4	GPIO4
Type													RW	RW	RW	RW
Reset													0	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO4	GPIO4	GPIO4			GPIO4	GPIO4	GPIO4	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3
Type	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	0	1			1	1	1	1	1	1	1	1	1	1	0

Overview: Configures GPIO PUPD control

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_PUPD	PUPD for GPIO51 0: 1: Enable	Disable
18	GPIO50	GPIO50_PUPD	PUPD for GPIO50 0: 1: Enable	Disable
17	GPIO49	GPIO49_PUPD	PUPD for GPIO49 0: 1: Enable	Disable
16	GPIO48	GPIO48_PUPD	PUPD for GPIO48 0: 1: Enable	Disable
15	GPIO47	GPIO47_PUPD	PUPD for GPIO47 0: 1: Enable	Disable
14	GPIO46	GPIO46_PUPD	PUPD for GPIO46 0: 1: Enable	Disable
13	GPIO45	GPIO45_PUPD	PUPD for GPIO45 0: 1: Enable	Disable
10	GPIO42	GPIO42_PUPD	PUPD for GPIO42 0:	Disable

Bit(s)	Mnemonic	Name	Description
9	GPIO41	GPIO41_PUPD	PUPD for GPIO41 1: Enable 0: Disable
8	GPIO40	GPIO40_PUPD	PUPD for GPIO40 1: Enable 0: Disable
7	GPIO39	GPIO39_PUPD	PUPD for GPIO39 1: Enable 0: Disable
6	GPIO38	GPIO38_PUPD	PUPD for GPIO38 1: Enable 0: Disable
5	GPIO37	GPIO37_PUPD	PUPD for GPIO37 1: Enable 0: Disable
4	GPIO36	GPIO36_PUPD	PUPD for GPIO36 1: Enable 0: Disable
3	GPIO35	GPIO35_PUPD	PUPD for GPIO35 1: Enable 0: Disable
2	GPIO34	GPIO34_PUPD	PUPD for GPIO34 1: Enable 0: Disable
1	GPIO33	GPIO33_PUPD	PUPD for GPIO33 1: Enable 0: Disable
0	GPIO32	GPIO32_PUPD	PUPD for GPIO32 1: Enable 0: Disable

A0020A14 **GPIO_PUPD1 SET** **GPIO PUPD Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	WO	WO	WO			WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0			0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_PUPD1

Bit(s)	Mnemonic	Name	Description
19	GPIO51	GPIO51_PUPD	Bitwise SET operation of GPIO51 PUPD 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
18	GPIO50	GPIO50_PUPD	Bitwise SET operation of GPIO50 PUPD 0: 1: SET bits	Keep
17	GPIO49	GPIO49_PUPD	Bitwise SET operation of GPIO49 PUPD 0: 1: SET bits	Keep
16	GPIO48	GPIO48_PUPD	Bitwise SET operation of GPIO48 PUPD 0: 1: SET bits	Keep
15	GPIO47	GPIO47_PUPD	Bitwise SET operation of GPIO47 PUPD 0: 1: SET bits	Keep
14	GPIO46	GPIO46_PUPD	Bitwise SET operation of GPIO46 PUPD 0: 1: SET bits	Keep
13	GPIO45	GPIO45_PUPD	Bitwise SET operation of GPIO45 PUPD 0: 1: SET bits	Keep
10	GPIO42	GPIO42_PUPD	Bitwise SET operation of GPIO42 PUPD 0: 1: SET bits	Keep
9	GPIO41	GPIO41_PUPD	Bitwise SET operation of GPIO41 PUPD 0: 1: SET bits	Keep
8	GPIO40	GPIO40_PUPD	Bitwise SET operation of GPIO40 PUPD 0: 1: SET bits	Keep
7	GPIO39	GPIO39_PUPD	Bitwise SET operation of GPIO39 PUPD 0: 1: SET bits	Keep
6	GPIO38	GPIO38_PUPD	Bitwise SET operation of GPIO38 PUPD 0: 1: SET bits	Keep
5	GPIO37	GPIO37_PUPD	Bitwise SET operation of GPIO37 PUPD 0: 1: SET bits	Keep
4	GPIO36	GPIO36_PUPD	Bitwise SET operation of GPIO36 PUPD 0: 1: SET bits	Keep
3	GPIO35	GPIO35_PUPD	Bitwise SET operation of GPIO35 PUPD 0: 1: SET bits	Keep
2	GPIO34	GPIO34_PUPD	Bitwise SET operation of GPIO34 PUPD 0: 1: SET bits	Keep
1	GPIO33	GPIO33_PUPD	Bitwise SET operation of GPIO33 PUPD 0: 1: SET bits	Keep
0	GPIO32	GPIO32_PUPD	Bitwise SET operation of GPIO32 PUPD 0: 1: SET bits	Keep

A0020A18 GPIO_PUPD1_C GPIO PUPD Control 00000000
LR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	WO	WO	WO			WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0			0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_PUPD1

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_PUPD	Bitwise CLR operation of GPIO51 PUPD 0: 1: CLR bits	Keep
18	GPIO50	GPIO50_PUPD	Bitwise CLR operation of GPIO50 PUPD 0: 1: CLR bits	Keep
17	GPIO49	GPIO49_PUPD	Bitwise CLR operation of GPIO49 PUPD 0: 1: CLR bits	Keep
16	GPIO48	GPIO48_PUPD	Bitwise CLR operation of GPIO48 PUPD 0: 1: CLR bits	Keep
15	GPIO47	GPIO47_PUPD	Bitwise CLR operation of GPIO47 PUPD 0: 1: CLR bits	Keep
14	GPIO46	GPIO46_PUPD	Bitwise CLR operation of GPIO46 PUPD 0: 1: CLR bits	Keep
13	GPIO45	GPIO45_PUPD	Bitwise CLR operation of GPIO45 PUPD 0: 1: CLR bits	Keep
10	GPIO42	GPIO42_PUPD	Bitwise CLR operation of GPIO42 PUPD 0: 1: CLR bits	Keep
9	GPIO41	GPIO41_PUPD	Bitwise CLR operation of GPIO41 PUPD 0: 1: CLR bits	Keep
8	GPIO40	GPIO40_PUPD	Bitwise CLR operation of GPIO40 PUPD 0: 1: CLR bits	Keep
7	GPIO39	GPIO39_PUPD	Bitwise CLR operation of GPIO39 PUPD 0: 1: CLR bits	Keep
6	GPIO38	GPIO38_PUPD	Bitwise CLR operation of GPIO38 PUPD 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
5	GPIO37	GPIO37_PUPD	Bitwise CLR operation of GPIO37 PUPD 0: 1: CLR bits	Keep
4	GPIO36	GPIO36_PUPD	Bitwise CLR operation of GPIO36 PUPD 0: 1: CLR bits	Keep
3	GPIO35	GPIO35_PUPD	Bitwise CLR operation of GPIO35 PUPD 0: 1: CLR bits	Keep
2	GPIO34	GPIO34_PUPD	Bitwise CLR operation of GPIO34 PUPD 0: 1: CLR bits	Keep
1	GPIO33	GPIO33_PUPD	Bitwise CLR operation of GPIO33 PUPD 0: 1: CLR bits	Keep
0	GPIO32	GPIO32_PUPD	Bitwise CLR operation of GPIO32 PUPD 0: 1: CLR bits	Keep

A0020B00 GPIO_RESEN0 **GPIO R0 Control** **B83FF400**
0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31		GPIO29	GPIO28	GPIO27	GPIO26					GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	RW		RW	RW	RW	RW					RW	RW	RW	RW	RW	RW
Reset	1		1	1	1	0					1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12		GPIO10										
Type	RW	RW	RW	RW		RW										
Reset	1	1	1	1		1										

Overview: Configures GPIO R0 control

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_R0	R0 for GPIO31 0: 1: Enable	Disable
29	GPIO29	GPIO29_R0	R0 for GPIO29 0: 1: Enable	Disable
28	GPIO28	GPIO28_R0	R0 for GPIO28 0: 1: Enable	Disable
27	GPIO27	GPIO27_R0	R0 for GPIO27 0: 1: Enable	Disable
26	GPIO26	GPIO26_R0	R0 for GPIO26 0: 1: Enable	Disable

Bit(s)	Mnemonic	Name	Description
21	GPIO21	GPIO21_R0	R0 for GPIO21 0: Disable 1: Enable
20	GPIO20	GPIO20_R0	R0 for GPIO20 0: Disable 1: Enable
19	GPIO19	GPIO19_R0	R0 for GPIO19 0: Disable 1: Enable
18	GPIO18	GPIO18_R0	R0 for GPIO18 0: Disable 1: Enable
17	GPIO17	GPIO17_R0	R0 for GPIO17 0: Disable 1: Enable
16	GPIO16	GPIO16_R0	R0 for GPIO16 0: Disable 1: Enable
15	GPIO15	GPIO15_R0	R0 for GPIO15 0: Disable 1: Enable
14	GPIO14	GPIO14_R0	R0 for GPIO14 0: Disable 1: Enable
13	GPIO13	GPIO13_R0	R0 for GPIO13 0: Disable 1: Enable
12	GPIO12	GPIO12_R0	R0 for GPIO12 0: Disable 1: Enable
10	GPIO10	GPIO10_R0	R0 for GPIO10 0: Disable 1: Enable

A0020B04 **GPIO_RESEN0** **GPIO R0 Control** **00000000**
0 SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO3 1		GPIO2 9	GPIO2 8	GPIO2 7	GPIO2 6					GPIO2 1	GPIO2 0	GPIO1 9	GPIO1 8	GPIO1 7	GPIO1 6
Type	WO		WO	WO	WO	WO					WO	WO	WO	WO	WO	WO
Reset	0		0	0	0	0					0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO1 5	GPIO1 4	GPIO1 3	GPIO1 2		GPIO1 0										
Type	WO	WO	WO	WO		WO										
Reset	0	0	0	0		0										

Overview: For bitwise access of GPIO_RESEN0_0

Bit(s)	Mnemonic	Name	Description
--------	----------	------	-------------

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_R0	Bitwise SET operation of GPIO31 R0 0: 1: SET bits	Keep
29	GPIO29	GPIO29_R0	Bitwise SET operation of GPIO29 R0 0: 1: SET bits	Keep
28	GPIO28	GPIO28_R0	Bitwise SET operation of GPIO28 R0 0: 1: SET bits	Keep
27	GPIO27	GPIO27_R0	Bitwise SET operation of GPIO27 R0 0: 1: SET bits	Keep
26	GPIO26	GPIO26_R0	Bitwise SET operation of GPIO26 R0 0: 1: SET bits	Keep
21	GPIO21	GPIO21_R0	Bitwise SET operation of GPIO21 R0 0: 1: SET bits	Keep
20	GPIO20	GPIO20_R0	Bitwise SET operation of GPIO20 R0 0: 1: SET bits	Keep
19	GPIO19	GPIO19_R0	Bitwise SET operation of GPIO19 R0 0: 1: SET bits	Keep
18	GPIO18	GPIO18_R0	Bitwise SET operation of GPIO18 R0 0: 1: SET bits	Keep
17	GPIO17	GPIO17_R0	Bitwise SET operation of GPIO17 R0 0: 1: SET bits	Keep
16	GPIO16	GPIO16_R0	Bitwise SET operation of GPIO16 R0 0: 1: SET bits	Keep
15	GPIO15	GPIO15_R0	Bitwise SET operation of GPIO15 R0 0: 1: SET bits	Keep
14	GPIO14	GPIO14_R0	Bitwise SET operation of GPIO14 R0 0: 1: SET bits	Keep
13	GPIO13	GPIO13_R0	Bitwise SET operation of GPIO13 R0 0: 1: SET bits	Keep
12	GPIO12	GPIO12_R0	Bitwise SET operation of GPIO12 R0 0: 1: SET bits	Keep
10	GPIO10	GPIO10_R0	Bitwise SET operation of GPIO10 R0 0: 1: SET bits	Keep

A0020B08 **GPIO_RESEN0** **GPIO R0 Control**

00000000

0_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31		GPIO29	GPIO28	GPIO27	GPIO26					GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	WO		WO	WO	WO	WO					WO	WO	WO	WO	WO	WO
Reset	0		0	0	0	0					0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12		GPIO10										
Type	WO	WO	WO	WO		WO										
Reset	0	0	0	0		0										

Overview: For bitwise access of GPIO_RESEN0_0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_R0	Bitwise CLR operation of GPIO31 R0 0: 1: CLR bits	Keep
29	GPIO29	GPIO29_R0	Bitwise CLR operation of GPIO29 R0 0: 1: CLR bits	Keep
28	GPIO28	GPIO28_R0	Bitwise CLR operation of GPIO28 R0 0: 1: CLR bits	Keep
27	GPIO27	GPIO27_R0	Bitwise CLR operation of GPIO27 R0 0: 1: CLR bits	Keep
26	GPIO26	GPIO26_R0	Bitwise CLR operation of GPIO26 R0 0: 1: CLR bits	Keep
21	GPIO21	GPIO21_R0	Bitwise CLR operation of GPIO21 R0 0: 1: CLR bits	Keep
20	GPIO20	GPIO20_R0	Bitwise CLR operation of GPIO20 R0 0: 1: CLR bits	Keep
19	GPIO19	GPIO19_R0	Bitwise CLR operation of GPIO19 R0 0: 1: CLR bits	Keep
18	GPIO18	GPIO18_R0	Bitwise CLR operation of GPIO18 R0 0: 1: CLR bits	Keep
17	GPIO17	GPIO17_R0	Bitwise CLR operation of GPIO17 R0 0: 1: CLR bits	Keep
16	GPIO16	GPIO16_R0	Bitwise CLR operation of GPIO16 R0 0: 1: CLR bits	Keep
15	GPIO15	GPIO15_R0	Bitwise CLR operation of GPIO15 R0 0: 1: CLR bits	Keep
14	GPIO14	GPIO14_R0	Bitwise CLR operation of GPIO14 R0 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description
13	GPIO13	GPIO13_R0	Bitwise CLR operation of GPIO13 R0 0: 1: CLR bits
12	GPIO12	GPIO12_R0	Bitwise CLR operation of GPIO12 R0 0: 1: CLR bits
10	GPIO10	GPIO10_R0	Bitwise CLR operation of GPIO10 R0 0: 1: CLR bits

A0020B10 GPIO_RESEN0 GPIO R0 Control 0007A7FF
1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													RW	RW	RW	RW
Reset													0	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34
Type	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	0	1			1	1	1	1	1	1	1	1	1	1	1

Overview: Configures GPIO R0 control

Bit(s)	Mnemonic	Name	Description
19	GPIO51	GPIO51_R0	R0 for GPIO51 0: 1: Enable
18	GPIO50	GPIO50_R0	R0 for GPIO50 0: 1: Enable
17	GPIO49	GPIO49_R0	R0 for GPIO49 0: 1: Enable
16	GPIO48	GPIO48_R0	R0 for GPIO48 0: 1: Enable
15	GPIO47	GPIO47_R0	R0 for GPIO47 0: 1: Enable
14	GPIO46	GPIO46_R0	R0 for GPIO46 0: 1: Enable
13	GPIO45	GPIO45_R0	R0 for GPIO45 0: 1: Enable
10	GPIO42	GPIO42_R0	R0 for GPIO42 0: 1: Enable

Bit(s)	Mnemonic	Name	Description
9	GPIO41	GPIO41_R0	R0 for GPIO41 0: Disable 1: Enable
8	GPIO40	GPIO40_R0	R0 for GPIO40 0: Disable 1: Enable
7	GPIO39	GPIO39_R0	R0 for GPIO39 0: Disable 1: Enable
6	GPIO38	GPIO38_R0	R0 for GPIO38 0: Disable 1: Enable
5	GPIO37	GPIO37_R0	R0 for GPIO37 0: Disable 1: Enable
4	GPIO36	GPIO36_R0	R0 for GPIO36 0: Disable 1: Enable
3	GPIO35	GPIO35_R0	R0 for GPIO35 0: Disable 1: Enable
2	GPIO34	GPIO34_R0	R0 for GPIO34 0: Disable 1: Enable
1	GPIO33	GPIO33_R0	R0 for GPIO33 0: Disable 1: Enable
0	GPIO32	GPIO32_R0	R0 for GPIO32 0: Disable 1: Enable

A0020B14 **GPIO_RESEN0_1_SET** **GPIO R0 Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34
Type	WO	WO	WO			WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0			0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_RESEN0_1

Bit(s)	Mnemonic	Name	Description
19	GPIO51	GPIO51_R0	Bitwise SET operation of GPIO51 R0 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
18	GPIO50	GPIO50_R0	Bitwise SET operation of GPIO50 R0 0: 1: SET bits	Keep
17	GPIO49	GPIO49_R0	Bitwise SET operation of GPIO49 R0 0: 1: SET bits	Keep
16	GPIO48	GPIO48_R0	Bitwise SET operation of GPIO48 R0 0: 1: SET bits	Keep
15	GPIO47	GPIO47_R0	Bitwise SET operation of GPIO47 R0 0: 1: SET bits	Keep
14	GPIO46	GPIO46_R0	Bitwise SET operation of GPIO46 R0 0: 1: SET bits	Keep
13	GPIO45	GPIO45_R0	Bitwise SET operation of GPIO45 R0 0: 1: SET bits	Keep
10	GPIO42	GPIO42_R0	Bitwise SET operation of GPIO42 R0 0: 1: SET bits	Keep
9	GPIO41	GPIO41_R0	Bitwise SET operation of GPIO41 R0 0: 1: SET bits	Keep
8	GPIO40	GPIO40_R0	Bitwise SET operation of GPIO40 R0 0: 1: SET bits	Keep
7	GPIO39	GPIO39_R0	Bitwise SET operation of GPIO39 R0 0: 1: SET bits	Keep
6	GPIO38	GPIO38_R0	Bitwise SET operation of GPIO38 R0 0: 1: SET bits	Keep
5	GPIO37	GPIO37_R0	Bitwise SET operation of GPIO37 R0 0: 1: SET bits	Keep
4	GPIO36	GPIO36_R0	Bitwise SET operation of GPIO36 R0 0: 1: SET bits	Keep
3	GPIO35	GPIO35_R0	Bitwise SET operation of GPIO35 R0 0: 1: SET bits	Keep
2	GPIO34	GPIO34_R0	Bitwise SET operation of GPIO34 R0 0: 1: SET bits	Keep
1	GPIO33	GPIO33_R0	Bitwise SET operation of GPIO33 R0 0: 1: SET bits	Keep
0	GPIO32	GPIO32_R0	Bitwise SET operation of GPIO32 R0 0: 1: SET bits	Keep

A0020B18 GPIO_RESEN0_1_CLR GPIO R0 Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	WO	WO	WO			WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0			0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_RESEN0_1

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_R0	Bitwise CLR operation of GPIO51 R0 0: 1: CLR bits	Keep
18	GPIO50	GPIO50_R0	Bitwise CLR operation of GPIO50 R0 0: 1: CLR bits	Keep
17	GPIO49	GPIO49_R0	Bitwise CLR operation of GPIO49 R0 0: 1: CLR bits	Keep
16	GPIO48	GPIO48_R0	Bitwise CLR operation of GPIO48 R0 0: 1: CLR bits	Keep
15	GPIO47	GPIO47_R0	Bitwise CLR operation of GPIO47 R0 0: 1: CLR bits	Keep
14	GPIO46	GPIO46_R0	Bitwise CLR operation of GPIO46 R0 0: 1: CLR bits	Keep
13	GPIO45	GPIO45_R0	Bitwise CLR operation of GPIO45 R0 0: 1: CLR bits	Keep
10	GPIO42	GPIO42_R0	Bitwise CLR operation of GPIO42 R0 0: 1: CLR bits	Keep
9	GPIO41	GPIO41_R0	Bitwise CLR operation of GPIO41 R0 0: 1: CLR bits	Keep
8	GPIO40	GPIO40_R0	Bitwise CLR operation of GPIO40 R0 0: 1: CLR bits	Keep
7	GPIO39	GPIO39_R0	Bitwise CLR operation of GPIO39 R0 0: 1: CLR bits	Keep
6	GPIO38	GPIO38_R0	Bitwise CLR operation of GPIO38 R0 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
5	GPIO37	GPIO37_R0	Bitwise CLR operation of GPIO37 R0 0: 1: CLR bits	Keep
4	GPIO36	GPIO36_R0	Bitwise CLR operation of GPIO36 R0 0: 1: CLR bits	Keep
3	GPIO35	GPIO35_R0	Bitwise CLR operation of GPIO35 R0 0: 1: CLR bits	Keep
2	GPIO34	GPIO34_R0	Bitwise CLR operation of GPIO34 R0 0: 1: CLR bits	Keep
1	GPIO33	GPIO33_R0	Bitwise CLR operation of GPIO33 R0 0: 1: CLR bits	Keep
0	GPIO32	GPIO32_R0	Bitwise CLR operation of GPIO32 R0 0: 1: CLR bits	Keep

A0020B20 **GPIO_RESEN1** **GPIO R1 Control** **00000000**
0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31		GPIO29	GPIO28	GPIO27	GPIO26					GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	RW		RW	RW	RW	RW					RW	RW	RW	RW	RW	RW
Reset	0		0	0	0	0					0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12		GPIO10										
Type	RW	RW	RW	RW		RW										
Reset	0	0	0	0		0										

Overview: Configures GPIO R1 control

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_R1	R1 for GPIO31 0: 1: Enable	Disable
29	GPIO29	GPIO29_R1	R1 for GPIO29 0: 1: Enable	Disable
28	GPIO28	GPIO28_R1	R1 for GPIO28 0: 1: Enable	Disable
27	GPIO27	GPIO27_R1	R1 for GPIO27 0: 1: Enable	Disable
26	GPIO26	GPIO26_R1	R1 for GPIO26 0: 1: Enable	Disable

Bit(s)	Mnemonic	Name	Description
21	GPIO21	GPIO21_R1	R1 for GPIO21 0: Disable 1: Enable
20	GPIO20	GPIO20_R1	R1 for GPIO20 0: Disable 1: Enable
19	GPIO19	GPIO19_R1	R1 for GPIO19 0: Disable 1: Enable
18	GPIO18	GPIO18_R1	R1 for GPIO18 0: Disable 1: Enable
17	GPIO17	GPIO17_R1	R1 for GPIO17 0: Disable 1: Enable
16	GPIO16	GPIO16_R1	R1 for GPIO16 0: Disable 1: Enable
15	GPIO15	GPIO15_R1	R1 for GPIO15 0: Disable 1: Enable
14	GPIO14	GPIO14_R1	R1 for GPIO14 0: Disable 1: Enable
13	GPIO13	GPIO13_R1	R1 for GPIO13 0: Disable 1: Enable
12	GPIO12	GPIO12_R1	R1 for GPIO12 0: Disable 1: Enable
10	GPIO10	GPIO10_R1	R1 for GPIO10 0: Disable 1: Enable

A0020B24 **GPIO_RESEN1** **GPIO R1 Control** **00000000**
0 SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO3 1		GPIO2 9	GPIO2 8	GPIO2 7	GPIO2 6					GPIO2 1	GPIO2 0	GPIO1 9	GPIO1 8	GPIO1 7	GPIO1 6
Type	WO		WO	WO	WO	WO					WO	WO	WO	WO	WO	WO
Reset	0		0	0	0	0					0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO1 5	GPIO1 4	GPIO1 3	GPIO1 2		GPIO1 0										
Type	WO	WO	WO	WO		WO										
Reset	0	0	0	0		0										

Overview: For bitwise access of GPIO_RESEN1_0

Bit(s)	Mnemonic	Name	Description
--------	----------	------	-------------

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_R1	Bitwise SET operation of GPIO31 R1 0: 1: SET bits	Keep
29	GPIO29	GPIO29_R1	Bitwise SET operation of GPIO29 R1 0: 1: SET bits	Keep
28	GPIO28	GPIO28_R1	Bitwise SET operation of GPIO28 R1 0: 1: SET bits	Keep
27	GPIO27	GPIO27_R1	Bitwise SET operation of GPIO27 R1 0: 1: SET bits	Keep
26	GPIO26	GPIO26_R1	Bitwise SET operation of GPIO26 R1 0: 1: SET bits	Keep
21	GPIO21	GPIO21_R1	Bitwise SET operation of GPIO21 R1 0: 1: SET bits	Keep
20	GPIO20	GPIO20_R1	Bitwise SET operation of GPIO20 R1 0: 1: SET bits	Keep
19	GPIO19	GPIO19_R1	Bitwise SET operation of GPIO19 R1 0: 1: SET bits	Keep
18	GPIO18	GPIO18_R1	Bitwise SET operation of GPIO18 R1 0: 1: SET bits	Keep
17	GPIO17	GPIO17_R1	Bitwise SET operation of GPIO17 R1 0: 1: SET bits	Keep
16	GPIO16	GPIO16_R1	Bitwise SET operation of GPIO16 R1 0: 1: SET bits	Keep
15	GPIO15	GPIO15_R1	Bitwise SET operation of GPIO15 R1 0: 1: SET bits	Keep
14	GPIO14	GPIO14_R1	Bitwise SET operation of GPIO14 R1 0: 1: SET bits	Keep
13	GPIO13	GPIO13_R1	Bitwise SET operation of GPIO13 R1 0: 1: SET bits	Keep
12	GPIO12	GPIO12_R1	Bitwise SET operation of GPIO12 R1 0: 1: SET bits	Keep
10	GPIO10	GPIO10_R1	Bitwise SET operation of GPIO10 R1 0: 1: SET bits	Keep

A0020B28 **GPIO_RESEN1** **GPIO R1 Control**

00000000

0_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO3 1		GPIO2 9	GPIO2 8	GPIO2 7	GPIO2 6					GPIO2 1	GPIO2 0	GPIO1 9	GPIO1 8	GPIO1 7	GPIO1 6
Type	WO		WO	WO	WO	WO					WO	WO	WO	WO	WO	WO
Reset	0		0	0	0	0					0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO1 5	GPIO1 4	GPIO1 3	GPIO1 2		GPIO1 0										
Type	WO	WO	WO	WO		WO										
Reset	0	0	0	0		0										

Overview: For bitwise access of GPIO_RESEN1_0

Bit(s)	Mnemonic	Name	Description	
31	GPIO31	GPIO31_R1	Bitwise CLR operation of GPIO31 R1 0: 1: CLR bits	Keep
29	GPIO29	GPIO29_R1	Bitwise CLR operation of GPIO29 R1 0: 1: CLR bits	Keep
28	GPIO28	GPIO28_R1	Bitwise CLR operation of GPIO28 R1 0: 1: CLR bits	Keep
27	GPIO27	GPIO27_R1	Bitwise CLR operation of GPIO27 R1 0: 1: CLR bits	Keep
26	GPIO26	GPIO26_R1	Bitwise CLR operation of GPIO26 R1 0: 1: CLR bits	Keep
21	GPIO21	GPIO21_R1	Bitwise CLR operation of GPIO21 R1 0: 1: CLR bits	Keep
20	GPIO20	GPIO20_R1	Bitwise CLR operation of GPIO20 R1 0: 1: CLR bits	Keep
19	GPIO19	GPIO19_R1	Bitwise CLR operation of GPIO19 R1 0: 1: CLR bits	Keep
18	GPIO18	GPIO18_R1	Bitwise CLR operation of GPIO18 R1 0: 1: CLR bits	Keep
17	GPIO17	GPIO17_R1	Bitwise CLR operation of GPIO17 R1 0: 1: CLR bits	Keep
16	GPIO16	GPIO16_R1	Bitwise CLR operation of GPIO16 R1 0: 1: CLR bits	Keep
15	GPIO15	GPIO15_R1	Bitwise CLR operation of GPIO15 R1 0: 1: CLR bits	Keep
14	GPIO14	GPIO14_R1	Bitwise CLR operation of GPIO14 R1 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
13	GPIO13	GPIO13_R1	Bitwise CLR operation of GPIO13 R1 0: 1: CLR bits	Keep
12	GPIO12	GPIO12_R1	Bitwise CLR operation of GPIO12 R1 0: 1: CLR bits	Keep
10	GPIO10	GPIO10_R1	Bitwise CLR operation of GPIO10 R1 0: 1: CLR bits	Keep

A0020B30 **GPIO RESEN1** GPIO R1 Control **00000000**
1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													RW	RW	RW	RW
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34
Type	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0			0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO R1 control

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_R1	R1 for GPIO51 0: 1: Enable	Disable
18	GPIO50	GPIO50_R1	R1 for GPIO50 0: 1: Enable	Disable
17	GPIO49	GPIO49_R1	R1 for GPIO49 0: 1: Enable	Disable
16	GPIO48	GPIO48_R1	R1 for GPIO48 0: 1: Enable	Disable
15	GPIO47	GPIO47_R1	R1 for GPIO47 0: 1: Enable	Disable
14	GPIO46	GPIO46_R1	R1 for GPIO46 0: 1: Enable	Disable
13	GPIO45	GPIO45_R1	R1 for GPIO45 0: 1: Enable	Disable
10	GPIO42	GPIO42_R1	R1 for GPIO42 0: 1: Enable	Disable

Bit(s)	Mnemonic	Name	Description
9	GPIO41	GPIO41_R1	R1 for GPIO41 0: Disable 1: Enable
8	GPIO40	GPIO40_R1	R1 for GPIO40 0: Disable 1: Enable
7	GPIO39	GPIO39_R1	R1 for GPIO39 0: Disable 1: Enable
6	GPIO38	GPIO38_R1	R1 for GPIO38 0: Disable 1: Enable
5	GPIO37	GPIO37_R1	R1 for GPIO37 0: Disable 1: Enable
4	GPIO36	GPIO36_R1	R1 for GPIO36 0: Disable 1: Enable
3	GPIO35	GPIO35_R1	R1 for GPIO35 0: Disable 1: Enable
2	GPIO34	GPIO34_R1	R1 for GPIO34 0: Disable 1: Enable
1	GPIO33	GPIO33_R1	R1 for GPIO33 0: Disable 1: Enable
0	GPIO32	GPIO32_R1	R1 for GPIO32 0: Disable 1: Enable

A0020B34 **GPIO_RESEN1_1_SET** **GPIO R1 Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34
Type	WO	WO	WO			WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0			0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_RESEN1_1

Bit(s)	Mnemonic	Name	Description
19	GPIO51	GPIO51_R1	Bitwise SET operation of GPIO51 R1 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description	
18	GPIO50	GPIO50_R1	Bitwise SET operation of GPIO50 R1 0: 1: SET bits	Keep
17	GPIO49	GPIO49_R1	Bitwise SET operation of GPIO49 R1 0: 1: SET bits	Keep
16	GPIO48	GPIO48_R1	Bitwise SET operation of GPIO48 R1 0: 1: SET bits	Keep
15	GPIO47	GPIO47_R1	Bitwise SET operation of GPIO47 R1 0: 1: SET bits	Keep
14	GPIO46	GPIO46_R1	Bitwise SET operation of GPIO46 R1 0: 1: SET bits	Keep
13	GPIO45	GPIO45_R1	Bitwise SET operation of GPIO45 R1 0: 1: SET bits	Keep
10	GPIO42	GPIO42_R1	Bitwise SET operation of GPIO42 R1 0: 1: SET bits	Keep
9	GPIO41	GPIO41_R1	Bitwise SET operation of GPIO41 R1 0: 1: SET bits	Keep
8	GPIO40	GPIO40_R1	Bitwise SET operation of GPIO40 R1 0: 1: SET bits	Keep
7	GPIO39	GPIO39_R1	Bitwise SET operation of GPIO39 R1 0: 1: SET bits	Keep
6	GPIO38	GPIO38_R1	Bitwise SET operation of GPIO38 R1 0: 1: SET bits	Keep
5	GPIO37	GPIO37_R1	Bitwise SET operation of GPIO37 R1 0: 1: SET bits	Keep
4	GPIO36	GPIO36_R1	Bitwise SET operation of GPIO36 R1 0: 1: SET bits	Keep
3	GPIO35	GPIO35_R1	Bitwise SET operation of GPIO35 R1 0: 1: SET bits	Keep
2	GPIO34	GPIO34_R1	Bitwise SET operation of GPIO34 R1 0: 1: SET bits	Keep
1	GPIO33	GPIO33_R1	Bitwise SET operation of GPIO33 R1 0: 1: SET bits	Keep
0	GPIO32	GPIO32_R1	Bitwise SET operation of GPIO32 R1 0: 1: SET bits	Keep

A0020B38 GPIO_RESEN1 GPIO R1 Control 00000000
1_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													GPIO51	GPIO50	GPIO49	GPIO48
Type													WO	WO	WO	WO
Reset													0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45			GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	WO	WO	WO			WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0			0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_RESEN1_1

Bit(s)	Mnemonic	Name	Description	
19	GPIO51	GPIO51_R1	Bitwise CLR operation of GPIO51 R1 0: 1: CLR bits	Keep
18	GPIO50	GPIO50_R1	Bitwise CLR operation of GPIO50 R1 0: 1: CLR bits	Keep
17	GPIO49	GPIO49_R1	Bitwise CLR operation of GPIO49 R1 0: 1: CLR bits	Keep
16	GPIO48	GPIO48_R1	Bitwise CLR operation of GPIO48 R1 0: 1: CLR bits	Keep
15	GPIO47	GPIO47_R1	Bitwise CLR operation of GPIO47 R1 0: 1: CLR bits	Keep
14	GPIO46	GPIO46_R1	Bitwise CLR operation of GPIO46 R1 0: 1: CLR bits	Keep
13	GPIO45	GPIO45_R1	Bitwise CLR operation of GPIO45 R1 0: 1: CLR bits	Keep
10	GPIO42	GPIO42_R1	Bitwise CLR operation of GPIO42 R1 0: 1: CLR bits	Keep
9	GPIO41	GPIO41_R1	Bitwise CLR operation of GPIO41 R1 0: 1: CLR bits	Keep
8	GPIO40	GPIO40_R1	Bitwise CLR operation of GPIO40 R1 0: 1: CLR bits	Keep
7	GPIO39	GPIO39_R1	Bitwise CLR operation of GPIO39 R1 0: 1: CLR bits	Keep
6	GPIO38	GPIO38_R1	Bitwise CLR operation of GPIO38 R1 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
5	GPIO37	GPIO37_R1	Bitwise CLR operation of GPIO37 R1 0: 1: CLR bits	Keep
4	GPIO36	GPIO36_R1	Bitwise CLR operation of GPIO36 R1 0: 1: CLR bits	Keep
3	GPIO35	GPIO35_R1	Bitwise CLR operation of GPIO35 R1 0: 1: CLR bits	Keep
2	GPIO34	GPIO34_R1	Bitwise CLR operation of GPIO34 R1 0: 1: CLR bits	Keep
1	GPIO33	GPIO33_R1	Bitwise CLR operation of GPIO33 R1 0: 1: CLR bits	Keep
0	GPIO32	GPIO32_R1	Bitwise CLR operation of GPIO32 R1 0: 1: CLR bits	Keep

KeyPad

KCOL0-4

PUPD	R1	R0	WEAK PULL UP/DOWN STATE
0	0	0	Disable both resistors
0	0	1	PU-36K ohms
0	1	0	PU-1200K ohms
0	1	1	PU- 1200K//36K ohms
1	0	0	Disable both resistors
1	0	1	PD-36K ohms
1	1	0	PD-1200K ohms
1	1	1	PD- 1200K//36K ohms

KROW0-4

PUPD	R1	R0	WEAK PULL UP/DOWN STATE
0	0	0	Disable both resistors
0	0	1	PU-36K ohms
0	1	0	PU-1K ohms
0	1	1	PU- 1K//36K ohms
1	0	0	Disable both resistors
1	0	1	PD-36K ohms
1	1	0	PD-1K ohms
1	1	1	PD- 1K//36K ohms

URXD:

CMPDN/CMCSD0/CMCSD1/CMMCLK

MCCK/MCCM0/MCDA0/MCDA1/MCDA2/MCDA3

TESTMODE_D/LSCE_B/LSCK/LSDA/LSA0/LPTE
RESETB:

PUPD	R1	R0	WEAK PULL UP/DOWN STATE
0	0	0	Disable both resistors
0	0	1	PU-47K ohms
0	1	0	PU-47K ohms
0	1	1	PU- 23.5K ohms
1	0	0	Disable both resistors
1	0	1	PD-47K ohms
1	1	0	PD-47K ohms
1	1	1	PD- 23.5K ohms

A0020C00 GPIO_MODE0 GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO7				GPIO6				GPIO5				GPIO4			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO3				GPIO2				GPIO1				GPIO0			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28	GPIO7		Aux. mode of GPIO_7 0: GPIO7 (IO) 1: EINT6 (I) 2: Reserved 3: BPI_BUS5 (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
26:24	GPIO6		Aux. mode of GPIO_6 0: GPIO6 (IO) 1: EINT5 (I) 2: MCINS (I) 3: BPI_BUS4 (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
22:20	GPIO5		Aux. mode of GPIO_5 0: GPIO5 (IO) 1: EINT4 (I) 2: Reserved 3: BPI_BUS3 (O) 4: Reserved 5: Reserved 6: Reserved

Bit(s)	Mnemonic	Name	Description
			7: Reserved
18:16		GPIO4	Aux. mode of GPIO_4
		0:	GPIO4 (IO)
		1:	EINT3 (I)
		2:	Reserved
		3:	Reserved
		4:	U1RTS (O)
		5:	Reserved
		6:	Reserved
		7:	Reserved
15:12		GPIO3	Aux. mode of GPIO_3
		0:	GPIO3 (IO)
		1:	MCINS (I)
		2:	YM (AIO)
		3:	Reserved
		4:	PWM1 (O)
		5:	CMCSD1 (I)
		6:	EDICK (O)
		7:	JTDO (O)
		8:	BTJTDO (O)
		9:	FMJTDO (O)
11:8		GPIO2	Aux. mode of GPIO_2
		0:	GPIO2 (IO)
		1:	EINT2 (I)
		2:	YP (AIO)
		3:	GPSFSYNC (O)
		4:	PWM0 (O)
		5:	CMCSD0 (I)
		6:	EDIWS (O)
		7:	JTRST_B (I)
		8:	BTJTRSTB (I)
		9:	FMJTRSTB (I)
7:4		GPIO1	Aux. mode of GPIO_1
		0:	GPIO1 (IO)
		1:	EINT1 (I)
		2:	XM (AIO)
		3:	U3TXD (O)
		4:	U1CTS (I)
		5:	CMMCLK (O)
		6:	EDIDI (I)
		7:	JTMS (I)
		8:	BTJTMS (I)
		9:	FMJTMS (I)
3:0		GPIO0	Aux. mode of GPIO_0
		0:	GPIO0 (IO)
		1:	EINT0 (I)
		2:	XP (AIO)
		3:	U3RXD (I)
		4:	CMCSD2 (I)
		5:	CMCSK (I)
		6:	EDIDO (O)
		7:	JTDI (I)
		8:	BTJTDI (I)
		9:	FMJTDI (I)

A0020C04 GPIO_MODE0 GPIO Mode Control

00000000

SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO7				GPIO6				GPIO5				GPIO4			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO3				GPIO2				GPIO1				GPIO0			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE0

Bit(s)	Mnemonic	Name	Description	
30:28		GPIO7	Bitwise SET operation for Aux. mode of GPIO_7 0: 1: SET bits	Keep
26:24		GPIO6	Bitwise SET operation for Aux. mode of GPIO_6 0: 1: SET bits	Keep
22:20		GPIO5	Bitwise SET operation for Aux. mode of GPIO_5 0: 1: SET bits	Keep
18:16		GPIO4	Bitwise SET operation for Aux. mode of GPIO_4 0: 1: SET bits	Keep
15:12		GPIO3	Bitwise SET operation for Aux. mode of GPIO_3 0: 1: SET bits	Keep
11:8		GPIO2	Bitwise SET operation for Aux. mode of GPIO_2 0: 1: SET bits	Keep
7:4		GPIO1	Bitwise SET operation for Aux. mode of GPIO_1 0: 1: SET bits	Keep
3:0		GPIO0	Bitwise SET operation for Aux. mode of GPIO_0 0: 1: SET bits	Keep

A0020C08 GPIO_MODE0 **GPIO Mode Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO7				GPIO6				GPIO5				GPIO4			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO3				GPIO2				GPIO1				GPIO0			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE0

Bit(s)	Mnemonic	Name	Description
30:28		GPIO7	Bitwise CLR operation for Aux. mode of GPIO_7 0: Keep 1: CLR bits
26:24		GPIO6	Bitwise CLR operation for Aux. mode of GPIO_6 0: Keep 1: CLR bits
22:20		GPIO5	Bitwise CLR operation for Aux. mode of GPIO_5 0: Keep 1: CLR bits
18:16		GPIO4	Bitwise CLR operation for Aux. mode of GPIO_4 0: Keep 1: CLR bits
15:12		GPIO3	Bitwise CLR operation for Aux. mode of GPIO_3 0: Keep 1: CLR bits
11:8		GPIO2	Bitwise CLR operation for Aux. mode of GPIO_2 0: Keep 1: CLR bits
7:4		GPIO1	Bitwise CLR operation for Aux. mode of GPIO_1 0: Keep 1: CLR bits
3:0		GPIO0	Bitwise CLR operation for Aux. mode of GPIO_0 0: Keep 1: CLR bits

A0020C10 GPIO_MODE1 GPIO Mode Control 00001100

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
Name		GPIO15					GPIO14					GPIO13					GPIO12			
Type		RW					RW					RW					RW			
Reset		0	0	0		0	0	0		0	0	0		0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name		GPIO11					GPIO10					GPIO9					GPIO8			
Type		RW					RW					RW					RW			
Reset		0	0	1		0	0	1		0	0	0		0	0	0				

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO15	Aux. mode of GPIO_15 0: (IO) 1: (IO) 2: (O) 3: (I) 4: (I) 5: (I) 6: (I) 7: Reserved
26:24		GPIO14	Aux. mode of GPIO_14 0: (IO) 1: (IO) 2: (I) 3: (I)

Bit(s)	Mnemonic	Name	Description	
			4:	Reserved
			5:	Reserved
			6:	Reserved
			7: Reserved	
22:20		GPIO13	Aux. mode of GPIO_13	
			0: GPIO13	(IO)
			1: KCOL3	(IO)
			2: EINT11	(I)
			3: PWM0	(O)
			4: FMJTMS	(I)
			5: JTMS	(I)
			6: BTJTMS	(I)
			7: Reserved	
18:16		GPIO12	Aux. mode of GPIO_12	
			0: GPIO12	(IO)
			1: KCOL4	(IO)
			2: U2RXD	(I)
			3: EDIDI	(I)
			4: FMJTDI	(I)
			5: JTDI	(I)
			6: BTJTDI	(I)
			7: Reserved	
14:12		GPIO11	Aux. mode of GPIO_11	
			0: GPIO11	(IO)
			1: U1TXD	(O)
			2: CMPDN	(O)
			3: EINT10	(I)
			4:	Reserved
			5:	Reserved
			6:	Reserved
			7: Reserved	
10:8		GPIO10	Aux. mode of GPIO_10	
			0: GPIO10	(IO)
			1: U1RXD	(I)
			2: CMRST	(O)
			3: EINT9	(I)
			4: MCINS	(I)
			5:	Reserved
			6:	Reserved
			7: Reserved	
6:4		GPIO9	Aux. mode of GPIO_9	
			0: GPIO9	(IO)
			1: EINT8	(I)
			2: SDA	(IO)
			3:	Reserved
			4:	Reserved
			5:	Reserved
			6:	Reserved
			7: Reserved	
2:0		GPIO8	Aux. mode of GPIO_8	
			0: GPIO8	(IO)
			1: EINT7	(I)
			2: SCL	(IO)
			3:	Reserved
			4:	Reserved
			5:	Reserved
			6:	Reserved
			7: Reserved	

A0020C14 GPIO_MODE1 SET GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO15				GPIO14				GPIO13				GPIO12			
Type	WO				WO				WO				WO			
Reset	0	0	0		0	0	0		0	0	0		0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO11				GPIO10				GPIO9				GPIO8			
Type	WO				WO				WO				WO			
Reset	0	0	0		0	0	0		0	0	0		0	0	0	

Overview: For bitwise access of GPIO_MODE1

Bit(s)	Mnemonic	Name	Description	
30:28		GPIO15	Bitwise SET operation for Aux. mode of KCOL1 0: 1: SET bits	Keep
26:24		GPIO14	Bitwise SET operation for Aux. mode of KCOL2 0: 1: SET bits	Keep
22:20		GPIO13	Bitwise SET operation for Aux. mode of KCOL3 0: 1: SET bits	Keep
18:16		GPIO12	Bitwise SET operation for Aux. mode of KCOL4 0: 1: SET bits	Keep
14:12		GPIO11	Bitwise SET operation for Aux. mode of UTXD1 0: 1: SET bits	Keep
10:8		GPIO10	Bitwise SET operation for Aux. mode of URXD1 0: 1: SET bits	Keep
6:4		GPIO9	Bitwise SET operation for Aux. mode of GPIO_9 0: 1: SET bits	Keep
2:0		GPIO8	Bitwise SET operation for Aux. mode of GPIO_8 0: 1: SET bits	Keep

A0020C18 GPIO_MODE1 CLR GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO15				GPIO14				GPIO13				GPIO12			
Type	WO				WO				WO				WO			
Reset	0	0	0		0	0	0		0	0	0		0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO11				GPIO10				GPIO9				GPIO8			
Type	WO				WO				WO				WO			
Reset	0	0	0		0	0	0		0	0	0		0	0	0	

Overview: For bitwise access of GPIO_MODE1

Bit(s)	Mnemonic	Name	Description	
30:28		GPIO15	Bitwise CLR operation for Aux. mode of KCOL1 0: 1: CLR bits	Keep
26:24		GPIO14	Bitwise CLR operation for Aux. mode of KCOL2 0: 1: CLR bits	Keep
22:20		GPIO13	Bitwise CLR operation for Aux. mode of KCOL3 0: 1: CLR bits	Keep
18:16		GPIO12	Bitwise CLR operation for Aux. mode of KCOL4 0: 1: CLR bits	Keep
14:12		GPIO11	Bitwise CLR operation for Aux. mode of UTXD1 0: 1: CLR bits	Keep
10:8		GPIO10	Bitwise CLR operation for Aux. mode of URXD1 0: 1: CLR bits	Keep
6:4		GPIO9	Bitwise CLR operation for Aux. mode of GPIO_9 0: 1: CLR bits	Keep
2:0		GPIO8	Bitwise CLR operation for Aux. mode of GPIO_8 0: 1: CLR bits	Keep

A0020C20 GPIO_MODE2 GPIO Mode Control 11000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO23				GPIO22				GPIO21				GPIO20		
Type		RW				RW				RW				RW		
Reset		0	0	1		0	0	1		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO19				GPIO18				GPIO17				GPIO16		
Type		RW				RW				RW				RW		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO23	Aux. mode of GPIO_23 0: 1: 2: 3: 4: 5: 6: 7: Reserved
			GPIO23 (IO) BPI_BUS1 (O) Reserved Reserved Reserved Reserved Reserved

Bit(s)	Mnemonic	Name	Description
26:24		GPIO22	Aux. mode of GPIO_22
			0: GPIO22 (IO)
			1: BPI_BUS2 (O)
			2: Reserved
			3: Reserved
			4: Reserved
			5: Reserved
			6: Reserved
22:20		GPIO21	Aux. mode of GPIO_21
			0: GPIO21 (IO)
			1: KROW0 (IO)
			2: Reserved
			3: Reserved
			4: Reserved
			5: MCINS (I)
			6: BTDBGIN (I)
18:16		GPIO20	Aux. mode of GPIO_20
			0: GPIO20 (IO)
			1: KROW1 (IO)
			2: EINT14 (I)
			3: EDIDO (O)
			4: BTPRI (IO)
			5: JTRCK (O)
			6: BTDBGACKN (O)
14:12		GPIO19	Aux. mode of GPIO_19
			0: GPIO19 (IO)
			1: KROW2 (IO)
			2: PWM1 (O)
			3: EDIWS (O)
			4: FMJTDO (O)
			5: JTDO (O)
			6: BTJTDO (O)
10:8		GPIO18	Aux. mode of GPIO_18
			0: GPIO18 (IO)
			1: KROW3 (IO)
			2: EINT13 (I)
			3: CLK00 (O)
			4: FMJTRSTB (I)
			5: JTRST_B (I)
			6: BTJTRSTB (I)
6:4		GPIO17	Aux. mode of GPIO_17
			0: GPIO17 (IO)
			1: KROW4 (IO)
			2: U2TXD (O)
			3: EDICK (O)
			4: Reserved
			5: Reserved
			6: Reserved
2:0		GPIO16	Aux. mode of GPIO_16
			0: GPIO16 (IO)
			1: KCOL0 (IO)
			2: Reserved

Bit(s)	Mnemonic	Name	Description
			3: Reserved
			4: Reserved
			5: Reserved
			6: Reserved
			7: Reserved

A0020C24 **GPIO_MODE2** **GPIO Mode Control** **00000000**
SET

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO23				GPIO22				GPIO21				GPIO20			
Type	WO				WO				WO				WO			
Reset	0 0 0				0 0 0				0 0 0				0 0 0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO19				GPIO18				GPIO17				GPIO16			
Type	WO				WO				WO				WO			
Reset	0 0 0				0 0 0				0 0 0				0 0 0			

Overview: For bitwise access of GPIO_MODE2

Bit(s)	Mnemonic	Name	Description	
30:28		GPIO23	Bitwise SET operation for Aux. mode of BPI_BUS1 0: 1: SET bits	Keep
26:24		GPIO22	Bitwise SET operation for Aux. mode of BPI_BUS2 0: 1: SET bits	Keep
22:20		GPIO21	Bitwise SET operation for Aux. mode of KROW0 0: 1: SET bits	Keep
18:16		GPIO20	Bitwise SET operation for Aux. mode of KROW1 0: 1: SET bits	Keep
14:12		GPIO19	Bitwise SET operation for Aux. mode of KROW2 0: 1: SET bits	Keep
10:8		GPIO18	Bitwise SET operation for Aux. mode of KROW3 0: 1: SET bits	Keep
6:4		GPIO17	Bitwise SET operation for Aux. mode of KROW4 0: 1: SET bits	Keep
2:0		GPIO16	Bitwise SET operation for Aux. mode of KCOL0 0: 1: SET bits	Keep

A0020C28 **GPIO_MODE2** **GPIO Mode Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO23				GPIO22				GPIO21				GPIO20			

Type		WO					WO					WO					WO				
Reset		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	GPIO19				GPIO18				GPIO17				GPIO16								
Type	WO				WO				WO				WO								
Reset	0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		

Overview: For bitwise access of GPIO_MODE2

Bit(s)	Mnemonic	Name	Description	
30:28		GPIO23	Bitwise CLR operation for Aux. mode of BPI_BUS1 0: 1: CLR bits	Keep
26:24		GPIO22	Bitwise CLR operation for Aux. mode of BPI_BUS2 0: 1: CLR bits	Keep
22:20		GPIO21	Bitwise CLR operation for Aux. mode of KROW0 0: 1: CLR bits	Keep
18:16		GPIO20	Bitwise CLR operation for Aux. mode of KROW1 0: 1: CLR bits	Keep
14:12		GPIO19	Bitwise CLR operation for Aux. mode of KROW2 0: 1: CLR bits	Keep
10:8		GPIO18	Bitwise CLR operation for Aux. mode of KROW3 0: 1: CLR bits	Keep
6:4		GPIO17	Bitwise CLR operation for Aux. mode of KROW4 0: 1: CLR bits	Keep
2:0		GPIO16	Bitwise CLR operation for Aux. mode of KCOL0 0: 1: CLR bits	Keep

A0020C30 GPIO_MODE3 GPIO Mode Control 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Name	GPIO31				GPIO30				GPIO29				GPIO28				
Type	RW				RW				RW				RW				
Reset	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	GPIO27				GPIO26				GPIO25				GPIO24				
Type	RW				RW				RW				RW				
Reset	0	0	0	0		0	0	0		0	0	0		0	0	1	

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO31	Aux. mode of GPIO_31 0: GPIO31 (IO) 1: MCK (O) 2: Reserved

Bit(s)	Mnemonic	Name	Description
			3: Reserved
			4: U2RXD (I)
			5: Reserved
			6: Reserved
			7: Reserved
26:24		GPIO30	Aux. mode of GPIO_30
			0: GPIO30 (IO)
			1: CMCSK (I)
			2: LPTE (I)
			3: CMCS2 (I)
			4: EINT16 (I)
			5: Reserved
			6: JTRCK (O)
			7: Reserved
23:20		GPIO29	Aux. mode of GPIO_29
			0: GPIO29 (IO)
			1: CMMCLK (O)
			2: LSA0DA1 (O)
			3: DAISYNC (O)
			4: SPIMISO (IO)
			5: FMJTDO (O)
			6: JTDO (O)
			7: Reserved
			8: MC2DA0 (IO)
19:16		GPIO28	Aux. mode of GPIO_28
			0: GPIO28 (IO)
			1: CMCS1 (I)
			2: LSDA1 (IO)
			3: DAIPCMOUT (O)
			4: SPIMOSI (IO)
			5: FMJTRSTB (I)
			6: JTRST_B (I)
			7: Reserved
			8: MC2CK (O)
15:12		GPIO27	Aux. mode of GPIO_27
			0: GPIO27 (IO)
			1: CMCS0 (I)
			2: LSCE_B1 (O)
			3: DAIPCMIN (I)
			4: SPISCK (IO)
			5: FMJTCK (I)
			6: JTCK (I)
			7: Reserved
			8: MC2CM0 (O)
10:8		GPIO26	Aux. mode of GPIO_26
			0: GPIO26 (IO)
			1: CMPDN (O)
			2: LSCK1 (O)
			3: DAICLK (O)
			4: SPICS (IO)
			5: FMJTMS (I)
			6: JTMS (I)
			7: Reserved
6:4		GPIO25	Aux. mode of GPIO_25
			0: GPIO25 (IO)
			1: CMRST (O)
			2: TESTMODE_D (O)
			3: CLKO1 (O)
			4: EINT15 (I)

Bit(s)	Mnemonic	Name	Description
			5: FMJTDI (I)
			6: JTDI (I)
			7: Reserved
2:0		GPIO24	Aux. mode of GPIO_24
			0: GPIO24 (IO)
			1: BPI_BUS0 (O)
			2: Reserved
			3: Reserved
			4: Reserved
			5: Reserved
			6: Reserved
			7: Reserved

A0020C34 **GPIO_MODE3 SET** GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31				GPIO30				GPIO29				GPIO28			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO27				GPIO26				GPIO25				GPIO24			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE3

Bit(s)	Mnemonic	Name	Description
30:28		GPIO31	Bitwise SET operation for Aux. mode of MCCK 0: Keep 1: SET bits
26:24		GPIO30	Bitwise SET operation for Aux. mode of CMCSK 0: Keep 1: SET bits
23:20		GPIO29	Bitwise SET operation for Aux. mode of CMMCLK 0: Keep 1: SET bits
19:16		GPIO28	Bitwise SET operation for Aux. mode of CMCSD1 0: Keep 1: SET bits
15:12		GPIO27	Bitwise SET operation for Aux. mode of CMCSD0 0: Keep 1: SET bits
10:8		GPIO26	Bitwise SET operation for Aux. mode of CMPDN 0: Keep 1: SET bits
6:4		GPIO25	Bitwise SET operation for Aux. mode of CMRST 0: Keep 1: SET bits
2:0		GPIO24	Bitwise SET operation for Aux. mode of BPI_BUS0 0: Keep 1: SET bits

A0020C38 **GPIO_MODE3** **GPIO Mode Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO31				GPIO30				GPIO29				GPIO28			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO27				GPIO26				GPIO25				GPIO24			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE3

Bit(s)	Mnemonic	Name	Description	
30:28		GPIO31	Bitwise CLR operation for Aux. mode of MCKK 0: 1: CLR bits	Keep
26:24		GPIO30	Bitwise CLR operation for Aux. mode of CMCSK 0: 1: CLR bits	Keep
23:20		GPIO29	Bitwise CLR operation for Aux. mode of CMMCLK 0: 1: CLR bits	Keep
19:16		GPIO28	Bitwise CLR operation for Aux. mode of CMCSK1 0: 1: CLR bits	Keep
15:12		GPIO27	Bitwise CLR operation for Aux. mode of CMCSK0 0: 1: CLR bits	Keep
10:8		GPIO26	Bitwise CLR operation for Aux. mode of CMPDN 0: 1: CLR bits	Keep
6:4		GPIO25	Bitwise CLR operation for Aux. mode of CMRST 0: 1: CLR bits	Keep
2:0		GPIO24	Bitwise CLR operation for Aux. mode of BPI_BUS0 0: 1: CLR bits	Keep

A0020C40 **GPIO_MODE4** **GPIO Mode Control** **11100000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO39				GPIO38				GPIO37				GPIO36			
Type	RW				RW				RW				RW			
Reset	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO35				GPIO34				GPIO33				GPIO32			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28	GPIO39	GPIO39	Aux. mode of GPIO_39
			0: GPIO39 (IO)
			1: SIM1_SCLK (IO)
			2: Reserved
			3: Reserved
			4: Reserved
			5: Reserved
			6: Reserved
26:24	GPIO38	GPIO38	Aux. mode of GPIO_38
			0: GPIO38 (IO)
			1: SIM1_SRST (IO)
			2: Reserved
			3: Reserved
			4: Reserved
			5: Reserved
			6: Reserved
22:20	GPIO37	GPIO37	Aux. mode of GPIO_37
			0: GPIO37 (IO)
			1: SIM1_SIO (IO)
			2: Reserved
			3: Reserved
			4: Reserved
			5: Reserved
			6: Reserved
18:16	GPIO36	GPIO36	Aux. mode of GPIO_36
			0: GPIO36 (IO)
			1: MCDA3 (IO)
			2: EINT19 (I)
			3: CLK02 (O)
			4: DAIPCMOUT (O)
			5: Reserved
			6: Reserved
14:12	GPIO35	GPIO35	Aux. mode of GPIO_35
			0: GPIO35 (IO)
			1: MCDA2 (IO)
			2: EINT18 (I)
			3: Reserved
			4: DAICLK (O)
			5: Reserved
			6: Reserved
10:8	GPIO34	GPIO34	Aux. mode of GPIO_34
			0: GPIO34 (IO)
			1: MCDA1 (IO)
			2: EINT17 (I)
			3: Reserved
			4: DAIPCMIN (I)
			5: Reserved
			6: Reserved
6:4	GPIO33	GPIO33	Aux. mode of GPIO_33
			0: GPIO33 (IO)

Bit(s)	Mnemonic	Name	Description
			1: MCDA0 (IO)
			2: Reserved
			3: Reserved
			4: DAISYNC (O)
			5: Reserved
			6: Reserved
			7: Reserved
2:0		GPIO32	Aux. mode of GPIO_32
			0: GPIO32 (IO)
			1: MCCM0 (O)
			2: Reserved
			3: Reserved
			4: U2TXD (O)
			5: Reserved
			6: Reserved
			7: Reserved

A0020C44 **GPIO_MODE4 SET** GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO39				GPIO38				GPIO37				GPIO36			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO35				GPIO34				GPIO33				GPIO32			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE4

Bit(s)	Mnemonic	Name	Description
30:28		GPIO39	Bitwise SET operation for Aux. mode of SIM1_SCLK 0: Keep 1: SET bits
26:24		GPIO38	Bitwise SET operation for Aux. mode of SIM1_SRST 0: Keep 1: SET bits
22:20		GPIO37	Bitwise SET operation for Aux. mode of SIM1_SIO 0: Keep 1: SET bits
18:16		GPIO36	Bitwise SET operation for Aux. mode of MCDA3 0: Keep 1: SET bits
14:12		GPIO35	Bitwise SET operation for Aux. mode of MCDA2 0: Keep 1: SET bits
10:8		GPIO34	Bitwise SET operation for Aux. mode of MCDA1 0: Keep 1: SET bits
6:4		GPIO33	Bitwise SET operation for Aux. mode of MCDA0 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description
2:0		GPIO32	Bitwise SET operation for Aux. mode of MCCM0 0: 1: SET bits

A0020C48 **GPIO_MODE4** **GPIO Mode Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO39				GPIO38				GPIO37				GPIO36			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO35				GPIO34				GPIO33				GPIO32			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE4

Bit(s)	Mnemonic	Name	Description
30:28		GPIO39	Bitwise CLR operation for Aux. mode of SIM1_SCLK 0: 1: CLR bits
26:24		GPIO38	Bitwise CLR operation for Aux. mode of SIM1_SRST 0: 1: CLR bits
22:20		GPIO37	Bitwise CLR operation for Aux. mode of SIM1_SIO 0: 1: CLR bits
18:16		GPIO36	Bitwise CLR operation for Aux. mode of MCDA3 0: 1: CLR bits
14:12		GPIO35	Bitwise CLR operation for Aux. mode of MCDA2 0: 1: CLR bits
10:8		GPIO34	Bitwise CLR operation for Aux. mode of MCDA1 0: 1: CLR bits
6:4		GPIO33	Bitwise CLR operation for Aux. mode of MCDA0 0: 1: CLR bits
2:0		GPIO32	Bitwise CLR operation for Aux. mode of MCCM0 0: 1: CLR bits

A0020C50 **GPIO_MODE5** **GPIO Mode Control** **01000111**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO47				GPIO46				GPIO45				GPIO44			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	GPIO43			GPIO42			GPIO41			GPIO40		
Type	RW			RW			RW			RW		
Reset	0	0	0	0	0	1	0	0	1	0	0	1

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28	GPIO47	Aux. mode of GPIO_47	0: GPIO47 (IO) 1: LSCK0 (O) 2: Reserved 3: CMPDN (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
26:24	GPIO46	Aux. mode of GPIO_46	0: GPIO46 (IO) 1: LSCE_B0 (O) 2: EINT20 (I) 3: CMCS0 (I) 4: CLK04 (O) 5: Reserved 6: Reserved 7: Reserved
22:20	GPIO45	Aux. mode of GPIO_45	0: GPIO45 (IO) 1: TESTMODE_D (O) 2: Reserved 3: CMRST (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
18:16	GPIO44	Aux. mode of GPIO_44	0: GPIO44 (IO) 1: SDA (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
14:12	GPIO43	Aux. mode of GPIO_43	0: GPIO43 (IO) 1: SCL (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
10:8	GPIO42	Aux. mode of GPIO_42	0: GPIO42 (IO) 1: SIM2_SCLK (IO) 2: LSCE1_B1 (O) 3: Reserved 4: SDA18 (IO) 5: Reserved

Bit(s)	Mnemonic	Name	Description
6:4		GPIO41	6: Reserved 7: Reserved Aux. mode of GPIO_41 0: GPIO41 (IO) 1: SIM2_SRST (IO) 2: CLK03 (O) 3: U2CTS (I) 4: SCL18 (IO) 5: Reserved 6: Reserved 7: Reserved
2:0		GPIO40	Aux. mode of GPIO_40 0: GPIO40 (IO) 1: SIM2_SIO (IO) 2: Reserved 3: U2RTS (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved

A0020C54 **GPIO_MODE5 SET** GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO47				GPIO46				GPIO45				GPIO44			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO43				GPIO42				GPIO41				GPIO40			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE5

Bit(s)	Mnemonic	Name	Description
30:28		GPIO47	Bitwise SET operation for Aux. mode of LSCK 0: Keep 1: SET bits
26:24		GPIO46	Bitwise SET operation for Aux. mode of LSCE_B 0: Keep 1: SET bits
22:20		GPIO45	Bitwise SET operation for Aux. mode of TESTMODE_D 0: Keep 1: SET bits
18:16		GPIO44	Bitwise SET operation for Aux. mode of SDA28 0: Keep 1: SET bits
14:12		GPIO43	Bitwise SET operation for Aux. mode of SCL28 0: Keep 1: SET bits
10:8		GPIO42	Bitwise SET operation for Aux. mode of SIM2_SCLK 0: Keep

Bit(s)	Mnemonic	Name	Description
6:4		GPIO41	1: SET bits Bitwise SET operation for Aux. mode of SIM2_SRST 0: 1: SET bits
			Keep
2:0		GPIO40	Bitwise SET operation for Aux. mode of SIM2_SIO 0: 1: SET bits
			Keep

A0020C58 **GPIO_MODE5** **GPIO Mode Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
Name		GPIO47					GPIO46					GPIO45					GPIO44			
Type		WO					WO					WO					WO			
Reset		0	0	0	0		0	0	0	0		0	0	0	0		0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name		GPIO43					GPIO42					GPIO41					GPIO40			
Type		WO					WO					WO					WO			
Reset		0	0	0	0		0	0	0	0		0	0	0	0		0	0	0	0

Overview: For bitwise access of GPIO_MODE5

Bit(s)	Mnemonic	Name	Description
30:28		GPIO47	Bitwise CLR operation for Aux. mode of LSCK 0: 1: CLR bits
			Keep
26:24		GPIO46	Bitwise CLR operation for Aux. mode of LSCE_B 0: 1: CLR bits
			Keep
22:20		GPIO45	Bitwise CLR operation for Aux. mode of TESTMODE_D 0: 1: CLR bits
			Keep
18:16		GPIO44	Bitwise CLR operation for Aux. mode of SDA28 0: 1: CLR bits
			Keep
14:12		GPIO43	Bitwise CLR operation for Aux. mode of SCL28 0: 1: CLR bits
			Keep
10:8		GPIO42	Bitwise CLR operation for Aux. mode of SIM2_SCLK 0: 1: CLR bits
			Keep
6:4		GPIO41	Bitwise CLR operation for Aux. mode of SIM2_SRST 0: 1: CLR bits
			Keep
2:0		GPIO40	Bitwise CLR operation for Aux. mode of SIM2_SIO 0: 1: CLR bits
			Keep

A0020C60 **GPIO_MODE6** **GPIO Mode Control** **00001000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO55				GPIO54				GPIO53				GPIO52			
Type	RW				RW				RW				RW			
Reset	0	0	0		0	0	0		0	0	0		0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO51				GPIO50				GPIO49				GPIO48			
Type	RW				RW				RW				RW			
Reset	0	0	1	0	0	0	0	0		0	0	0		0	0	0

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO55	Aux. mode of GPIO_55 0: AGPIO55 (AGIO) 1: Reserved 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
26:24		GPIO54	Aux. mode of GPIO_54 0: AGPIO54 (AGIO) 1: Reserved 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
22:20		GPIO53	Aux. mode of GPIO_53 0: AGPIO53 (AGI) 1: SRCLKENAI (I) 2: EINT24 (I) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
18:16		GPIO52	Aux. mode of GPIO_52 0: AGPI52 (AGI) 1: Reserved 2: EINT23 (I) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
14:12		GPIO51	Aux. mode of GPIO_51 0: GPIO51 (IO) 1: RESETB (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
11:8		GPIO50	Aux. mode of GPIO_50 0: GPIO50 (IO)

Bit(s)	Mnemonic	Name	Description
			1: LPTE (I)
			2: EINT22 (I)
			3: CMCSK (I)
			4: CMCS2 (I)
			5: Reserved
			6: MCINS (I)
			7: Reserved
			8: Reserved
			9: CLK05 (O)
6:4		GPIO49	Aux. mode of GPIO_49
			0: GPIO49 (IO)
			1: LSA0DA0 (O)
			2: LSCE1_B0 (O)
			3: CMMCLK (O)
			4: Reserved
			5: Reserved
			6: Reserved
			7: Reserved
2:0		GPIO48	Aux. mode of GPIO_48
			0: GPIO48 (IO)
			1: LSDA0 (IO)
			2: EINT21 (I)
			3: CMCS1 (I)
			4: WIFITOB (I)
			5: Reserved
			6: Reserved
			7: Reserved

A0020C64 GPIO_MODE6 SET GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO55				GPIO54				GPIO53				GPIO52			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO51				GPIO50				GPIO49				GPIO48			
Type	WO				WO				WO				WO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_MODE6

Bit(s)	Mnemonic	Name	Description
30:28		GPIO55	Bitwise SET operation for Aux. mode of SRCLKENAI 0: Keep 1: SET bits
26:24		GPIO54	Bitwise SET operation for Aux. mode of EINT 0: Keep 1: SET bits
22:20		GPIO53	Bitwise SET operation for Aux. mode of TP4 0: Keep 1: SET bits
18:16		GPIO52	Bitwise SET operation for Aux. mode of TP3 0: Keep

Bit(s)	Mnemonic	Name	Description	
14:12		GPIO51	1: SET bits Bitwise SET operation for Aux. mode of RESETB 0: 1: SET bits	Keep
11:8		GPIO50	1: SET bits Bitwise SET operation for Aux. mode of LPTE 0: 1: SET bits	Keep
6:4		GPIO49	1: SET bits Bitwise SET operation for Aux. mode of LSA0 0: 1: SET bits	Keep
2:0		GPIO48	1: SET bits Bitwise SET operation for Aux. mode of LSDA 0: 1: SET bits	Keep

A0020C68 **GPIO_MODE6** **GPIO Mode Control** **00000000**
CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Name		GPIO55					GPIO54					GPIO53				GPIO52		
Type		WO					WO					WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name		GPIO51					GPIO50					GPIO49				GPIO48		
Type		WO					WO					WO				WO		
Reset		0	0	0	0	0	0	0		0	0	0		0	0	0		

Overview: For bitwise access of GPIO_MODE6

Bit(s)	Mnemonic	Name	Description	
30:28		GPIO55	1: CLR bits Bitwise CLR operation for Aux. mode of SRCLKENAI 0: 1: CLR bits	Keep
26:24		GPIO54	1: CLR bits Bitwise CLR operation for Aux. mode of EINT 0: 1: CLR bits	Keep
22:20		GPIO53	1: CLR bits Bitwise CLR operation for Aux. mode of TP4 0: 1: CLR bits	Keep
18:16		GPIO52	1: CLR bits Bitwise CLR operation for Aux. mode of TP3 0: 1: CLR bits	Keep
14:12		GPIO51	1: CLR bits Bitwise CLR operation for Aux. mode of RECLRB 0: 1: CLR bits	Keep
11:8		GPIO50	1: CLR bits Bitwise CLR operation for Aux. mode of LPTE 0: 1: CLR bits	Keep
6:4		GPIO49	1: CLR bits Bitwise CLR operation for Aux. mode of LSA0 0: 1: CLR bits	Keep
2:0		GPIO48	1: CLR bits Bitwise CLR operation for Aux. mode of LSDA	

Bit(s)	Mnemonic	Name	Description
			0: 1: CLR bits

A0020D10 GPIO_TDSEL **GPIO TDSEL Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Name																GPIO50	
Type																RW	
Reset																0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	GPIO46		GPIO42		GPIO41		GPIO36		GPIO25		GPIO18		GPIO9		GPIO8		
Type	RW		RW		RW		RW		RW		RW		RW		RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Overview: GPIO TX duty control register

Bit(s)	Mnemonic	Name	Description
17:16	GPIO50	GPIO50_TDSEL	GPIO50 Tx duty control
15:14	GPIO46	GPIO46_TDSEL	GPIO46 Tx duty control
13:12	GPIO42	GPIO42_TDSEL	GPIO42 Tx duty control
11:10	GPIO41	GPIO41_TDSEL	GPIO41 Tx duty control
9:8	GPIO36	GPIO36_TDSEL	GPIO36 Tx duty control
7:6	GPIO25	GPIO25_TDSEL	GPIO25 Tx duty control
5:4	GPIO18	GPIO18_TDSEL	GPIO18 Tx duty control
3:2	GPIO9	GPIO9_TDSEL	GPIO9 Tx duty control
1:0	GPIO8	GPIO8_TDSEL	GPIO8 Tx duty control

A0020D14 GPIO_TDSEL_S
ET **GPIO TDSEL Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Name																GPIO50	
Type																WO	
Reset																0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	GPIO46		GPIO42		GPIO41		GPIO36		GPIO25		GPIO18		GPIO9		GPIO8		
Type	WO		WO		WO		WO		WO		WO		WO		WO		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Overview: For bitwise access of GPIO_TDSEL

Bit(s)	Mnemonic	Name	Description
17:16	GPIO50	GPIO50_TDSEL	Bitwise SET operation of GPIO50_TDSEL Tx duty control 0: 1: SET bits
15:14	GPIO46	GPIO46_TDSEL	Bitwise SET operation of GPIO46_TDSEL Tx duty control 0: 1: SET bits
13:12	GPIO42	GPIO42_TDSEL	Bitwise SET operation of GPIO42_TDSEL Tx duty control

Bit(s)	Mnemonic	Name	Description	
			0: 1: SET bits	Keep
11:10	GPIO41	GPIO41_TDSEL	Bitwise SET operation of GPIO41_TDSEL Tx duty control 0: 1: SET bits	Keep
9:8	GPIO36	GPIO36_TDSEL	Bitwise SET operation of GPIO36_TDSEL Tx duty control 0: 1: SET bits	Keep
7:6	GPIO25	GPIO25_TDSEL	Bitwise SET operation of GPIO25_TDSEL Tx duty control 0: 1: SET bits	Keep
5:4	GPIO18	GPIO18_TDSEL	Bitwise SET operation of GPIO18_TDSEL Tx duty control 0: 1: SET bits	Keep
3:2	GPIO9	GPIO9_TDSEL	Bitwise SET operation of GPIO9_TDSEL Tx duty control 0: 1: SET bits	Keep
1:0	GPIO8	GPIO8_TDSEL	Bitwise SET operation of GPIO8_TDSEL Tx duty control 0: 1: SET bits	Keep

A0020D18 GPIO TDSEL C LR GPIO TDSEL Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Name																GPIO50	
Type																WO	
Reset																0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	GPIO46		GPIO42		GPIO41		GPIO36		GPIO25		GPIO18		GPIO9		GPIO8		
Type	WO		WO		WO		WO		WO		WO		WO		WO		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Overview: For bitwise access of GPIO_TDSEL

Bit(s)	Mnemonic	Name	Description	
17:16	GPIO50	GPIO50_TDSEL	Bitwise CLR operation of GPIO50_TDSEL Tx duty control 0: 1: CLR bits	Keep
15:14	GPIO46	GPIO46_TDSEL	Bitwise CLR operation of GPIO46_TDSEL Tx duty control 0: 1: CLR bits	Keep
13:12	GPIO42	GPIO42_TDSEL	Bitwise CLR operation of GPIO42_TDSEL Tx duty control 0: 1: CLR bits	Keep
11:10	GPIO41	GPIO41_TDSEL	Bitwise CLR operation of GPIO41_TDSEL Tx duty control 0: 1: CLR bits	Keep
9:8	GPIO36	GPIO36_TDSEL	Bitwise CLR operation of GPIO36_TDSEL Tx duty control 0: 1: CLR bits	Keep

Bit(s)	Mnemonic	Name	Description	
7:6	GPIO25	GPIO25_TDSEL	Bitwise CLR operation of GPIO25_TDSEL Tx duty control 0: 1: CLR bits	Keep
5:4	GPIO18	GPIO18_TDSEL	Bitwise CLR operation of GPIO18_TDSEL Tx duty control 0: 1: CLR bits	Keep
3:2	GPIO9	GPIO9_TDSEL	Bitwise CLR operation of GPIO9_TDSEL Tx duty control 0: 1: CLR bits	Keep
1:0	GPIO8	GPIO8_TDSEL	Bitwise CLR operation of GPIO8_TDSEL Tx duty control 0: 1: CLR bits	Keep

A0020E00 CLK_OUT0 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														CLK_OUT0		
Type														RW		
Reset													0	1	0	0

Overview: CLK_OUT0 Setting

Bit(s)	Mnemonic	Name	Description	
3:0	CLK_OUT0	CFG0	Selects clock output for CLKO_0 [1]: 26Mhz [4]: 32Khz_clock others: debug clock	clock

A0020E10 CLK_OUT1 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														CLK_OUT1		
Type														RW		
Reset													0	1	0	0

Overview: CLK_OUT1 setting

Bit(s)	Mnemonic	Name	Description	
3:0	CLK_OUT1	CFG1	Selects clock output for CLKO_1 [1]: 26Mhz [4]: 32Khz_clock others: debug clock	clock

A0020E20 **CLK_OUT2** **CLK Out Selection Control** **00000004**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT2			
Type													RW			
Reset													0	1	0	0

Overview: CLK_OUT2 setting

Bit(s)	Mnemonic	Name	Description
3:0	CLK_OUT2	CFG2	Selects clock output for CLKO_2 [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E30 **CLK_OUT3** **CLK Out Selection Control** **00000004**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT3			
Type													RW			
Reset													0	1	0	0

Overview: CLK_OUT3 setting

Bit(s)	Mnemonic	Name	Description
3:0	CLK_OUT3	CFG3	Selects clock output for CLKO_3 [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E40 **CLK_OUT4** **CLK Out Selection Control** **00000004**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT4			
Type													RW			
Reset													0	1	0	0

Overview: CLK_OUT4 setting

Bit(s)	Mnemonic	Name	Description
3:0	CLK_OUT4	CFG4	Selects clock output for CLKO_4 [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E50 CLK_OUT5 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Name																	
Type																	
Reset																	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name														CLK_OUT5			
Type														RW			
Reset													0	1	0	0	

Overview: CLK OUT5 setting

Bit(s)	Mnemonic	Name	Description
3:0	CLK_OUT5	CFG5	Selects clock output for CLKO_5 [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

3.5 General-purpose Timer

3.5.1 General Descriptions

Three general-purpose timers are provided. Two timers are 16 bits long and one timer is 32 bits long. Each runs independently. GPT1 ~ 2 use 32k clock source to count, whereas GPT4 uses 26M clock source. The 26M clock source can be gated when the system enters the sleep mode, and this will cause GPT4 to stop counting, whereas 32k clock source is always toggling. GPT1 and GPT2 can operate in one of the two modes: one-shot mode and auto-repeat mode. GPT4 are free running timer. In the one-shot mode, when the timer counts down and reaches 0, it will be halted. In the auto-repeat mode, when the timer reaches 0, it will simply be reset to counting down the initial value and repeating the count-down to 0. This loop keeps repeating until the disabling signal is set to 1. Regardless of the timer's mode, if the countdown initial value (i.e. GPTIMER1_DAT for GPT1 or GPTIMER_DAT2 for GPT2) is written when the timer is running, the new initial value will not take effect until the next time the timer is restarted. In the auto-repeat mode, the new countdown start value is used on the next countdown iteration. Therefore, before enabling the timer, the desired values for GPTIMER_DAT and the GPTIMER_PRESCALER registers must first be set.

Note that GPTimer's design is inherited form MT6260, GPT3 is removed for achieving lower cost.

For other timers, register addresses are the same with those of MT6260.

3.5.2 Register Definition

Module name: GPTimer base address: (+A00C000h)

Address	Name	Width	Register function
A00C0000	<u>GPTIMER1_CON</u>	32	GPT1 control register
A00C0004	<u>GPTIMER1_DAT</u>	32	GPT1 time-out interval register
A00C0008	<u>GPTIMER2_CON</u>	32	GPT2 control register
A00C000C	<u>GPTIMER2_DAT</u>	32	GPT2 time-out Interval register
A00C0010	<u>GPTIMER_STA</u>	32	GPT status register
A00C0014	<u>GPTIMER1_PRESCALER</u>	32	GPT1 prescaler register
A00C0018	<u>GPTIMER2_PRESCALER</u>	32	GPT2 prescaler register
A00C0028	<u>GPTIMER4_CON</u>	32	GPT4 control register
A00C002C	<u>GPTIMER4_DAT</u>	32	GPT4 data register

A00C0000 **GPTIMER1_CON** **GPT1 Control Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN	MODE														
Type	RW	RW														
Reset	0	0														

Bit(s)	Mnemonic	Name	Description
15	EN	EN	Controls GPT1 to start counting or to stop 0: Disable GPT1 1: Enable GPT1
14	MODE	MODE	Controls GPT1 to count repeatedly (in a loop) or just one-shot 0: One-shot mode is selected. 1: Auto-repeat mode is selected.

A00C0004 **GPTIMER1_DAT** **GPT1 Time-out Interval Register** **0000FFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0	CNT	CNT	Initial counting value GPT1 counts down from GPTIMER1_DAT. When GPT1 counts down to 0, a GPT1 interrupt will be generated.

A00C0008 GPTIMER2_CON GPT2 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN	MODE														
Type	RW	RW														
Reset	0	0														

Bit(s)	Mnemonic	Name	Description
15	EN	EN	Controls GPT2 to start counting or to stop 0: Disable GPT2 1: Enable GPT2
14	MODE	MODE	Controls GPT2 to count repeatedly (in a loop) or just one-shot 0: One-shot mode is selected. 1: Auto-repeat mode is selected.

A00C000C GPTIMER2_DAT GPT2 Time-out Interval Register 0000FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0	CNT	CNT	Initial counting value GPT2 counts down from GPTIMER2_DAT. When GPT2 counts down to 0, a GPT2 interrupt will be generated.

A00C0010 GPTIMER_STA GPT Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															GPT2	GPT1
Type															RC	RC
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	GPT2	GPT2	GP timer time-out status Each flag is set when the corresponding timer count-down is completed. Can be cleared when the CPU reads the status register.
0	GPT1	GPT1	GP timer time-out status Each flag is set when the corresponding timer count-down is completed. Can be cleared when the CPU reads the status register.

A00C0014 GPTIMER1 PRE SCALER GPT1 Prescaler Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRESCALER		
Type														RW		
Reset														1	0	0

Bit(s)	Mnemonic	Name	Description
2:0	PRESCALE R	PRESCALER	Controls the counting clock for GP timer 1 0: 16,384Hz 1: 8,192Hz 2: 4,096Hz 3: 2,048Hz 4: 1,024Hz 5: 512Hz 6: 256Hz 7: 128Hz

A00C0018 GPTIMER2 PRE SCALER GPT2 Prescaler Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRESCALER		
Type														RW		
Reset														1	0	0

Bit(s)	Mnemonic	Name	Description
2:0	PRESCALE R	PRESCALER	Controls the counting clock for GP timer 2 0: 16,384Hz 1: 8,192Hz 2: 4,096Hz 3: 2,048Hz 4: 1,024Hz 5: 512Hz 6: 256Hz 7: 128Hz

A00C0028 **GPTIMER4_CON** **GPT4 Control Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															LOCK	EN
Type															RW	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	LOCK	LOCK	Controls GPT4 EN bit can be modified or not If LOCK = 0, EN can be modified. If LOCK = 1, the EN value will be fixed, and the LOCK bit will always be 1 and cannot be modified until hardware reset. 0: Unlock 1: Lock
0	EN	EN	Controls GPT4 to start counting or to stop 0: Disable GPT4 1: Enable GPT4

A00C002C **GPTIMER4_DAT** **GPT4 Data Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CNT[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	CNT	CNT	If EN = 1, GPT4 will be a free running timer. This register records the GPT4 value. If EN = 0, this register will be cleared to 0. This register does not allow continuous read. It requires at least 1 26M clock cycle between 2 APB reads.

3.5.3 Application Note

When the GPT is in running status, GPTIMER_DAT cannot be configured. To start GPT1 or GPT2, SW should make sure the timer has finished counting for at least 3 cycles of the 32kHz clock.

3.6 MCU OSTIMER

3.6.1 Overview

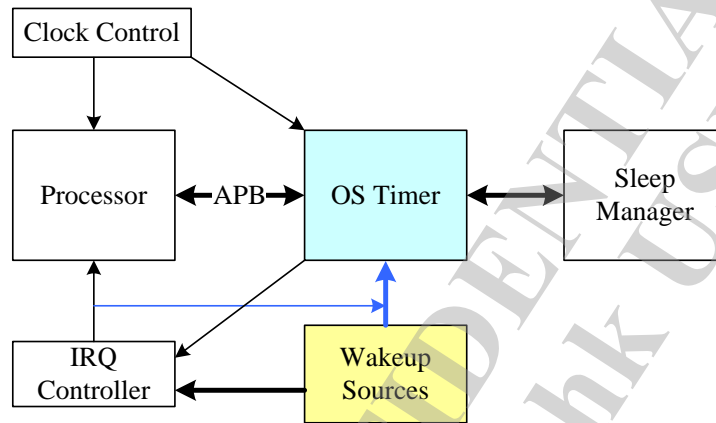


Figure 31. OS timer system view

The OS timer is a hardware timer which specifies the OS time frame duration and generates time-out interrupts by programming the frame counter number. The OS timer has the pause mode. The user can specify pause duration period before the pause mode, and the timer will resume from the pause mode by the external wakeup sources or when the pause duration is timed out.

3.6.2 Terminology

Table 46. Abbreviations

Abbreviation	
R/W	Read/Write
RO	Read only
WO	Write only
W1C	Write 1 to clear
RW1C	Read/Write 1 to clear
FRC	Free running counter in the system
OST	OS timer

3.6.3 Introduction to Wakeup Source

The OS timer only accepts level trigger wakeup source. The wakeup sources are all treated as asynchronous input (will be changed) and will be synchronized by OST clock in OST wakeup source controller.

The possible wakeup sources are listed in the table below.

Table 47. Wakeup sources

No	Wakeup source
0	GPT
1	EINT
2	Timer trigger
3	KP
4	MSDC
5	ANALOG
6	DSP
7	MSDC2
8	SPISLV
9	Reserved
10	Reserved
11	DSP_ASYNC
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved

Table 48. Characteristics of wakeup sources

No	Wakeup source	Edge/Level	SW clear	Clock domain
0	GPT	Edge		F32k_CK
1	EINT	Edge/Level		F32k_CK
2	Timer trigger	Level		F32k_CK
3	KP	Edge		
4	MSDC	Level		
5	ANALOG	Edge/Level		
6	DSP	Level		DSP_CK
7	MSDC2	Level		
8	SPISLV	Level		
9	Reserved			
10	Reserved			
11	DSP_ASYNC	Level		DSP_CK
12	Reserved			
13	Reserved			
14	Reserved			
15	Reserved			
16	Reserved			

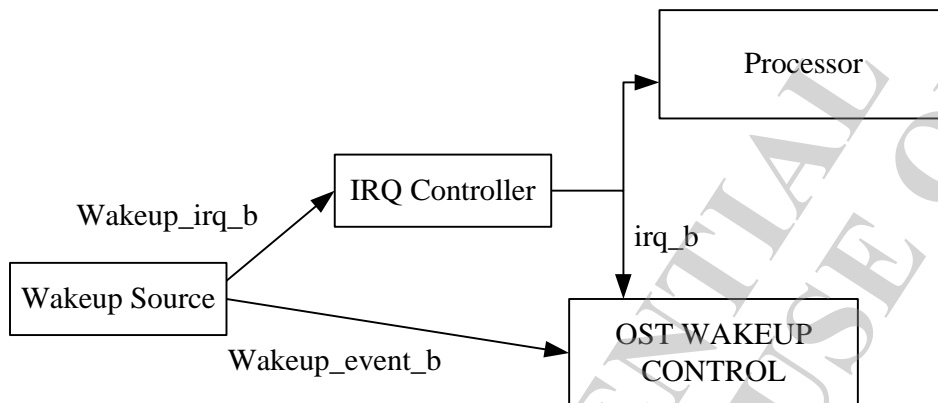


Figure 32. Wakeup event and Irq_b integration diagram

wakeup_irq_b may be asserted before wakeup_event_b is asserted from the wakeup source peripheral to ensure the software will enter wakeup ISR after the pause command is set.

Recommended software programming sequence:

1. I-BIT is set when a pause command is executed.
2. Set up IRQ mask registers to select IRQ wakeup sources
3. Pause criteria are met.
4. Set up pause command.
5. Confirm the pause command is executed at OST, and check if the pause command is pending or not.
6. Set up processor in request for interrupt state.
7. The processor clock will be off if there is no interrupt.
8. Check if the pause request command is completed (after the processor clock is active or resumes).
9. Clear I-BIT.

3.6.4 Register Definition

Address	Name	Width	Register function
0XA01F0000	<u>OST_CON</u>	32	OS timer control register Only updated when OST_CMD.OST_WR is set and OST_STA.WR_RDY is high.
0XA01F0004	<u>OST_CMD</u>	32	OS timer command Only valid when the write data BIT31 to BIT16 is 0x1153.
0XA01F0008	<u>OST_STA</u>	32	OS timer status
0XA01F000C	<u>OST_FRM</u>	32	OS timer frame duration Specifies OS timer frame duration by micron second resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before OS timer is enabled. The hardware will set up OST_ISR[0] periodically at frame duration period when the OS timer is enabled.
0XA01F0010	<u>OST_FRM_F32K</u>	32	OS timer frame duration by 32K clock

Address	Name	Width	Register function
			<p>Specifies OS timer frame duration with 30.5176 micron second (32kHz) resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before the OS timer is enabled.</p> <p>$OST_FRAM_F32K * 30.5176\mu s$ should be less than $OST_FRM_NUM * OST_FRM * 1\mu s - 30.5176\mu s$.</p> <p>Set up OST_FRM_NUM if the frame duration is shorter than system settling time.</p>
0XA01F0014	<u>OST_UFN</u>	32	<p>OS timer un-alignment frame number</p> <p>Specifies OS timer un-alignment event frame number count. This register value is updated to OS timer only when $OST_CMD.OST_WR$ is set. The read value of this register may not be the current OS time Un-alignment frame number. The hardware will count down the OST_UFN counter at frame time-out and stop the count-down when the current value is 0. When the OS timer is in the pause mode, this counter is still active and wakes up the OS timer when OST_UFN becomes 0 at frame time-out boundary. The software should enable the OS timer pause mode when OST_UFN is larger than 1, or the pause command will be ignored by the hardware.</p>
0XA01F0018	<u>OST_AFN</u>	32	<p>OS timer alignment frame number</p> <p>Specifies OS timer alignment event frame number count. This register value is updated to the OS timer only when $OST_CMD.OST_WR$ is set. The read value of this register may not be the current OS time alignment frame number. The hardware will count down the OST_AFN counter at frame time-out and stop the count-down when the current value is 0. The hardware will set $OST_ISR[1]$ when OST_AFN is 1 or 0 at frame time-out, and the OS timer is in normal mode. OST_AFN is current hardware AFN down counter value, and the software cannot read this register directly. The software can only read the current frame number through the OST_AFN_R register, and there is synchronization latency from OST_AFN to OST_AFN_R.</p>
0XA01F001C	<u>OST_AFN_DLY</u>	32	<p>OS timer alignment frame delay number count</p> <p>Specifies the OS timer alignment event frame delay count due to OS timer pause mode. The software has to use the $OST_CMD.OST_RD$ command to update this register value, or the value may be out of date.</p>
0XA01F0020	<u>OST_UFN_R</u>	32	<p>Current OS timer un-alignment frame number</p> <p>Specifies the OS timer current un-alignment frame number. The software has to use the $OST_CMD.OST_RD$ command to update this register value, or the value may be out of date.</p>
0XA01F0024	<u>OST_AFN_R</u>	32	<p>Current OS timer alignment frame number</p> <p>Specifies the OS timer current alignment frame number. The software has to use the $OST_CMD.OST_RD$ command to update this register value, or the value may be out of date.</p>
0XA01F0030	<u>OST_INT_MASK</u>	32	<p>OS timer interrupt mask</p> <p>Specifies the OS timer interrupt mask control.</p>
0XA01F0040	<u>OST_ISR</u>	32	<p>OS timer interrupt status</p> <p>Specifies the OS timer interrupt status. The software has to write 1 at the corresponding bit to clear the interrupt status bit. $OST_ISR[2-0]$ are also cleared when the OST_WR command is executed and AFN, UFN are updated.</p>

Address	Name	Width	Register function
0XA01F0050	OST_EVENT_MASK	32	OS timer event mask Specifies which wakeup event will be masked to wake up the OS timer in the OS timer pause mode.
0XA01F0054	OST_WAKEUP_STA	32	OS timer event wakeup status
0XA01F0060	OST_DBG_WAKEUP	32	OS timer debug wakeup

0XA01F0000 OS Timer Control Register OST_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														OST_DBG	UFN_DOWN	EN
Type														RW	RW	RW
Reset														0	1	0

Overview: This registers is only updated when OST_CMD.OST_WR is set and OST_STA.WR_RDY is high.

Bit(s)	Name	Description
2	OST_DBG	Enables OST wakeup debugging function 0: Disable 1: Enable
1	UFN_DOWN	Enables OST_UFN count-down 0: Disable OST_UFN count-down 1: Enable OST_UFN count-down
0	EN	Enables OS timer 0: OS timer is disabled. Then all internal timers are stopped. 1: OS Timer is enabled. The software has to ensure OST_AFN, OST_UFN, OST_FRAM are configured before enabling the OS timer.

0XA01F0004 OS Timer Command OST_CMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_KEY															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_CON_WR	OST_AFN_WR	OST_UFN_WR											OST_WR	OST_RD	PAUSE_STR
Type	RW	RW	RW											WO	WO	WO
Reset	0	0	0											0	0	0

Overview: The command is only valid when the write data BIT31 to BIT16 is 0x1153.

Bit(s)	Name	Description
31:16	OST_KEY	
15	OST_CON_WR	Updates OST_CON when the OST_WR command is active.
14	OST_AFN_WR	Updates OST_AFN when the OST_WR command is active.
13	OST_UFN_WR	Updates OST_UFN when the OST_WR command is active.
2	OST_WR	Write 1 to this bit to update the bus clock domain OS timer configuration into the OST SYSCLK domain
1	OST_RD	Write 1 to this bit to update the current OS timer status to the bus clock domain
0	PAUSE_STR	Write 1 to this bit to enable the OS timer pause function. The command will be ignored if the current OST UFN is less than 2. The software has to ensure OST_CMD.OST_WR is completed before the next software pause sequence.

0XA01F0008		OS Timer Status										OST_STA				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CPU_SLEEP									AFN_DLY_OVER		PAUSE_REQ			CMD_CPL	READY
Type	RO									RO		RO			RO	RO
Reset	0									0		0	0		1	0

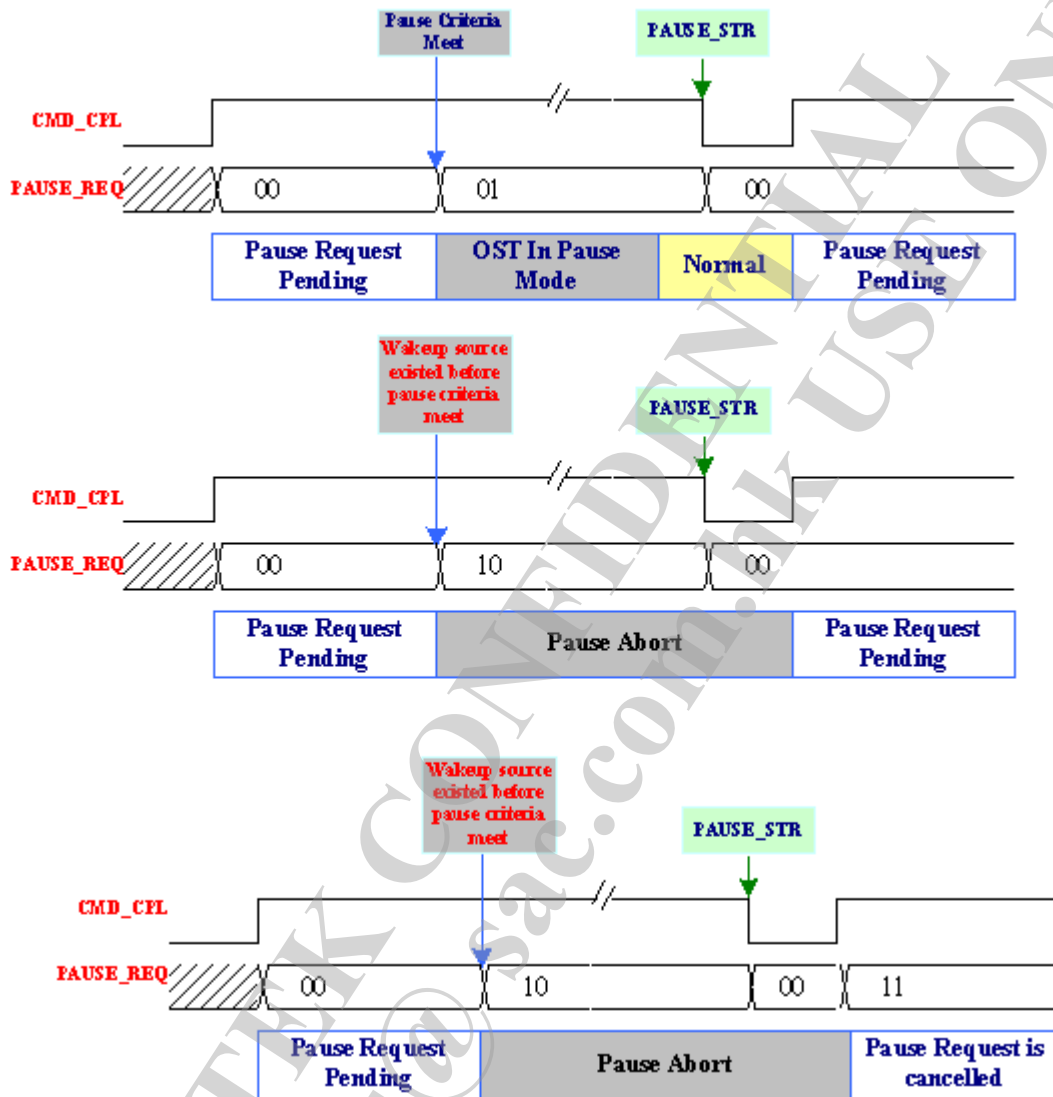


Figure 33. Pause command complete and pause request state

Bit(s)	Name	Description
15	CPU_SLEEP	The processor is in the sleep mode. (for debugging) 0: Active 1: Sleep
6	AFN_DLY_OVER	AFN_DLY counter overflows. This bit is cleared when AFN is updated. 0: Does not overflow 1: Overflow
4:3	PAUSE_REQ	An OS timer pause request is pending. A pause command will be completed when the pause command is set. 1. Processor is in the sleep mode (processor clock is off), UFN ≥ 2 and no wakeup sources 2. Any wakeup sources are sensed after the pause command is set. 3. UFN < 2 00: The last pause command request is not completed yet (CP15 is not enable)

Bit(s)	Name	Description
1	CMD_CPL	and no wakeup source). 01: The last pause command request is completed with OST pause mode being active. 10: The last pause command request is completed with wakeup sources. 11: The last pause command request is completed with UFN < 2. OST command is completed. It takes several clocks from OST_CMD being updated to command being active. 0: OST command is not completed. 1: OST command is completed.
0	READY	OS timer status 0: OST is in pause mode. 1: OST is in normal mode.

0XA01F000C OS Timer Frame Duration OST_FRM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_FRM															
Type	RW															
Reset				1	0	0	0	0	0	1	0	0	1	0	0	1

Overview: This register specifies the OS timer frame duration by micron second resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before the OS timer is enabled. The hardware will set up OST_ISR[0] periodically at frame duration period when the OS timer is enabled.

Bit(s)	Name	Description
12:0	OST_FRM	

0XA01F0010 OS Timer Frame Duration by 32K Clock OST_FRM_F32K

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN								OST_FRM_F32K							
Type	RW								RW							
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer frame duration with 30.5176 micron second (32kHz) resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set up before the OS timer is enabled. OST_FRAM_F32K*30.5176us should be less than OST_FRM_NUM*OST_FRM*1us - 30.5176us. Set up OST_FRM_NUM if the frame duration is shorter than the system settling time.

Bit(s)	Name	Description
--------	------	-------------

Bit(s)	Name	Description
15:12	OST_UFN	
11:0	OST_FRM_F32K	

0XA01F0014 OS Timer Un-alignment Frame Number OST_UFN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_UFN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer un-alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time un-alignment frame number. The hardware will count down the OST_UFN counter at frame time-out and stop the count-down when the current value is 0. When the OS timer is in the pause mode, this counter is still active and wakes up the OS timer when OST_UFN becomes 0 at frame time-out boundary. The software should enable the OS timer pause mode when OST_UFN is larger than 1, or the pause command will be ignored by hardware.

Bit(s)	Name	Description
31:0	OST_UFN	

0XA01F0018 OS Timer Alignment Frame Number OST_AFN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time alignment frame number. The hardware will count down the OST_AFN counter at frame time-out and stop the count-down when the current value is 0. The hardware will set up OST_ISR[1] when OST_AFN is 1 or 0 at frame time-out and the OS timer is in the normal mode. OST_AFN is the current hardware AFN down counter value, and the software cannot read this register directly. The software can only read the current frame number through the OST_AFN_R register, and there is synchronization latency from OST_AFN to OST_AFN_R.

Bit(s)	Name	Description
31:0	OST_AFN	

0XA01F001C OS Timer Alignment Frame Delay Number Count OST_AFN_DLY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN_DLY[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN_DLY[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS Timer Alignment event frame delay count due to OS timer pause mode. Software has to use OST_CMD.OST_RD command to update this register value, or the value maybe out of date.

Bit(s)	Name	Description
31:0	OST_AFN_DLY	

0XA01F0020 Current OS Timer Un-alignment Frame Number OST_UFN_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_UFN_R[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN_R[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer current un-alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.

Bit(s)	Name	Description
31:0	OST_UFN_R	

0XA01F0024 Current OS Timer Alignment Frame Number OST_AFN_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN_R[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN_R[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer current alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.

Bit(s)	Name	Description
31:0	OST_AFN_R	

0XA01F0030 OS Timer Interrupt Mask OST_INT_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													OST_INT_MASK			
Type													RW			
Reset												1	1	1	1	1

Overview: This register specifies the OS timer interrupt mask control.

Bit(s)	Name	Description
4:0	OST_INT_MASK	0: Mask OS timer frame time-out interrupt 1: Mask OS timer alignment frame time-out interrupt 2: Mask OS timer un-alignment frame time-out interrupt 3: Mask OS timer pause abort interrupt 4: Mask OS timer pause interrupt

0XA01F0040 OS Timer Interrupt Status OST_ISR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												OST_ISR				
Type												W1C				
Reset												0	0	0	0	0

Overview: This register specifies the OS timer interrupt status. The software has to write 1 to the corresponding bit to clear the interrupt status bit. OSR_ISR[2-0] are also cleared when the OSR_WR command is executed and AFN, UFN are updated.

Bit(s)	Name	Description
4:0	OST_ISR	0: OS timer frame time-out interrupt status 1: OS timer alignment frame time-out interrupt status 2: OS timer un-alignment frame time-out interrupt status 3: OS timer pause abort interrupt status 4: OS timer pause interrupt status

0XA01F0050 OS Timer Event Mask OST_EVENT_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Name																			OST_EVENT_MASK [18:16]	
Type																				RW
Reset																				0 1 1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	OST_EVENT_MASK[15:0]																			
Type	RW																			
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Overview: This register specifies which wakeup event will be masked to wake up the OS timer in the OS timer pause mode.

Bit(s)	Name	Description
18:0	OST_EVENT_MASK	

0XA01F0054	OS Timer Event Wakeup Status												OST_WAKEUP_STA							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
Name																	OST_WAKEUP_STA [18:16]			
Type																	RO			
Reset																	0 0 0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	OST_WAKEUP_STA[15:0]																			
Type	RO																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit(s)	Name	Description
18:0	OST_WAKEUP_STA	

0XA01F0060	OS Timer Debug Wakeup												OST_DBG_WAKEOSUP							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
Name	CIRQ_MASK_EN																OST_DBG_WAKEU P[18:16]			
Type	RW																			
Reset	0																0 0 0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	OST_DBG_WAKEUP[15:0]																			
Type																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit(s)	Name	Description
31	CIRQ_MASK_EN	0: Disable cirq mask function 1: Enable cirq mask function
18:0	OST_DBG_WAKEUP	Controls wakeup status in debug timing event1

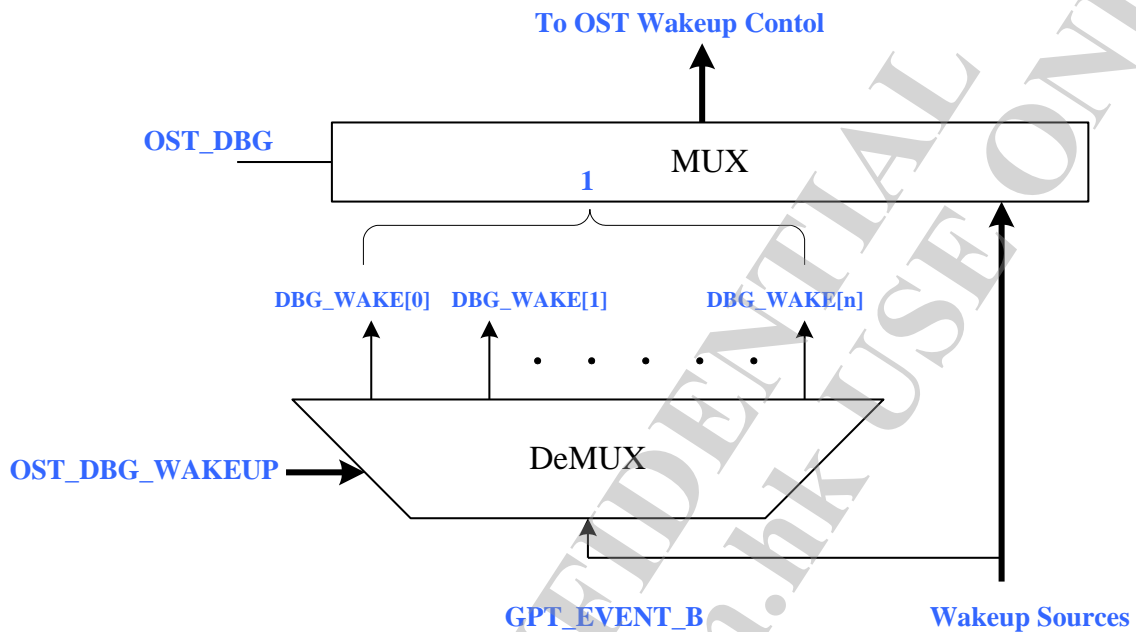


Figure 34. Debug wakeup events

3.7 UART1

3.7.1 General Description

The baseband chipset houses two UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external dev ices.

The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths **from 5 to 8 bits, an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. 8 modem control lines and a diagnostic loop-back mode are provided. UART also includes two DMA handshake lines, indicating when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.

Note that UART is designed so that all internal operation is synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and industry standard 16550A device, which means that the core is not clock for clock identical to the original device.

After a hardware reset, UART will be in M16C450 mode. Its FIFOs can then be enabled and UART can enter M16550A mode. UART has further functions beyond the M16550A mode. Each of the

extended functions can be selected individually under software control.

UART provides more powerful enhancements than the industry-standard 16550:

- Hardware flow control. This feature is very useful when the ISR latency is hard to predict and control in the embedded applications. The MCU is relieved of having to fetch the received data within a fixed amount of time.

Note that in order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The enhanced mode bit ensures that UART is backward compatible with the software that has been written for 16C450 and 16550A devices. **Figure 35** is the block diagram of the UART1 device.

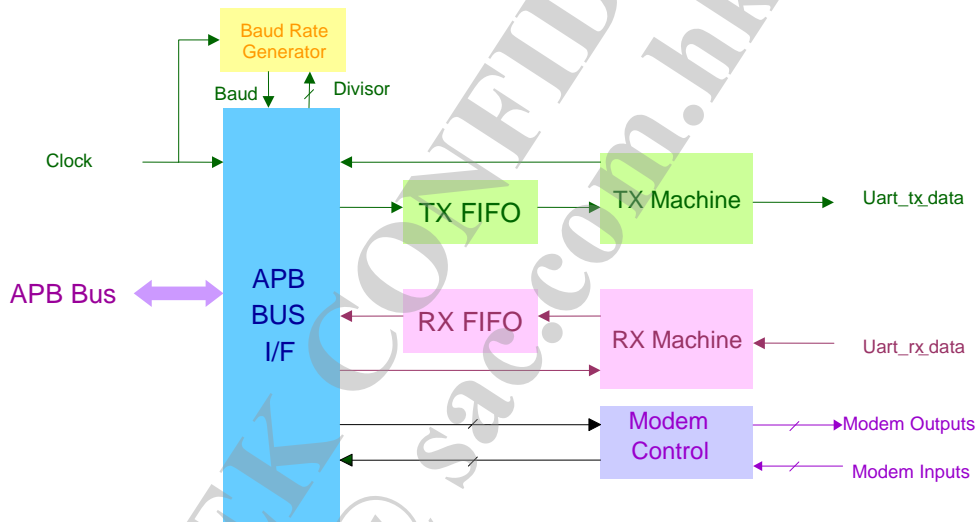


Figure 35. Block Diagram of UART1

3.7.2 Register Definition

Module name: UART1 Base address: (+A0080000h)

Address	Name	Width	Register Function
A0080000	<u>RBR</u>	8	RX Buffer Register Note: Only when LCR[7] = 0.
A0080000	<u>THR</u>	8	TX Holding Register Note: Only when LCR[7] = 0
A0080000	<u>DLL</u>	8	Divisor Latch (LS) Divides the bclk frequency Note: Modified when LCR[7]!=0
A0080004	<u>IER</u>	8	Interrupt Enable Register Note: Only when LCR[7] = 0. By storing 1 to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt will be disabled.

Address	Name	Width	Register Function
			IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.
A0080004	<u>DLM</u>	8	Divisor Latch (MS) used to divid the bclk frequency . *NOTE: modified when LCR[7]!=0
A0080008	<u>IIR</u>	8	Interrupt Identification Register Note: Only when LCR!=BF'h. Priority is from high to low as the following. IIR[5:0]=0X1: No interrupt pending IIR[5:0]=0X6: Line status interrupt (under IER[2]=1) IIR[5:0]=0Xc: RX data time-out interrupt (under IER[0]=1) IIR[5:0]=0X4: RX data are placed in the RX bBuffer register or the RX trigger level is reached. (under IER[0]=1). IIR[5:0]=0X2: TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level (under IER[1]=1). IIR[5:0]=0X10: XOFF character received (under IER[5]=1,EFR[4] = 1).
A0080008	<u>FCR</u>	8	FIFO Control Register FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh
A0080008	<u>EFR</u>	8	Enhanced Feature Register Note: Only when LCR=BF'h
A008000C	<u>LCR</u>	8	Line Control Register Determines characteristics of serial communication signals.
A0080010	<u>MCR</u>	8	Modem Control Register Controls interface signals of the UART. MCR[5:0] are modified when LCR != 8'hBF, MCR[7] can be read when LCR != 8'hBF & EFR[4] = 1.
A0080010	<u>XON1</u>	8	XON1 Char Register Note: XON1 is modified only when LCR=BF'h.
A0080014	<u>LSR</u>	8	Line Status Register Modified when LCR != BFh.
A0080018	<u>XOFF1</u>	8	XOFF1 Char Register *Note: XOFF1 is modified only when LCR=BF'h.
A008001C	<u>SCR</u>	8	Scratch Register A general purpose read/write register. After reset, its value will be un-defined. Modified when LCR != BFh.
A0080020	<u>AUTOBAUD EN</u>	8	Auto Baud Detect Enable Register
A0080024	<u>HIGHSPEED</u>	8	High Speed Mode Register
A0080028	<u>SAMPLE COUNT</u>	8	Sample Counter Register When HIGHSPEED=3, sample_count will be the threshold value for UART sample counter (sample_num). Count from 0 to sample_count.
A008002C	<u>SAMPLE POINT</u>	8	Sample Point Register When HIGHSPEED=3, UART gets the input data when sample_count=sample_num, e.g. system clock = 13MHz,

Address	Name	Width	Register Function
			921600 = 13000000/14. Therefore, sample_count = 13, and sample point = 6 (sampling the central point to decrease the inaccuracy) SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.
A0080030	<u>AUTOBAUD_REG</u>	8	Auto Baud Monitor Register the autobaud detection state ,it will not change until enable the autobaud_en again.
A0080034	<u>RATEFIX_AD</u>	8	Clock Rate Fix Register
A0080038	<u>AUTOBAUDSAMPL E</u>	8	Auto Baud Sample Register Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13. When system clock = 52MHz, autobaudsample = 27.
A008003C	<u>GUARD</u>	8	Guard Time Added Register
A0080040	<u>ESCAPE_DAT</u>	8	Escape Character Register
A0080044	<u>ESCAPE_EN</u>	8	Escape Enable Register
A0080048	<u>SLEEP_EN</u>	8	Sleep Enable Register
A008004C	<u>DMA_EN</u>	8	DMA Enable Register
A0080050	<u>RXTRI_AD</u>	8	Rx Trigger Address
A0080054	<u>FRACDIV_L</u>	8	Fractional Divider LSB Address
A0080058	<u>FRACDIV_M</u>	8	Fractional Divider MSB Address
A008005C	<u>FCR_RD</u>	8	FIFO Control Register

A0080000 RBR RX Buffer Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RBR							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	RBR	Read-only register The received data can be read by accessing this register. Only when LCR[7] = 0.

A0080000 THR TX Holding Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									THR							
Type									WO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	THR	TX Holding Register Write-only register. The data to be transmitted are written to this register and sent to the PC via serial communication.

Bit(s)	Name	Description
		Only when LCR[7] = 0.

A0080000 **DLL** **Divisor Latch (LS)** **01**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLL							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	DLL	Divisor latch low 8-bit data <i>Note: Modified when LCR[7]!=0.</i>

A0080004 **IER** **Interrupt Enable Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CTSI	RTSI	XOFFI		EDSSI	ELSI	ETBEI	ERBFI
Type									RW	RW	RW		RW	RW	RW	RW
Reset									0	0	0		0	0	0	0

Bit(s)	Name	Description
7	CTSI	Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line. <i>Note: This interrupt is only enabled when hardware flow control is enabled.</i> 0: Mask an interrupt that is generated when a rising edge is detected on the CTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the CTS modem control line.
6	RTSI	Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line. <i>Note: This interrupt is only enabled when hardware flow control is enabled.</i> 0: Mask an interrupt that is generated when a rising edge is detected on the RTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the RTS modem control line.
5	XOFFI	Masks an interrupt that is generated when an XOFF character is received. <i>Note: This interrupt is only enabled when software flow control is enabled.</i> 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.
3	EDSSI	When set to 1, an interrupt will be generated if DCTS (MSR[0]) becomes set. 0: No interrupt is generated if DCTS (MSR[0]) becomes set. 1: An interrupt is generated if DCTS (MSR[0]) becomes set.
2	ELSI	When set to 1, an interrupt will be generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
1	ETBEI	When set to 1, an interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level. 0: No interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level. 1: An interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.
0	ERBFI	When set to 1, an interrupt will be generated if RX data are placed in RX buffer register or the RX trigger level is reached.

Bit(s)	Name	Description
		0: No interrupt will be generated if RX data are placed in the RX buffer register or the RX trigger level is reached. 1: An interrupt will be generated if RX Data are placed in the RX buffer register or the RX trigger level is reached.

A0080004 DLM Divisor Latch (MS) 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLM							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	DLM	<p>Divisor latch high 8-bit data</p> <p>Note: Modified when LCR[7]!=0. DLL & DLM can only be updated when DLAB(LCR[7]) is set to 1. Division by 1 will generate a BAUD signal that is constantly high. DLL & DLM setting formula is {DLH,DLL}=(system clock frequency/baud_pulse/baud_rate). When RATE_FIX(RATEFIX_AD[0])=0, system clock frequency = 52MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=0, system clock frequency = 26MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=1, system clock frequency = 13MHz. For baud_pulse value, refer to HIGH_SPEED(offset=24H) register For example, when at 52MHz, default speed mode and 115200 baud rate, {DLH,DLL}=52MHz/16/115200=28.</p>

A0080008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE			ID				
Type									RO			RU				
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:6	FIFOE	
5:0	ID	<p>IIR[5:0] Priority level interrupt source</p> <p>000001 - No interrupt pending 000110 1 Line status interrupt: BI, FE, PE or OE set in LSR. (Under IER[2]=1) 001100 2 RX data time-out: Time-out on character in RX FIFO. (Under IER[0]=1) 000100 3 RX data received: RX data received or RX trigger level reached. (Under IER[0]=1) 000010 4 TX holding register empty: 000000 5 Modem status change: DCTS set in MSR. (Under IER[3]=1) TX Holding Register empty or TX FIFO trigger level reached. (Under IER[1]=1) 010000 6 Software flow control: XOFF Character received. (Under IER[5]=1) 100000 7 Hardware flow control: CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1)</p> <p>Line status interrupt: A RX line status interrupt (IIR[5:0] = 000110b) will be generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the line status register.</p> <p>RX data time-out interrupt: When the virtual FIFO mode is disabled, RX data time-out</p>

Bit(s)	Name	Description
		<p>interrupt will be generated if all of the following conditions are applied:</p> <ol style="list-style-type: none"> 1. FIFO contains at least one character. 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits); 3. The most recent CPU read of the FIFO is longer than four character periods ago. <p>The timeout timer is restarted upon receipt of a new byte from the RX shift register or upon a CPU read from the RX FIFO. The RX data time-out interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading RX FIFO.</p> <p>When the virtual FIFO mode is enabled, RX data time-out interrupt will be generated if all of the following conditions are applied:</p> <ol style="list-style-type: none"> 1. FIFO is empty. 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits). 3. The most recent CPU read of the FIFO is longer than four character periods ago. <p>The timeout timer is restarted upon receipt of a new byte from the RX shift register or reading DMA_EN register. The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading DMA_EN register.</p> <p>RX data received interrupt: A RX received interrupt (IER[5:0] = 000100b) is generated if EFRBI (IER[0]) is set and either RX data are placed in the RX buffer register or the RX trigger level is reached. The interrupt is cleared by reading the RX buffer register or the RX FIFO (if enabled).</p> <p>TX holding register empty interrupt: A TX holding register empty interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX holding register is empty or the contents of the TX FIFO are reduced to its trigger level. The interrupt is cleared by writing to the TX holding register or TX FIFO if FIFO is enabled.</p> <p>Modem status change interrupt: A modem status change Interrupt (IIR[5:0] = 000000b) will be generated if EDSSI (IER[3]) is set and e DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the modem status register.</p> <p>Software flow control interrupt: A software flow control interrupt (IIR[5:0] = 010000b) will be generated if the software flow control is enabled and XOFF1 (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the interrupt identification register.</p> <p>Hardware flow control interrupt: A hardware flow control interrupt (IER[5:0] = 100000b) will be generated if the hardware flow control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS modem control line. The interrupt is cleared by reading the interrupt identification register.</p>

A0080008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									WO		WO			WO	WO	WO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	<p>RX FIFO trigger threshold</p> <p>RX FIFO contains total 32 bytes.</p> <p>0: 1</p> <p>1: 6</p> <p>2: 12</p>

Bit(s)	Name	Description	
		3: RXTRIG	
5:4	TFTL1_TFTL0	TX FIFO trigger threshold TX FIFO contains total 32 bytes.	
		0:	1
		1:	4
		2:	8
		3: 14	
2	CLRT	Control bit to clear TX FIFO	
		0: No	effect
		1: Clear TX FIFO	
1	CLRR	Control bit to clear RX FIFO	
		0: No	effect
		1: Clear RX FIFO	
0	FIFOE	Enables FIFO This bit must be set to 1 for any of other bits in the registers to have any effect.	
		0: Disable both RX and TX FIFOs.	
		1: Enable both RX and TX FIFOs.	

A0080008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTO_CTS	AUTO_RTS		ENABLE_E	SW_FLOW_CONT			
Type									RW	RW		RW	RW			
Reset									0	0		0	0	0	0	0

Bit(s)	Name	Description	
7	AUTO_CTS	Enables hardware transmission flow control	
		0: Disable	
		1: Enable	
6	AUTO_RTS	Enables hardware reception flow control	
		0: Disable	
		1: Enable	
4	ENABLE_E	Enables enhancement feature	
		0: Disable	
		1: Enable	
3:0	SW_FLOW_CONT	Software flow control bits	
		00xx: No TX flow control bytes	
		01xx: No TX flow control bytes	
		10xx: Transmit XON1/XOFF1 as flow control bytes	
		xx00: No RX flow control	
		xx01: No RX flow control	
		xx10: Receive XON1/XOFF1 as flow control bytes	

A008000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
Type									RW	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	DLAB	Divisor latch access bit

Bit(s)	Name	Description
6	SB	<p>0: RX and TX registers are read/written at Address 0 and the IER register is read/written at Address 4.</p> <p>1: Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.</p> <p>Set break</p> <p>0: No effect</p> <p>1: SOUT signal is forced to the 0 state.</p>
5	SP	<p>Stick parity</p> <p>0: No effect.</p> <p>1: The parity bit is forced to a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the parity bit will be set and checked = 0. If EPS=0 & PEN=1, the parity bit will be set and checked = 1.</p>
4	EPS	<p>Selects even parity</p> <p>0: When EPS=0, an odd number of ones is sent and checked.</p> <p>1: When EPS=1, an even number of ones is sent and checked.</p>
3	PEN	<p>Enables parity</p> <p>0: The parity is neither transmitted nor checked.</p> <p>1: The parity is transmitted and checked.</p>
2	STB	<p>Number of STOP bits</p> <p>0: One STOP bit is always added.</p> <p>1: Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.</p>
1:0	WLS1_WLS0	<p>Selects word length</p> <p>0: 5 bits</p> <p>1: 6 bits</p> <p>2: 7 bits</p> <p>3: 8 bits</p>

A0080010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF_STATU S			Loop			RTS	
Type									RU			RW			RW	
Reset									0			0			0	

Bit(s)	Name	Description
7	XOFF_STATUS	<p>Read-only bit</p> <p>0: When an XON character is received.</p> <p>1: When an XOFF character is received.</p>
4	Loop	<p>Loop-back control bit</p> <p>0: No loop-back is enabled.</p> <p>1: Loop-back mode is enabled.</p>
1	RTS	<p>Controls the state of the output NRTS, even in loop mode.</p> <p>0: RTS will always output 1.</p> <p>1: RTS's output will be controlled by flow control condition.</p>

A0080010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name									XON1									
Type									RW									
Reset									0	0	0	0	0	0	0	0		

Bit(s)	Name	Description
7:0	XON1	XON1 character for software flow control Modified only when LCR=BF'h.

A0080014 **LSR** **Line Status Register** **60**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE RR	TEMT	THRE	BI	FE	PE	OE	DR
Type									RU	RU	RU	RU	RU	RU	RU	RU
Reset									0	1	1	0	0	0	0	0

Bit(s)	Name	Description
7	FIFOERR	RX FIFO error indicator 0: No PE, FE, BI set in the RX FIFO. 1: Set to 1 when there is at least one PE, FE or BI in the RX FIFO.
6	TEMT	TX holding register (or TX FIFO) and the TX shift register are empty. 0: Empty conditions below are not met. 1: If FIFOs are enabled, the bit will be set whenever the TX FIFO and the TX shift register are empty. If FIFOs are disabled, the bit will be set whenever TX holding register and TX shift register are empty.
5	THRE	Indicates if there is room for TX holding register or TX FIFO is reduced to its trigger level 0: Reset whenever the contents of the TX FIFO are more than its trigger level (FIFOs are enabled), or whenever TX holding register is not empty (FIFOs are disabled). 1: Set whenever the contents of the TX FIFO are reduced to its trigger level (FIFOs are enabled), or whenever TX holding register is empty and ready to accept new data (FIFOs are disabled).
4	BI	Break interrupt 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). If the FIFOs are enabled, this error will be associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: The next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.
3	FE	Framing error 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid STOP bit. If the FIFOs are enabled, the state of this bit will be revealed when the byte it refers to is the next to be read.
2	PE	Parity error 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid parity bit. If the FIFOs are enabled, the state of this bit will be revealed when the referred byte is the next to be read.
1	OE	Overrun error 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the RX buffer is not read by the CPU before the new data from the RX shift register overwrites the previous contents. If the FIFOs are enabled, an overrun error will occur when the RX FIFO is full and the RX shift register becomes full. OE will be set as soon as this happens. The character in the shift register is then overwritten, but not transferred to the FIFO.
0	DR	Data ready 0: Cleared by the CPU reading the RX buffer or by reading all the FIFO bytes. 1: Set by the RX buffer becoming full or by the FIFO becoming no empty.

A0080018 **XOFF1** **XOFF1 Char Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XOFF1															
Type	RW															
Reset	0 0 0 0 0 0 0 0															

Bit(s)	Name	Description
7:0	XOFF1	XOFF1 character for software flow control Modified only when LCR=BF'h.

A008001C **SCR** **Scratch Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCR															
Type	RW															
Reset	0 0 0 0 0 0 0 0															

Bit(s)	Name	Description
7:0	SCR	General purpose read/write register After reset, its value will be undefined. Modified when LCR != BFh.

A0080020 **AUTOBAUD_EN** **Auto Baud Detect Enable Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														SLEEP_ACK_SEL	AUTOBAUD_SEL	AUTOBAUD_EN
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	SLEEP_ACK_SEL	Selects sleep ack when autobaud_en 0: Support sleep_ack when autobaud_en is opened . 1: Does not support sleep_ack when autobaud_en is opened .
1	AUTOBAUD_SEL	Selects auto-baud 0: Support standard baud rate detection 1: Support non_standard baud rate detection (support baud from 110 to 115200; recommended to use 52MHz to auto fix) .
0	AUTOBAUD_EN	Auto-baud enabling signal 0: Disable auto-baud function 1: Enable auto-baud function (UARTn+0024h SPEED should be set to 0.) <i>Note: When AUTOBAUD_EN is active, there should not be A*/a* char before the auto baud char AT/at. If A*/a* is Inevitable, autobaud will fail and please disable AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.</i>

A0080024 **HIGHSPEED** **High Speed Mode Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPEED															
Type	RW															

Bit(s)	Name	Description
11:		at_8E1
12:		at_801
13:		Autobaud detection fails
3:0	BAUD_RATE	Autobaud baud rate (only true value in standard autobaud detection)
0:		115,200
1:		57,600
2:		38,400
3:		19,200
4:		9,600
5:		4,800
6:		2,400
7:		1,200
8:		300
9:		110

A0080034 **RATEFIX_AD** **Clock Rate Fix Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														FREQ_SEL	AUTOB AUD_R ATE_FI X	RATE_ FIX
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	FREQ_SEL	0: Select 26MHz as system clock 1: Select 13MHz as system clock
1	AUTOBAUD_RATE_FIX	0: Use 52MHz as system clock for UART auto baud detection 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART auto baud detection
0	RATE_FIX	0: Use 52MHz as system clock for UART TX/RX 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART TX/RX

A0080038 **AUTOBAUDSA
MPLE** **Auto Baud Sample Register** **0D**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset											0	0	1	1	0	1

Bit(s)	Name	Description
5:0	AUTOBAUDSAMPLE	clk diveision for autobaud rate detection For standard baud rate detection. System clk 52m: 'd 27 System clk 26m: 'd 13 System clk 13m: 'd 6 For non-standard baud rate detection. :15.

A008003C **GUARD** **Guard Time Added Register** **0F**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUAR D_EN	GUARD_CNT			
Type												RW	RW			
Reset												0	1	1	1	1

Bit(s)	Name	Description
4	GUARD_EN	Guard interval add enabling signal 0: No guard interval added 1: Add guard interval after stop bit.
3:0	GUARD_CNT	Guard interval count value Guard interval = $[1/(\text{system clock}/\text{div_step}/\text{div})] * \text{GUARD_CNT}$.

A0080040 ESCAPE_DAT Escape Character Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ESCAPE_DAT															
Type	RW															
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	ESCAPE_DAT	Escape character added before software flow control data and escape character If TX data are xon (31h), with esc_en =1, UART will transmit data as esc + CEh (~xon).

A0080044 ESCAPE_EN Escape Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ESC_EN															
Type	RW															
Reset	0															

Bit(s)	Name	Description
0	ESC_EN	Adds escape character in transmitter and removes escape character in receiver by UART 0: Does not deal with the escape character 1: Add escape character in transmitter and remove escape character in receiver

A0080048 SLEEP_EN Sleep Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SLEEP_EN															
Type	RW															
Reset	0															

Bit(s)	Name	Description
0	SLEEP_EN	For sleep mode issue 0: Does not deal with sleep mode indicate signal 1: Activate hardware flow control or software control according to software initial setting when the chip enters sleep mode. Release hardware flow when the chip wakes up. However, for software control, UART sends xon when awoken and when FIFO does not

Bit(s)	Name	Description
		reach threshold level.

A008004C DMA_EN DMA Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FIFO_lsr_sel	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Name	Description
3	FIFO_lsr_sel	Selects FIFO LSR mode 0: LSR will hold the first line status error state until you read the LSR register. 1: LSR will update automatically.
2	TO_CNT_AUTORST	Time-out counter auto reset register 0: After RX time-out happens, SW shall reset the interrupt by reading UART 0x4C. 1: The time-out counter will be auto reset. Set this register when Rain's new DMA is used.
1	TX_DMA_EN	TX_DMA mechanism enabling signal 0: Does not use DMA in TX 1: Use DMA in TX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt for DMA.for DMA.
0	RX_DMA_EN	RX_DMA mechanism enabling signal 0: Does not use DMA in RX 1: Use DMA in RX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt

A0080050 RXTRI_AD Rx Trigger Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RXTRIG			
Type													RW			
Reset													0	0	0	0

Bit(s)	Name	Description
3:0	RXTRIG	When {rtm,rtl}=2'b11, the RX FIFO threshold will be Rxtrig. The value is suggested to be smaller than half of RX FIFO size, which is 32 bytes.

A0080054 FRACDIV_L Fractional Divider LSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FRACDIV_L	Adds sampling count (+1) from state data7 to data0 to contribute fractional divisor.only when high_speed=3.

A0080058 **FRACDIV_M** **Fractional Divider MSB Address** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name																FRACDIV_M	
Type																RW	
Reset																0	0

Bit(s)	Name	Description
1:0	FRACDIV_M	Adds sampling count when in state stop to parity to contribute fractional divisor. only when high_speed=3.

A008005C **FCR_RD** **FIFO Control Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									RO		RO			RO	RO	RO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	RX FIFO trigger threshold RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	TX FIFO trigger threshold TX FIFO contains total 32 bytes. 0: 1 1: 4 2: 8 3: 14
2	CLRT	0: TX FIFO is not cleared. 1: TX FIFO is cleared.
1	CLRR	0: RX FIFO is not cleared. 1: RX FIFO is cleared.
0	FIFOE	Enables FIFO This bit must be set to 1 for any of other bits in the registers to have any effect. 0: RX and TX FIFOs are not enabled. 1: RX and TX FIFOs are enabled.

3.8 UART2

3.8.1 General Description

The baseband chipset houses two UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external dev ices.

The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths **from 5 to 8 bits, an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. 8 modem control lines and a diagnostic loop-back mode are provided. UART also includes two DMA handshake lines, indicating when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.

Note that UART is designed so that all internal operation is synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and industry standard 16550A device, which means that the core is not clock for clock identical to the original device.

After a hardware reset, UART will be in M16C450 mode. Its FIFOs can then be enabled and UART can enter M16550A mode. UART has further functions beyond the M16550A mode. Each of the extended functions can be selected individually under software control.

UART provides more powerful enhancements than the industry-standard 16550:

- **Hardware flow control.** This feature is very useful when the ISR latency is hard to predict and control in the embedded applications. The MCU is relieved of having to fetch the received data within a fixed amount of time.

Note that in order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The enhanced mode bit ensures that UART is backward compatible with the software that has been written for 16C450 and 16550A devices. **Figure 35** is the block diagram of the UART2 device.

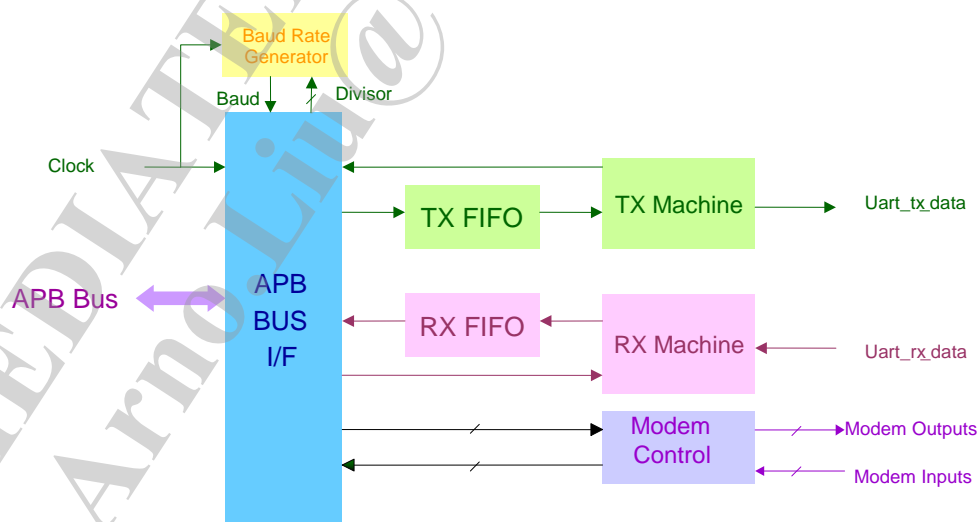


Figure 36. Block Diagram of UART2

3.8.2 Register Definition

Module name: UART2 Base address: (+A0090000h)

Address	Name	Width	Register Function
A0090000	<u>RBR</u>	8	RX Buffer Register Note: Only when LCR[7] = 0.
A0090000	<u>THR</u>	8	TX Holding Register Note: Only when LCR[7] = 0
A0090000	<u>DLL</u>	8	Divisor Latch (LS) Divides the bclk frequency Note: Modified when LCR[7]!=0
A0090004	<u>IER</u>	8	Interrupt Enable Register Note: Only when LCR[7] = 0. By storing 1 to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt will be disabled. IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.
A0090004	<u>DLM</u>	8	Divisor Latch (MS) Used to divide the bclk frequency. *NOTE: modified when LCR[7]!=0
A0090008	<u>IIR</u>	8	Interrupt Identification Register Note: Only when LCR!=BF'h. Priority is from high to low as the following. IIR[5:0]=0X1: No interrupt pending IIR[5:0]=0X6: Line status interrupt (under IER[2]=1) IIR[5:0]=0Xc: RX data time-out interrupt (under IER[0]=1) IIR[5:0]=0X4: RX data are placed in the RX bBuffer register or the RX trigger level is reached. (under IER[0]=1). IIR[5:0]=0X2: TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level (under IER[1]=1). IIR[5:0]=0X10: XOFF character received (under IER[5]=1, EFR[4] = 1).
A0090008	<u>FCR</u>	8	FIFO Control Register FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh
A0090008	<u>EFR</u>	8	Enhanced Feature Register Note: Only when LCR=BF'h
A009000C	<u>LCR</u>	8	Line Control Register Determines characteristics of serial communication signals.
A0090010	<u>MCR</u>	8	Modem Control Register Controls interface signals of the UART. MCR[5:0] are modified when LCR != 8'hBF, MCR[7] can be read when LCR != 8'hBF & EFR[4] = 1.
A0090010	<u>XON1</u>	8	XON1 Char Register Note: XON1 is modified only when LCR=BF'h.
A0090014	<u>LSR</u>	8	Line Status Register Modified when LCR != BFh.

Address	Name	Width	Register Function
A0090018	<u>XOFF1</u>	8	XOFF1 Char Register *Note: XOFF1 is modified only when LCR=BF'h.
A009001C	<u>SCR</u>	8	Scratch Register A general purpose read/write register. After reset, its value will be un-defined. Modified when LCR != BFh.
A0090020	<u>AUTOBAUD EN</u>	8	Auto Baud Detect Enable Register
A0090024	<u>HIGHSPEED</u>	8	High Speed Mode Register
A0090028	<u>SAMPLE_COUNT</u>	8	Sample Counter Register When HIGHSPEED=3, sample_count will be the threshold value for UART sample counter (sample_num). Count from 0 to sample_count.
A009002C	<u>SAMPLE_POINT</u>	8	Sample Point Register When HIGHSPEED=3, UART gets the input data when sample_count=sample_num, e.g. system clock = 13MHz, 921600 = 13000000/14. Therefore, sample_count = 13, and sample point = 6 (sampling the central point to decrease the inaccuracy) SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.
A0090030	<u>AUTOBAUD REG</u>	8	Auto Baud Monitor Register the autobaud detection state ,it will not change until enable the autobaud_en again.
A0090034	<u>RATEFIX AD</u>	8	Clock Rate Fix Register
A0090038	<u>AUTOBAUDSAMPL E</u>	8	Auto Baud Sample Register Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13. When system clock = 52MHz, autobaudsample = 27.
A009003C	<u>GUARD</u>	8	Guard Time Added Register
A0090040	<u>ESCAPE DAT</u>	8	Escape Character Register
A0090044	<u>ESCAPE EN</u>	8	Escape Enable Register
A0090048	<u>SLEEP EN</u>	8	Sleep Enable Register
A009004C	<u>DMA EN</u>	8	DMA Enable Register
A0090050	<u>RXTRI AD</u>	8	Rx Trigger Address
A0090054	<u>FRACDIV L</u>	8	Fractional Divider LSB Address
A0090058	<u>FRACDIV M</u>	8	Fractional Divider MSB Address
A009005C	<u>FCR RD</u>	8	FIFO Control Register

A0090000 RBR **RX Buffer Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name									RBR									
Type									RU									
Reset									0	0	0	0	0	0	0	0		

Bit(s)	Name	Description
--------	------	-------------

Bit(s)	Name	Description
7:0	RBR	Read-only register The received data can be read by accessing this register. Only when LCR[7] = 0.

A0090000 **THR** **TX Holding Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name									THR								
Type									WO								
Reset									0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	THR	TX Holding Register Write-only register. The data to be transmitted are written to this register and sent to the PC via serial communication. Only when LCR[7] = 0.

A0090000 **DLL** **Divisor Latch (LS)** **01**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLL							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	DLL	Divisor latch low 8-bit data <i>Note: Modified when LCR[7] != 0.</i>

A0090004 **IER** **Interrupt Enable Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name									CTSI	RTSI	XOFF1					EDSSI	ELSI	ETBEI	ERBFI
Type									RW	RW	RW					RW	RW	RW	RW
Reset									0	0	0					0	0	0	0

Bit(s)	Name	Description
7	CTSI	Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line. <i>Note: This interrupt is only enabled when hardware flow control is enabled.</i> 0: Mask an interrupt that is generated when a rising edge is detected on the CTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the CTS modem control line.
6	RTSI	Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line. <i>Note: This interrupt is only enabled when hardware flow control is enabled.</i> 0: Mask an interrupt that is generated when a rising edge is detected on the RTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the

Bit(s)	Name	Description
5	XOFFI	<p>RTS modem control line.</p> <p>Masks an interrupt that is generated when an XOFF character is received.</p> <p><i>Note: This interrupt is only enabled when software flow control is enabled.</i> 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.</p>
3	EDSSI	<p>When set to 1, an interrupt will be generated if DCTS (MSR[0]) becomes set.</p> <p>0: No interrupt is generated if DCTS (MSR[0]) becomes set. 1: An interrupt is generated if DCTS (MSR[0]) becomes set.</p>
2	ELSI	<p>When set to 1, an interrupt will be generated if BI, FE, PE or OE (LSR[4:1]) becomes set.</p> <p>0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.</p>
1	ETBEI	<p>When set to 1, an interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.</p> <p>0: No interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level. 1: An interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.</p>
0	ERBFI	<p>When set to 1, an interrupt will be generated if RX data are placed in RX buffer register or the RX trigger level is reached.</p> <p>0: No interrupt will be generated if RX data are placed in the RX buffer register or the RX trigger level is reached. 1: An interrupt will be generated if RX Data are placed in the RX buffer register or the RX trigger level is reached.</p>

A0090004 **DLM** **Divisor Latch (MS)** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLM							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	DLM	<p>Divisor latch high 8-bit data</p> <p><i>Note: Modified when LCR[7]≠0. DLL & DLM can only be updated when DLAB(LCR[7]) is set to 1. Division by 1 will generate a BAUD signal that is constantly high. DLL & DLM setting formula is {DLH,DLL}=(system clock frequency/ baud_pulse/ baud_rate).</i></p> <p>When RATE_FIX(RATEFIX_AD[0])=0, system clock frequency = 52MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=0, system clock frequency = 26MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=1, system clock frequency = 13MHz.</p> <p>For baud_pulse value, refer to HIGH_SPEED(offset=24H) register</p> <p>For example, when at 52MHz, default speed mode and 115200 baud rate, {DLH,DLL}=52MHz/16/115200=28.</p>

A0090008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE		ID					
Type									RO		RU					
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
--------	------	-------------

7:6 FIFOE
5:0 ID

IIR[5:0] Priority level interrupt source

- 000001 - No interrupt pending
- 000110 1 Line status interrupt: BI, FE, PE or OE set in LSR. (Under IER[2]=1)
- 001100 2 RX data time-out: Time-out on character in RX FIFO. (Under IER[0]=1)
- 000100 3 RX data received: RX data received or RX trigger level reached. (Under IER[0]=1)
- 000010 4 TX holding register empty: Modem status change: DCTS set in MSR. (Under IER[3]=1)
- 000000 5 TX Holding Register empty or TX FIFO trigger level reached. (Under IER[1]=1)
- 010000 6 Software flow control: XOFF Character received. (Under IER[5]=1)
- 100000 7 Hardware flow control: CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1)

Line status interrupt: A RX line status interrupt (IIR[5:0] = 000110b) will be generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the line status register.

RX data time-out interrupt: When the virtual FIFO mode is disabled, RX data time-out interrupt will be generated if all of the following conditions are applied:
 1. FIFO contains at least one character.
 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits);
 3. The most recent CPU read of the FIFO is longer than four character periods ago.

The timeout timer is restarted upon receipt of a new byte from the RX shift register or upon a CPU read from the RX FIFO. The RX data time-out interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading RX FIFO.

When the virtual FIFO mode is enabled, RX data time-out interrupt will be generated if all of the following conditions are applied:
 1. FIFO is empty.
 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits).
 3. The most recent CPU read of the FIFO is longer than four character periods ago.

The timeout timer is restarted upon receipt of a new byte from the RX shift register or reading DMA_EN register. The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading DMA_EN register.

RX data received interrupt: A RX received interrupt (IER[5:0] = 000100b) is generated if EFRBI (IER[0]) is set and either RX data are placed in the RX buffer register or the RX trigger level is reached. The interrupt is cleared by

Bit(s)	Name	Description
		reading the RX buffer register or the RX FIFO (if enabled).
		TX holding register empty interrupt: A TX holding register empty interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX holding register is empty or the contents of the TX FIFO are reduced to its trigger level. The interrupt is cleared by writing to the TX holding register or TX FIFO if FIFO is enabled.
		Modem status change interrupt: A modem status change Interrupt (IIR[5:0] = 000000b) will be generated if EDSSI (IER[3]) is set and e DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the modem status register.
		Software flow control interrupt: A software flow control interrupt (IIR[5:0] = 010000b) will be generated if the software flow control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the interrupt identification register.
		Hardware flow control interrupt: A hardware flow control interrupt (IER[5:0] = 100000b) will be generated if the hardware flow control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS modem control line. The interrupt is cleared by reading the interrupt identification register.

A0090008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL	TFTL1_TFTL				CLRT	CLRR	FIFOE
Type									0	0				WO	WO	WO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	RX FIFO trigger threshold RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	TX FIFO trigger threshold TX FIFO contains total 32 bytes. 0: 1 1: 4 2: 8 3: 14
2	CLRT	Control bit to clear TX FIFO 0: No effect 1: Clear TX FIFO
1	CLRR	Control bit to clear RX FIFO 0: No effect 1: Clear RX FIFO
0	FIFOE	Enables FIFO This bit must be set to 1 for any of other bits in the registers to have any effect. 0: Disable both RX and TX FIFOs. 1: Enable both RX and TX FIFOs.

A0090008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTO_CTS	AUTO_RTS		ENABLE_E	SW_FLOW_CONT			
Type									RW	RW		RW	RW			
Reset									0	0		0	0	0	0	0

Bit(s)	Name	Description
7	AUTO_CTS	Enables hardware transmission flow control 0: Disable 1: Enable
6	AUTO_RTS	Enables hardware reception flow control 0: Disable 1: Enable
4	ENABLE_E	Enables enhancement feature 0: Disable 1: Enable
3:0	SW_FLOW_CONT	Software flow control bits 00xx: No TX flow control bytes 01xx: No TX flow control bytes 10xx: Transmit XON1/XOFF1 as flow control bytes xx00: No RX flow control xx01: No RX flow control xx10: Receive XON1/XOFF1 as flow control bytes

A009000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1_WLS0	
Type									RW	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	DLAB	Divisor latch access bit 0: RX and TX registers are read/written at Address 0 and the IER register is read/written at Address 4. 1: Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.
6	SB	Set break 0: No effect 1: SOUT signal is forced to the 0 state.
5	SP	Stick parity 0: No effect. 1: The parity bit is forced to a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the parity bit will be set and checked = 0. If EPS=0 & PEN=1, the parity bit will be set and checked = 1.
4	EPS	Selects even parity 0: When EPS=0, an odd number of ones is sent and checked. 1: When EPS=1, an even number of ones is sent and checked.

Bit(s)	Name	Description
3	PEN	Enables parity 0: The parity is neither transmitted nor checked. 1: The parity is transmitted and checked.
2	STB	Number of STOP bits 0: One STOP bit is always added. 1: Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.
1:0	WLS1_WLS0	Selects word length 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits

A0090010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF_STAT US			Loop			RTS	
Type									RU			RW			RW	
Reset									0			0			0	

Bit(s)	Name	Description
7	XOFF_STATUS	Read-only bit 0: When an XON character is received. 1: When an XOFF character is received.
4	Loop	Loop-back control bit 0: No loop-back is enabled. 1: Loop-back mode is enabled.
1	RTS	Controls the state of the output NRTS, even in loop mode. 0: RTS will always output 1. 1: RTS's output will be controlled by flow control condition.

A0090010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XON1	XON1 character for software flow control Modified only when LCR=BF'h.

A0090014 LSR Line Status Register 60

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE RR	TEMT	THRE	BI	FE	PE	OE	DR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XOFF1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XOFF1	XOFF1 character for software flow control Modified only when LCR=BF'h.

A009001C SCR **Scratch Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCR															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SCR	General purpose read/write register After reset, its value will be undefined. Modified when LCR != BFh.

A0090020 AUTOBAUD_EN **Auto Baud Detect Enable Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														SLEEP_ACK_SEL	AUTOBAUD_SEL	AUTOBAUD_EN
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	SLEEP_ACK_SEL	Selects sleep ack when autobaud_en 0: Support sleep_ack when autobaud_en is opened . 1: Does not support sleep_ack when autobaud_en is opened .
1	AUTOBAUD_SEL	Selects auto-baud 0: Support standard baud rate detection 1: Support non_standard baud rate detection (support baud from 110 to 115200; recommended to use 52MHz to auto fix) .
0	AUTOBAUD_EN	Auto-baud enabling signal 0: Disable auto-baud function 1: Enable auto-baud function (UARTn+0024h SPEED should be set to 0.) <i>Note: When AUTOBAUD_EN is active, there should not be A*/a* char before the auto baud char AT/at. If A*/a* is Inevitable, autobaud will fail and please disable AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.</i>

A0090024 HIGHSPEED **High Speed Mode Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name																SPEED	
Type																RW	
Reset																0	0

Bit(s)	Name	Description
1:0	SPEED	UART sample counter base 0: Based on 16*baud_pulse, baud_rate = system clock frequency/16/{DLH, DLL} 1: Based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL} 2: Based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL} 3: Based on sampe_count * baud_pulse, baud_rate = system clock frequency / (sampe_count+1){DLM, DLL}

A0090028 **SAMPLE_COUN** **Sample Counter Register** **00**
I

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLECOUNT							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SAMPLECOUNT	Only useful when HIGHSPEED mode = 3.

A009002C **SAMPLE_POINT** **Sample Point Register** **FF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLEPOINT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	SAMPLEPOINT	SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal. Effective only when HIGHSPEED=3.

A0090030 **AUTOBAUD_RE** **Auto Baud Monitor Register** **00**
G

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BAUD_STAT				BAUD_RATE			
Type									RU				RU			
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:4	BAUD_STAT	Autobaud state (only true value in standard autobaud detection) 0: Autobaud is detecting. 1: AT_7N1 2: AT_7O1 3: AT_7E1 4: AT_8N1 5: AT_8O1 6: AT_8E1 7: at_7N1

Bit(s)	Name	Description
		8: at_7E1
		9: at_7O1
		10: at_8N1
		11: at_8E1
		12: at_8O1
		13: Autobaud detection fails
3:0	BAUD_RATE	Autobaud baud rate (only true value in standard autobaud detection)
		0: 115,200
		1: 57,600
		2: 38,400
		3: 19,200
		4: 9,600
		5: 4,800
		6: 2,400
		7: 1,200
		8: 300
		9: 110

A0090034 RATEFIX_AD Clock Rate Fix Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														FREQ_SEL	AUTO BAUD RATE	
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	FREQ_SEL	0: Select 26MHz as system clock 1: Select 13MHz as system clock
1	AUTOBAUD_RATE_FIX	0: Use 52MHz as system clock for UART auto baud detection 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART auto baud detection
0	RATE_FIX	0: Use 52MHz as system clock for UART TX/RX 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART TX/RX

A0090038 AUTOBAUDSAMPLE Auto Baud Sample Register 0D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											AUTOBAUDSAMPLE					
Type											RW					
Reset											0	0	1	1	0	1

Bit(s)	Name	Description
5:0	AUTOBAUDSAMPLE	clk division for autobaud rate detection
		For standard baud rate detection.
		System clk 52m: 'd 27
		System clk 26m: 'd 13
		System clk 13m: 'd 6

Bit(s)	Name	Description
		For non-standard baud rate detection.

A009003C **GUARD** **Guard Time Added Register** **0F**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT			
Type												RW	RW			
Reset												0	1	1	1	1

Bit(s)	Name	Description
4	GUARD_EN	Guard interval add enabling signal 0: No guard interval added 1: Add guard interval after stop bit.
3:0	GUARD_CNT	Guard interval count value Guard interval = $[1/(\text{system clock}/\text{div_step}/\text{div})] * \text{GUARD_CNT}$.

A0090040 **ESCAPE_DAT** **Escape Character Register** **FF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ESCAPE_DAT				
Type												RW				
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	ESCAPE_DAT	Escape character added before software flow control data and escape character If TX data are xon (31h), with esc_en =1, UART will transmit data as esc + CEh (~xon).

A0090044 **ESCAPE_EN** **Escape Enable Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	ESC_EN	Adds escape character in transmitter and removes escape character in receiver by UART 0: Does not deal with the escape character 1: Add escape character in transmitter and remove escape character in receiver

A0090048 SLEEP_EN Sleep Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name																	SLEEP_EN
Type																	RW
Reset																	0

Bit(s)	Name	Description
0	SLEEP_EN	<p>For sleep mode issue</p> <p>0: Does not deal with sleep mode indicate signal 1: Activate hardware flow control or software control according to software initial setting when the chip enters sleep mode. Release hardware flow when the chip wakes up. However, for software control, UART sends xon when awoken and when FIFO does not reach threshold level.</p>

A009004C DMA_EN DMA Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FIFO_LSR_sel	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Name	Description
3	FIFO_lsr_sel	<p>Selects FIFO LSR mode</p> <p>0: LSR will hold the first line status error state until you read the LSR register. 1: LSR will update automatically.</p>
2	TO_CNT_AUTORST	<p>Time-out counter auto reset register</p> <p>0: After RX time-out happens, SW shall reset the interrupt by reading UART 0x4C. 1: The time-out counter will be auto reset. Set this register when Rain's new DMA is used.</p>
1	TX_DMA_EN	<p>TX_DMA mechanism enabling signal</p> <p>0: Does not use DMA in TX 1: Use DMA in TX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt for DMA.for DMA.</p>
0	RX_DMA_EN	<p>RX_DMA mechanism enabling signal</p> <p>0: Does not use DMA in RX 1: Use DMA in RX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt</p>

A0090050 RXTRI_AD Rx Trigger Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name																	RXTRIG
Type																	RW
Reset																	0

Bit(s)	Name	Description
--------	------	-------------

Bit(s)	Name	Description
3:0	RXTRIG	When {rtm,rtl}=2'b11, the RX FIFO threshold will be Rxtrig. The value is suggested to be smaller than half of RX FIFO size, which is 32 bytes.

A0090054 **FRACDIV_L** **Fractional Divider LSB Address** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FRACDIV_L	Adds sampling count (+1) from state data7 to data0 to contribute fractional divisor.only when high_speed=3.

A0090058 **FRACDIV_M** **Fractional Divider MSB Address** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset															0	0

Bit(s)	Name	Description
1:0	FRACDIV_M	Adds sampling count when in state stop to parity to contribute fractional divisor.only when high_speed=3.

A009005C **FCR_RD** **FIFO Control Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	RX FIFO trigger threshold RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	TX FIFO trigger threshold TX FIFO contains total 32 bytes. 0: 1 1: 4 2: 8 3: 14
2	CLRT	0: TX FIFO is not cleared. 1: TX FIFO is cleared.

Bit(s)	Name	Description
1	CLRR	0: RX FIFO is not cleared. 1: RX FIFO is cleared.
0	FIFOE	Enables FIFO This bit must be set to 1 for any of other bits in the registers to have any effect. 0: RX and TX FIFOs are not enabled. 1: RX and TX FIFOs are enabled.

3.9 UART3

3.9.1 General Description

The baseband chipset houses two UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external devices.

The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths **from 5 to 8 bits, an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided. The UART also includes two DMA handshake lines, used to indicate when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.

Note: The UART has been designed so that all internal operations are synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and the industry standard 16550A device, which means that the core is not clock for clock identical to the original device.

After a hardware reset, the UART is in M16C450 mode. Its FIFOs can be enabled and the UART can then enter M16550A mode. The UART adds further functionality beyond M16550A mode. Each of the extended functions can be selected individually under software control.

The UART provides more powerful enhancements than the industry-standard 16550:

Note: In order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The Enhanced Mode bit ensures that the UART is backward compatible with software that has been written for 16C450 and 16550A devices.

Figure 35 shows the block diagram of the UART3 device.

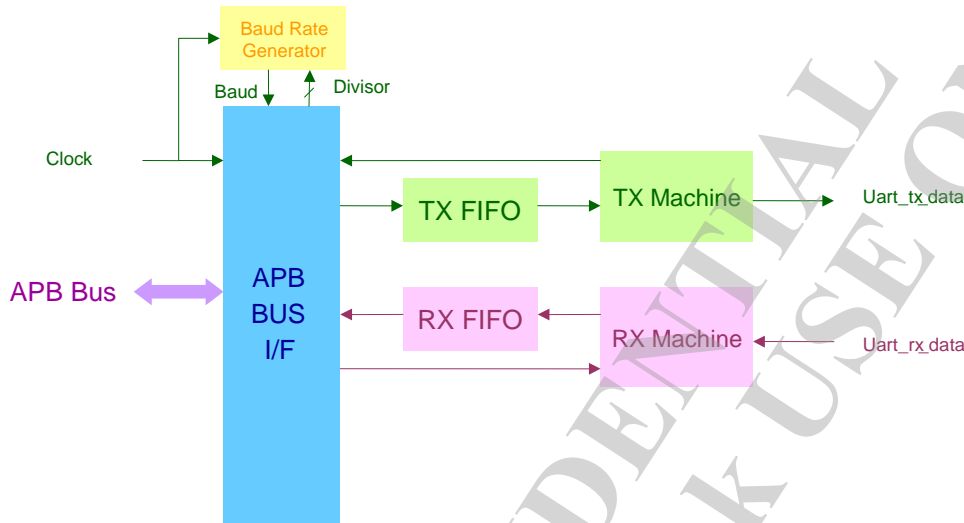


Figure 37. Block Diagram of UART3

3.9.2 Register Definitions

Module name: UART3 Base address: (+A00A0000h)

Address	Name	Width	Register Function
A00A0000	RBR	8	RX Buffer Register *NOTE: onnly when LCR[7] = 0.
A00A0000	THR	8	TX Holding Register *NOTE: only when LCR[7] = 0.
A00A0000	DLL	8	Divisor Latch (LS) used to divid the bclk frequency . *NOTE: modified when LCR[7] != 0
A00A0004	IER	8	Interrupt Enable Register *NOTE: onnly when LCR[7] = 0. By storing a '1' to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt is disabled. IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.
A00A0004	DLM	8	Divisor Latch (MS) used to divid the bclk frequency . *NOTE: modified when LCR[7] != 0
A00A0008	IIR	8	Interrupt Identification Register *NOTE: only when LCR != BF'h. priority is from high to low as following . IIR[5:0] == 0X1: no interrupt pending . IIR[5:0] == 0X6: line status interrup(Under IER[2]=1). IIR[5:0] == 0Xc: rx data timeout interrump(Under IER[0]=1). IIR[5:0] == 0X4: RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached.(Under IER[0]=1) . IIR[5:0] == 0X2: TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level(Under IER[1]=1). IIR[5:0] == 0X10: XOFF character recieved (Under IER[5]=1, EFR[4] = 1).
A00A0008	FCR	8	FIFO Control Register FCR is used to control the trigger levels of the FIFOs, or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh
A00A0008	EFR	8	Enhanced Feature Register *NOTE: Only when LCR=BF'h
A00A000C	LCR	8	Line Control Register Line Control Register. Determines characteristics of serial communication signals.

A00A0010	<u>MCR</u>	8	Modem Control Register Modem Control Register. Control interface signals of the UART. MCR[5:0] are modified when LCR != 8'hBF, MCR[7] can be read when LCR != 8'hBF & EFR[4] = 1.
A00A0010	<u>XON1</u>	8	XON1 Char Register *Note: XON1 modified only when LCR=BF'h.
A00A0014	<u>LSR</u>	8	Line Status Register Line Status Register. Modified when LCR != BFh.
A00A0018	<u>XOFF1</u>	8	XOFF1 Char Register *Note: , XOFF1 modified only when LCR=BF'h.
A00A001C	<u>SCR</u>	8	Scratch Register A general purpose read/write register. After reset, its value is un-defined. Modified when LCR != BFh.
A00A0020	<u>AUTOBAUD_EN</u>	8	Auto Baud Detect Enable Register
A00A0024	<u>HIGHSPEED</u>	8	High Speed Mode Register
A00A0028	<u>SAMPLE_COUNT</u>	8	Sample Counter Register When HIGHSPEED=3, the sample_count is the threshold value for UART sample counter (sample_num). Count from 0 to sample_count.
A00A002C	<u>SAMPLE_POINT</u>	8	Sample Point Register When HIGHSPEED=3, UART gets the input data when sample_count=sample_num. e.g. system clock = 13MHz, 921600 = 13000000 / 14 sample_count = 13 and sample point = 6 (sample the central point to decrease the inaccuracy) The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 and remove the decimal.
A00A0030	<u>AUTOBAUD_REG</u>	8	Auto Baud Monitor Register the autobaud detection state ,it will not change until enable the autobaud_en again.
A00A0034	<u>RATEFIX_AD</u>	8	Clock Rate Fix Register
A00A0038	<u>AUTOBAUDSAMPLE</u>	8	Auto Baud Sample Register Since the system clock may change, autobaud sample duration should change as system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13;When system clock = 52MHz, autobaudsample = 27.
A00A003C	<u>GUARD</u>	8	Guard time added register
A00A0040	<u>ESCAPE_DAT</u>	8	Escape character register
A00A0044	<u>ESCAPE_EN</u>	8	Escape enable register
A00A0048	<u>SLEEP_EN</u>	8	Sleep enable register
A00A004C	<u>DMA_EN</u>	8	DMA enable register
A00A0050	<u>RXTRI_AD</u>	8	Rx Trigger Address
A00A0054	<u>FRACDIV_L</u>	8	Fractional Divider LSB Address
A00A0058	<u>FRACDIV_M</u>	8	Fractional Divider MSB Address
A00A005C	<u>FCR_RD</u>	8	FIFO Control Register

A00A0000 RBR RX Buffer Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	RBR	Read-only register. The received data can be read by accessing this register. Only when LCR[7] = 0.

A00A0000 **THR** **TX Holding Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									THR							
Type									WO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	THR	TX Holding Register. Write-only register. The data to be transmitted is written to this register, and then sent to the PC via serial communication. only when LCR[7] = 0.

A00A0000 **DLL** **Divisor Latch (LS)** **01**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLL							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	DLL	divisor Latch low 8bit data. *NOTE: modified when LCR[7]!=0

A00A0004 **IER** **Interrupt Enable Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											XOFFI			ELSI	ETBEI	ERBFI
Type											RW			RW	RW	RW
Reset											0			0	0	0

Bit(s)	Name	Description
5	XOFFI	Masks an interrupt that is generated when an XOFF character is received. Note: This interrupt is only enabled when software flow control is enabled. 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.
2	ELSI	When set ("1"), an interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
1	ETBEI	When set ("1"), an interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level. 0: No interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level. 1: An interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level.

0 ERBFI

When set ("1"), an interrupt is generated if RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached.

0: No interrupt is generated if RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached.
1: An interrupt is generated if RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached.

A00A0004 DLM Divisor Latch (MS) 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DLM															
Type	RW															
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit(s)	Name	Description
7:0	DLM	<p>divisor Latch high 8bit data..</p> <p>*NOTE: modified when LCR[7]!=0 Note: DLL & DLM can only be updated when DLAB(LCR[7]) is set to 1, Note to that division by 1 generates a BAUD signal that is constantly high. Note: DLL & DLM setting formula is {DLH,DLL}=(system clock frequency/baud_pulse/baud_rate). When RATE_FIX(RATEFIX_AD[0])=0, system clock frequency = 52MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=0, system clock frequency = 26MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=1, system clock frequency = 13MHz. For baud_pulse value refer to HIGH_SPEED(offset=24H) register. e.g. When 52MHz,default speed mode and 115200 baud rate, the {DLH,DLL}=52MHz/16/115200=28.</p>

A00A0008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE				ID			
Type									RO				RU			
Reset									0 0 0 0 0 0 0 0				0 0 0 0 0 0 0 1			

Bit(s)	Name	Description
7:6	FIFOE	
5:0	ID	<p>IIR[5:0] Priority Level Interrupt Source</p> <p>000001 - No interrupt pending 000110 1 Line Status Interrupt: BI, FE, PE or OE set in LSR. (Under IER[2]=1) 001100 2 RX Data Timeout: Timeout on character in RX FIFO. (Under IER[0]=1) 000100 3 RX Data Received: RX Data received or RX Trigger Level reached. (Under IER[0]=1) 000010 4 TX Holding Register Empty: 000000 5 Modem Status change: DCTS set in MSR. (Under IER[3]=1) TX Holding Register empty or TX FIFO Trigger Level reached. (Under IER[1]=1) 010000 6 Software Flow Control: XOFF Character received. (Under IER[5]=1) 100000 7 Hardware Flow Control: CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1)</p>

Line Status Interrupt: A RX Line Status Interrupt (IIR[5:0] == 000110b) is generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the Line Status Register.

RX Data Timeout Interrupt: When virtual FIFO mode is disabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO contains at least one character;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register, or on a CPU read from the RX FIFO.

The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1, and is cleared by reading RX FIFO.

When virtual FIFO mode is enabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO is empty;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register or reading DMA_EN register.

The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1, and is cleared by reading DMA_EN register.

RX Data Received Interrupt: A RX Received interrupt (IER[5:0] == 000100b) is generated if EFRBI (IER[0]) is set and either RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached. The interrupt is cleared by reading the RX Buffer Register or the RX FIFO (if enabled).

TX Holding Register Empty Interrupt: A TX Holding Register Empty Interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level. The interrupt is cleared by writing to the TX Holding Register or TX FIFO if FIFO enabled.

Modem Status Change Interrupt: A Modem Status Change Interrupt (IIR[5:0] = 000000b) is generated if EDSSI (IER[3]) is set and e DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the Modem Status Register.

Software Flow Control Interrupt: A Software Flow Control Interrupt (IIR[5:0] = 010000b) is generated if Software Flow Control is enabled and XOFF1 (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the Interrupt Identification Register.

Hardware Flow Control Interrupt: A Hardware Flow Control Interrupt (IER[5:0] = 100000b) is generated if Hardware Flow Control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS Modem Control line. The interrupt is cleared by reading the Interrupt Identification Register.

A00A0008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL		TFTL1_TFTL			CLRT	CLRR	FIFOE
Type									0		0			WO	WO	WO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description	
7:6	RFTL1_RFTL0	RX FIFO trigger threshold. (RX FIFO contains total 32 bytes.)	
		0:	1
		1:	6
		2:	12
		3: RXTRIG	
5:4	TFTL1_TFTL0	TX FIFO trigger threshold (TX FIFO contains total 32 bytes.)	
		0:	1
		1:	4
		2:	8
		3: 14	

2	CLRT	control bit to clear tx fifo 0: no effect 1: clear TX FIFO
1	CLRR	control bit to clear rx fifo 0: no effect 1: clear RX FIFO
0	FIFOE	FIFO Enabled. This bit must be set to 1 for any of the other bits in the registers to have any effect. 0: Disable both the RX and TX FIFOs. 1: Enable both the RX and TX FIFOs.

A00A0008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ENABLE_E	SW_FLOW_CONT			
Type												RW	RW			
Reset												0	0	0	0	0

Bit(s)	Name	Description
4	ENABLE_E	Enables enhancement feature 0: Disabled. 1: Enabled.
3:0	SW_FLOW_CONT	Software flow control bits. 00xx: No TX Flow Control 01xx: No TX Flow Control 10xx: Transmit XON1/XOFF1 as flow control bytes xx00: No RX Flow Control xx01: No RX Flow Control xx10: Receive XON1/XOFF1 as flow control bytes

A00A000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	DLAB	Divisor Latch Access Bit. 0: The RX and TX Registers are read/written at Address 0 and the IER register is read/written at Address 4. 1: The Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.
6	SB	Set Break 0: No effect 1: SOUT signal is forced into the "0" state.
5	SP	Stick Parity 0: No effect. 1: The Parity bit is forced into a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the Parity bit is set and checked = 0. If EPS=0 & PEN=1, the Parity bit is set and checked = 1.
4	EPS	Even Parity Select

- 3 PEN
0: When EPS=0, an odd number of ones is sent and checked.
1: When EPS=1, an even number of ones is sent and checked.
Parity Enable
0: The Parity is neither transmitted nor checked.
1: The Parity is transmitted and checked.
- 2 STB
Number of STOP bits
0: One STOP bit is always added.
1: Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.
- 1:0 WLS1_WLS0
Word Length Select.
0: 5 bits
1: 6 bits
2: 7 bits
3: 8 bits

A00A0010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF_STAT_US			Loop				
Type									RU			RW				
Reset									0			0				

Bit(s)	Name	Description
7	XOFF_STATUS	This is a read-only bit. 0: When an XON character is received. 1: When an XOFF character is received.
4	Loop	Loop-back control bit. 0: No loop-back is enabled. 1: Loop-back mode is enabled.

A00A0010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XON1	XON1 character for software flow control. modified only when LCR=BF'h.

A00A0014 LSR Line Status Register 60

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE RR	TEMT	THRE	BI	FE	PE	OE	DR
Type									RU	RU	RU	RU	RU	RU	RU	RU
Reset									0	1	1	0	0	0	0	0

Bit(s)	Name	Description
--------	------	-------------

7	FIFOERR	<p>RX FIFO Error Indicator. 0: No PE, FE, BI set in the RX FIFO. 1: Set to 1 when there is at least one PE, FE or BI in the RX FIFO.</p>
6	TEMT	<p>TX Holding Register (or TX FIFO) and the TX Shift Register are empty. 0: Empty conditions below are not met. 1: If FIFOs are enabled, the bit is set whenever the TX FIFO and the TX Shift Register are empty. If FIFOs are disabled, the bit is set whenever TX Holding Register and TX Shift Register are empty.</p>
5	THRE	<p>Indicates if there is room for TX Holding Register or TX FIFO is reduced to its Trigger Level. 0: Reset whenever the contents of the TX FIFO are more than its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is not empty(FIFOs are disabled). 1: Set whenever the contents of the TX FIFO are reduced to its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is empty and ready to accept new data (FIFOs are disabled).</p>
4	BI	<p>Break Interrupt. 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit is set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). If the FIFOs are enabled, this error is associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: the next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.</p>
3	FE	<p>Framing Error. 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit. If the FIFOs are enabled, the state of this bit is revealed when the byte it refers to is the next to be read.</p>
2	PE	<p>Parity Error 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit is set if the received data did not have a valid parity bit. If the FIFOs are enabled, the state of this bit is revealed when the referred byte is the next to be read.</p>
1	OE	<p>Overrun Error. 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit is set if the RX Buffer was not read by the CPU before new data from the RX Shift Register overwrote the previous contents. If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX Shift Register becomes full. OE is set as soon as this happens. The character in the Shift Register is then overwritten, but not transferred to the FIFO.</p>
0	DR	<p>Data Ready. 0: Cleared by the CPU reading the RX Buffer or by reading all the FIFO bytes. 1: Set by the RX Buffer becoming full or by the FIFO becoming no empty.</p>

A00A0018 XOFF1 XOFF1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XOFF1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XOFF1	XOFF1 character for software flow control. modified only when LCR=BF'h.

A00A001C SCR Scratch Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Name																	SCR									
Type																		RW								
Reset																		0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SCR	A general purpose read/write register. After reset, its value is un-defined. Modified when LCR != BFh.

A00A0020 AUTOBAUD_EN Auto Baud Detect Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														SLEEP_ACK_SEL	AUTOBAUD_SEL	AUTOBAUD_EN
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	SLEEP_ACK_SEL	Sleep ack select when autobaud_en 0: support sleep_ack when autobaud_en is opened . 1: not support sleep_ack when autobaud_en is opened .
1	AUTOBAUD_SEL	Auto-baud select 0: support standard baud rate detection . 1: support non_standard baud rate detection(support baud from 110 to 115200, it is recommend to use 52MHZ to auto fix) .
0	AUTOBAUD_EN	Auto-baud enable signal 0: Auto-baud function disable 1: Auto-baud function enable (UARTn+0024h SPEED should be set 0) Note: when AUTOBAUD_EN is active, there should not A*/a* char before the auto baud char AT/at, if the A*/a* is inevitable, the autobaud will fail and please disable the AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.

A00A0024 HIGHPEED High Speed Mode Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name																SPEED	
Type																RW	
Reset																0	0

Bit(s)	Name	Description
1:0	SPEED	UART sample counter base 0: based on 16*baud_pulse, baud_rate = system clock frequency/16/{DLH, DLL} 1: based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL} 2: based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL} 3: based on sampe_count * baud_pulse, baud_rate = system clock frequency / (sampe_count+1) / {DLM, DLL}

A00A0028 SAMPLE_COUN Sample Counter Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name																SAMPLECOUNT	
Type																RW	
Reset																0	0

Bit(s)	Name	Description
7:0	SAMPLECOUNT	Just be useful when HIGHSPEED mode = 3.

A00A002C SAMPLE_POINT Sample Point Register **FF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLEPOINT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	SAMPLEPOINT	The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 and remove the decimal. sample point , is effective only When HIGHSPEED=3

A00A0030 AUTOBAUD_RE Auto Baud Monitor Register **00**
G

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BAUD_STAT				BAUD_RATE			
Type									RU				RU			
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:4	BAUD_STAT	Autobaud state(only true value in standard autobaud detection)
		0: Autobaud is detecting
		1: AT_7N1
		2: AT_7O1
		3: AT_7E1
		4: AT_8N1
		5: AT_8O1
		6: AT_8E1
		7: at_7N1
		8: at_7E1
		9: at_7O1
		10: at_8N1
		11: at_8E1
		12: at_8O1
		13: Autobaud detection fails
3:0	BAUD_RATE	Autobaud baud rate(only true value in standard autobaud detection)
		0: 115200
		1: 57600
		2: 38400
		3: 19200
		4: 9600
		5: 4800
		6: 2400
		7: 1200
		8: 300
		9: 110

A00A0034 RATEFIX_AD Clock Rate Fix Register **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														FREQ	AUTO	RATE

Bit(s)	Name	Description
7:0	ESCAPE_DAT	Escape character added before software flow control data and escape character, i.e. if tx data is xon (31h), with esc_en =1, uart transmits data as esc + CEh (~xon).

A00A0044 ESCAPE_EN Escape enable register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	ESC_EN	Add escape character in transmitter and remove escape character in receiver by UART. 0: Do not deal with the escape character. 1: Add escape character in transmitter and remove escape character in receiver.

A00A0048 SLEEP_EN Sleep enable register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	SLEEP_EN	For sleep mode issue 0: Do not deal with sleep mode indicate signal 1: To activate hardware flow control or software control according to software initial setting when chip enters sleep mode. Releasing hardware flow when chip wakes up; but for software control, uart sends xon when awoken and when FIFO does not reach threshold level.

A00A004C DMA_EN DMA enable register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FIFO_Isr_sel	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Name	Description
3	FIFO_Isr_sel	fifo lsr mode selection 0: lsr will hold the first line status error state until you read the lsr register. 1: lsr will update automatically.
2	TO_CNT_AUTORST	Timeout counter auto reset register 0: After RX timeout happen, SW shall reset the interrupt by reading UART 0x4C. 1: The timeout counter will be auto reset. Set this register when Rain's new DMA is used.

- 1 TX_DMA_EN **TX_DMA mechanism enable signal**
0: Do not use DMA in TX.
1: Use DMA in TX. When this register is enabled, the flow control is based on the DMA threshold, and generates a timeout interrupt for DMA.
- 0 RX_DMA_EN **RX_DMA mechanism enable signal**
0: Do not use DMA in RX.
1: Use DMA in RX. When this register is enabled, the flow control is based on the DMA threshold, and generates a timeout interrupt

A00A0050 RXTRI_AD Rx Trigger Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name														RXTRIG			
Type														RW			
Reset														0	0	0	0

Bit(s)	Name	Description
3:0	RXTRIG	When {rtm,rtl}=2'b11, The Rx FIFO threshold will be Rxtrig. The value is suggested to be less than half of RX FIFO size, which is 32 Bytes.

A00A0054 FRACDIV_L Fractional Divider LSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name														FRACDIV_L							
Type														RW							
Reset														0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FRACDIV_L	Add sampling count (+1) from state data7 to data0, in order to contribute fractional divisor.only when high_speed==3.

A00A0058 FRACDIV_M Fractional Divider MSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														FRACDIV_M		
Type														RW		
Reset														0	0	

Bit(s)	Name	Description
1:0	FRACDIV_M	Add sampling count when in state stop to parity, in order to contribute fractional divisor.only when high_speed==3.

A00A005C FCR_RD FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								RFTL1_RFTL		TFTL1_TFTL		CLRT		CLRR	FIFOE	
Type								RO		RO		RO		RO	RO	
Reset								0	0	0	0	0	0	0	0	

Bit(s)	Name	Description	
7:6	RFTL1_RFTL0	RX FIFO trigger threshold. (RX FIFO contains total 32 bytes.) 0: 1: 2: 3: RXTRIG	1 6 12
5:4	TFTL1_TFTL0	TX FIFO trigger threshold (TX FIFO contains total 32 bytes.) 0: 1: 2: 3: 14	1 4 8
2	CLRT	0: TX FIFO is not cleared 1: TX FIFO is cleared	
1	CLRR	0: RX FIFO is not cleared 1: RX FIFOs are cleared	
0	FIFOE	FIFO Enabled. This bit must be set to 1 for any of the other bits in the registers to have any effect. 0: the RX and TX FIFOs are not enabled.. 1: RX and TX FIFOs are enabled.	

3.10 I2C/SCCB Controller

3.10.1 General Description

I2C (Inter-IC)/SCCB (Serial Camera Control Bus) is a two-wire serial interface. The two signals are SCL and SDA. SCL is a clock signal that is driven by the master. SDA is a bi-directional data signal that can be driven by either the master or the slave. This generic controller supports the master role and conforms to the I2C specification.

3.10.1.1 Feature

- I2C compliant master mode operation
- Adjustable clock speed for LS/FS mode operation
- Supports 7-bit/10-bit addressing
- Supports high-speed mode
- Supports slave clock extension
- START/STOP/REPEATED START condition
- Manual transfer mode
- Multi-write per transfer (up to 8 data bytes for non-DMA mode)
- Multi-read per transfer (up to 8 data bytes for non-DMA mode)
- Multi-transfer per transaction
- Combined format transfer with length change capability
- Active drive/wired-and I/O configuration

3.10.1.2 Manual Transfer Mode

The controller offers one transfer mode, the manual mode.

When the manual mode is selected, in addition to the slave address register, the controller has a built-in 8byte deep FIFO which allows MCU to prepare up to 8 bytes of data for a write transfer, or read up to 8 bytes of data for a read transfer.


3.10.1.3 Transfer Format Support

This controller has been designed to be as generic as possible in order to support a wide range of devices that may utilize different combinations of transfer formats. Here are the transfer format types that can be supported through different software configuration:

Wording convention note

- Transfer = Anything encapsulated within a Start and Stop or Repeated Start.
- Transfer length = Number of bytes within the transfer
- Transaction = This is the top unit. Everything combined equals 1 transaction.
- Transaction length = Number of transfers to be conducted.

 Master to slave dir

 Slave to master dir

Single-byte access

Single Byte Write

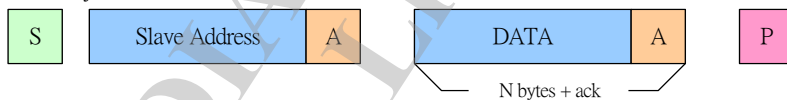


Single Byte Read

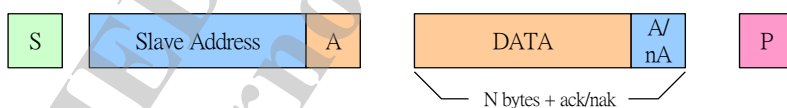


Multi-byte access

Multi Byte Write

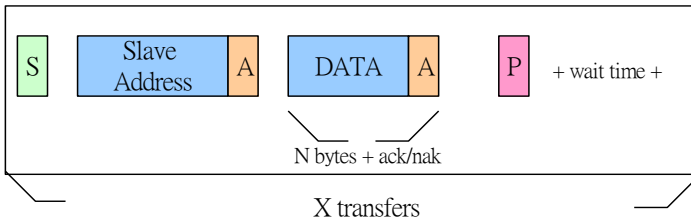


Multi Byte Read

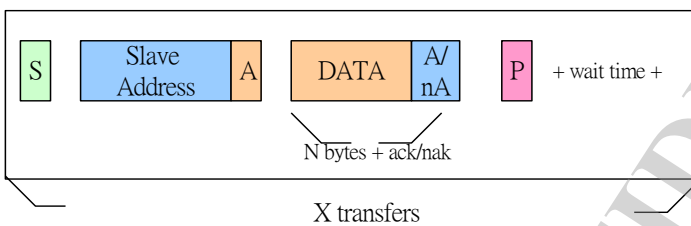


Multi-byte transfer + multi-transfer (same direction)

Multi Byte Write + Multi Transfer

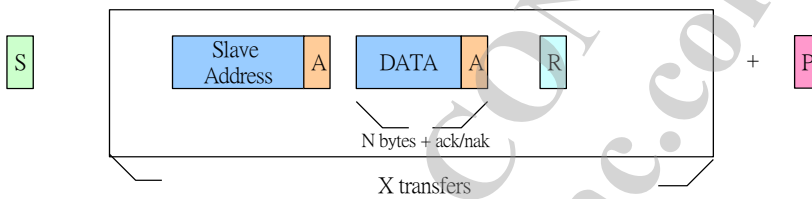


Multi Byte Read + Multi Transfer

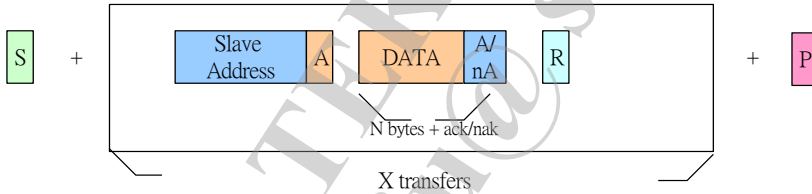


Multi-byte transfer + multi-transfer w RS (same direction)

Multi Byte Write + Multi Transfer + Repeated Start



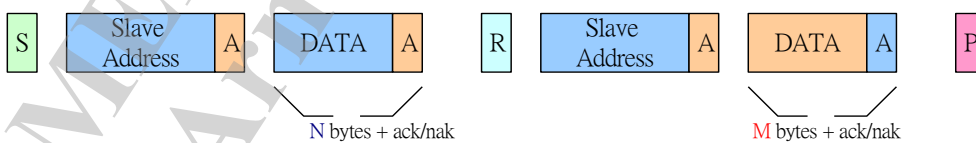
Multi Byte Read + Multi Transfer + Repeated Start



Combined write/read with Repeated Start (direction change)

Note: Only supports write and then read sequence. Read and then write is not supported.

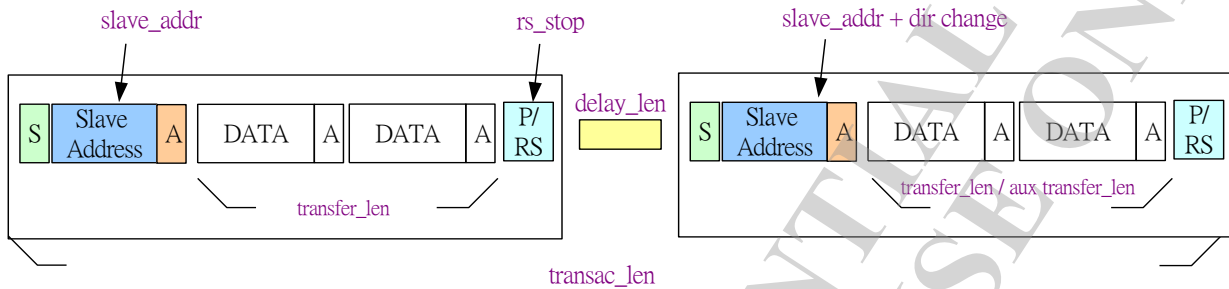
Combined Multi Byte Write + Multi Byte Read



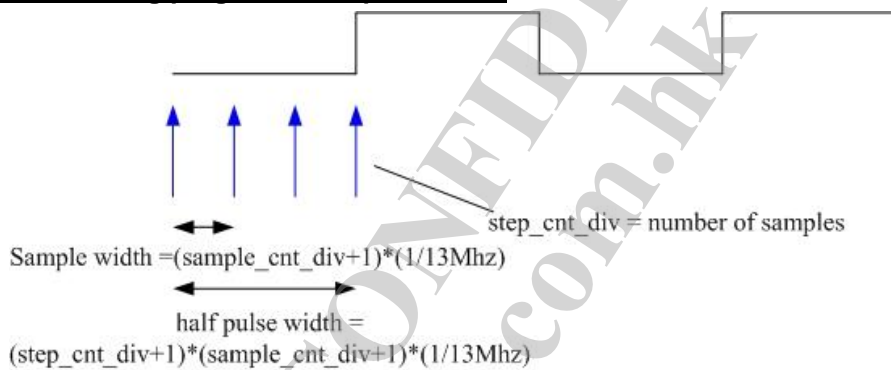
3.10.2 Programming Examples

Common transfer programmable parameters

Programmable Parameters



Output waveform timing programmable parameters



3.10.3 Register Definition

Module name: I2C_SCCB_Controller base address: (+A0120000h)

Address	Name	Width	Register function
A0120000	DATA_PORT	16	Data port register
A0120004	SLAVE_ADDR	16	Slave address register
A0120008	INTR_MASK	16	<p>Interrupt mask register</p> <p>This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register.</p> <p>1 = Allow interrupt 0 = Disable interrupt</p> <p><i>Note: While disabled, the corresponding interrupt will not be asserted, however intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.</i></p>
A012000C	INTR_STAT	16	<p>Interrupt status register</p> <p>When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.</p>
A0120010	CONTROL	16	Control register

Address	Name	Width	Register function
A0120014	TRANSFER_LEN	16	Transfer length register (number of bytes per transfer)
A0120018	TRANSAC_LEN	16	Transaction length register (number of transfers per transaction)
A012001C	DELAY_LEN	16	Inter delay length register
A0120020	TIMING	16	Timing control register LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to $(step_cnt_div+1)*(sample_cnt_div + 1)/13MHz$
A0120024	START	16	Start register
A0120030	FIFO_STAT	16	FIFO status register
A0120038	FIFO_ADDR_CLR	16	FIFO address clear register
A0120040	IO_CONFIG	16	IO config register This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode, or open-drain mode to support wired-and bus.
A0120048	HS	16	High speed mode register This register contains options for supporting high speed operation features. Each HS half pulse width, i.e. each high or low pulse, is equal to $(step_cnt_div+1)*(sample_cnt_div + 1)/13MHz$
A0120050	SOFTRESET	16	Soft reset register
A0120060	SPARE	16	SPARE
A0120064	DEBUGSTAT	16	Debug status register
A0120068	DEBUGCTRL	16	Debug control register
A012006C	TRANSFER_LEN_A UX	16	Transfer length register (number of bytes per transfer)
A0120074	TIMEOUT	16	Timeout timing register

A0120000 [DATA_PORT](#) Data Port Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA_PORT															
Type	RW															
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit(s)	Mnemonic	Name	Description
7:0	DATA_PORT	DATA_PORT	<p>FIFO access port</p> <p>During master write sequences ($slave_addr[0] = 0$), this port can be written by APB, and during master read sequences ($slave_addr[0] = 1$), this port can be read by APB.</p> <p><i>Note: Slave_addr must be set correctly before accessing FIFO.</i></p>

Bit(s)	Mnemonic	Name	Description
			For debugging only: If the fifo_apb_debug bit is set, FIFO can be read and written by the APB.

A0120004 **SLAVE_ADDR** Slave Address Register **00BF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												SLAVE_ADDR				
Type												RW				
Reset									1	0	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
7:0	SLAVE_ADDR	SLAVE_ADDR	<p>Specifies the slave address of the device to be accessed</p> <p>Bit 0 is defined by the I2C protocol as a bit that indicates the direction of transfer.</p> <p>0: Master write</p> <p>1: Master read</p>

A0120008 **INTR_MASK** Interrupt Mask Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												MASK_TIMEOUT		MASK_HS_NACKERR	MASK_ACKERR	MASK_TRANSAC_COMP
Type												RW		RW	RW	RW
Reset												0		0	0	0

Overview: This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = allow interrupt; 0 = disable interrupt Note that while disabled, the corresponding interrupt will not be asserted. However, intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.

Bit(s)	Mnemonic	Name	Description
4	MASK_TIMEOUT	MASK_TIMEOUT	Setting this value to 0 will mask TIMEOUT interrupt signal.
2	MASK_HS_NACKERR	MASK_HS_NACKERR	Setting this value to 0 will mask HS_NACKERR interrupt signal.
1	MASK_ACKERR	MASK_ACKERR	Setting this value to 0 will mask ACK_ERR interrupt signal.
0	MASK_TRANSAC_COMP	MASK_TRANSAC_COMP	Setting this value to 0 will mask TRANSAC_COMP interrupt signal.

A012000C **INTR_STAT** Interrupt Status Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												TIMEOUT	ARB_LOST	HS_NACKERR	ACKERR	TRANSAC_COMP
Type												W1C	W1C	W1C	W1C	W1C

Reset												0	0	0	0	0
-------	--	--	--	--	--	--	--	--	--	--	--	---	---	---	---	---

Overview: When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.

Bit(s)	Mnemonic	Name	Description
4	TIMEOUT	TIMEOUT_IRQ	This status is asserted if time-out is enabled and the timer expires. The internal master state machine will stop, and MCU will need to manually clear the state machine by either issuing software reset by disabling the <code>transact_en</code> bit. Time-out can be used to detect PCB or I2C malfunction.
3	ARB_LOST	SPARE	Reserved
2	HS_NACKERR	HS_NACKERR	This status is asserted if HS master code NACK error detection is enabled. If enabled, HS master code NACK err will cause transaction to end, and stop will be issued.
1	ACKERR	ACKERR	This status is asserted if ACK error detection is enabled. If enabled, ACKERR will cause transaction to end, and stop will be issued.
0	TRANSAC_COMP	TRANSAC_COMP	This status is asserted when a transaction is completed successfully.

A0120010 CONTROL Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								TIMEOUT_EN	RESET_BUSS_BUSY_EN	TRANSFER_LEN_CHANGE	ACKERR_DET_EN	DIR_CHANGE	CLK_EXTEN			RS_STOP
Type								RW	RW	RW	RW	RW	RW			RW
Reset								0	0	0	0	0	0			0

Bit(s)	Mnemonic	Name	Description
8	TIMEOUT_EN	TIMEOUT_EN	Enables time-out mechanism When enabled, if SCL stays at 0 for too long period of time due to I2C slave holds SCL at 0 for too long or SCL sticks at 0 due to PCB issue, the master shall terminate the transaction, stop the internal state machine and assert time-out interrupt. MCU shall handle this case appropriately and reset the master and FIFO address before reissuing the transaction again. If this option is disabled, the HW timer will not count and nor expire forever. 0: Disable 1: Enable
7	RESET_BUSS_BUSY_EN	SPARE	Reserved
6	TRANSFER_LEN_CHANGE	TRANSFER_LEN_CHANGE	Specifies whether or not to change the transfer length after the fist transfer is completed If enabled, the transfers after the first transfer will use the <code>transfer_len_aux</code> parameter. 0: Disable 1: Enable
5	ACKERR_DET_EN	ACKERR_DET_EN	Enables slave ACK error detection When enabled, if slave ACK error is detected, the master shall

Bit(s)	Mnemonic	Name	Description
			terminate the transaction by issuing a STOP condition and then assert the ACKERR interrupt. MCU handles this case appropriately then resets the FIFO address before reissuing transaction. If this option is disabled, the controller will ignore slave ACK error and keep on scheduled transaction. 0: Disable 1: Enable
4	DIR_CHANG	DIR_CHANGE E	Combined transfer format, where the direction of transfer is to be changed from write to read after the FIRST RS condition <i>Note: When set to 1, the transfers after the direction change will be based on the transfer_len_aux parameter.</i> 0: Disable 1: Enable
3	CLK_EXT_E	CLK_EXT_EN N	I2C spec allows slaves to hold the SCL line low if it is not yet ready for further processing Therefore, if this bit is set to 1, the master controller will enter a high wait state until the slave releases the SCL line.
1	RS_STOP	RS_STOP	In LS/FS mode, this bit affects multi-transfer transaction only. It controls whether or not the REPEATED-START condition is used between transfers. The last ending transfer always ends with a STOP. In HS mode, this bit must be set to 1. 0: Use STOP 1: Use REPEATED-START

A0120014 [TRANSFER_LEN](#) Transfer Length Register (Number of Bytes per Transfer) **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TRANSFER_LEN_AUX			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	TRANSFER_LEN_AUX	TRANSFER_LEN	Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte) <i>Note: The value must be set to be bigger than 1; otherwise no transfer will take place.</i>

A0120018 [TRANSAC_LEN](#) Transaction Length Register (Number of Transfers per Transaction) **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									TRANSAC_LEN							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
7:0	TRANSAC_LEN	TRANSAC_LEN	Indicates the number of transfers to be transferred in 1 transaction <i>Note: The value must be set to be bigger than 1; otherwise no</i>

Bit(s)	Mnemonic	Name	Description
			<i>transfer will take place.</i>

A012001C **DELAY_LEN** Inter Delay Length Register **0002**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DELAY_LEN															
Type	RW															
Reset									0	0	0	0	0	0	1	0

Bit(s)	Mnemonic	Name	Description
7:0	DELAY_LEN	DELAY_LEN	Sets up wait delay between consecutive transfers when RS_STOP bit is set to 0 Unit: Half the pulse width

A0120020 **TIMING** Timing Control Register **1303**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA_READ_ADJ	DATA_READ_TIME				SAMPLE_CNT_DIV					STEP_CNT_DIV					
Type	RW	RW				RW					RW					
Reset	0	0	0	1		0	1	1			0	0	0	0	1	1

Overview: LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz

Bit(s)	Mnemonic	Name	Description
15	DATA_READ_ADJ	DATA_READ_ADJ	When set to 1, data latch in sampling time during master reads are adjusted according to the DATA_READ_TIME value. Otherwise, by default, the data are latched in at half of the high pulse width. This value must be set to be smaller than or equal to half the high pulse width.
14:12	DATA_READ_TIME	DATA_READ_TIME	This value is valid only when DATA_READ_ADJ is set to 1. This can be used to adjust so that the data are latched in at earlier sampling points (assuming data are settled by then).
10:8	SAMPLE_CNT_DIV	SAMPLE_CNT_DIV	Used for LS/FS only. This adjusts the width of each sample. Sample width = (sample_cnt_div + 1)/13MHz
5:0	STEP_CNT_DIV	STEP_CNT_DIV	Specifies the number of samples per half pulse width, i.e. each high or low pulse

A0120024 **START** Start Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																START
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	START	START	Starts the transaction on the bus It is auto de-asserted at the end of the transaction.

A0120030 FIFO_STAT **FIFO Status Register** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RD_ADDR				WR_ADDR				FIFO_OFFSET						WR_FULL	RD_EMPTY
Type	RO				RU				RU						RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0			0	1

Bit(s)	Mnemonic	Name	Description
15:12	RD_ADDR	RD_ADDR	Current RD address pointer Only bit [2:0] have physical meanings.
11:8	WR_ADDR	WR_ADDR	Current WR address pointer Only bit [2:0] have physical meanings.
7:4	FIFO_OFFSET	FIFO_OFFSET	wr_addr[3:0] - rd_addr[3:0]
1	WR_FULL	WR_FULL	Indicates FIFO is full
0	RD_EMPTY	RD_EMPTY	Indicates FIFO is empty

A0120038 FIFO_ADDR_CLR **FIFO Address Clear Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																FIFO_ADDR_CLR
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	FIFO_ADDR_CLR	FIFO_ADDR_CLR	When written with 1'b1, a 1 pulse fifo_addr_clr is generated to clear the FIFO address back to 0.

A0120040 IO_CONFIG **IO Config Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													IDLE_OE_EN		SDA_IO_CONFIG	SCL_IO_CONFIG
Type													RW		RW	RW
Reset													0		0	0

Overview: This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode or open-drain mode to support wired-and bus.

Bit(s)	Mnemonic	Name	Description
3	IDLE_OE_EN	IDLE_OE_EN	0: Does not drive bus in idle state

Bit(s)	Mnemonic	Name	Description
N			1: Drive bus in idle state
1	SDA_IO_CONFIG_NFIG	SDA_IO_CONFIG	0: Normal 1: Open-drain mode
0	SCL_IO_CONFIG_NFIG	SCL_IO_CONFIG	0: Normal 1: Open-drain mode

A0120048 **HS** **High Speed Mode Register** **0102**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		HS_SAMPLE_CNT_DIV				HS_STEP_CNT_DIV				MASTER_CODE					HS_NACKERR_DET_EN	HS_EN
Type		RW				RW				RW					RW	RW
Reset		0	0	0		0	0	1		0	0	0			1	0

Overview: This register contains options for supporting high speed operation features Each HS half pulse width, i.e. each high or low pulse, is equal to $(step_cnt_div+1) * (sample_cnt_div + 1) / 13MHz$.

Bit(s)	Mnemonic	Name	Description
14:12	HS_SAMPL E_CNT_DIV	HS_SAMPLE_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the sample width will become dependent on this parameter.
10:8	HS_STEP_CNT_DIV	HS_STEP_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the number of samples per half pulse width will become dependent on this value.
6:4	MASTER_CODE	MASTER_CODE	This is the 3 bit programmable value for the master code to be transmitted.
1	HS_NACKERR_DET_EN	HS_NACKERR_DET_EN	Enables NACKERR detection during the master code transmission When enabled, if NACK is not received after the master code is transmitted, the transaction will be terminated with a STOP condition.
0	HS_EN	HS_EN	Enables high-speed transaction <i>Note: rs_stop must be set to 1.</i>

A0120050 **SOFTRESET** **Soft Reset Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SOFT_RESET
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	SOFT_RESET	SOFT_RESET	When written with 1'b1, a 1 pulse soft reset is used as synchronous reset to reset the I2C internal hardware circuits.

A0120060 **SPARE** SPARE 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	SPARE																
Type	RW																
Reset														0	0	0	0

Bit(s)	Mnemonic	Name	Description
3:0	SPARE	SPARE	Reserved for future use

A0120064 **DEBUGSTAT** Debug Status Register 0020

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								BUS_BUSY	MASTER_WRITE	MASTER_READ	MASTER_STATE					
Type								RO	RO	RO	RO					
Reset								0	0	1	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7	BUS_BUSY	SPARE	Reserved
6	MASTER_WRITE	MASTER_WRITE	For debugging only 1: Current transfer is in the master write dir.
5	MASTER_READ	MASTER_READ	For debugging only 1: Current transfer is in the master read dir.
4:0	MASTER_STATE	MASTER_STATE	(For debugging only) Reads back the current master_state. 0: Idle state 1: I2c master is preparing to send out the start bit, SCL=1, SDA=1. 2: I2C master is sending out the start bit, SCL=1, SDA=0. 3: I2C master/slave is preparing to transmit data bit, SCL=0, SDA=data bit. (Data bit can be changed when SCL=0.) 4: I2C master/slave is transmitting data bit, SCL=1, SDA=data bit. (Data bit is stable when SCL=1.) 5: I2C master/slave is preparing to transmit the ACK bit, SCL=0, SDA=ack. (The ACK bit can be changed when SCL=0.) 6: I2C master/slave is transmitting the ACK bit, SCL=1, SDA=0. (The ACK bit is stable when SCL=1.) 7: I2C master is preparing to send out stop bit or repeated-start bit, SCL=0, SDA=0/1. (0: Stop bit; 1: Repeated-start bit) 8: I2C master is sending out stop bit or repeated-start bit, SCL=1, SDA=1/0. (0: Repeated-start bit; 1: Stop bit) 9: I2C master is in delay start between two transfers, SCL=1, SDA=1. 10: I2C master is in FIFO wait state; For writing transaction, it means FIFO is empty and I2C master is waiting for DMA controller to write data into FIFO. For reading transaction, it means FIFO is full and I2C master is waiting for DMA controller to read data from FIFO, SCL=0, SDA=don't care. 12: I2C master is preparing to send out data bit of master code. This state is used only in high-speed transaction, SCL=0, SDA=data bit of master code. (Data bit of master code can be changed when SCL=0.)

Bit(s)	Mnemonic	Name	Description
			13: I2C master is sending out data bit of master code. This state is used only in high-speed transaction, SCL=1, SDA=data bit of master code. (Data bit of master code is stable when SCL=1.)
			14: I2C master/slave is preparing to transmit the NACK bit, SCL=0, SDA=nack bit. (The NACK bit can be changed when SCL=0.) This state is used only in high-speed transaction.
			15: I2C master/slave is transmitting the NACK bit, SCL=1, SDA=1. This state is used only in high-speed transaction.

A0120068 **DEBUGCTRL** **Debug Control Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															APB_DEBU G_RD	FIFO_APB_ DEBU G
Type															WO	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	APB_DEBU G_RD	APB_DEBUG_RD	Only valid when fifo_apb_debug is set to 1 Writing to this register will generate a 1 pulsed FIFO APB RD signal for reading the FIFO data.
0	FIFO_APB_ DEBU	FIFO_APB_DEBUG	Used for trace 32 debugging When using trace 32, and the memory map is shown, turning this bit on will block the normal APB read access. The APB read access to the FIFO will then be enabled by writing to apb_debug_rd. 0: Disable 1: Enable

A012006C **TRANSFER_LEN_AUX** **Transfer Length Register (Number of Bytes per Transfer)** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset															0	1

Bit(s)	Mnemonic	Name	Description
3:0	TRANSFER_LEN	TRANSFER_LEN_AUX	Only valid when dir_change or transfer_len_change is set to 1. Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte) for the transfers following the direction change or transfer_len_change If dir_change =1, the first write transfer length will depend on transfer_len, while the second read transfer length will depend on transfer_len_aux. Dir change is always after the first transfer. Similarly, transfer length change is always after the first transfer. <i>Note: The value must be set to be bigger than 1; otherwise no transfer will take place.</i>

A0120074 **TIMEOUT** Timeout Timing Register FFFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIMEOUT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0		TIMEOUT	Indicates the number of steps to count before time-out The time-out counter counts only when the time-out mechanism is enabled and has started a transaction.

3.11 I2C/SCCB Controller (I2C_SCCB_Controller_V18)

3.11.1 General Description

I2C (Inter-IC)/SCCB (Serial Camera Control Bus) is a two-wire serial interface. The two signals are SCL and SDA. SCL is a clock signal that is driven by the master. SDA is a bi-directional data signal that can be driven by either the master or the slave. This generic controller supports the master role and conforms to the I2C specification.

3.11.1.1 Feature

- I2C compliant master mode operation
- Adjustable clock speed for LS/FS mode operation
- Supports 7-bit/10-bit addressing
- Supports high-speed mode
- Supports slave clock extension
- START/STOP/REPEATED START condition
- Manual transfer mode
- Multi-write per transfer (up to 8 data bytes for non-DMA mode)
- Multi-read per transfer (up to 8 data bytes for non-DMA mode)
- Multi-transfer per transaction
- Combined format transfer with length change capability
- Active drive/wired-and I/O configuration

3.11.1.2 Manual Transfer Mode

The controller offers one transfer mode, the manual mode.

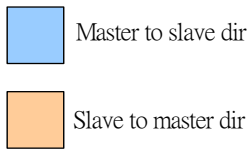
When the manual mode is selected, in addition to the slave address register, the controller has a built-in 8byte deep FIFO which allows MCU to prepare up to 8 bytes of data for a write transfer, or read up to 8 bytes of data for a read transfer.

3.11.1.3 Transfer Format Support

This controller has been designed to be as generic as possible in order to support a wide range of devices that may utilize different combinations of transfer formats. Here are the transfer format types that can be supported through different software configuration:

Wording convention note

- Transfer = Anything encapsulated within a Start and Stop or Repeated Start.
- Transfer length = Number of bytes within the transfer
- Transaction = This is the top unit. Everything combined equals 1 transaction.
- Transaction length = Number of transfers to be conducted.



Single-byte access

Single Byte Write

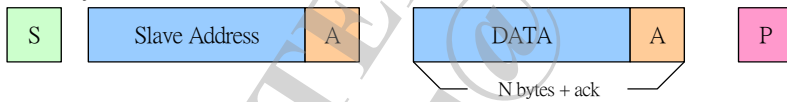


Single Byte Read

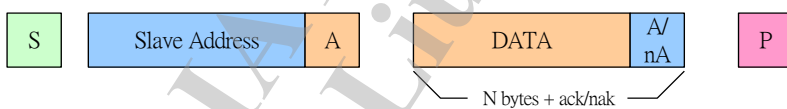


Multi-byte access

Multi Byte Write

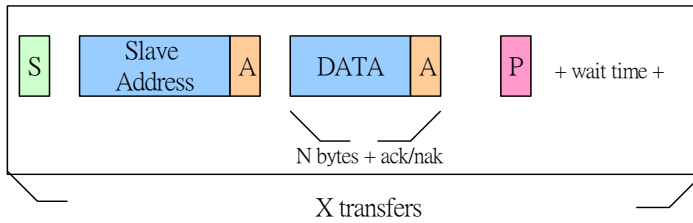


Multi Byte Read

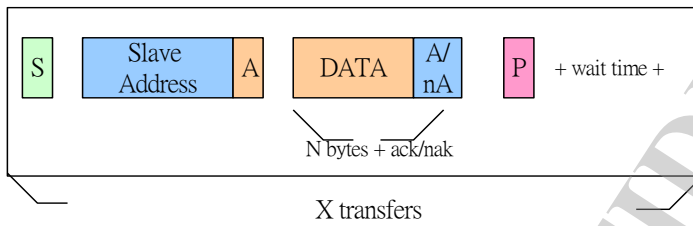


Multi-byte transfer + multi-transfer (same direction)

Multi Byte Write + Multi Transfer

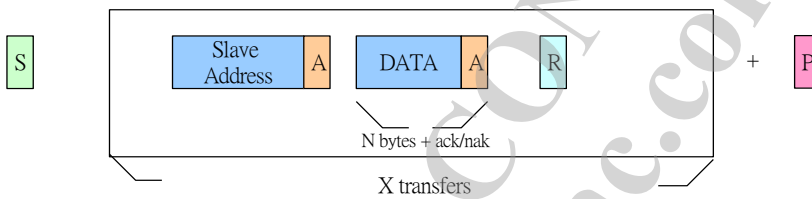


Multi Byte Read + Multi Transfer

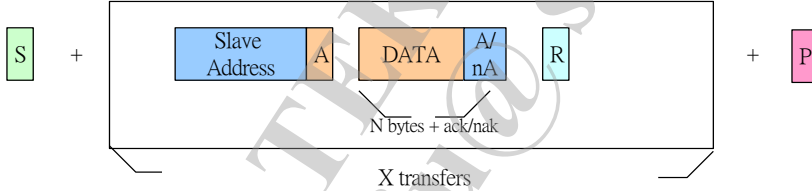


Multi-byte transfer + multi-transfer w RS (same direction)

Multi Byte Write + Multi Transfer + Repeated Start



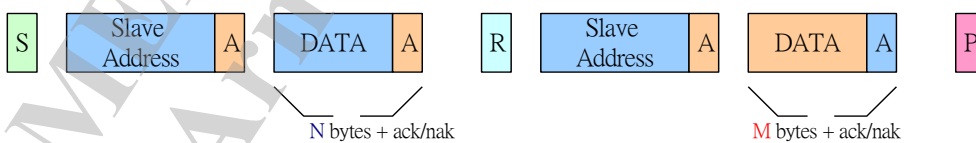
Multi Byte Read + Multi Transfer + Repeated Start



Combined write/read with Repeated Start (direction change)

Note: Only supports write and then read sequence. Read and then write is not supported.

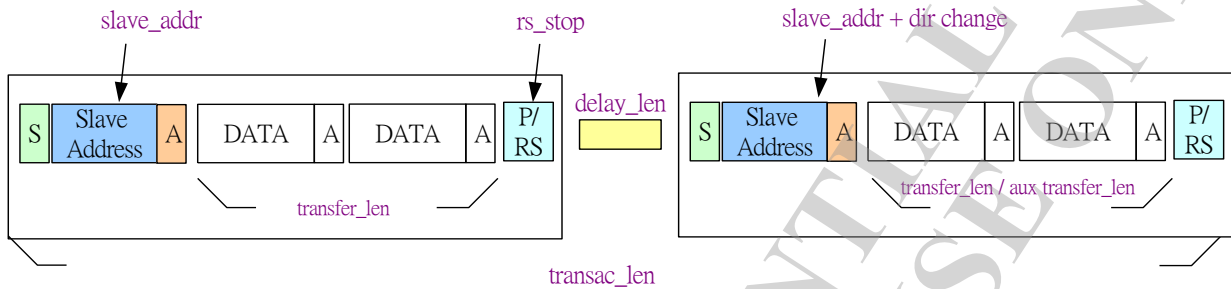
Combined Multi Byte Write + Multi Byte Read



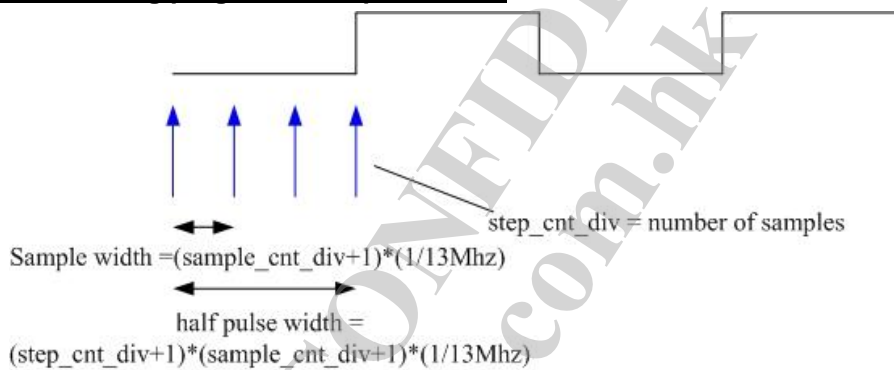
3.11.2 Programming Examples

Common transfer programmable parameters

Programmable Parameters



Output waveform timing programmable parameters



3.11.3 Register Definition

Module name: I2C_SCCB_Controller_V18 base address: (+A02A0000h)

Address	Name	Width	Register function
A02A0000	DATA_PORT	16	Data port register
A02A0004	SLAVE_ADDR	16	Slave address register
A02A0008	INTR_MASK	16	<p>Interrupt mask register</p> <p>This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register.</p> <p>1 = Allow interrupt 0 = Disable interrupt</p> <p><i>Note: While disabled, the corresponding interrupt will not be asserted, however intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.</i></p>
A02A000C	INTR_STAT	16	<p>Interrupt status register</p> <p>When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.</p>
A02A0010	CONTROL	16	Control register

Address	Name	Width	Register function
A02A0014	TRANSFER_LEN	16	Transfer length register (number of bytes per transfer)
A02A0018	TRANSAC_LEN	16	Transaction length register (number of transfers per transaction)
A02A001C	DELAY_LEN	16	Inter delay length register
A02A0020	TIMING	16	Timing control register LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to $(step_cnt_div+1)*(sample_cnt_div + 1)/13MHz$
A02A0024	START	16	Start register
A02A0030	FIFO_STAT	16	FIFO status register
A02A0038	FIFO_ADDR_CLR	16	FIFO address clear register
A02A0040	IO_CONFIG	16	IO config register This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode, or open-drain mode to support wired-and bus.
A02A0048	HS	16	High speed mode register This register contains options for supporting high speed operation features. Each HS half pulse width, i.e. each high or low pulse, is equal to $(step_cnt_div+1)*(sample_cnt_div + 1)/13MHz$
A02A0050	SOFTRESET	16	Soft reset register
A02A0060	SPARE	16	SPARE
A02A0064	DEBUGSTAT	16	Debug status register
A02A0068	DEBUGCTRL	16	Debug control register
A02A006C	TRANSFER_LEN_A UX	16	Transfer length register (number of bytes per transfer)
A02A0074	TIMEOUT	16	Timeout timing register

A02A0000 [DATA_PORT](#) Data Port Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA_PORT															
Type	RW															
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit(s)	Mnemonic	Name	Description
7:0	DATA_PORT	DATA_PORT	<p>FIFO access port</p> <p>During master write sequences ($slave_addr[0] = 0$), this port can be written by APB, and during master read sequences ($slave_addr[0] = 1$), this port can be read by APB.</p> <p><i>Note: Slave_addr must be set correctly before accessing FIFO.</i></p>

Bit(s)	Mnemonic	Name	Description
			For debugging only: If the fifo_apb_debug bit is set, FIFO can be read and written by the APB.

A02A0004 **SLAVE_ADDR** Slave Address Register **00BF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SLAVE_ADDR															
Type	RW															
Reset									1	0	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
7:0	SLAVE_ADDR	SLAVE_ADDR	<p>Specifies the slave address of the device to be accessed</p> <p>Bit 0 is defined by the I2C protocol as a bit that indicates the direction of transfer.</p> <p>0: Master write</p> <p>1: Master read</p>

A02A0008 **INTR_MASK** Interrupt Mask Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												MASK_TIMEOUT		MASK_HS_NACKERR	MASK_ACKERR	MASK_TRANSAC_COMP
Type												RW		RW	RW	RW
Reset												0		0	0	0

Overview: This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = allow interrupt; 0 = disable interrupt Note that while disabled, the corresponding interrupt will not be asserted. However, intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.

Bit(s)	Mnemonic	Name	Description
4	MASK_TIMEOUT	MASK_TIMEOUT	Setting this value to 0 will mask TIMEOUT interrupt signal.
2	MASK_HS_NACKERR	MASK_HS_NACKERR	Setting this value to 0 will mask HS_NACKERR interrupt signal.
1	MASK_ACKERR	MASK_ACKERR	Setting this value to 0 will mask ACK_ERR interrupt signal.
0	MASK_TRANSAC_COMP	MASK_TRANSAC_COMP	Setting this value to 0 will mask TRANSAC_COMP interrupt signal.

A02A000C **INTR_STAT** Interrupt Status Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												TIMEOUT	ARB_LOST	HS_NACKERR	ACKERR	TRANSAC_COMP
Type												W1C	W1C	W1C	W1C	W1C

Reset														0	0	0	0	0
-------	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---	---	---	---

Overview: When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.

Bit(s)	Mnemonic	Name	Description
4	TIMEOUT	TIMEOUT_IRQ	This status is asserted if time-out is enabled and the timer expires. The internal master state machine will stop, and MCU will need to manually clear the state machine by either issuing software reset by disabling the <code>transact_en</code> bit. Time-out can be used to detect PCB or I2C malfunction.
3	ARB_LOST	SPARE	Reserved
2	HS_NACKERR	HS_NACKERR	This status is asserted if HS master code NACK error detection is enabled. If enabled, HS master code NACK err will cause transaction to end, and stop will be issued.
1	ACKERR	ACKERR	This status is asserted if ACK error detection is enabled. If enabled, ACKERR will cause transaction to end, and stop will be issued.
0	TRANSAC_COMP	TRANSAC_COMP	This status is asserted when a transaction is completed successfully.

A02A0010 CONTROL Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								TIMEOUT_EN	RESET_BUSS_BUSY_EN	TRANSFER_LEN_CHANGE	ACKERR_DET_EN	DIR_C HANGE	CLK_EXT_EN		RS_STOP	
Type								RW	RW	RW	RW	RW	RW		RW	
Reset								0	0	0	0	0	0		0	

Bit(s)	Mnemonic	Name	Description
8	TIMEOUT_EN	TIMEOUT_EN	Enables time-out mechanism When enabled, if SCL stays at 0 for too long period of time due to I2C slave holds SCL at 0 for too long or SCL sticks at 0 due to PCB issue, the master shall terminate the transaction, stop the internal state machine and assert time-out interrupt. MCU shall handle this case appropriately and reset the master and FIFO address before reissuing the transaction again. If this option is disabled, the HW timer will not count and nor expire forever. 0: Disable 1: Enable
7	RESET_BUSS_BUSY_EN	SPARE	Reserved
6	TRANSFER_LEN_CHANGE	TRANSFER_LEN_CHANGE	Specifies whether or not to change the transfer length after the fist transfer is completed If enabled, the transfers after the first transfer will use the <code>transfer_len_aux</code> parameter. 0: Disable 1: Enable
5	ACKERR_DET_EN	ACKERR_DET_EN	Enables slave ACK error detection When enabled, if slave ACK error is detected, the master shall

Bit(s)	Mnemonic	Name	Description
			terminate the transaction by issuing a STOP condition and then assert the ACKERR interrupt. MCU handles this case appropriately then resets the FIFO address before reissuing transaction. If this option is disabled, the controller will ignore slave ACK error and keep on scheduled transaction. 0: Disable 1: Enable
4	DIR_CHANG	DIR_CHANGE E	Combined transfer format, where the direction of transfer is to be changed from write to read after the FIRST RS condition <i>Note: When set to 1, the transfers after the direction change will be based on the transfer_len_aux parameter.</i> 0: Disable 1: Enable
3	CLK_EXT_E	CLK_EXT_EN N	I2C spec allows slaves to hold the SCL line low if it is not yet ready for further processing Therefore, if this bit is set to 1, the master controller will enter a high wait state until the slave releases the SCL line.
1	RS_STOP	RS_STOP	In LS/FS mode, this bit affects multi-transfer transaction only. It controls whether or not the REPEATED-START condition is used between transfers. The last ending transfer always ends with a STOP. In HS mode, this bit must be set to 1. 0: Use STOP 1: Use REPEATED-START

A02A0014 [TRANSFER_LEN](#) Transfer Length Register (Number of Bytes per Transfer) **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TRANSFER_LEN_AUX			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	TRANSFER_LEN_AUX	TRANSFER_LEN	Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte) <i>Note: The value must be set to be bigger than 1; otherwise no transfer will take place.</i>

A02A0018 [TRANSAC_LEN](#) Transaction Length Register (Number of Transfers per Transaction) **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset									0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
7:0	TRANSAC_LEN	TRANSAC_LEN	Indicates the number of transfers to be transferred in 1 transaction <i>Note: The value must be set to be bigger than 1; otherwise no</i>

Bit(s)	Mnemonic	Name	Description
			<i>transfer will take place.</i>

A02A001C **DELAY_LEN** Inter Delay Length Register **0002**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DELAY_LEN															
Type	RW															
Reset									0	0	0	0	0	0	1	0

Bit(s)	Mnemonic	Name	Description
7:0	DELAY_LEN	DELAY_LEN	Sets up wait delay between consecutive transfers when RS_STOP bit is set to 0 Unit: Half the pulse width

A02A0020 **TIMING** Timing Control Register **1303**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA_READ_ADJ	DATA_READ_TIME				SAMPLE_CNT_DIV					STEP_CNT_DIV					
Type	RW	RW				RW					RW					
Reset	0	0	0	1		0	1	1			0	0	0	0	1	1

Overview: LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz

Bit(s)	Mnemonic	Name	Description
15	DATA_READ_ADJ	DATA_READ_ADJ	When set to 1, data latch in sampling time during master reads are adjusted according to the DATA_READ_TIME value. Otherwise, by default, the data are latched in at half of the high pulse width point. This value must be set to be smaller than or equal to half the high pulse width.
14:12	DATA_READ_TIME	DATA_READ_TIME	This value is valid only when DATA_READ_ADJ is set to 1. This can be used to adjust so that the data are latched in at earlier sampling points (assuming data are settled by then).
10:8	SAMPLE_CNT_DIV	SAMPLE_CNT_DIV	Used for LS/FS only. This adjusts the width of each sample. Sample width = (sample_cnt_div + 1)/13MHz
5:0	STEP_CNT_DIV	STEP_CNT_DIV	Specifies the number of samples per half pulse width, i.e. each high or low pulse

A02A0024 **START** Start Register **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																START
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	START	START	Starts the transaction on the bus It is auto de-asserted at the end of the transaction.

A02A0030 FIFO_STAT **FIFO Status Register** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RD_ADDR				WR_ADDR				FIFO_OFFSET						WR_FULL	RD_EMPTY
Type	RO				RU				RU						RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0			0	1

Bit(s)	Mnemonic	Name	Description
15:12	RD_ADDR	RD_ADDR	Current RD address pointer Only bit [2:0] have physical meanings.
11:8	WR_ADDR	WR_ADDR	Current WR address pointer Only bit [2:0] have physical meanings.
7:4	FIFO_OFFSET	FIFO_OFFSET	wr_addr[3:0] - rd_addr[3:0]
1	WR_FULL	WR_FULL	Indicates FIFO is full
0	RD_EMPTY	RD_EMPTY	Indicates FIFO is empty

A02A0038 FIFO_ADDR_CLR **FIFO Address Clear Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																FIFO_ADDR_CLR
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	FIFO_ADDR_CLR	FIFO_ADDR_CLR	When written with 1'b1, a 1 pulse fifo_addr_clr is generated to clear the FIFO address back to 0.

A02A0040 IO_CONFIG **IO Config Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													IDLE_OE_EN		SDA_IO_CONFIG	SCL_IO_CONFIG
Type													RW		RW	RW
Reset													0		0	0

Overview: This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode or open-drain mode to support wired-and bus.

Bit(s)	Mnemonic	Name	Description
3	IDLE_OE_EN	IDLE_OE_EN	0: Does not drive bus in idle state

Bit(s)	Mnemonic	Name	Description
N			1: Drive bus in idle state
1	SDA_IO_CONFIG_NFIG	SDA_IO_CONFIG	0: Normal 1: Open-drain mode
0	SCL_IO_CONFIG_NFIG	SCL_IO_CONFIG	0: Normal 1: Open-drain mode

A02A0048 **HS** **High Speed Mode Register** **0102**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		HS_SAMPLE_CNT_DIV				HS_STEP_CNT_DIV				MASTER_CODE					HS_NACKERR_DET_EN	HS_EN
Type		RW				RW				RW					RW	RW
Reset		0	0	0		0	0	1		0	0	0			1	0

Overview: This register contains options for supporting high speed operation features Each HS half pulse width, i.e. each high or low pulse, is equal to $(step_cnt_div+1) * (sample_cnt_div + 1) / 13MHz$.

Bit(s)	Mnemonic	Name	Description
14:12	HS_SAMPL E_CNT_DIV	HS_SAMPLE_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the sample width will become dependent on this parameter.
10:8	HS_STEP_CNT_DIV	HS_STEP_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the number of samples per half pulse width will become dependent on this value.
6:4	MASTER_CODE	MASTER_CODE	This is the 3 bit programmable value for the master code to be transmitted.
1	HS_NACKERR_DET_EN	HS_NACKERR_DET_EN	Enables NACKERR detection during the master code transmission When enabled, if NACK is not received after the master code is transmitted, the transaction will be terminated with a STOP condition.
0	HS_EN	HS_EN	Enables high-speed transaction <i>Note: rs_stop must be set to 1.</i>

A02A0050 **SOFTRESET** **Soft Reset Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SOFT_RESET
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	SOFT_RESET	SOFT_RESET	When written with 1'b1, a 1 pulse soft reset is used as synchronous reset to reset the I2C internal hardware circuits.

A02A0060 **SPARE** SPARE 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPARE															
Type	RW															
Reset													0	0	0	0

Bit(s)	Mnemonic	Name	Description
3:0	SPARE	SPARE	Reserved for future use

A02A0064 **DEBUGSTAT** Debug Status Register 0020

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								BUS_BUSY	MASTER_WRITE	MASTER_READ	MASTER_STATE					
Type								RO	RO	RO	RO					
Reset								0	0	1	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7	BUS_BUSY	SPARE	Reserved
6	MASTER_WRITE	MASTER_WRITE	For debugging only 1: Current transfer is in the master write dir.
5	MASTER_READ	MASTER_READ	For debugging only 1: Current transfer is in the master read dir.
4:0	MASTER_STATE	MASTER_STATE	(For debugging only) Reads back the current master_state. 0: Idle state 1: I2c master is preparing to send out the start bit, SCL=1, SDA=1. 2: I2C master is sending out the start bit, SCL=1, SDA=0. 3: I2C master/slave is preparing to transmit data bit, SCL=0, SDA=data bit. (Data bit can be changed when SCL=0.) 4: I2C master/slave is transmitting data bit, SCL=1, SDA=data bit. (Data bit is stable when SCL=1.) 5: I2C master/slave is preparing to transmit the ACK bit, SCL=0, SDA=ack. (The ACK bit can be changed when SCL=0.) 6: I2C master/slave is transmitting the ACK bit, SCL=1, SDA=0. (The ACK bit is stable when SCL=1.) 7: I2C master is preparing to send out stop bit or repeated-start bit, SCL=0, SDA=0/1. (0: Stop bit; 1: Repeated-start bit) 8: I2C master is sending out stop bit or repeated-start bit, SCL=1, SDA=1/0. (0: Repeated-start bit; 1: Stop bit) 9: I2C master is in delay start between two transfers, SCL=1, SDA=1. 10: I2C master is in FIFO wait state; For writing transaction, it means FIFO is empty and I2C master is waiting for DMA controller to write data into FIFO. For reading transaction, it means FIFO is full and I2C master is waiting for DMA controller to read data from FIFO, SCL=0, SDA=don't care. 12: I2C master is preparing to send out data bit of master code. This state is used only in high-speed transaction, SCL=0, SDA=data bit of master code. (Data bit of master code can be changed when SCL=0.)

Bit(s)	Mnemonic	Name	Description
			13: I2C master is sending out data bit of master code. This state is used only in high-speed transaction, SCL=1, SDA=data bit of master code. (Data bit of master code is stable when SCL=1.)
			14: I2C master/slave is preparing to transmit the NACK bit, SCL=0, SDA=nack bit. (The NACK bit can be changed when SCL=0.) This state is used only in high-speed transaction.
			15: I2C master/slave is transmitting the NACK bit, SCL=1, SDA=1. This state is used only in high-speed transaction.

A02A0068 **DEBUGCTRL** **Debug Control Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															APB_DEBU G_RD	FIFO_APB_ DEBU G
Type															WO	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	APB_DEBU G_RD	APB_DEBUG_RD	Only valid when fifo_apb_debug is set to 1 Writing to this register will generate a 1 pulsed FIFO APB RD signal for reading the FIFO data.
0	FIFO_APB_ DEBU	FIFO_APB_DEBUG	Used for trace 32 debugging When using trace 32, and the memory map is shown, turning this bit on will block the normal APB read access. The APB read access to the FIFO will then be enabled by writing to apb_debug_rd. 0: Disable 1: Enable

A02A006C **TRANSFER_LEN_AUX** **Transfer Length Register (Number of Bytes per Transfer)** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
3:0	TRANSFER_LEN	TRANSFER_LEN_AUX	Only valid when dir_change or transfer_len_change is set to 1. Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte) for the transfers following the direction change or transfer_len_change If dir_change =1, the first write transfer length will depend on transfer_len, while the second read transfer length will depend on transfer_len_aux. Dir change is always after the first transfer. Similarly, transfer length change is always after the first transfer. <i>Note: The value must be set to be bigger than 1; otherwise no transfer will take place.</i>

A02A0074 **TIMEOUT** Timeout Timing Register FFFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIMEOUT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0		TIMEOUT	Indicates the number of steps to count before time-out The time-out counter counts only when the time-out mechanism is enabled and has started a transaction.

3.12 Real Time Clock

3.12.1 General Descriptions

The Real-Time Clock (RTC) module provides time and data information, as well as 32.768kHz clock. By configuring pin XOSC32_ENB, the use of the 32k crystal can be determined, i.e. using a 32k crystal, or not to use a 32k crystal. The RTC block has an independent power supply. When the mobile handset is powered off, a dedicated regulator supplies the RTC block. If the main battery is not present, a backup supply such as a small mercury cell battery or a large capacitor is used. In addition to providing timing data, an alarm interrupt is generated and can be used to power up the baseband core. Regulator interrupts corresponding to seconds, minutes, hours and days can be generated whenever the time counter value reaches a maximum value (e.g., 59 for seconds and minutes, 23 for hours, etc.). The year span is supported up to 2127. The maximum day-of-month values, which depend on the leap year condition, are stored in the RTC block.

3.12.2 Register Definition

Module name: RTC Base address: (+A0710000h)

Address	Name	Width	Register function
A0710000	<u>RTC_BBPU</u>	16	Baseband power up
A071 0004	<u>RTC_IRQ_STA</u>	16	RTC IRQ status This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A071 0008	<u>RTC_IRQ_EN</u>	16	RTC IRQ enable This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A071 000C	<u>RTC_CII_EN</u>	16	Counter increment IRQ enable This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.
A071 0010	<u>RTC_AL_MAS K</u>	16	RTC alarm mask The alarm condition for alarm IRQ generation depends on whether or not the corresponding bit in this register is masked. Warning: If you set all bits to 1 in

Address	Name	Width	Register function
			RTC_AL_MASK (i.e. RTC_AL_MASK=0x7f) and PWREN=1 in RTC_BBPU, it means the alarm will come every second, not disabled.
A071 0014	<u>RTC TC SEC</u>	16	RTC seconds time counter register
A071 0018	<u>RTC TC MIN</u>	16	RTC minutes time counter register
A071001C	<u>RTC TC HOU</u>	16	RTC hours time counter register
A0710020	<u>RTC TC DOM</u>	16	RTC day-of-month time counter register
A0710024	<u>RTC TC DOW</u>	16	RTC day-of-week time counter register
A0710028	<u>RTC TC MTH</u>	16	RTC month time counter register
A071002C	<u>RTC TC YEA</u>	16	RTC year time counter register
A0710030	<u>RTC AL SEC</u>	16	RTC second alarm setting register
A0710034	<u>RTC AL MIN</u>	16	RTC minute alarm setting register
A0710038	<u>RTC AL HOU</u>	16	RTC hour alarm setting register
A071003C	<u>RTC AL DOM</u>	16	RTC day-of-month alarm setting register
A0710040	<u>RTC AL DOW</u>	16	RTC day-of-week alarm setting register
A0710044	<u>RTC AL MTH</u>	16	RTC month alarm setting register
A0710048	<u>RTC AL YEA</u>	16	RTC year alarm setting register
A071004C	<u>RTC OSC32 ON</u>	16	OSC32 control The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.
A0710050	<u>RTC POWERKEY1</u>	16	RTC_POWERKEY1 register
A0710054	<u>RTC POWERKEY2</u>	16	RTC_POWERKEY2 register
A0710058	<u>RTC PDN1</u>	16	PDN1
A071005C	<u>RTC PDN2</u>	16	PDN2
A0710060	<u>RTC SPAR0</u>	16	Spare register for specific purpose
A0710064	<u>RTC SPAR1</u>	16	Spare register for specific purpose
A0710068	<u>RTC PROT</u>	16	Lock/unlock scheme to prevent RTC miswriting
A071006C	<u>RTC DIFF</u>	16	One-time calibration offset This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A0710070	<u>RTC CALI</u>	16	Repeat calibration offset This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A0710074	<u>RTC WRTGR</u>	16	Enable transfers from core to RTC in queue
A0710078	<u>RTC CON</u>	16	Other RTC control registers Note: LPRST and LPEN are tied to 0 internally when RTC_POWERKEY1 & RTC_POWERKEY2 do not match the correct values. After changing RTC_CON, write WRTGR = 1 to make it take effect.

A0710000 RTC_BBPU Baseband Power Up 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	KEY_BBPU									CBUSY	RELOAD	CLRPKY	AUTO	BBPU		PWREN
Type	WO									RO	WO	WO	RW	RW		RW
Reset	0	0	0	0	0	0	0	0		0	0	0	0	0		0

Bit (s)	Mnemonic	Name	Description
15:8	KEY_BBPU	KEY_BBPU	A bus write is acceptable only when KEY_BBPU=0x43.
6	CBUSY	CBUSY	The read/write channels between RTC/Core is busy. This bit indicates high after the software program sequence to anyone of RTC data registers and enables the transfer by RTC_WRTGR = 1. In addition, it is high after the reset from low to high due to RTC reload process.
5	RELOAD	RELOAD	Reloads the values from RTC domain to core domain Generally the RTC will reload and synchronize the data from RTC to core when being reset from 0 to 1. This bit can be treated as a debugging bit.
4	CLRPKY	CLRPKY	Clears powerkey1 and powerkey2 at the same time In some cases, the software may clear powerkey1 & powerkey2. BBWAKEUP depends on the matching specific patterns of powerkey1 and powerkey2. If any one of powerkey1 or powerkey2 or BBPU is cleared, BBWAKEUP will go low immediately. The software cannot program the other control bits without power. By programming RTC_BBPU with CLRPKY = 1 and BBPU = 0 condition, RTC can clear powerkey1, powerkey2 and BBPU at the same time.
3	AUTO	AUTO	Controls if BBWAKEUP is automatically in the low state when SYSRST transitions from high to low. 0: BBWAKEUP is not automatically in the low state when SYSRST# transitions are from high to low. 1: BBWAKEUP is automatically in the low state when SYSRST# transitions are from high to low. The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.
2	BBPU	BBPU	Controls the power of PMU If powerkey1 = A357h and powerkey2 = 67D2h, PMU takes on the value programmed by software; otherwise PMU is low. 0: Power down 1: Power on
0	PWREN	PWREN	0: RTC alarm has no action on power switch. 1: When an RTC alarm occurs, BBPU is set to 1 and the system is powered on by RTC alarm wakeup.

A0710004 RTC_IRQ_STA RTC IRQ Status 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													LPSTA		TCSTA	ALSTA
Type													RC		RC	RC
Reset													0		0	0

Overview: This register is read-cleared and is fixed to 0 when RTC_POWERKEY1 and RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
3	LPSTA	LPSTA	Indicates the IRQ status and whether or not the LPD is asserted (LPD function is either provided by XOSC32 or EOSC32 ⁶ , depending on XOSC32_ENB) 0: No IRQ occurred; the 32K clock is good. 1: IRQ occurred; the 32K clock stops. This can be masked by LP_EN or cleared by initializing LPD.
1	TCSTA	TCSTA	Indicates the IRQ status and whether or not the tick condition has been met. 0: No IRQ occurred; the tick condition has not been met. 1: IRQ occurred; the tick condition has been met.
0	ALSTA	ALSTA	Indicates the IRQ status and whether or not the alarm condition has been met. 0: No IRQ occurred; the alarm condition has not been met. 1: IRQ occurred; the alarm condition has been met.

A0710008 **RTC_IRQ_EN** **RTC IRQ Enable** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													LP_EN	ONESHOT	TC_EN	AL_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Overview: This register is fixed to 0 when RTC_POWERKEY1 and RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
3	LP_EN	LP_EN	Enables the control bit for IRQ generation if low power detected (32k clock off). 0: Disable IRQ generations 1: Enable LPD
2	ONESHOT	ONESHOT	Controls automatic reset of AL_EN and TC_EN
1	TC_EN	TC_EN	Enables the control bit for IRQ generation if the tick condition has been met. 0: Disable IRQ generations 1: Enable the tick time match interrupt. Clear the interrupt when ONESHOT is high upon generation of the corresponding IRQ.
0	AL_EN	AL_EN	Enables the control bit for IRQ generation if the alarm condition has been met. 0: Disable IRQ generations

Bit (s)	Mnemonic	Name	Description
			1: Enable the alarm time match interrupt. Clear the interrupt when ONESHOT is high upon generation of the corresponding IRQ.

A071000C RTC_CII_EN Counter Increment IRQ Enable **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne							1/8SECCII	1/4SECCII	1/2SECCII	YEACII	MTHCII	DOWCII	DOMCII	HOUCII	MINCII	SECCII
Type							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset							0	0	0	0	0	0	0	0	0	0

Overview: This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.

Bit (s)	Mnemonic	Name	Description
9	1/8SECCII	SECCII_1_8	Set the bit to 1 to activate the IRQ at each 1/8 of a second update
8	1/4SECCII	SECCII_1_4	Set the bit to 1 to activate the IRQ at each 1/4 of a second update
7	1/2SECCII	SECCII_1_2	Set the bit to 1 to activate the IRQ at each 1/2 of a second update
6	YEACII	YEACII	Set the bit to 1 to activate the IRQ at each year update
5	MTHCII	MTHCII	Set the bit to 1 to activate the IRQ at each month update
4	DOWCII	DOWCII	Set the bit to 1 to activate the IRQ at each day-of-week update
3	DOMCII	DOMCII	Set the bit to 1 to activate the IRQ at each day-of-month update
2	HOUCII	HOUCII	Set the bit to 1 to activate the IRQ at each hour update
1	MINCII	MINCII	Set the bit to 1 to activate the IRQ at each minute update
0	SECCII	SECCII	Set this bit to 1 to activate the IRQ at each second update

A0710010 RTC_AL_MAS
K RTC Alarm Mask **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne										YEA_MSK	MTH_MSK	DOW_MSK	DOM_MSK	HOU_MSK	MIN_MSK	SEC_MSK
Type										RW	RW	RW	RW	RW	RW	RW
Reset										0	0	0	0	0	0	0

Overview: The alarm condition for alarm IRQ generation depends on whether or not the corresponding bit in this register is masked. Note that if all bits 1 in RTC_AL_MASK are set (i.e. RTC_AL_MASK = 0x7f) and PWREN = 1 in RTC_BBPU, it means the alarm will come every second, not disabled.

Bit (s)	Mnemonic	Name	Description
6	YEA_MSK	YEA_MSK	0: Condition (RTC_TC_YEA = RTC_AL_YEA) is checked to generate the alarm signal.

Bit (s)	Mnemonic	Name	Description
5	MTH_MSK	MTH_MSK	0: Condition (RTC_TC_YEA = RTC_AL_YEA) is masked, i.e. the value of RTC_TC_YEA does not affect the alarm IRQ generation. 1: Condition (RTC_TC_MTH = RTC_AL_MTH) is checked to generate the alarm signal.
4	DOW_MSK	DOW_MSK	0: Condition (RTC_TC_DOW = RTC_AL_DOW) is checked to generate the alarm signal. 1: Condition (RTC_TC_DOW = RTC_AL_DOW) is masked, i.e. the value of RTC_TC_DOW does not affect the alarm IRQ generation.
3	DOM_MSK	DOM_MSK	0: Condition (RTC_TC_DOM = RTC_AL_DOM) is checked to generate the alarm signal. 1: Condition (RTC_TC_DOM = RTC_AL_DOM) is masked, i.e. the value of RTC_TC_DOM does not affect the alarm IRQ generation.
2	HOU_MSK	HOU_MSK	0: Condition (RTC_TC_HOU = RTC_AL_HOU) is checked to generate the alarm signal. 1: Condition (RTC_TC_HOU = RTC_AL_HOU) is masked, i.e. the value of RTC_TC_HOU does not affect the alarm IRQ generation.
1	MIN_MSK	MIN_MSK	0: Condition (RTC_TC_MIN = RTC_AL_MIN) is checked to generate the alarm signal. 1: Condition (RTC_TC_MIN = RTC_AL_MIN) is masked, i.e. the value of RTC_TC_MIN does not affect the alarm IRQ generation.
0	SEC_MSK	SEC_MSK	0: Condition (RTC_TC_SEC = RTC_AL_SEC) is checked to generate the alarm signal. 1: Condition (RTC_TC_SEC = RTC_AL_SEC) is masked, i.e. the value of RTC_TC_SEC does not affect the alarm IRQ generation.

A0710014 **RTC TC SEC** **RTC Seconds Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											TC_SECOND					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	TC_SECOND	TC_SECOND	Second initial value for the time counter Range: 0 ~ 59

A0710018 **RTC TC MIN** **RTC Minutes Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											TC_MINUTE					
Type											RW					

Reset																	0	0	0	0	0	0
-------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---	---	---	---	---

Bit (s)	Mnemonic	Name	Description
5:0	TC_MINUTE	TC_MINUTE	Minute initial value for the time counter Range: 0 ~ 59

A071001C **RTC TC_HOU** **RTC Hours Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Mne													TC_HOUR				
Type													RW				
Reset													0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	TC_HOUR	TC_HOUR	Hour initial value for the time counter Range: 0 ~ 23

A0710020 **RTC TC_DOM** **RTC Day-of-month Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Mne													TC_DOM				
Type													RW				
Reset													0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	TC_DOM	TC_DOM	Day-of-month initial value for the time counter The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

A0710024 **RTC TC_DOW** **RTC Day-of-week Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													TC_DOW			
Type													RW			
Reset													0	0	0	

Bit (s)	Mnemonic	Name	Description
2:0	TC_DOW	TC_DOW	Day-of-week initial value for the time counter Range: 1 ~ 7

A0710028 **RTC TC_MTH** **RTC Month Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Mne																	TC_MONTH			
Type																	RW			
Reset																	0	0	0	0

Bit (s)	Mnemonic	Name	Description
3:0	TC_MONTH	TC_MONTH	Month initial value for the time counter Range: 1 ~ 12

A071002C RTC TC_YEA **RTC Year Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	TC_YEAR							
Mne																	RW							
Type																	RW							
Reset																	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
6:0	TC_YEAR	TC_YEAR	Year initial value for the time counter Range: 0 ~ 127 (2000-2127). The software can bias the year as multiples of 4 for the internal leap-year formula. Here are 3 examples: 2000 ~ 2127, 1972 ~ 2099 and 1904 ~ 2031. To simplify the process, the RTC hardware treats all 4-multiple as leap years. If the range you define includes non-leap 4-multiple year (e.g. 2100), please adjust it to the correct date by yourselves. (e.g. change Feb. 29th, 2100 to Mar. 1st, 2100). It is suggested to bias the range to be bigger than 1900 and smaller than 2100 to evade the manual adjustment, i.e. the bias values are suggested to be in the range of [-28,-96], that are (1972~ 2099) ~ (1904~ 2031). The formal leap formula: If year modulo 400 is 0 then leap Else if year modulo 100 is 0 then no_leap Else if year modulo 4 is 0 then leap Else no_leap

A0710030 RTC AL_SEC **RTC Second Alarm Setting Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	AL_SECOND							
Mne																	RW							
Type																	RW							
Reset																	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	AL_SECOND	AL_SECOND	Second value of the alarm counter setting Range: 0 ~ 59

A0710034 RTC AL_MIN **RTC Minute Alarm Setting Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													AL_MINUTE			
Type													RW			
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	AL_MINUTE	AL_MINUTE	Minute value of the alarm counter setting Range: 0 ~ 59

A0710038 [RTC AL HOU](#) **RTC Hour Alarm Setting Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE0								AL_HOUR							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0				0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	NEW_SPARE0	NEW_SPARE0	New spare-register 0
4:0	AL_HOUR	AL_HOUR	Hour value of the alarm counter setting Range: 0 ~ 23

A071003C [RTC AL DOM](#) **RTC Day-of-month Alarm Setting Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE1								AL_DOM							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0				0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	NEW_SPARE1	NEW_SPARE1	New spare-register 1
4:0	AL_DOM	AL_DOM	Day-of-month value of the alarm counter setting The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

A0710040 [RTC AL DO W](#) **RTC Day-of-week Alarm Setting Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE2								AL_DOW							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0						0	0	0

Bit (s)	Mnemonic	Name	Description
---------	----------	------	-------------

Bit (s)	Mnemonic	Name	Description
15:8	NEW_SPARE 2	NEW_SPARE 2	New spare-register 2
2:0	AL_DOW	AL_DOW	Day-of-week value of the alarm counter setting Range: 1 ~ 7

A0710044 **RTC_AL_MTH** **RTC Month Alarm Setting Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE3												AL_MONTH			
Type	RW												RW			
Reset	0	0	0	0	0	0	0	0					0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	NEW_SPARE 3	NEW_SPARE 3	New spare-register 3.
3:0	AL_MONTH	AL_MONTH	Month value of the alarm counter setting. Range: 1 ~ 12

A0710048 **RTC_AL_YEA** **RTC Year Alarm Setting Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE4										AL_YEAR					
Type	RW										RW					
Reset	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	NEW_SPARE 4	NEW_SPARE 4	New spare-register 4
6:0	AL_YEAR	AL_YEAR	Year value of the alarm counter setting Range: 0 ~ 127 (2000-2127)

A071004C **RTC_OSC32C_ON** **OSC32 Control** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	EOSC32_RSV							EMBCK_SEL_MODE		XOSC32_ENB	XOSCCALI					
Type	RW							RW		RO	RW					
Reset	-	0	0	0	0	1	0	0	0	0	-	0	1	1	1	1

Overview: The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

Bit (s)	Mnemonic	Name	Description
15:9	OSC32_RSV	OSC32_RSV	OSC32 reserved bits Keep it at 0x2.
8:6	EMB_MODE	EMBCK_SEL	Mode setting for crystal removal case

Bit (s)	Mnemonic	Name	Description
5	XOSC32_ENB	XOSC32_ENB	XOSC32_ENB Reads pin XOSC32_ENB configuration to know the 32k crystal usage. 0: Use 32k crystal 1: Does not use 32k crystal.
4:0	XOSCCALI	XOSCCALI	Controls XOSC32/EOSC32 calibration If XOSC32_ENB(RTC_OSC32CON[5]) = 0, XOSCCALI controls the bias current for 32k xtal in XOSC32. When powerkeys do not match, the default value is 0x7. If XOSC32_ENB(RTC_OSC32CON[5]) = 1, XOSCCALI is the trimming value for EOSC32. SW needs to find the best trimming value for EOSC32 when the 1 st power-on by frequency meter. When powerkeys do not match, the default value is 0xf.

OSC32CON is not protected by RTC_PROT because RTC_PROT needs 32.768kHz clock to unlock. Similarly, to modify RTC_OSC32CON, writing RTC_WRTGR is not needed, neither. To protect the OSC32 control bits, follow the *update sequence* to update RTC_OSC32CON. After the updating sequence is completed, reload to acquire the internal RTC_OSC32CON. This register needs to be initialized **before** writing powerkeys match.

Update sequence:

- Step 1: Write RTC_OSC32CON = 0x1a57 and wait until CBUSY=0
- Step 2: Write RTC_OSC32CON = 0x2b68 and wait until CBUSY=0
- Step 3: Write the real value you would like to write RTC_OSC32CON and wait until CBUSY=0
- Step 4: Return to step 1 if you need to modify RTC_OSC32CON again.

Note: RTC_OSC32CON should be set before writing POWERKEY1 and POWERKEY2 to the correct value.

A0710050 RTC_POWERKEY
1 **RTC_POWERKEY1 Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_POWERKEY1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	RTC_POWERKEY1	RTC_POWERKEY1	

A0710054 RTC_POWERKEY
2 **RTC_POWERKEY2 Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_POWERKEY2															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	RTC_POWER KEY2	RTC_POWER KEY2	

These register sets are used to determine if the real-time clock has been programmed by the software, i.e. the time value in real time clock is correct. When the real-time clock is first powered on, the register contents are all undefined, and therefore the time values shown are incorrect. The software needs to know if the real-time clock has been programmed. Hence, the two registers are defined to solve this power-on issue. After the software programs the correct value, the two register sets will not need to be updated. In addition to programming the correct time value, when the contents of the register sets are wrong, the interrupt will not be generated. Therefore, the real-time clock will not generate the interrupts before the software programs the registers, and unwanted interrupt due to wrong time value will not occur. The correct values of these two register sets are:

RTC_POWERKEY1 A357h
RTC_POWERKEY2 67D2h

A0710058 RTC_PDN1 PDN1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_PDN1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	RTC_PDN1	RTC_PDN1	Spare registers for software to keep the power-on and power-off state information

A071005C RTC_PDN2 PDN2 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_PDN2															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	RTC_PDN2	RTC_PDN2	Spare registers for software to keep power-on and power-off state information

A0710060 RTC_SPAR0 Spare Register For Specific Purpose 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_SPAR0															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bit (s)	Mnemonic	Name	Description
15:0	RTC_SPAR0	RTC_SPAR0	Reserved for specific purposes

A0710064 RTC_SPAR1 Spare Register For Specific Purpose **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_SPAR1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	RTC_SPAR1	RTC_SPAR1	Reserved for specific purposes

A0710068 RTC_PROT Lock/Unlock Scheme to Prevent RTC Miswriting **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_PROT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
---------	----------	------	-------------

Protects RTC write interface by RTC_PROT

Whether the RTC writing interface is enabled or not is decided by RTC_PROT contents. When RTC_POWERKEY1 & RTC_POWERKEY2 are not equal to the correct values, the RTC writing interface will always be enabled. However, when they match, the user has to perform the unlock flow to enable the writing interface.

15:0	RTC_PROT	RTC_PROT	<p>Unlock flow:</p> <p>Step1: *RTC_PROT=0x586a;</p> <p>Step2: *RTC_WRTGR=1;</p> <p>Step3: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec</p> <p>Step4: *RTC_PROT=0x9136;</p> <p>Step5: *RTC_WRTGR=1;</p> <p>Step6: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec</p>
------	----------	----------	---

Note: Always keep RTC in the unlock state in the power-on mode. Once the normal RTC content writing is completed, DO NOT modify the RTC_PROT content to lock RTC. The RTC_PROT contents will be cleared automatically when being powered off immediately.

A071006C RTC_DIFF One-time Calibration Offset **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RESE RVED	RESE RVED			RTC_DIFF											
Type	RW	RO			RW											
Reset	-	-			0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register is fixed to 0 when RTC_POWERKEY1 and RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
11:0	RTC_DIFF	RTC_DIFF	<p>Adjusts internal counter of RTC</p> <p>It takes effect once and returns to 0 when done. In some cases, RTC is faster or slower than the standard. To change RTC_TC_SEC being coarse may cause alarm problem. RTC_DIFF provides a finer time unit. An internal 15-bit counter accumulates in each 32,768Hz clock. Entering a non-0 value to RTC_DIFF will cause the internal RTC counter to increase or decrease RTC_DIFF when RTC_DIFF changes to 0 again. RTC_DIFF represents 2's complement. For example, if you fill in 0xff into RTC_DIFF, the internal counter will decrease by 1 when RTC_DIFF returns to 0. In other words, you can only use RTC_DIFF continuously if RTC_DIFF is equal to 0 now. <i>Note: RTC_DIFF ranges from 0x800 (-2048) to 0x7fd (2045). 0x7ff & 0x7fe are forbidden.</i></p>

A0710070 **RTC_CALI** **Repeat Calibration Offset** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RESE RVED	RESE RVED	RTC_CALI													
Type	RO	RW	RW													
Reset	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
13:0	RTC_CALI	RTC_CALI	<p>Provides a repeated calibration scheme</p> <p>RTC_CALI provides 7-bit calibration capability in 8-second duration, i.e. 5-bit calibration capability in each second. RTC_CALI represents 2's complement form for the user to adjust RTC increase or decrease. Due to RTC_CALI is revealed in 8 seconds, the resolution is less than a 1/32768 clock. Avg. resolution: 1/32768/8 = 3.81us Avg. adjust range: -0.244~0.240ms/sec in 2's complement: -0x2000 ~ 0x1fff (-8192 ~ 8191)</p>

A0710074 **RTC_WRTGR** **Enable Transfers From Core to RTC in Queue** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Mne																	WRTGR
Type																	WO
Reset																	0

Bit (s)	Mnemonic	Name	Description
0	WRTGR	WRTGR	<p>Enables the transfers from core to RTC</p> <p>After you modify all the RTC registers and would like to change it, write 1 to RTC_WRTGR to trigger the transfer. The prior writing operation is queued in the core power domain. The pending data will not be transferred to the RTC domain until WRTGR = 1.</p> <p>After WRTGR=1, the pending data will be transferred to the RTC domain sequentially in order of register addresses, from low to high. For example: RTC_BBPU -> RTC_IRQ_EN -> RTC_CII_EN -> RTC_AL_MASK -> RTC_TC_SEC -> etc. CBUSY in RTC_BBPU is equal to 1 in the writing process. Observe CBUSY to determine when the transmission is completed.</p>

A0710078 RTC_CON Other RTC Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	LPST_A_RAW	RESE_RVED	RESE_RVED	POWEROF_SF_SEQ_EN	RESE_RVED	RESE_RVED	RESE_RVED	RESE_RVED	RESE_RVED	RESE_RVED	RESE_RVED	RESE_RVED	LPRST	LPEN		VBAT_LPST_A_RAW
Type	W1C	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW
Reset																

Bit (s)	Mnemonic	Name	Description
15	LPSTA_RAW	LPSTA_RAW	<p>Raw status of LP_STA</p> <p>Re-initialize LPD to clear this bit.</p> <p><i>Note: This bit is always high before LPD initialization sequence after the first power-on.</i></p>
3	LPRST	LPRST	<p>Resets LPDETB</p> <p>Only takes effect when LPEN = 1.</p>
2:1	LPEN	LPEN	<p>Enables LPDETB</p> <p>LP initialization sequence:</p> <ol style="list-style-type: none"> Write LPEN = 1, LPRST = 0. Write RTC_WRTGR = 1. wait cbusy down. Write LPEN = 1, LPRST = 1. Write RTC_WRTGR = 1. wait cbusy down. Write LPEN = 1, LPRST = 0. Write RTC_WRTGR = 1. wait cbusy down.
0	VBAT_LPSTA_RAW	VBAT_LPSTA_RAW	<p>Indicates the battery has been in LP state</p> <p>Software needs to clear this bit for the next time use</p> <p>0: VBAT has not been in LP state. 1: VBAT has been in LP state.</p> <p><i>Note: VBAT LP state = VBAT < 2.5V</i></p>

When Vcore always exists, the software can trust the registers and wait for the LP interrupt from RTC if the 32.768 kHz clock stopped or has been stopped.

However, nothing can be trusted after the battery is off (Vrtc may drop). In every boot time, the software checks if LPSTA_RAW = 1. (LP_STA = LPSTA_RAW & LP_IRQ_EN) If true, the RTC contents will no longer be trusted just like powerkeys do not match. You have to initialize the RTC contents in this case.

3.13 Auxiliary ADC Unit

The auxiliary ADC unit is used to monitor the status of the battery and charger, to identify the plugged peripheral, and to perform temperature measurement.

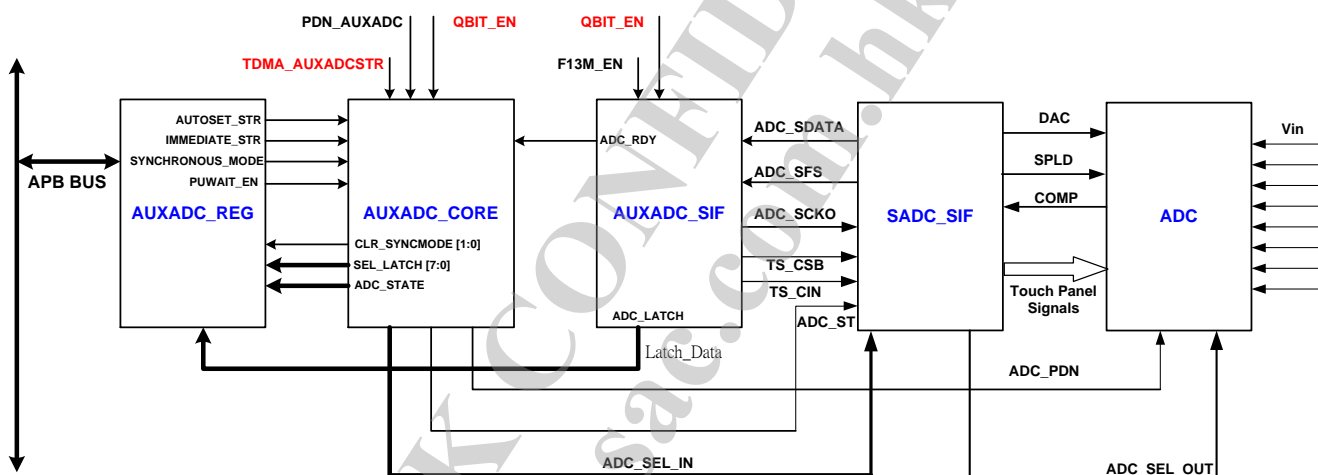


Figure 38. AUXADC architecture

Each channel operates in one of the two modes: immediate mode or timer-triggered mode. The mode of each channel can be individually selected through register AUXADC_CON0. For example, if the flag SYN0 in register AUXADC_CON0 is set, channel 0 will be set in the timer-triggered mode. Otherwise, the channel will operate in the immediate mode.

In the immediate mode, the A/D converter samples the value once only when the flag in the AUXADC_CON1 register is set. For example, if the flag IMM0 in AUXADC_CON1 is set, the A/D converter will sample the data for channel 0. The IMM flags must be cleared and set again to initialize another sampling.

The value sampled for channel 0 is stored in register AUXADC_DAT0, and the value for channel 1 is stored in register AUXADC_DAT1, and so on.

If the AUTOSSET flag in register AUXADC_CON3 is set, the auto-sample function will be enabled. The A/D converter samples the data for the channel in which the corresponding data register is read. For

example, in the case where the SYN1 flag is not set, the AUTOSET flag is set. When the data register AUXADC_DAT0 is read, the A/D converter will sample the next value for channel 1 immediately.

If multiple channels are selected at the same time, the task will be performed sequentially on every selected channel. For example, if AUXADC_CON1 is set to 0x3f, i.e. 6 channels are selected, the state machine in the unit will start sampling from channel 6 to channel 0 and save the values of each input channel in respective registers. The same process also applies to the timer-triggered mode.

In the timer-triggered mode, the A/D converter samples the value for the channels in which the corresponding SYN flags are set when the TDMA timer counts to the value specified in register TDMA_AUXEV1 placed in the TDMA timer. For example, if AUXADC_CON0 is set to 0x3f, the 6 channels will be selected to be in the timer-triggered mode. The state machine will sample the 6 channels sequentially and save the values in registers from AUXADC_DAT0 to AUXADC_DAT5, as it does in the immediate mode.

AUTOCLR_n in register AUXADC_CON3 is set when it is intended to sample only once after setting up the timer-triggered mode. If the AUTOCLR1 flag is set, after the data for the channels in the timer-triggered mode are stored, the SYN_n flags in register AUXADC_CON0 will be cleared.

The uses of the immediate mode and timer-triggered mode are mutually exclusive in terms of individual channels.

There are only two external pins (channel 4 ~ 5) for voltage detection. Other channels (0 ~ 3) are for battery voltage, battery current, charger and battery temperature respectively. Channel 9 is used for audio.

Touch panel

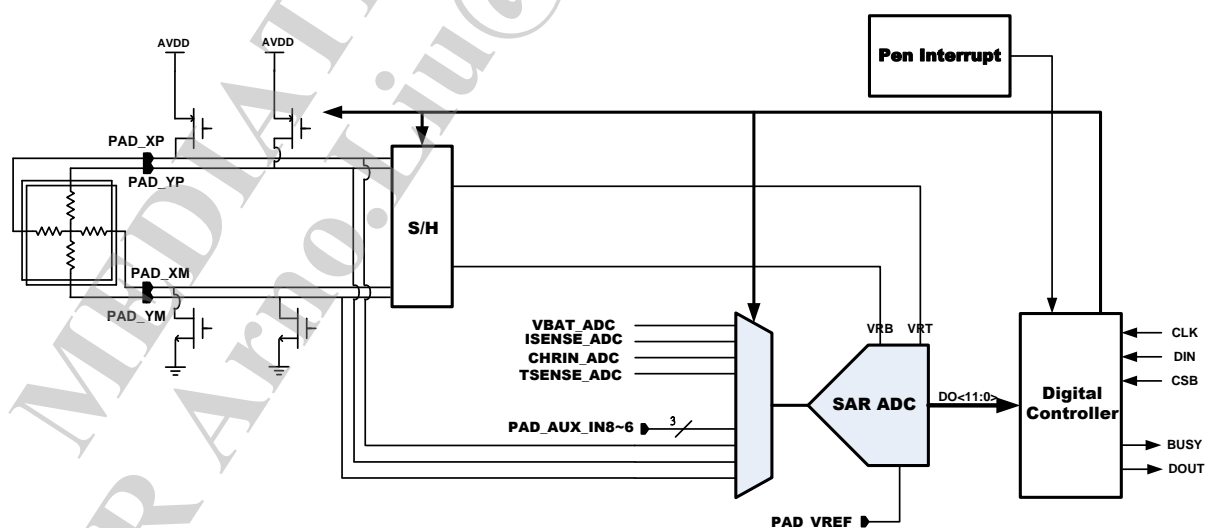


Figure 39. Touch panel circuit structure

Besides the normal sampling of external input voltage, the AUXADC includes the sampling of the touch panel function. For specified axis, the software should program AUX_TS_CMD first then trigger the sample of touch panel in register AUX_TS_CON. The touch panel sampling waveform is shown as the following. After the software polls the status bit in register AUXADC_CON3 to know that the touch panel sample is finished, the software can read back the specified axis value from register AUX_TS_DAT0.

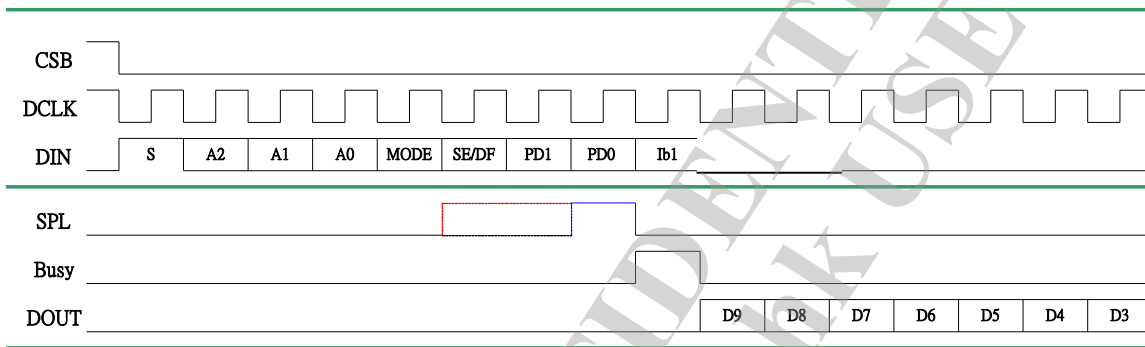


Figure 40. Touch panel sampling waveform

S: Start bit
A2 ~ A0: Addressing bits
Mode: 10-bit or 8-bit
SE/DF: Single end or differential mode
PD1 ~ 0: Power down command

These values are defined in register AUX_TS_CMD. The table below shows the relationship between AUX_TS_CMD and touch panel control signals.

Table 49. Relationship between commands and touch panel control signals

A2	A1	A0	SE/DFB	CHN_SEL	ADC In	X switches	Y switches	+REF	-REF
0	0	1	0	C	X+	OFF	ON	Y+	Y-
0	1	0	0	F	X-	OFF	ON	Y+	Y-
0	1	1	0	C	X+	X+ OFF X- ON	Y+ ON Y- OFF	Y+	X-
1	0	0	0	E	Y-	X+ OFF X- ON	Y+ ON Y- OFF	Y+	X-
1	0	1	0	D	Y+	ON	OFF	X+	X-
1	1	0	0	E	Y-	ON	OFF	X+	X-

Table 2. AUXADC channel description

AuxADC Channel ID	Description
Channel 0	VBAT

AuxADC Channel ID	Description
Channel 1	ISENSE
Channel 2	CHRIN
Channel 3	BATON (BATtemp)
Channel 4	AUXIN4 (external)
Channel 5	ACCDET (external)
Channel 9	ClassAB
Channel 12	XP / External
Channel 13	YP / External
Channel 14	YM / External
Channel 15	XM / External

3.13.1 Register Definition

Module name: AUXADC Base address: (+A0790000h)

Address	Name	Width	Register Function
A0790000	<u>AUXADC_CON0</u>	16	AuxiliaryADC Control Register 0 These bits define whether the corresponding channel is sampled or not in the timer-triggered mode. It is associated with the timing offset register TDMA_AUXEV1. It supports multiple flags. The flags can be automatically cleared after those channels are sampled if AUTOCLR1 in register AUXADC_CON3 is set.
A0790004	<u>AUXADC_CON1</u>	16	AuxiliaryADC Control Register 1 These bits are set individually to sample the data for the corresponding channel. It supports multiple flags.
A079000C	<u>AUXADC_CON3</u>	16	AuxiliaryADC Control Register 3
A0790010	<u>AUXADC_DAT0</u>	16	AuxiliaryADC Channel 0 Register (VBAT)
A0790014	<u>AUXADC_DAT1</u>	16	AuxiliaryADC Channel 1 Register (ISENSE)
A0790018	<u>AUXADC_DAT2</u>	16	AuxiliaryADC Channel 2 Register (CHRIN)
A079001C	<u>AUXADC_DAT3</u>	16	AuxiliaryADC Channel 3 Register (VBATTMP)
A0790020	<u>AUXADC_DAT4</u>	16	AuxiliaryADC Channel 4 Register (External)
A0790024	<u>AUXADC_DAT5</u>	16	AuxiliaryADC Channel 5 Register (External/ACCDET)
A0790034	<u>AUXADC_DAT9</u>	16	AuxiliaryADC Channel 9 Register (ClassAB)
A0790040	<u>AUXADC_DAT12</u>	16	AuxiliaryADC Channel 12 Register (External)
A0790044	<u>AUXADC_DAT13</u>	16	AuxiliaryADC Channel 13 Register (External)
A0790048	<u>AUXADC_DAT14</u>	16	AuxiliaryADC Channel 14 Register (External)
A079004C	<u>AUXADC_DAT15</u>	16	AuxiliaryADC Channel 15 Register (External)
A0790054	<u>AUX_TS_CMD0</u>	16	Touch Screen Sample Command 0
A0790058	<u>AUX_TS_CON</u>	16	Touch Screen Control
A079005C	<u>AUX_TS_DAT0</u>	16	Touch Screen Sample DATA 0
A0790070	<u>AUXADC_DAT_ZCV</u>	16	AuxiliaryADC ZCV Sample DATA
A07900D0	<u>AUXADC_CON4</u>	16	AuxiliaryADC Control Register 4

Address	Name	Width	Register Function
A07900D4	<u>AUX_TS_CMD1</u>	16	Touch Screen Sample Command 1
A07900D8	<u>AUX_TS_DAT1</u>	16	Touch Screen Sample DATA 1

A0790000 **AUXADC_CON0** AuxiliaryADC Control Register 0 **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SYN15	SYN14	SYN13	SYN12			SYN9				SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
Type	R/W	R/W	R/W	R/W			R/W				R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0			0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	SYN15	SYN15	Channel 15 sync mode 0: The channel is not selected. 1: The channel is selected.
14	SYN14	SYN14	Channel 14 sync mode 0: The channel is not selected. 1: The channel is selected.
13	SYN13	SYN13	Channel 13 sync mode 0: The channel is not selected. 1: The channel is selected.
12	SYN12	SYN12	Channel 12 sync mode 0: The channel is not selected. 1: The channel is selected.
9	SYN9	SYN9	Channel 9 sync mode 0: The channel is not selected. 1: The channel is selected.
5	SYN5	SYN5	Channel 5 sync mode 0: The channel is not selected. 1: The channel is selected.
4	SYN4	SYN4	Channel 4 sync mode 0: The channel is not selected. 1: The channel is selected.
3	SYN3	SYN3	Channel 3 sync mode 0: The channel is not selected. 1: The channel is selected.
2	SYN2	SYN2	Channel 2 sync mode 0: The channel is not selected. 1: The channel is selected.
1	SYN1	SYN1	Channel 1 sync mode 0: The channel is not selected. 1: The channel is selected.
0	SYN0	SYN0	Channel 0 sync mode 0: The channel is not selected. 1: The channel is selected.

A0790004 **AUXADC_CON1** AuxiliaryADC Control Register 1 **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IMM15	IMM14	IMM13	IMM12			IMM9				IMM5	IMM4	IMM3	IMM2	IMM1	IMM0
Type	R/W	R/W	R/W	R/W			R/W				R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0			0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
--------	----------	------	-------------

15	IMM15	IMM15	Channel 15 immediate mode 0: The channel is not selected. 1: The channel is selected.
14	IMM14	IMM14	Channel 14 immediate mode 0: The channel is not selected. 1: The channel is selected.
13	IMM13	IMM13	Channel 13 immediate mode 0: The channel is not selected. 1: The channel is selected.
12	IMM12	IMM12	Channel 12 immediate mode 0: The channel is not selected. 1: The channel is selected.
9	IMM9	IMM9	Channel 9 immediate mode 0: The channel is not selected. 1: The channel is selected.
5	IMM5	IMM5	Channel 5 immediate mode 0: The channel is not selected. 1: The channel is selected.
4	IMM4	IMM4	Channel 4 immediate mode 0: The channel is not selected. 1: The channel is selected.
3	IMM3	IMM3	Channel 3 immediate mode 0: The channel is not selected. 1: The channel is selected.
2	IMM2	IMM2	Channel 2 immediate mode 0: The channel is not selected. 1: The channel is selected.
1	IMM1	IMM1	Channel 1 immediate mode 0: The channel is not selected. 1: The channel is selected.
0	IMM0	IMM0	Channel 0 immediate mode 0: The channel is not selected. 1: The channel is selected.

A079000C AUXADC_CON3 Auxiliary ADC Control Register 3 0010

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AUTO SET				RSV		AUTO CLR1		SOFT _RST			BYPA SS_SL P_ZC V_TRI GGER				AUXA DC_S TA
Type	R/W				R/W		R/W		R/W			R/W				RO
Reset	0				0		0		0			1				0

Bit(s)	Mnemonic	Name	Description
15	AUTOSET	AUTOSET	Defines the auto-sample mode of the module. In auto-sample mode, each channel with its sample register being read can start sampling immediately without configuring the control register AUXADC_CON1 again.
11	RSV	RSV	Please keep 1'b0
9	AUTOCLR1	AUTOCLR1	Defines the auto-clear mode of the module for event 1. In the auto-clear mode, each timer-triggered channel acquires samples of specified channels once the SYNn bit in register AUXADC_CON0 is set. The SYNn bits are automatically cleared and the channel is not enabled again by the timer event except when the SYNn flags are set again. 0: The automatic clear mode is not enabled. 1: The automatic clear mode is enabled.

7	SOFT_RST SOFT_RST	Software reset AUXADC state machine 0: Normal function 1: Reset AUXADC state machine
4	BYPASS_SL BYPASS_SLP_ZCV P_ZCV_TRI _TRIGGER GGER	Bypass zcv triggering after sleep mode setting 0: trigger zcv measuring after sleep mode 1: bypass zcv measuring trigger after sleep mode
0	AUXADC_S AUXADC_STA TA	Defines the state of the module 0: This module is idle. 1: This module is busy.

A0790010 AUXADC_DAT0 AuxiliaryADC Channel 0 Register (VBAT) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT0															
Type	RO															
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT0	DAT0	Sampled data for channel0

A0790014 AUXADC_DAT1 AuxiliaryADC Channel 1 Register (ISENSE) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT1															
Type	RO															
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT1	DAT1	Sampled data for channel1

A0790018 AUXADC_DAT2 AuxiliaryADC Channel 2 Register (CHRIN) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT2															
Type	RO															
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT2	DAT2	Sampled data for channel2

A079001C AUXADC_DAT3 AuxiliaryADC Channel 3 Register (VBATTMP) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT3															
Type	RO															
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT3	DAT3	Sampled data for channel3

A0790020 AUXADC_DAT4 AuxiliaryADC Channel 4 Register (External) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT4									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT4	DAT4	Sampled data for channel4

A0790024 **AUXADC_DAT5** AuxiliaryADC Channel 5 Register (External/ACCDET) **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT5									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT5	DAT5	Sampled data for channel5

A0790034 **AUXADC_DAT9** AuxiliaryADC Channel 9 Register (ClassAB) **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT9									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT9	DAT9	Sampled data for channel9

A0790040 **AUXADC_DAT1** AuxiliaryADC Channel 12 Register (External) **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT12									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT12	DAT12	Sampled data for channel12

A0790044 **AUXADC_DAT1** AuxiliaryADC Channel 13 Register (External) **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT13									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT13	DAT13	Sampled data for channel13

A0790048 AUXADC_DAT1 AuxiliaryADC Channel 14 Register (External) **0000**
4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT14															
Type	RO															
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT14	DAT14	Sampled data for channel14

A079004C AUXADC_DAT1 AuxiliaryADC Channel 15 Register (External) **0000**
5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT15															
Type	RO															
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT15	DAT15	Sampled data for channel15

A0790054 AUX_TS_CMD0 Touch Screen Sample Command 0 **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_S PL	TS_MAGIC_KEY						ADDRESS			MODE	SEDF	PD			
Type	R/W	WO						R/W			R/W	R/W	R/W			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	TS_SPL	TS_SPL	Touch screen sample trigger For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.
14:7	TS_MAGIC_KEY	TS_MAGIC_KEY	The TS commands in AUX_TS_CMD0 can only take effect when the TS_MAGIC_KEY matches the correct value. TS_MAGIC_KEY=0xaa .
6:4	ADDRESS	ADDRESS	Defines which x or y or z data will be sampled 001: Y position 011: Z1 position 100: Z2 position 101: X position Others: Reserved
3	MODE	MODE	Selects sample resolution 0: 10-bit 1: 8-bit
2	SEDF	SEDF	Selects mode 0: Differential mode 1: Single-end mode
1:0	PD	PD	Power-down control for analog IRQ signal and touch screen sample control signal

00: Turn on Y_drive signal and PDN_sh_ref
 01: Turn on PDN_IRQ and PDN_sh_ref
 10: Reserved
 11: Turn on PDN_IRQ

A0790058 AUX_TS_CON Touch Screen Control **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																TS_SPL
Type																R/W
Reset																0

Bit(s)	Mnemonic	Name	Description
0	TS_SPL	TS_SPL	Touch screen sample trigger For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.

A079005C AUX_TS_DAT0 Touch Screen Sample DATA 0 **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL															TS_DAT
Type	RO															RO
Reset	0						0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	TS_SPL	TS_SPL	Touch screen sample trigger For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status"
9:0	TS_DAT	TS_DAT	Touch screen sample result data

A0790070 AUXADC_DAT_ZCV AuxiliaryADC ZCV Sample DATA **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																DAT_ZCV
Type																RO
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT_ZCV	DAT_ZCV	Sampled data for ZCV

A07900D0 AUXADC_CON4 AuxiliaryADC Control Register 4 **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL	AUXADC_STA					SYN9				SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
Type	RO	RO					RO				RO	RO	RO	RO	RO	RO
Reset	0	0					0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	TS_SPL	TS_SPL	Touch screen sample trigger For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status" action 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.
14	AUXADC_STA	AUXADC_STA	Defines the state of the module 0: This module is idle. 1: This module is busy.
9	SYN9	SYN9	Channel 9 sync mode 0: The channel is not selected. 1: The channel is selected.
5	SYN5	SYN5	Channel 5 sync mode 0: The channel is not selected. 1: The channel is selected.
4	SYN4	SYN4	Channel 4 sync mode 0: The channel is not selected. 1: The channel is selected.
3	SYN3	SYN3	Channel 3 sync mode 0: The channel is not selected. 1: The channel is selected.
2	SYN2	SYN2	Channel 2 sync mode 0: The channel is not selected. 1: The channel is selected.
1	SYN1	SYN1	Channel 1 sync mode 0: The channel is not selected. 1: The channel is selected.
0	SYN0	SYN0	Channel 0 sync mode 0: The channel is not selected. 1: The channel is selected.

A07900D4 AUX_TS_CMD1 Touch Screen Sample Command 1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										ADDRESS			MODE	SEDF	PD	
Type										R/W			R/W	R/W	R/W	
Reset										0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
6:4	ADDRESS	ADDRESS	Defines which x or y or z data will be sampled 001: Y position 011: Z1 position 100: Z2 position 101: X position Others: Reserved
3	MODE	MODE	Selects sample resolution 0: 10-bit 1: 8-bit
2	SEDF	SEDF	Selects mode 0: Differential mode 1: Single-end mode
1:0	PD	PD	Power-down control for analog IRQ signal and touch screen sample control signal

00: Turn on Y-drive signal and PDN_sh_ref
 01: Turn on PDN_IRQ and PDN_sh_ref
 10: Reserved
 11: Turn on PDN_IRQ

A07900D8 AUX_TS_DAT1 Touch Screen Sample DATA 1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							TS_DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	TS_DAT	TS_DAT	Touch screen sample result data

※ Please refer to “PMU” datasheet for other AuxADC related settings:

- Sampling cycle control: Please see “AUX_CON4” and “AUX_CON5”

3.13.2 General Programming Guide

All register writes will occur in sequence. However, due to synchronization time, any reads after writes need to be delayed by three dummy reads.

3.13.3 Usage Programming Guide

There are two modes to program the AUXADC to sample: immediate mode and synchronous mode. The following are notes on programming each mode:

Immediate mode sampling is accomplished by programming AUXADC_CON1 with the channels to be sampled. After programming, it is necessary to perform three dummy reads on AUXADC_CON3. After the dummy reads, the next read of AUXADC_CON3 will be valid. After sampling is done, it is necessary to program AUXADC_CON1 back to zero before sampling again.

Synchronous mode sampling is accomplished by programming AUXADC_CON0 with the channels to be sampled. Then it is necessary to program TDMA_EVTENA7 to 0x2. After the sample is done, TDMA_EVTENA7 must be programmed to 0x0 with waiting of two frames before sampling again.

3.13.4 Performance Programming Guide

For details on adjusting the performance of ADC sampling, please refer to registers AUX_CON4, AUX_CON5 and AUX_CON6 in the PMU datasheet.

3.13.5 AUXADC PDN

AUXADC is located in A-die. Due to limitation, one of UART1,2,3 clocks must be turned on in order to use AUXADC. Please clear the corresponding PDN bits to enable the AUXADC clock (bit 2 or 3 of ACFG_CLK_CG).

3.13.6 Notice

The 4 TP pins – PAD_XP, PAD_XM, PAD_YP, PAD_YM, are used as EINT pins when no R-touch is used in the system. Therefore, please make sure the pin settings are correct.

3.14 USB Device Controller

3.14.1 General Description

This chip provides a USB function interface which complies with Universal Serial Bus Specification Rev 1.1. The USB device controller supports only full-speed (12Mbps) operation. The chip can make use of this widely available USB interface to transmit/receive data with USB hosts, typically PC/laptop. There are 6 endpoints in the USB device controller besides the mandatory control endpoint, 4 of which are for IN transactions and 2 endpoints are for OUT transactions. Word, half-word, and byte access are all allowed for loading and unloading the FIFO. The controller features 4 DMA channels to accelerate the data transfer for ACL and SCO data streams. The features of the endpoints are:

1. Endpoint 0: The control endpoint feature 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. DMA transfer is not supported.
2. IN endpoint 1: It features 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. GDMA Channel 4 Write Transfer is supported.
3. IN endpoint 2: It features 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. GDMA Channel 6 Write Transfer is supported.
4. IN endpoint 3: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.
5. IN endpoint 4: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.

6. OUT endpoint 1: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Channel 5 Read Transfer is supported.
7. OUT endpoint 2: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Channel 7 Read Transfer is supported.

For each endpoint except for the control endpoint, when the packet size is smaller than half the size of the FIFO, at most 2 packets can be buffered.

This unit is highly software configurable. All endpoints except for the control endpoint can be configured to be a bulk, interrupt or isochronous endpoints. Composite devices are also supported. IN endpoint 1 and OUT endpoint 1 share the same endpoint number but they can be used separately. So is the situation for the endpoint 2.

The USB device uses the cable-powered feature for the transceiver but only drains little current. An internal pull-up resistor is integrated across Vbus and D+ signal. The switch on/off of the pull-up resistor can be configured through the internal register. Two additional external serial resistors might be needed to place on the output of D+ and D- signals to make the output impedance equivalent to 28~44Ohm.

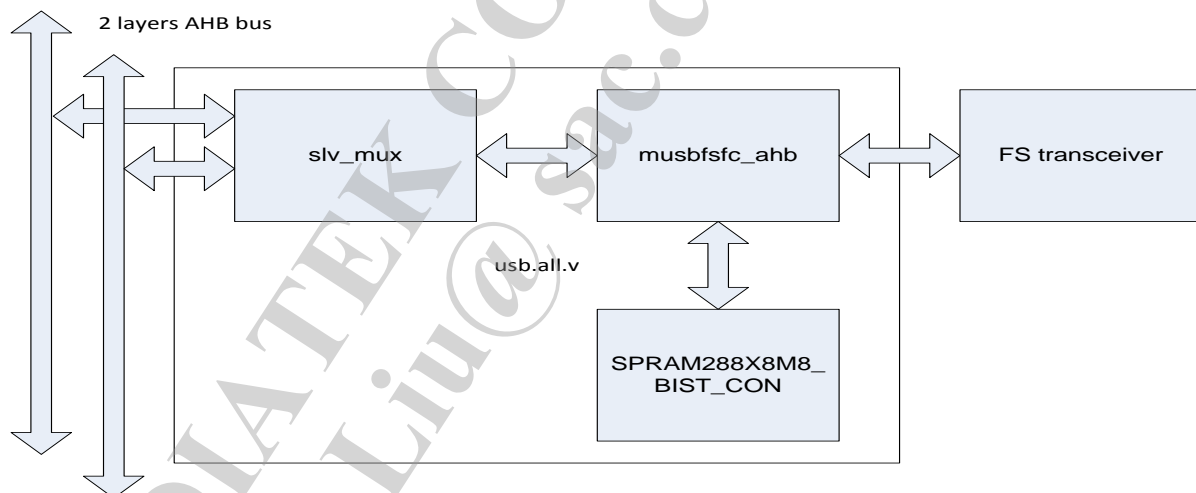


Figure 1. USB11 controller system diagram

3.14.2 Terminology

RW: Writable, Readable.

RO: Read-only. Value never changes.

WO: Write-only.

W1: Write-once. Readable.

RU: Read-only but value updated by the design.

W1C: Readable. Write 1 to bitwise-clear.

RC: Clear on read.

A1: Auto-set by the design. Can be read and write 0 to clear.

A0: Auto-cleared by the design. Can be read and write 1 to set.

DC: Don't care.

OTHER: Others. Mixed attribute. Refer to bit description.

RSV: Reserved. The read/write behavior to this bit is undefined.

3.14.3 Register Definition

00	<u>USB_FADDR</u> USB Function Address Register														00	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									UPD							FADDR
Type									RW							RW
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	UPD	This is an 8bit register that should be written with the functions 7-bit address (received through a SET_ADDRESS description). It is then used for decoding the function address in subsequent token packets. When set by the MCU, the core will wait for an SOF token from the time INPKTRDY is set before sending the packet
6:0	FADDR	Function address of device

01	<u>USB_POWER</u> USB Power Control Register														00	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ISOU PDAT E			SWR STEN AB	RESE T	RESU ME	SUSP MOD E	SUSP ENA B
Type									RW			RU	RU	RW	RU	RW
Reset									0			0	0	0	0	0

Bit(s)	Name	Description
7	ISOUPDATE	When set by the MCU, the core will wait for an SOF token from the time INPKTRDY is set before sending the packet
4	SWRSTENAB	Set by the MCU to enable the mode in which the device can only be reset by the software after reset signals are detected on the bus. In case the software is delayed by other high priority processes and cannot make it to read the command from the buffer before the hardware reset the device after the reset signal is detected on the bus, the command will be lost. That is why the software reset mode is effective. When the flag is enabled, the hardware state machine cannot reset itself but by the software. In that sense, the software and hardware can keep synchronous detecting the reset signal.
3	RESET	The read-only bit is set when Reset signaling is present on the bus
2	RESUME	Set by the MCU to generate Resume signaling when the function is in the suspend mode. The MCU should clear this bit after 10ms (maximum 15ms) to end Resume signaling
1	SUSPMODE	Set by the USB core when the Suspend mode is entered Cleared when the Resume bit of this register is set.
0	SUSPENAB	Set by the MCU to enable device into the Suspend mode when Suspend

Bit(s)	Name	Description
		signaling is received on the bus

0002 USB_INTRIN USB IN Endpoints Interrupt Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												EP4_IN	EP3_IN	EP2_IN	EP1_IN	EP0
Type												RC	RC	RC	RC	RC
Reset												0	0	0	0	0

Bit(s)	Name	Description
4	EP4_IN	IN Endpoint 4 interrupt event
3	EP3_IN	IN Endpoint 3 interrupt event
2	EP2_IN	IN Endpoint 2 interrupt event
1	EP1_IN	IN Endpoint 1 interrupt event
0	EP0	Endpoint 0 interrupt event

0004 USB_INTROUT USB OUT Endpoints Interrupt Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														EP2_OUT	EP1_OUT	
Type														RC	RC	
Reset														0	0	

Bit(s)	Name	Description
2	EP2_OUT	OUT Endpoint 2 interrupt event
1	EP1_OUT	OUT Endpoint 1 interrupt event

06 USB_INTRUSB USB General Interrupt Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												POWERDWN	SOF	RESET	RESUME	SUSPEND
Type												RC	RC	RC	RC	RC
Reset												0	0	0	0	0

Bit(s)	Name	Description
4	POWERDWN	Set at SUSPMODE and LineState is JState. The programmer should have de bounce scheme in SW code when using this interrupt.
3	SOF	Set at the start of each frame
2	RESET	Set when Reset signaling is detected on the bus
1	RESUME	Set when Resume signaling is detected on the bus while the USB core is

Bit(s)	Name	Description
0	SUSPEND	in suspend mode Set when Suspend signaling is detected on the bus

0007 **USB_INTRINE** USB IN Endpoints Interrupt Enable Register 00FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												EP4_INE	EP3_INE	EP2_INE	EP1_INE	EP0_E
Type												RW	RW	RW	RW	RW
Reset												1	1	1	1	1

Bit(s)	Name	Description
4	EP4_INE	1b0: Disable IN Endpoint 4 interrupt event 1b1: Enable IN Endpoint 4 interrupt event
3	EP3_INE	1b0: Disable IN Endpoint 3 interrupt event 1b1: Enable IN Endpoint 3 interrupt event
2	EP2_INE	1b0: Disable IN Endpoint 2 interrupt event 1b1: Enable IN Endpoint 2 interrupt event
1	EP1_INE	1b0: Disable IN Endpoint 1 interrupt event 1b1: Enable IN Endpoint 1 interrupt event
0	EP0_E	1b0: Disable IN Endpoint 0 interrupt event 1b1: Enable IN Endpoint 0 interrupt event

0009 **USB_INTRROUTE** USB OUT Endpoints Interrupt Enable Register 00FE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														EP2_OUTE	EP1_OUTE	
Type														RW	RW	
Reset														1	1	

Bit(s)	Name	Description
2	EP2_OUTE	1b0: Disable OUT Endpoint 2 interrupt event 1b1: Enable OUT Endpoint 2 interrupt event
1	EP1_OUTE	1b0: Disable OUT Endpoint 1 interrupt event 1b1: Enable OUT Endpoint 1 interrupt event

0B **INTRUSBE** USB General Interrupt Enable Register 06

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												POWERDWN_E	SOF_E	RESET_E	RESUME_E	SUSPEND_E
Type												RW	RW	RW	RW	RW
Reset												0	0	1	1	0

Bit(s)	Name	Description
4	POWERDWN_E	Enables power-down interrupt

Bit(s)	Name	Description
3	SOF_E	Enables SOF interrupt
2	RESET_E	Enables reset/babble interrupt
1	RESEUM_E	Enables resume interrupt
0	SUSPEND_E	Enables suspend interrupt

0C **USB_FRAME1** **USB Frame Count #1 Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														NUML		
Type																
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	NUML	The lower 8 bits of the frame number

0D **USB_FRAME2** **USB Frame Count #2 Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														NUMH		
Type																
Reset														0	0	0

Bit(s)	Name	Description
2:0	NUMH	The upper 3 bits of the frame number

0E **USB_INDEX** **USB Endpoint Register Index** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														INDEX		
Type																
Reset													0	0	0	0

The register determines which endpoint control/status registers are to be accessed at addresses USB+10h to USB+17h. Each IN endpoint and OUT endpoint has its own set of control/status registers. Only one set of IN control/status and one set of OUT control/status registers appear in the memory map at a time. Before accessing the control/status registers of an endpoint, the endpoint number should be written to the USB_INDEX register to ensure that the correct control/status registers appear in the memory map.

Bit(s)	Name	Description
3:0	INDEX	Index of the endpoint

0F **USB_RSTCTRL** **USB Reset Control** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SWR ST					RSTCNR		
Type									RW							
Reset																

Reset									0					0	0	0	0
-------	--	--	--	--	--	--	--	--	---	--	--	--	--	---	---	---	---

Bit(s)	Name	Description
7	SWRST	If the flag SWRSTENAB in register USB_POWER is set to 1, the software enable mode will be enabled, and the device can be reset by writing 1 to this flag.
3:0	RSTCNTR	Signifies the duration of the reset operation after reset signal is detected on the bus. It is only enabled when software reset is not enabled. If the value is 0, the duration will be 2.5us. Otherwise, the duration will be this value multiplied by 341 then added by 2.5 in us unit. The range consequently starts from 2.5us to 5122.5us.

11 USB_EP0_CSR USB Control/Status Register for Endpoint 0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SSETUPEN D	SOUTPKTR DY	SENDSTAL L	SETUPEND	DATAEND	SENTSTALL	INPKTRDY	OUTPKTRDY
Type									A0	A0	A0	RU	A0	A1	A0	RU
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	SSETUPEND	The MCU writes 1 to this bit to clear the SETUPEND bit. It is cleared automatically. Only active when a transaction is started.
6	SOUTPKTRDY	The MCU writes 1 to this bit to clear the OUTPKTRDY bit. It is cleared automatically. Only active when an OUT transaction is started
5	SENDSTALL	The MCU writes 1 to this bit to terminate the current transaction. The STALL handshake will be transmitted, and this bit will be cleared automatically.
4	SETUPEND	This bit will be set when a control transaction ends before the DATAEND bit is set. An interrupt will be generated and FIFO flushed at this time. The bit is cleared by the MCU writing 1 to the SSETUPEND bit.
3	DATAEND	The MCU sets this bit: 1. When setting INPKTRDY for the last data packet. 2. When clearing OUTPKTRDY after unloading the last data packet. 3. When setting INPKTRDY for a zero length data packet. It is cleared automatically.
2	SENTSTALL	This bit is set when a STALL handshake is transmitted. The MCU should clear this bit by writing 0 to it.
1	INPKTRDY	The MCU sets up this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet is transmitted. An interrupt is generated when this bit is set
0	OUTPKTRDY	This bit is set when a data packet is received. An interrupt is generated when this bit is set. The MCU clears this bit by setting up the SOUTPKTRDY bit.

16 USB_EP0_COUNT_I EP0 Received Bytes Count Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Name																	EP0_COUNT										
Type																	RU										
Reset																	0	0	0	0	0	0	0	0	0	0	0

The register indicates the number of received data bytes in the endpoint 0. The value returned is valid while the OUTPKTRDY bit of the USB_EP0_CSR register is set. The register is active when the USB_INDEX register is set to 0.

Bit(s)	Name	Description
6:0	EP0_COUNT	Number of received data bytes in the endpoint 0

10 **USB EP_INMAXP** USB Maximum Packet Size Register for IN Endpoint **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
Name																	MAXP										
Type																	RW										
Reset																	0	0	0	0	0	0	0	0	0	0	0

The register holds the maximum packet size for transactions through the currently selected IN endpoint - in units of one byte. When setting up the value, the programmer should note the constraints placed by the USB Specification on the packet size for bulk interrupt and isochronous transactions in full speed operation. There is an INMAXP register for each IN endpoint except for endpoint 0. The registers are active when the USB_INDEX register is set to 1, 2, 3, and 4 respectively.

The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch may cause unexpected results. If a value is bigger than the configured IN FIFO size for the endpoint written to the register, the value will be automatically changed to the IN FIFO size. If the value written to the register is smaller than or equal to half the IN FIFO size, two IN packets can be buffered. The configured IN FIFO size for the endpoint 1, 2, and 3, are 16 bytes, 64 bytes, and 64 bytes respectively.

The register is reset to 0. If the register is changed after the packets are sent from the endpoint, the endpoint IN FIFO should be completely flushed after writing the new value to the register.

Bit(s)	Name	Description
7:0	MAXP	Maximum packet size (unit: byte)

11 **USB EP_INCSR1** USB Control/Status Register #1 for IN Endpoint **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ABORTPKT_EN	CLRDATA_OG	SENTSTAL_L	SENTSTAL_L	FLUSHFIFO	UNDERRUN	FIFOEMPTY	INPKTRDY
Type									RW	A0	A1	RW	A0	A1	RU	A0
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	ABORTPKT_EN	When MCU writes 1 to ABORTPKT_EN, FLUSHFIFO switches to abort the packet function. This bit should be enabled before FLUSHFIFO is set. If FLUSHFIFO is set and ABORTPKT_EN is enabled the data loaded into FIFO will be discarded. After

Bit(s)	Name	Description
		the packet is aborted, EP will issue an interrupt. The programmer should wait for this interrupt to make sure the packet is aborted
6	CLRDATATOG	The MCU writes 1 to this bit to reset the endpoint IN data toggle to 0
5	SENTSTALL	The bit is set when a STALL handshake is transmitted. The FIFO is flushed and the INPKTRDY bit is cleared. The MCU should clear this bit by writing 0 to this bit.
4	SENDSTALL	The MCU writes 1 to this bit to issue a STALL handshake to an IN token. The MCU clears this bit to terminate the stall condition
3	FLUSHFIFO	The MCU writes 1 to this bit to flush the next packet to be transmitted from the endpoint IN FIFO. The FIFO pointer is reset and the INPKTRDY bit is cleared. If the FIFO contains two packets, FLUSHFIFO will need to be set twice to completely clear the FIFO. FLUSHFIFO should only be used when INPKTRDY is set. At other times, it may cause data corruption. If ABORTPKT_EN is enabled and this bit is set, the function of this bit will become ABORTPKT to abort the next packet to be transmitted from the endpoint IN FIFO and does not need to set INPKTRDY. It is the same with the FLUSHFIFO function. This bit is only active when the endpoint is idle.
2	UNDERRUN	In isochronous mode, this bit is set when a zero length data packet is sent after receiving an IN token with the INPKTRDY bit not set. In Bulk/Interrupt mode, this bit is set when a NAK is returned in response to an IN token. The MCU should clear this bit by writing a 0 to this bit
1	FIFONOTEMPTY	This bit is set when there is at least 1 packet in the IN FIFO
0	INPKTRDY	The MCU sets up this bit after loading a data packet into the FIFO. Only active when an IN transaction is started. It is cleared automatically when a data packet is transmitted. An interrupt will be generated (if enabled) when the bit is cleared

12 **USB EP INCSR2** USB Control/Status Register #2 for IN Endpoint **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTO SET	ISO	MODE	DMA ENAB	FRCDATATOG			
Type									RW	RW	RW	RW	RW			
Reset									0	0	0	0	0			

Bit(s)	Name	Description
7	AUTOSET	If the MCU sets up the bit, INPKTRDY will be automatically set when the data of the maximum packet size (value in INMAXP) is loaded into the IN FIFO. If a packet of smaller than the maximum packet size is loaded, INPKTRDY will have to be set manually. When 2 packets are in the IN FIFO, INPKTRDY will also be automatically set when the first packet is sent if the second packet is the maximum packet size
6	ISO	The MCU sets up this bit to enable the IN endpoint for isochronous transfer and clears it to enable the IN endpoint for bulk/interrupt transfers
5	MODE	The MCU sets up this bit to enable the endpoint direction as IN and clears it to enable the endpoint direction as OUT. It is valid only when the same endpoint FIFO is used for both IN and OUT transactions.
4	DMAENAB	The MCU sets up this bit to enable the DMA request for the IN endpoint.
3	FRCDATATOG	The MCU sets up this bit to force the endpoints IN data toggle to switch after each data packet is sent regardless of whether an ACK has been received. This can be used by interrupt IN endpoints which are used to communicate

Bit(s)	Name	Description
		rate feedback for isochronous endpoints

13 [USB_EP_OUTMAXP](#) USB Maximum Packet Size Register for OUT Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MAXP							
Type									RW							
Reset									0	0	0	0	0	0	0	0

This register holds the maximum packet size for transactions through the currently selected OUT endpoint (unit: byte). When setting up this value, the programmer should note the constraints placed by the USB specification on packet sizes for bulk, interrupt, and isochronous transactions in full speed operations. There is an OUTMAXP register for each OUT endpoint except for endpoint 0. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch may cause unexpected results. The total amount of data represented by the value written to this register must not exceed the FIFO size for the OUT endpoint and should not exceed half the FIFO size if double buffering is required. If a value bigger than the configured OUT FIFO size for the endpoint is written to the register, the value will be automatically changed to the OUT FIFO size. If the value written to the register is smaller than or equal to half the OUT FIFO size, two OUT packets can be buffered. The configured IN FIFO size for the endpoint 1, 2, and 3 are both 16, 64, and 64 bytes, respectively.

Bit(s)	Name	Description
7:0	MAXP	Maximum packet size (unit: byte)

14 [USB_EP_OUTCSR1](#) USB Control/Status Register #1 for OUT Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CLRDTATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERROR	OVERRUN	FIFOFULL	RXPKTRDY
Type									A0	A1	RW	A0	RU	A1	RU	A1
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	CLRDTATOG	The MCU writes 1 to this bit to reset the endpoint data toggle to 0.
6	SENTSTALL	The bit is set when a STALL handshake is transmitted. The MCU should clear this bit by writing 0 to it.
5	SENDSTALL	The MCU writes 1 to this bit to issue a STALL handshake. The MCU clears this bit to terminate the stall condition. This bit will have no effect if the OUT endpoint is in the isochronous mode
4	FLUSHFIFO	The MCU writes 1 to this bit to flush the next packet to be read from the endpoint OUT FIFO. If the FIFO contains two packets, FLUSHFIFO will need to be set twice to completely clear the FIFO. FLUSHFIFO should only be used when OUTPKTRDY is set. At other times, it may cause data corruption.
3	DATAERROR	The bit is set when OUTPKTRDY is set if the data packet has a CRC or bit

Bit(s)	Name	Description
		stuff error.
2	OVERRUN	It is cleared when OUTPKTRDY is cleared. This bit is only valid in the isochronous mode. The bit will be set if an OUT packet cannot be loaded into the OUT FIFO. The MCU should clear the bit by writing 0 to it. This bit is only valid in the isochronous mode.
1	FIFOFULL	This bit is set when no more packets can be loaded into the OUT FIFO
0	RXPKTRDY	The bit is set when a data packet has been received. The MCU should clear (write 0 to it) the bit when the packet is unloaded from the OUT FIFO. An interrupt will be generated when the bit is set. When the receiving null packet is received, OUTPKTRDY will be set after USB_INTROUT1 is high.

15 [USB EP OUTCS R2](#) USB Control/Status Register #2 for OUT Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTO CLEAR	ISO	DMA ENAB	DMA MODE				
Type									RW	RW	RW	RW				
Reset									0	0	0	0				

Bit(s)	Name	Description
7	AUTOCLEAR	If the MCU sets up this bit, the OUTPKTRDY bit will be automatically cleared when a packet of OUTMAXP bytes is unloaded from the OUT FIFO. When packets of smaller than the maximum packet size are unloaded, OUTPKTRDY will have to be cleared manually.
6	ISO	The MCU sets up this bit to enable the OUT endpoint for isochronous transfers and clears it to enable the OUT endpoint for bulk/interrupt transfers
5	DMAENAB	The MCU sets up this bit to enable the DMA request for the OUT endpoint
4	DMAMODE	Two modes of DMA operation are supported: DMA mode 0 in which a DMA request is generated for all received packets, together with an interrupt (if enabled). DMA mode 1 in which a DMA request (but no interrupt) is generated for OUT packets of size OUTMAXP bytes and an interrupt (but no DMA request) is generated for OUT packets of any other size. The MCU sets up the bit to select DMA mode 1 and clears this bit to select DMA mode 0.

16 [USB EP COUNT 1](#) USB OUT Endpoint Byte Counter Register LSB Part for Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name									NUML											
Type									RU											
Reset									0	0	0	0	0	0	0	0	0			

The register holds the lower 8 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY in register USB_OUTCSR1 is set. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

Bit(s)	Name	Description
--------	------	-------------

Bit(s)	Name	Description
7:0	NUML	Lower 8 bits of the number of received data bytes for the OUT endpoint

17 [USB EP COUNT 2](#) USB OUT Endpoint Byte Counter Register MSB Part for Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															NUMH	
Type															RU	
Reset														0	0	0

The register holds the upper 3 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY in register USB_EP_OUTCSR1 is set. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

Bit(s)	Name	Description
2:0	NUMH	Upper 8 bits of the number of received data bytes for the OUT endpoint.

20 [USB EP0 FIFO DB0](#) USB Endpoint 0 FIFO Register DB0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO0_DB0				
Type												Other				
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 0. Writing to this register will load data to the FIFO for the endpoint 0. Reading this register unloads data from the FIFO for the endpoint 0.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO0_DB0	The first byte to be loaded to or unloaded from the FIFO.

21 [USB EP0 FIFO DB1](#) USB Endpoint 0 FIFO Register DB1 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO0_DB1				
Type												Other				
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO0_DB1	The second byte to be loaded to or unloaded from the FIFO.

22 [USB EP0 FIFO](#) USB Endpoint 0 FIFO Register DB2 00

DB2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO0_DB2															
Type	Other															
Reset	0 0 0 0 0 0 0 0 0															

Bit(s)	Name	Description
7:0	FIFO0_DB2	The third byte to be loaded to or unloaded from the FIFO.

23 USB EP0 FIFO DB3 USB Endpoint 0 FIFO Register DB3 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO0_DB3															
Type	Other															
Reset	0 0 0 0 0 0 0 0 0															

Bit(s)	Name	Description
7:0	FIFO0_DB3	The forth byte to be loaded to or unloaded from the FIFO.

24 USB EP1 FIFO DB0 USB Endpoint 1 FIFO Register DB0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO1_DB0															
Type	Other															
Reset	0 0 0 0 0 0 0 0 0															

The register provides MCU access to the FIFO for the endpoint 1. Writing to this register will load data to the FIFO for the endpoint 1. Reading this register will unload data from the FIFO for the endpoint 1.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO1_DB0	The first byte to be loaded to or unloaded from the FIFO.

25 USB EP1 FIFO DB1 USB Endpoint 1 FIFO Register DB1 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO1_DB1															
Type	Other															
Reset	0 0 0 0 0 0 0 0 0															

Bit(s)	Name	Description
7:0	FIFO1_DB1	The second byte to be loaded to or unloaded from the FIFO.

26 [USB_EP1_FIFO_DB2](#) **USB Endpoint 1 FIFO Register DB2** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO1_DB2				
Type												Other				
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB2	The third byte to be loaded to or unloaded from the FIFO.

27 [USB_EP1_FIFO_DB3](#) **USB Endpoint 1 FIFO Register DB3** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO1_DB3				
Type												Other				
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB3	The forth byte to be loaded to or unloaded from the FIFO.

28 [USB_EP2_FIFO_DB0](#) **USB Endpoint 2 FIFO Register DB0** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO2_DB0				
Type												Other				
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 2. Writing to this register will load data to the FIFO for the endpoint 2. Reading this register will unloads data from the FIFO for the endpoint 2.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO2_DB0	The first byte to be loaded to or unloaded from the FIFO.

29 [USB_EP2_FIFO_DB1](#) **USB Endpoint 2 FIFO Register DB1** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO2_DB1				
Type												Other				
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB1	The second byte to be loaded to or unloaded from the FIFO.

2A [USB_EP2_FIFO_DB2](#) **USB Endpoint 2 FIFO Register DB2** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO2_DB2				
Type												Other				
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB2	The third byte to be loaded to or unloaded from the FIFO.

2B [USB_EP2_FIFO_DB3](#) **USB Endpoint 2 FIFO Register DB3** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO2_DB3				
Type												Other				
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB3	The forth byte to be loaded to or unloaded from the FIFO.

2C [USB_EP3_FIFO_DB0](#) **USB Endpoint 3 FIFO Register DB0** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO3_DB0				
Type												Other				
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 3. Writing to this register will load data to the FIFO for the endpoint 3. Reading this register will unload data from the FIFO for the endpoint 3.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO3_DB0	The first byte to be loaded to or unloaded from the FIFO.

2D [USB_EP3_FIFO_DB1](#) **USB Endpoint 3 FIFO Register DB1** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												FIFO3_DB1				
Type												Other				
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB1	The second byte to be loaded to or unloaded from the FIFO.

2E [USB EP3 FIFO DB2](#) USB Endpoint 3 FIFO Register DB2 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO3_DB2															
Type	Other															
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB2	The third byte to be loaded to or unloaded from the FIFO.

2F [USB EP3 FIFO DB3](#) USB Endpoint 3 FIFO Register DB3 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO3_DB3															
Type	Other															
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB3	The forth byte to be loaded to or unloaded from the FIFO.

30 [USB EP4 FIFO DB0](#) USB Endpoint 4 FIFO Register DB0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO4_DB0															
Type	Other															
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 4. Writing to this register will load data to the FIFO for the endpoint 4. Reading this register will unload data from the FIFO for the endpoint 4.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO4_DB0	The first byte to be loaded to or unloaded from the FIFO.

31 [USB EP4 FIFO DB1](#) USB Endpoint 4 FIFO Register DB1 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIFO4_DB1															
Type	Other															
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
--------	------	-------------

Bit(s)	Name	Description
7:0	FIFO4_DB1	The second byte to be loaded to or unloaded from the FIFO.

32 USB EP4 FIFO DB2 USB Endpoint 4 FIFO Register DB2 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO4_DB2	The third byte to be loaded to or unloaded from the FIFO.

33 USB EP4 FIFO DB3 USB Endpoint 4 FIFO Register DB3 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO4_DB3	The forth byte to be loaded to or unloaded from the FIFO.

240 USB CON USB PHY Control 20

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											NULL PKT FIX				DMP ULLU P	DPP ULLU P
Type											RW				RW	RW
Reset											1				0	0

Bit(s)	Name	Description
5	NULLPKT_FIX	If NULLPKT_FIX is set to 1, the USB controller will not issue a DMAreq when a null packet is received.
1	DMPULLUP	Pull-up enabling pin. Enables the pull-up 1.5KOhm pull-up on D pin as a full speed device by setting it to high
0	DPPULLUP	Pull-up enabling pin. Enables the pull-up 1.5KOhm pull-up on D+ pin as a full speed device by setting it to high

3.14.4 System Integration Guide

3.14.4.1 USB device configuration

The target audience of this section is the software engineer.

The USB device controller features one control endpoint and 6 other endpoints. The configuration of interfaces and endpoints can be accommodated by software for specific functions, basically supporting Bluetooth HCI, and device firmware upgrade.

Bluetooth HCI transport layer defines the configuration of endpoints and interfaces.

- ✓ One voice channel with 16-bit encoding.

Endpoint	Endpoint type	Max. packet size (bytes)	Max. bandwidth (bytes/ms)	Min. bandwidth (bytes/ms)	Double buffer in controller	Generic DMA
Endpoint 0 (command)	Control	64			No	No
Endpoint 1 IN	Bulk (IN)	64		1024	No	Yes
Endpoint 2 OUT	Bulk (OUT)	64		1024	No	Yes
Endpoint 2 IN	Bulk (IN)	64	1024		No	Yes
Endpoint 2 OUT	Bulk (OUT)	64	1024		No	Yes
Endpoint 3 IN	Interrupt (IN)	16		16	No	No
Endpoint 4 IN	Interrupt (IN)	16		16	No	No

*When the maximum packet size is smaller than one half of the device FIFO size (64 bytes), the double buffer will be automatically enabled by hardware.

The pull-up resistor on the USB transceiver is initially disconnected when boot-up. No external resistor is required. The software should enable it after performing the configuration of the USB device controller.

3.14.4.2 System Infrastructure Configuration

The clock, interrupt and DMA are defined as the system infrastructure. It requires several steps to bring up the USB. Those steps should be done in sequence to prevent from malfunction.

Power-on

1. Enable USB PLL.
2. Enable USB clock after USB PLL is settled.
3. Unmask the USB interrupt in the interrupt controller.
4. Enable the pull-up resistor.

The USB device controller can generate the interrupt when conditions are met as defined in USB_INTRINE, USB_INTRROUTE, and USB_INTRUSBE.

The generic DMA controller is used to move data from or to the USB device controller. The USB device controller will use at most 4 DMA channels for ACL and SCO. The user should use the half-channel DMA since only the half channel DMA has the hardware flow control.

The USB FIFO provides byte and word accesses to the read/write port of USB_EP0_FIFO, USB_EP1_FIFO, USB_EP2_FIFO, USB_EP3_FIFO and USB_EP4_FIFO. If the data buffer allocated in memory is word aligned, the user can enable word transfer in the DMA controller. If the data buffer allocated in memory is not word aligned, the user should set to byte aligned and set B2W in DMAx_CON to 1 to enable fast byte-to-word transfer.

3.14.4.3 Power On/Off USB PHY and Controller Sequence

Power-on sequence after plug-in

1. Turn on Vusb(PHY 3.3v power) – The control register is in PMIC document.
2. Turn on USB AHB clock(78MHz) – The control register is in config document.
3. Turn on internal 48MHz PLL – the control register is in clock document.
4. Wait for 50 usec. (PHY 3.3v power stable time)
5. Turn on USB PHY BIAS current control → reg[USB+08C1h] bit3 = 1. (RG_USB11_FSLs_ENBGRI).
6. Wait for 10 usec.
7. Set up D+ pull up register for connecting Host → reg[USB+0240h] bit 0 =1(PUB)

Power-off sequence after plug-out

1. Release D+ pull up register for disconnecting Host → Setting reg[USB+0240h] bit 0 =0 (PUB)
2. Turn off USB PHY BIAS current control → reg[USB+08C1h] bit3 = 0. (RG_USB11_FSLs_ENBGRI).
3. Turn off Vusb (PHY 3.3v power) – The control register is in PMIC document

3.15 Accessory Detector

3.15.1 General Description

The hardware accessory detector (ACCDDET) detects plug-in/out of multiple types of external components. Based on the suggested circuit (see **Figure 41**), this design supports 3 types of external components, which are microphone, hook-switch and TV-OUT line. This design uses the internal 2-bit comparator to separate external components. The de-bounce scheme is also supported to resist

uncertain input noises. When the plug-in/out state is stable, the PWM unit of ACCDET will enable the comparator, MBIAS and threshold voltage of the comparator periodically for the plugging detection. With suitable PWM settings, very low-power consumption can be achieved when the detection feature is enabled. In order to compensate the delay between the detection login and comparator, the delay enabling scheme is adopted. Given the suitable delay number compared to the rising edge of PWM high pulse, the stable plugging state can be prorogated to digital detection logic. Then the correct plugging state can be detected and reported.

Figure 42 shows the state machine without TV-OUT mode. The state machine is executed by the software to control the ACCDET design. The ACCDET design will send one interrupt to acknowledge the software after the ACCDET input state is changed and the duration of the state is longer than debounce time. The software needs to read out the memorized ACCDET input state and follow the recommend state machine to program the register in it.

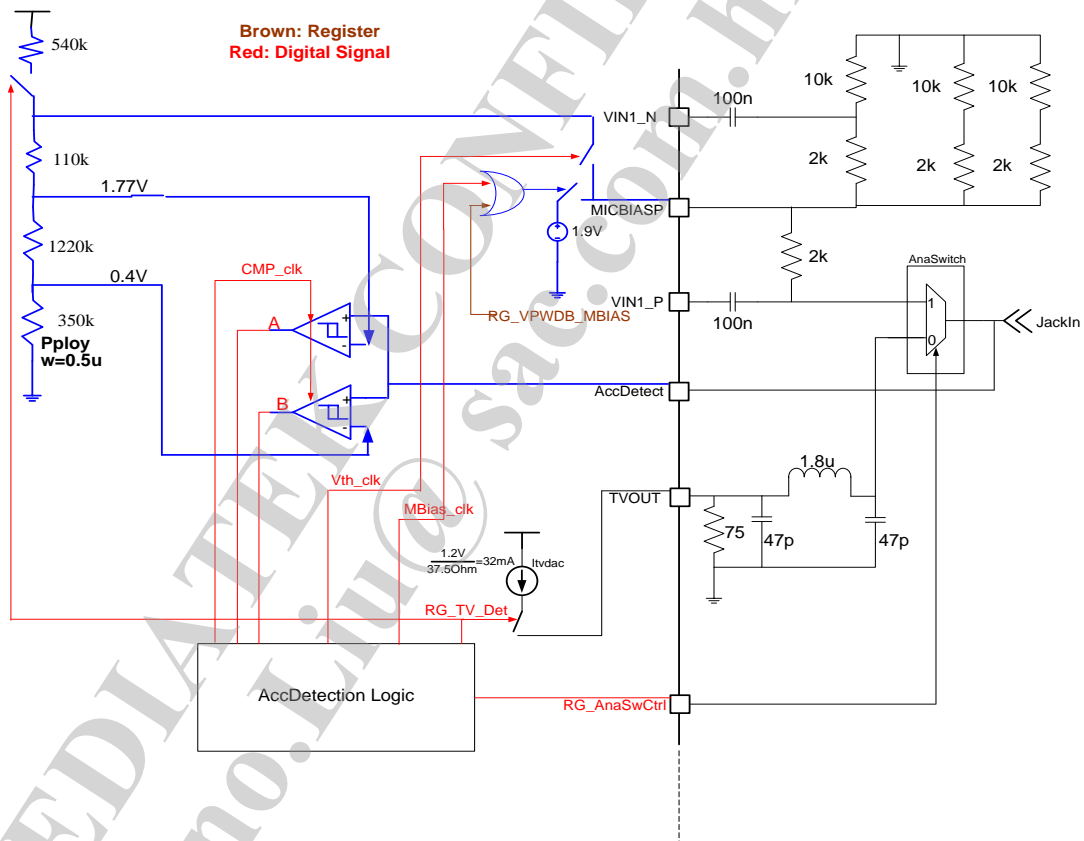


Figure 41. Suggested Accessory Detection Circuit.

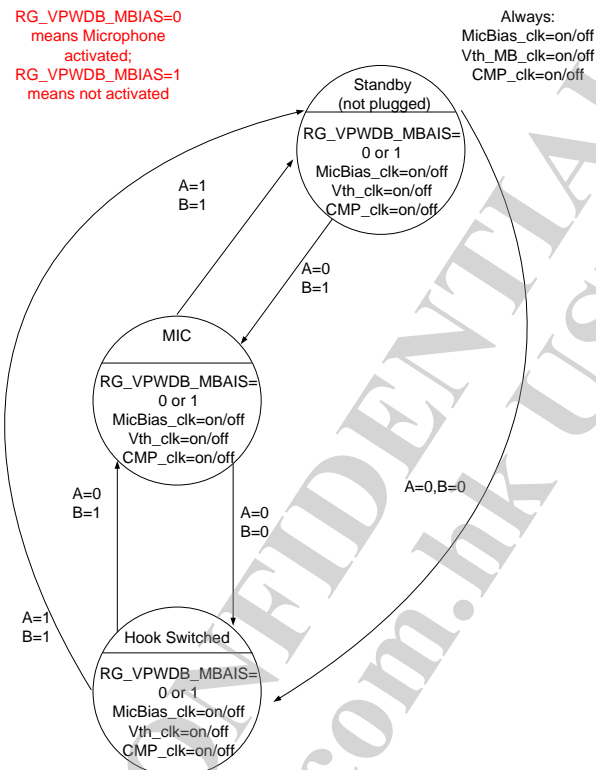


Figure 42. The State machine between Microphone and Hook-Switch plug-in/out change.

3.15.2 Pulse Width Modulation

The ACCDET design also provides one Pulse-Width-Modulation (PWM) to enable the comparator, microphone's bias current and the threshold voltage of the comparator periodically. With suitable PWM and settings for delayed enabling, the ACCDET can achieve very low power consumption and accurate plug-in/out detection. Figure 43 shows a timing diagram example of such PWM design. The output from PWM keeps being at "0" until the value of the counter is smaller than the programmed threshold.

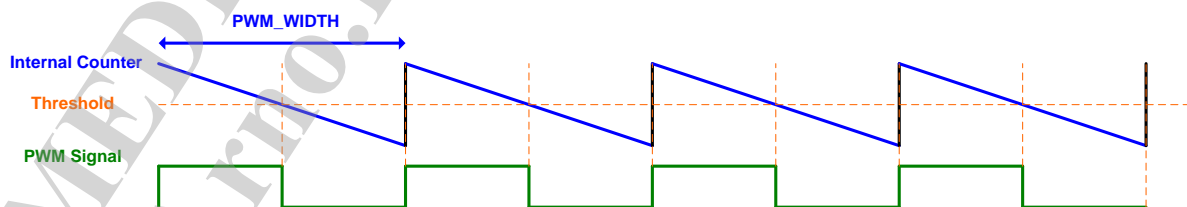


Figure 43. PWM waveform.

3.15.3 Register Definition

Module name: ACCDET base address: (+A0750000h)

Address	Name	Width	Register function
A0750000	ACCDDET_RSTB	32	ACCDDET software reset register
A0750004	ACCDDET_CTRL	32	ACCDDET control register
A0750008	ACCDDET_STATE_SWCTRL	32	ACCDDET state switch control register
A075000C	ACCDDET_PWM_WIDTH	32	ACCDDET PWM width register
A0750010	ACCDDET_PWM_THRESH	32	ACCDDET PWM threshold register
A0750024	ACCDDET_EN_DELAY_NUM	32	ACCDDET enable delay number register
A0750028	ACCDDET_PWM_IDLE_VALU E	32	ACCDDET PWM IDLE value register
A075002C	ACCDDET_DEBOUNCE0	32	ACCDDET debounce0 register
A0750030	ACCDDET_DEBOUNCE1	32	ACCDDET debounce1 register
A0750038	ACCDDET_DEBOUNCE3	32	ACCDDET debounce3 register
A075003C	ACCDDET_IRQ_STS	32	ACCDDET interrupt status register
A0750040	ACCDDET_CURR_IN	32	ACCDDET current input status register
A0750044	ACCDDET_SAMPLE_IN	32	ACCDDET sampled input status register
A0750048	ACCDDET_MEMOIZED_IN	32	ACCDDET memorized input status register
A075004C	ACCDDET_LAST_MEMOIZED _IN	32	ACCDDET last memorized input status register
A0750050	ACCDDET_FSM_STATE	32	ACCDDET FSM status register
A0750054	ACCDDET_CURR_DEBOUNC E	32	ACCDDET current de-bounce status register
A0750058	ACCDDET_VERSION	32	ACCDDET version code
A075005C	ACCDDET_IN_DEFAULT	32	default value of accdet_in

A0750000 ACCDET_RSTB ACCDET Software Reset Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																RSTB
Type																RW
Reset																1

Overview: After applying the setting to register, software reset is necessary for state initialization. Without this process, ACCDET may detect incorrect plug state.

Bit(s)	Mnemonic	Name	Description
0	RSTB	RSTB	Set to 0 to reset the ACCDET unit and set to 1 after the reset process is finished. This software reset will clear ACCDET's enable signal but keep all ACCDET's settings. After the reset process, ACCDET will return to the IDLE state.

A0750004 ACCDET_CTRL ACCDET Control Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																ACCD ET_E N
Type																RW
Reset																1

Bit(s)	Mnemonic	Name	Description
0	ACCD ET_E N	EN	Set to 1 to enable the ACCDET unit.

A0750008 ACCDET_STATE_SWCTRL ACCDET State Switch Control Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												MBIAS S_PWM M_EN	VTH_ PWM EN	CMP_ PWM EN		
Type												RW	RW	RW		
Reset												0	0	0		

Bit(s)	Mnemonic	Name	Description
4	MBIAS_PWM M_EN	MBIAS_PWM_EN	Enables PWM of ACCDET MBIAS unit
3	VTH_PWM_ EN	VTH_PWM_EN	Enables PWM of ACCDET voltage threshold unit
2	CMP_PWM_ EN	CMP_PWM_EN	Enables PWM of ACCDET comparator

A075000C ACCDET_PWM_WIDTH ACCDET PWM Width Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	PWM_WIDTH															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	PWM_WIDT H	PWM_WIDTH	<p>ACCDET PWM width</p> <p>It is PWM max. counter value. It will be the initial value for the internal counter. The PWM internal counter always counts down to zero to finish one complete period, and the value of the internal counter will return to the value of PWM_WIDTH.</p> <p>PWM output frequency = (32k/PWM_WIDTH) Hz.</p>

A0750010 ACCDET_PWM_THRESH ACCDET PWM Threshold Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	PWM_THRESH															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	PWM_THRE SH	PWM_THRESH	<p>ACCDET PWM threshold</p> <p>When the internal counter value is bigger than or equal to PWM_THRESH, the PWM output signal will be "0". When the internal counter is smaller than PWM_THRESH, the PWM output signal will be "1".</p> <p>PWM output duty cycle = (PWM_THRESH)x(1/32) ms.</p>

Figure 44 shows the PWM waveform with register value present.

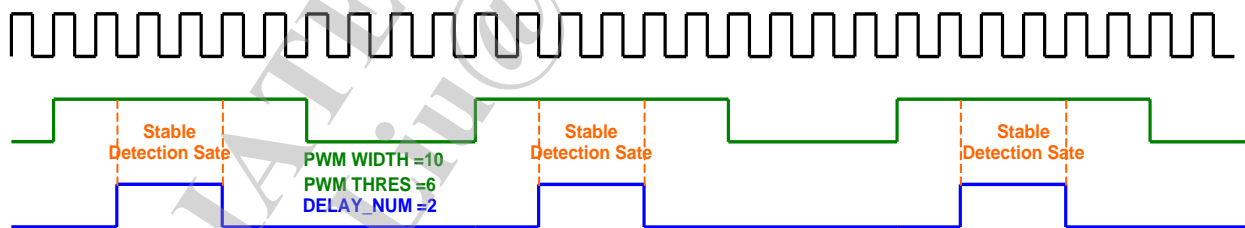


Figure 44. PWM waveform with register value present

A0750024 ACCDET_EN_D ELAY_NUM ACCDET Enable Delay Number Register 00000101

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	FALL_DELA	RISE_DELAY_NUM														

	Y_NUM																
Type	RW	RW															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
15	FALL_DELA Y_NUM	FALL_DELAY_NUM	Falling delay cycle compared to CMP PWM waveform In order to make sure the plug state is stable after disabling ACCDET, the suitable delay cycle is necessary. This number indicates the clock cycle number between the point when the digital part of ACCDET stops receiving accdet_in and the point when the analog part of ACCDET stops working.
14:0	RISE_DELA Y_NUM	RISE_DELAY_NUM	Rising delay cycle compared to PWM waveform In order to make sure the plug state is stable before activating ACCDET, the suitable delay cycle is necessary. This number indicates the clock cycle number between the point when the analog part of ACCDET starts working and the point when the digital part of ACCDET starts receiving stable accdet_in. This number should be fine tuned depending on different project requirements.

A0750028 ACCDET_PWM_IDLE_VALUE ACCDET PWM IDLE Value Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														MBIAS	VTH	CMP
Type														RW	RW	RW
Reset														0	0	1

Bit(s)	Mnemonic	Name	Description
2	MBIAS	MBIAS	IDLE value of MBIAS PWM
1	VTH	VTH	IDLE value of VTH PWM
0	CMP	CMP	IDLE value of CMP PWM

A075002C ACCDET_DEBO_UNCE0 ACCDET Debounce0 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE0															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE0	DEBOUNCE0	De-bounce time control of the next state = 2'b00 De-bounce time = DEBOUNCE/32 ms

A0750030 ACCDET_DEBOUNCE1 ACCDET Debounce1 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE1	DEBOUNCE1	De-bounce time control of the next state = 2'b01 De-bounce time = DEBOUNCE/32 ms

A0750034 ACCDET_DEBOUNCE2 ACCDET Debounce2 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE2															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE2	DEBOUNCE2	De-bounce time control of the next state = 2'b10 De-bounce time = DEBOUNCE/32 ms

A0750038 ACCDET_DEBO UNCE3 ACCDET Debounce3 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE3															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE3	DEBOUNCE3	De-bounce time control of the next state = 2'b11 De-bounce time = DEBOUNCE/32 ms

A075003C ACCDET_IRQ_STS ACCDET Interrupt Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne								IRQ_C LR								IRQ
Type								RW								RW
Reset								0								0

Overview: When the interrupt of ACCDET is asserted, IRQ_CLR must be set to 1 to clear the interrupt status. This bit will pause all activities in the ACCDET design until both interrupt status and IRQ_CLR are cleared. The software should write 1 to IRQ_CLR first to clear the interrupt (IRQ). After that, if pclk gating is enabled, the software should read ACCDET_IRQ_STS again to make IRQ_CLR self-reset to 0.

Bit(s)	Mnemonic	Name	Description
8	IRQ_CLR	IRQ_CLR	Clears interrupt status of ACCDET unit
0	IRQ	IRQ	Interrupt status of ACCDET unit Because this register will be cleared by hardware, the interrupt edge-sensitive scheme should be adopted for this design.

A0750040 ACCDET_CURR_IN ACCDET Current Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																CURR_IN
Type																RO
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	CURR_IN	CURR_IN	Current input status of ACCDET unit

A0750044 **ACCDET_SAMP**
LE_IN **ACCDET Sampled Input Status Register** **00000003**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																SAMPLE_IN
Type																RO
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	SAMPLE_IN	SAMPLE_IN	Samples input status of ACCDET unit When the plug-in/out state is changed, the ACCDET unit will do sampling.

A0750048 **ACCDET_MEM**
OIZED_IN **ACCDET Memorized Input Status Register** **00000003**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																MEMORIZED_IN
Type																RO
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	MEMORIZED_IN	MEMORIZED_IN	Memorized input status of ACCDET unit When the plug-in/out states is changed and held longer than the de-bounce time, the ACCDET unit will save the sampled input state to the memorized state. The interrupt will also be asserted to acknowledge the software.

A075004C **ACCDET_LAST**
_MEMORIZED_IN **ACCDET Last Memorized Input Status Register** **00000003**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																LAST_MEMO RIZED_IN
Type																RO
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	LAST_MEM ORIZED_IN	LAST_MEMORIZE D_IN	Last memorized input status of ACCDET unit

A0750050 ACCDET_FSM_STATE ACCDET FSM Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																FSM_STATE
Type																RO
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2:0	FSM_STATE	FSM_SATE	State of ACCDET unit finite-state-machine 0: ACCDET_IDLE 1: ACCDET_SAMPLE 2: ACCDET_DEBOUNCE 3: ACCDET_CHECK 4: ACCDET_MEMORIZED 5: ACCDET_IRQ

A0750054 ACCDET_CURR_DEBOUNCE ACCDET Current De-bounce Status Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																CURR_DEBOUNCE
Type																RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit(s)	Mnemonic	Name	Description
15:0	CURR_DEB OUNCE	CURR_DEBOUNC E	Currently used de-bounce time setting

A0750058 **ACCDET_VERSION** **ACCDET Version Code** **000000 03**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ACCDET_VERSION
Type																RO
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	ACCDET_VERSION	ACCDET_VERSI ON	Version code for ACCDET

A075005C **ACCDET_IN_DEFAULT** **Default Value of accdet_in** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ACCDET_IN_DEFAULT_REFRESH_EN				ACCDET_IN_DEFAULT
Type												RW				RW
Reset												0			0	0

Overview: The default value of sample_accdet_in and memorised_accdet_in can be set by software instead of using the default value set by hardware(i.e. 3). ACCDET_DEFAULT_REFRESH_EN is the enable bit controlling whether to use this additional function. The value of sample_accdet_in and memorized_accdet_in will change when accdet_en rises from low to high. *Note that if software reset is applied when accdet_en is high, the default value of sample_accdet_in and memorized_accdet_in will also be loaded when the software reset is de-asserted.*

Bit(s)	Mnemonic	Name	Description
4	ACCDET_IN_DEFAULT_REFRESH_EN	ACCDET_IN_DE FAULT_REFRES H_EN	Enable signal for whether to load accdet_in_default 0: accdet_in_default will not be loaded. 1: accdet_in_default will be loaded.
1:0	ACCDET_IN_DEFAULT	ACCDET_IN_DE FAULT	Default value of accdet_in set by software

3.16 SD Memory Card Controller (MSDC0)

3.16.1 Introduction

The controller fully supports the SD memory card bus protocol as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 2.0.

Furthermore, the controller also partially supports the SDIO card specification version 2.0. However, the controller can only be configured as the host of the SD memory card. Hereafter, the controller is also abbreviated as the SD controller. The following are the main features of the controller.

- Interface with MCU by APB bus
- 16/32-bit access on APB bus
- 16/32-bit access for control registers
- 32-bit access for FIFO
- Built-in 32 bytes FIFO buffers for transmit and receive, FIFO is shared for transmit and receive
- Built-in CRC circuit
- CRC generation can be disabled
- DMA supported
- Interrupt capabilities
- Data rate up to 48 Mbps in serial mode, 48x4 Mbps in parallel model, the module is targeted at 48 MHz operating clock
- Serial clock rate on SD bus is programmable
- Card detection capabilities during sleep mode
- Controllability of power for memory card
- Does not support SPI mode for SD memory card
- Does not support multiple SD memory cards

3.16.2 Overview

3.16.2.1 Pin Assignment

The following lists pins required for the SD memory card. Table 50 shows how the pins are shared. Note that all I/O pads have embedded both pull-up and pull-down resistors because they are shared by the SD memory card. The pull-down resistors for these pins can be used for power saving. If optimal pull-up or pull-down resistors are required on the system board, all embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers. The VDDPD pin is used for power saving. Power for the SD memory card can be shut down by programming the corresponding control register. The WP (Write Protection) pin is used to detect the status of the Write Protection Switch on the SD memory card.

Table 50. Sharing of pins for SD memory card controller

No.	Name	Type	MMC	SD	Description
-----	------	------	-----	----	-------------

No.	Name	Type	MMC	SD	Description
1	SD_CLK	O	CLK	CLK	Clock
2	SD_DAT3	I/O/PP		CD/DAT3	Data Line [Bit 3]
3	SD_DAT0	I/O/PP	DAT0	DAT0	Data Line [Bit 0]
4	SD_DAT1	I/O/PP		DAT1	Data Line [Bit 1]
5	SD_DAT2	I/O/PP		DAT2	Data Line [Bit 2]
6	SD_CMD	I/O/PP	CMD	CMD	Command Or Bus State
7	SD_PWRON	O			VDD ON/OFF
8	SD_WP	I			Write Protection Switch in SD
9	SD_INS	I	VSS2	VSS2	Card Detection

3.16.2.2 Card Detection

For SD memory card, detection of card insertion/removal by hardware is supported, and a dedicated pin “INS” is used to perform card insertion and removal for SD. The pin “INS” will be connected to the pin “VSS2” of a SD connector (see **Figure 45**).

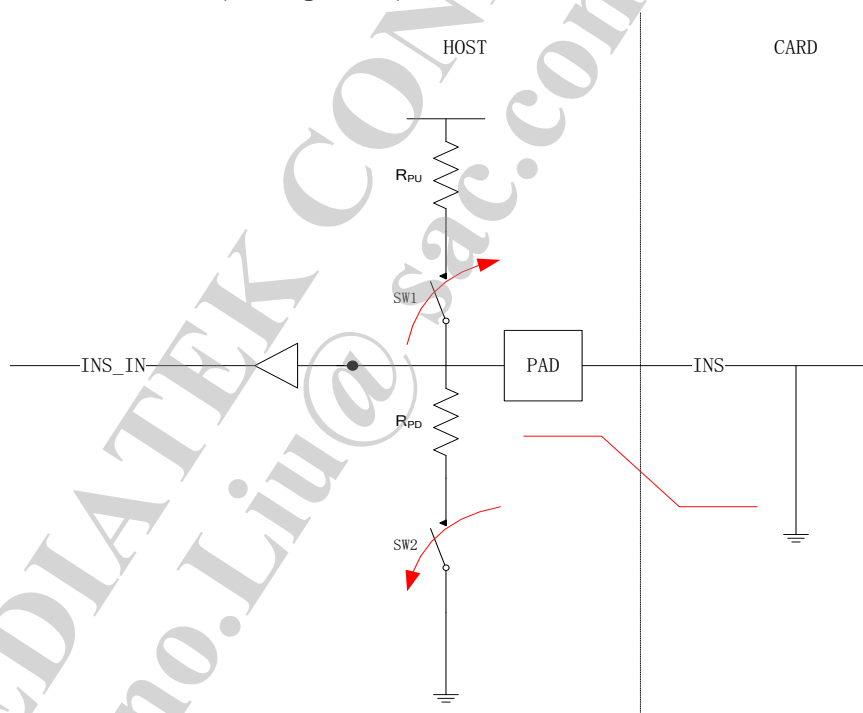


Figure 45. Card detection for SD memory card

3.16.3 Register Definition

Module name: MSDC0 base address: (+A0130000h)

Address	Name	Width	Register function
---------	------	-------	-------------------

Address	Name	Width	Register function
A0130000	<u>MSDC_CFG</u>	32	SD memory card controller configuration register For general configuration of the SD controller. <i>Note: MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.</i>
A0130004	<u>MSDC_STA</u>	32	SD memory card controller status register Contains the status of FIFO, interrupts and data requests.
A0130008	<u>MSDC_INT</u>	32	SD memory card controller interrupt register Contains the status of interrupts. Note that the register still shows the status of interrupt even though the interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting up register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.
A013000C	<u>MSDC_PS</u>	32	SD memory card pin status register Used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull down when the system is powered on. The scheme of card detection for MS is the same as that for SD. For detecting card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.
A0130010	<u>MSDC_DAT</u>	32	SD memory card controller data register Reads/Writes data from/to FIFO inside SD controller. Data access unit: 32 bits
A0130014	<u>MSDC_IOCON</u>	32	SD memory card controller IO control register Specifies output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current will depend on the PAD type selected for the chip.
A0130018	<u>MSDC_IOCON1</u>	32	SD memory card controller IO control register 1
A0130020	<u>SDC_CFG</u>	32	SD memory card controller configuration register Configures the SD memory card controller when it is configured as the host of SD. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the

Address	Name	Width	Register function
			controller. <i>Note: SDC_CFG[31:16] can be accessed by 16-bit APB bus access.</i>
A0130024	<u>SDC_CMD</u>	32	SD memory card controller command register Defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.
A0130028	<u>SDC_ARG</u>	32	SD memory card controller argument register Contains argument of the SD memory card command.
A013002C	<u>SDC_STA</u>	32	SD memory card controller status register Contains various statuses of SD controller as the controller is configured as the host of SD memory card.
A0130030	<u>SDC_RESP0</u>	32	SD memory card controller response register 0 Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0130034	<u>SDC_RESP1</u>	32	SD memory card controller response register 1 Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0130038	<u>SDC_RESP2</u>	32	SD memory card controller response register 2 Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A013003C	<u>SDC_RESP3</u>	32	SD memory card controller response register 3 Contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For responses of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.
A0130040	<u>SDC_CMDSTA</u>	32	SD memory card controller command status register Contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register will also be used as the interrupt source. The register is cleared when

Address	Name	Width	Register function
			being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0130044	<u>SDC_DATSTA</u>	32	SD memory card controller data status register Contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register is also used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0130048	<u>SDC_CSTA</u>	32	SD memory card status register After commands with R1 and R1b response this register containing the status of the SD card, it will be used as the response interrupt source. In all register fields, logic high indicates error, and logic low indicates there is no error. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A013004C	<u>SDC_IRQMASK0</u>	32	SD memory card IRQ mask register 0 Contains parts of SD memory card interrupt mask register. See the descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CMDSTA and SDC_DATSTA.
A0130050	<u>SDC_IRQMASK1</u>	32	SD memory card IRQ mask register 1 Contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then interrupt source from register field OUT_OF_RANGE of register SDC_CSTA will be masked. '0' in some bit does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.
A0130054	<u>SDIO_CFG</u>	32	SDIO configuration register Configures functions for SDIO.

Address	Name	Width	Register function
A0130058	<u>SDIO_STA</u>	32	SDIO status register Identifies if there is SDIO interrupt during the interrupt period on data line.
A0130080	<u>CLK_RED</u>	32	CLK latch configuration register Configures the MSDC sample data/response clock. <i>Note: When MSDC_IOCON[19] = 1, the host will latch response; otherwise MSDC FSM will handle the response from PAD directly.</i>
A0130098	<u>DAT_CHECKSUM</u>	32	MSDC Rx data checksum register Computes the checksum value of Rx read data

A0130000 MSDC_CFG SD Memory Card Controller Configuration Register 04000020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FIFOTH						VDDPD	RCDEN	DIRQEN	PINEN	DMAEN	INTEN
Type					RW						RW	RW	RW	RW	RW	RW
Reset					0		0	0			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCLKF								SCLKON	CRED	STDBY	CLKSRC	NOCRC	RST	MSDC	
Type	RW								RW	RW	RW	RW	RW	W1C	RW	
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Overview: For general configuration of the SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
27:24	FIFOTH	FIFOTH	FIFO threshold The register field determines when to issue a DMA request. For write transactions, DMA requests will be asserted if the number of free entries in FIFO are bigger than or equal to the value in the register field. For read transactions, DMA requests will be asserted if the number of valid entries in FIFO are bigger than or equal to the value in the register field. The register field must be set according to the setting of data transfer count in DMA burst mode. If single mode for DMA transfer is used, the register field shall be set to 0b0001. 0000: Invalid 0001: Threshold value is 1. 0010: Threshold value is 2. 0011~01111: ... 1000: Threshold value is 8. Others: Invalid
21	VDDPD	VDDPD	Controls output pin VDDPD used for power saving Output pin VDDPD controls the power for memory card. 0: Output pin VDDPD outputs logic low. The power for memory card will be turned off. 1: Output pin VDDPD outputs logic high. The power for memory card will be turned on.
20	RCDEN	RCDEN	Controls output pin RC DEN used for card identification process when the controller is for SD memory card Its output controls the pull-down resistor on the system board to

Bit(s)	Mnemonic	Name	Description
			connect to or disconnect from signal CD/DAT3. 0: The output pin RC DEN outputs logic low. 1: The output pin RC DEN outputs logic high.
19	DIRQEN	DIRQEN	Enables data request interrupt The register bit is used to control if data request is used as an interrupt source. 0: Data request is not used as an interrupt source. 1: Data request is used as an interrupt source.
18	PINEN	PINEN	Enables pin interrupt The register bit is used to control if the pin for card detection is used as an interrupt source. 0: The pin for card detection is not used as an interrupt source. 1: The pin for card detection is used as an interrupt source.
17	DMAEN	DMAEN	Enables DMA <i>Note: If DMA capability is disabled, the application software must poll the status of register MSDC_STA to check on any data transfer request. If DMA is desired, the register bit must be set up before command register is written.</i> 0: DMA request induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: DMA request induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
16	INTEN	INTEN	Enables interrupt <i>Note: If interrupt capability is disabled, the application software must poll the status of register MSDC_STA to check on any interrupt request.</i> 0: Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: Interrupt induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
15:8	SCLKF	SCLKF	Controls clock frequency of serial clock on SD bus and denotes clock frequency of SD bus serial clock as fslave and clock frequency of the SD controller as fhost which is 98.3 or 96.2 MHz <i>Note: The allowed maximum frequency of fslave is 49.15MHz.</i> While changing the clock rate, "1T clock period before change + 1T clock period after change" is required for HW signal to re-synchronize. 00000000b: fslave = (1/2)*fhost 00000001b: fslave = [1/(4*1)]*fhost 00000010b: fslave = [1/(4*2)]*fhost 00000011b: fslave = [1/(4*3)]*fhost 00000100b~11111110b: ... 11111111b: fslave = [1/(4*255)]*fhost
7	SCLKON	SCLKON	Serial clock always on For debugging. 0: Serial clock not always on 1: Serial clock always on
6	CRED	CRED	Rising edge data The register bit is used to determine the serial data input is latched at the falling edge or rising edge of the serial clock. The default

Bit(s)	Mnemonic	Name	Description
			setting is at the rising edge. If the serial data have bad timing, set the register bit to 1. When the memory card has bad timing on returned read data, set the register bit to 1. 0: Serial data input is latched at the rising edge of serial clock. 1: Serial data input is latched at the falling edge of serial clock.
5	STDBY	STDBY	Standby mode If the module is powered down, operating clock to the module will be stopped. At the same time, the clock to card detection circuitry will also be stopped. If detection on memory card insertion and removal is desired, write 1 to the register bit. If interrupt for detection on memory card insertion and removal is enabled, the interrupt will take place whenever the memory is inserted or removed. 0: Standby mode is disabled. 1: Standby mode is enabled.
4:3	CLKSRC	CLKSRC	Specifies which clock is used as source clock of memory card 00 : MPLL/5.5MHz clock 01 : MPLL/7MHz clock (this divider is default off, before switch controller source clock to this clock, you should switch controller source clock to 26Mhz first, and then enable divider MPLL/7, after a moment you can switch controller source clock to MPLL/ 7 to void source clock glitch. For detail setting please refer to configs document) 10 : MPLL/8MHz clock 11 : MPLL/10MHz clock For phone app. 00 : 94.5MHz clock Need to keep BT_APP_DIV_EN= 1'b0 in CLK_CONDA[15]. 01 : 74.3MHz clock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 : 65MHz clock 11 : Forbidden For BT app. 00 : Forbidden 01 : 89.1MHz clock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 : 78MHz clock 11 : 62.4MHz clock NOTE: Need to set POWERFUL_DIV_EN2 = 1'b1 first in CLK_CONDA[9].
2	NOCRC	NOCRC	Disable CRC '1' indicates data transfer without CRC is desired. For write data block, the data are transmitted without CRC. For read data block, CRC will not be checked. For testing purpose. 0: Data transfer with CRC is desired. 1: Data transfer without CRC is desired.
1	RST	RST	Software reset Writing 1 to the register bit will cause internal synchronous reset of SD controller but will not reset register settings, RST should only be set when RST equal to 0. 0: Read 0 stands for the reset process is finished. 1: Write 1 to reset SD controller.
0	MSDC	MSDC	Configures the controller as SD memory card mode CLK/CMD/DAT line is pulled low when SD memory card mode is disable. 0: Disable SD memory card 1: Enable SD memory card

A0130004 MSDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSY	FIFOC LR							FIFOCNT				INT	DRQ	BE	BF
Type	R	W1C							RO				RO	RO	RO	RO
Reset	0	0							0	0	0	0	0	0	0	0

Overview: The register contains the status of FIFO, interrupts and data requests.

Bit(s)	Mnemonic	Name	Description
15	BUSY	BUSY	<p>Status of the controller</p> <p>If the controller is in busy state, the register bit will be 1; otherwise 0. 0: The controller is in busy state. 1: The controller is in idle state.</p>
14	FIFOC LR	FIFOC LR	<p>Clears FIFO</p> <p>Writing 1 to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.</p> <p>0: Read 0 stands for the FIFO clear process is finished. 1: Write 1 to clear the content of FIFO clear and reset the status of FIFO controller.</p>
7:4	FIFOCNT	FIFOCNT	<p>FIFO count</p> <p>The register field shows how many valid entries are there in FIFO.</p> <p>0000: 0 valid entry in FIFO 0001: 1 valid entry in FIFO 0010: 2 valid entries in FIFO 0011~0111: ... 1000: 8 valid entries in FIFO</p>
3	INT	INT	<p>Indicates if there is any interrupt existing</p> <p>When there is interrupt existing, the register bit will still be active even if register bit INTEN in register MSDC_CFG is disabled. The SD controller can interrupt MCU by issuing interrupt request to the interrupt controller, or the software/application will poll the register endlessly to check if there is any interrupt request existing in the SD controller. When register bit INTEN in register MSDC_CFG is disabled, the second method is used. For read commands, it is possible that time-out error takes place. The software can read the status register to check if the time-out error takes place without OS time tick support or data request asserted.</p> <p><i>Note: The register bit will be cleared when register MSDC_INT is read.</i></p> <p>0: No interrupt request existing. 1: Interrupt request exists.</p>
2	DRQ	DRQ	<p>Indicates if any data transfer is required</p> <p>When a data transfer is required, the register bit will still be active even if register bit DIRQEN in register MSDC_CFG is disabled. Data transfer can be achieved by DMA channel alleviating MCU loading, or by polling the register bit to check if any data transfer is requested. When register bit DIRQEN in register MSDC_CFG is disabled, the second method is used.</p> <p>0: No DMA request existing. 1: DMA request exists.</p>
1	BE	BE	<p>Indicates if FIFO in SD controller is empty</p> <p>0: FIFO in SD controller is not empty.</p>

Bit(s)	Mnemonic	Name	Description
0	BF	BF	1: FIFO in SD controller is empty. Indicates if FIFO in SD controller is full 0: FIFO in SD controller is not full. 1: FIFO in SD controller is full.

A0130008 MSDC_INT SD Memory Card Controller Interrupt Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SDIOI RQ	SDR1 BIRQ		SDMC IRQ	SDDA TIRQ	SDCM DIRQ	PINIR Q	DIRQ
Type									RC	RC		RC	RC	RC	RC	RC
Reset									0	0		0	0	0	0	0

Overview: The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.

Bit(s)	Mnemonic	Name	Description
7	SDIOIRQ	SDIOIRQ	SDIO interrupt The register bit indicates if there is any interrupt for SDIO existing. Whenever an interrupt for SDIO exists, the register bit will be set to 1 if the interrupt is enabled. It will be reset when the register is read. 0: No SDIO interrupt 1: Interrupt for SDIO exists.
6	SDR1BIRQ	SDR1BIRQ	SD R1b response interrupt The register bit will be active when a SD command with R1b response is finished and the DAT0 line is transitioned from busy to idle state. Single block write commands with R1b response will cause interrupt when the command is completed either successfully or with CRC error. However, multi-block write commands with R1b response do not cause interrupt because multi-block write commands are always stopped by STOP_TRANS commands. STOP_TRANS commands (with R1b response) behind multi-block write commands will cause interrupt. Single block read command with R1b response will cause interrupt when the command is completed, but multi-block read commands do not. <i>Note: STOP_TRANS commands (with R1b response) behind multi-block read commands will cause interrupt.</i> 0: No interrupt for SD R1b response. 1: Interrupt for SD R1b response exists.
4	SDMCIRQ	SDMCIRQ	SD memory card interrupt The register bit indicates if there is any interrupt for SD memory card existing. Whenever an interrupt for SD memory card exists, i.e. any bit in register SDC_CSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. <i>Note: This bit will not trigger MSDC hardware interrupt.</i> 0: No SD memory card interrupt

Bit(s)	Mnemonic	Name	Description
3	SDDATIRQ	SDDATIRQ	<p>1: SD memory card interrupt exists.</p> <p>SD bus DAT interrupt</p> <p>The register bit indicates if there is any interrupt for SD DAT line existing. Whenever interrupt for SD DAT line exists, i.e. any bit in register SDC_DATSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read.</p> <p>0: No SD DAT line interrupt 1: SD DAT line interrupt exists.</p>
2	SDCMDIRQ	SDCMDIRQ	<p>SD bus CMD interrupt</p> <p>The register bit indicates if there is any interrupt for SD CMD line existing. Whenever interrupt for SD CMD line exists, i.e. any bit in the register SDC_CMDSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read.</p> <p>0: No SD CMD line interrupt 1: SD CMD line interrupt exists.</p>
1	PINIRQ	PINIRQ	<p>Pin change interrupt</p> <p>The register bit indicates if there is any interrupt for memory card insertion/removal existing. Whenever the memory card is inserted or removed and card detection interrupt is enabled, i.e. register bit PINEN in register MSDC_CFG is set to 1, the register bit will be set to 1. It will be reset when the register is read.</p> <p>0: Otherwise 1: Card is inserted or removed.</p>
0	DIRQ	DIRQ	<p>Data request interrupt</p> <p>The register bit indicates if there is any interrupt for data request existing. Whenever data request exists and data request as an interrupt source is enabled, i.e. register bit DIRQEN in register MSDC_CFG is set to 1, the register bit will be active. It will be reset when being read. For software, data requests can be recognized by polling register bit DRQ or by data request interrupt. Data request interrupts will be generated every FIFOTH data transfers.</p> <p>0: No data request interrupt 1: Data request interrupt occurs.</p>

A01300C **MSDC_PS** **SD Memory Card Pin Status Register** **0000008**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								CMD	DAT							
Type								RO	RO							
Reset								0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CDDEBOUNCE											PINCHG	PINO	POEN0	PIENO	CDEN
Type	RW											RC	RO	RW	RW	RW
Reset	0	0	0	0								0	1	0	0	0

Overview: The register is used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull-down when the system is powered on. The scheme of card detection for MS is the same as that for SD. To detect card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.

Bit(s)	Mnemonic	Name	Description
24	CMD	CMD	Memory card/SDIO card/MMC card command lines
23:16	DAT	DAT	Memory card/SDIO card/MMC card data lines
15:12	CDDEBOUN CE	CDDEBOUN CE	Specifies the time interval for card detection de-bounce Default value: 0. It means the de-bounce interval is 32-cycle time at 32kHz. The interval can extend one cycle time at 32kHz by increasing the counter by 1.
4	PINCHG	PINCHG	Pin change The register bit indicates the status of card insertion/removal. If the memory card is inserted or removed, the register bit will be set to 1 no matter pin change interrupt is enabled or not. It will be cleared when the register is read. 0: Otherwise 1: Card is inserted or removed.
3	PINO	PINO	Shows the value of input pin for card detection 0: The value of input pin for card detection is logic low. 1: The value of input pin for card detection is logic high.
2	POENO	POENO	Controls output of input pin for card detection 0: Output of input pin for card detection is disabled. 1: Output of input pin for card detection is enabled.
1	PIENO	PIENO	Controls input pin for card detection 0: Input pin for card detection is disabled. 1: Input pin for card detection is enabled.
0	CDEN	CDEN	Enables card detection The register bit is used to enable or disable card detection. 0: Card detection is disabled. 1: Card detection is enabled.

A0130010 MSDC_DAT SD Memory Card Controller Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to read/write data from/to FIFO inside the SD controller. Data access unit: 32 bits.

Bit(s)	Mnemonic	Name	Description
31:0	DATA	DATA	Reads/Writes data from/to FIFO inside SD controller Data access unit: 32 bits

A0130014 MSDC_IOCON SD Memory Card Controller IO Control Register 010000C3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							SAMPLEDLY	FIXDLY	SAMP ON	CRCDI S	CMDS EL	INTLH	DSW			
Type							RW	RW	RW	RW	RW	RW	RW			

Reset							0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDRE					HIGH SPEED D	DMABURST	SRCFG1	SRCFG0	ODCCFG1			ODCCFG0			
Type	RW					RW	RW	RW	RW	RW			RW			
Reset	0					0	0	0	1	1	0	0	0	0	1	1

Overview: The register specifies the output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current depends on the PAD type selected for the chip.

Bit(s)	Mnemonic	Name	Description
25:24	SAMPLEDL Y	SAMPLEDLY	<p>Used for SW to select the turn-around delay cycle between write data end bit and CRC status for SD card</p> <p>00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay</p>
23:22	FIXDLY	FIXDLY	<p>Used for SW to select the delay cycle after clock fix high for the host controller to SD card</p> <p>00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay</p>
21	SAMPON	SAMPON	<p>Data sample enabling always on</p> <p>The bit is suggested to be set to 1 when the feedback clock is used and to 0 when the multiple phase clock is used.</p> <p>0: Data sample enabling not always on 1: Data sample enabling always on</p>
20	CRCDIS	CRCDIS	<p>Switches off data CRC check for SD read data</p> <p>0: CRC check is on. 1: CRC check is off.</p>
19	CMDSEL	CMDSEL	<p>Determines whether the host should delay 1-T to latch response from card</p> <p>0: Host latches response without 1-T delay 1: Host latches response with 1-T delay.</p>
18:17	INTLH	INTLH	<p>Selects latch timing for SDIO multi-block read interrupt</p> <p>00: Host latches INT at the second backend clock after the end bit of the current data block from card is received. (Default) 01: Host latches INT at the first backend clock after the end bit of the current data block from card is received. 10: Host latches INT at the second backend clock after the end bit of the current data block from card is received. 11: Host latches INT at the third backend clock after the end bit of the current data block from card is received.</p>
16	DSW	DSW	<p>Determines whether the host should latch data with 1-T delay or not</p> <p>For SD card, this bit is suggested to be 0. For MSPRO cards, it is suggested to be 1.</p> <p>0: Host latches the data with 1-T delay. 1: Host latches the data without 1-T delay.</p>
15	CMDRE	CMDRE	<p>Determines whether the host should latch response token (sent from card on CMD line) at rising edge or falling edge of serial</p>

Bit(s)	Mnemonic	Name	Description
			clock (T.B.D this bit is un-useful) 0: Host latches response at rising edge of serial clock. 1: Host latches response at falling edge of serial clock.
10	HIGH_SPEE D	HIGH_SPEED	For high-speed mode when internal sample clock is used High-speed mode means the SD/MMC serial bus clock rate is bigger than 25MHz. The default speed mode means that the SD/MMC serial bus clock rate is bigger than 25MHz. 0: Default speed 1: High speed
9:8	DMABURST	DMABURST	Used for SW to select burst type when data are transferred by DMA <i>Note: Only single mode can support non-4N bytes data transfer in read operation.</i> 00: Single mode 01: 4-beat incrementing burst 10: 8-beat incrementing burst 11: Reserved
7	SRCFG1	SRCFG1	Output driving capability for pins DAT0, DAT1, DAT2 and DAT3 0: Fast slew rate 1: Slow slew rate
6	SRCFG0	SRCFG0	Output driving capability for pins CMD/BS and SCLK 0: Fast slew rate 1: Slow slew rate
5:3	ODCCFG1	ODCCFG1	Output driving capability for pins DAT0, DAT1, DAT2 and DAT3 000: 4mA 001: 8mA 010: 12mA 011: 16mA
2:0	ODCCFG0	ODCCFG0	Output driving capability for pins CMD/BS and SCLK 000: 4mA 001: 8mA 010: 12mA 011: 16mA

A0130018 MSDC IOCON1 SD Memory Card Controller IO Control Register 1 00022022

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		PRCF G_CK	PRVAL_CK			PRCF G_CM	PRVAL_CM			PRCF G_DA	PRVAL_DA					
Type		RW	RW			RW	RW			RW	RW					
Reset		0	1	0		0	0	0		0	1	0				

Bit(s)	Mnemonic	Name	Description
14	PRCFG_CK	PRCFG_CK	Pull-up/down register configuration for pin CK Default value: 0 0: Pull-up resistor in the I/O pad of pin CK is enabled. 1: Pull-down resistor in the I/O pad of pin CK is enabled.

Bit(s)	Mnemonic	Name	Description
13:12	PRVAL_CK	PRVAL_CK	Pull-up/down register value for pin CLK Default value: 10 00: Pull-up/down resistor in the I/O pad of pin CLK are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CLK value is 23.5k.
10	PRCFG_CM	PRCFG_CM	Pull-up/down register configuration for pin CM Default value is 0. 0: Pull-up resistor in the I/O pad of pin CM is enabled. 1: Pull-down resistor in the I/O pad of pin CM is enabled.
9:8	PRVAL_CM	PRVAL_CM	Pull-up/down register value for pin CMD/BS Default value: 00 00: Pull-up/down resistor in the I/O pad of pin CMD/BS are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 23.5k.
6	PRCFG_DA	PRCFG_DA	Pull-up/down register configuration for pin DAT0, DAT1, DAT2 and DAT3 Default value: 0 0: Pull-up resistor in the I/O pad of pin DAT is enabled. 1: Pull-down resistor in the I/O pad of pin DAT is enabled.
5:4	PRVAL_DA	PRVAL_DA	Pull-up/down register value for pin DAT0, DAT1, DAT2 and DAT3 Default value: 10 00: Pull-up/down resistor in the I/O pad of pin DAT are all disabled. 01: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 10: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 11: Pull-up/down resistor in the I/O pad of pin DAT value is 23.5k.

A0130020 SDC_CFG SD Memory Card Controller Configuration Register 00008000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIOC								WDOD				SDIO		MDLE N	SIEN
Type	RW								RW				RW		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BSYDLY				BLKLEN											
Type	RW				RW											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to configure the SD memory card controller when it is configured as the host of SD. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
31:24	DIOC	DIOC	Data time-out counter The period from finish of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit

Bit(s)	Mnemonic	Name	Description
			of 65,536 serial clock. See the register field descriptions of register bit RDINT for reference. 00000000: Extend 65,536 more serial clock cycles 00000001: Extend 65,536x2 more serial clock cycles 00000010: Extend 65,536x3 more serial clock cycles 00000011~11111110: ... 11111111: Extend 65,536x 256 more serial clock cycles
23:20	WDOD	WDOD	Write data output delay The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
19	SDIO	SDIO	Enables SDIO 0: Disable SDIO mode 1: Enable SDIO mode
17	MDLEN	MDLEN	Enables multiple data line The register can be enabled only when SD memory card is applied and detected by software application. It is the responsibility of the application to program the bit correctly when an multi-media card is applied. If an multi-media card is applied and 4-bit data line is enabled, the 4 bits will be output every serial clock. Therefore, data integrity will fail. 0: Disable 4-bit data line 1: Enable 4-bit data line
16	SIEN	SIEN	Enables serial interface It should be enabled as soon as possible before any command. 0: Disable serial interface for SD 1: Enable serial interface for SD
15:12	BSYDLY	BSYDLY	Only valid for the commands with R1b response If the command has a response of R1b type, the SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if the operation in SD memory card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, the controller will abandon the detection. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
11:0	BLKLEN	BLKLEN	Block length The register field is used to define the length of one block in unit of byte in a data transaction. The maximum value of block length is 2,048 bytes. 000000000000: Reserved 000000000001: Block length is 1 byte. 000000000010: Block length is 2 bytes.

Bit(s)	Mnemonic	Name	Description
			000000000011~011111111110: ... 011111111111: Block length is 2,047 bytes. 100000000000: Block length is 2,048 bytes.

A0130024 SDC_CMD SD Memory Card Controller Command Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																CMDFAIL
Type																RW
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTC	STOP	RW	DTYPE		IDRT	RSPTYP			BREAK	CMD					
Type	RW	RW	RW	RW		RW	RW			RW	RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto the SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.

Bit(s)	Mnemonic	Name	Description
16	CMDFAIL	CMDFAIL	If 4-bit SDIO mode is enabled and when CMD/DAT error occurs, set this bit to select whether to "wait stop command" or "wait data state machine idle". 0: Wait stop command 1: Wait data state machine idle
15	INTC	INTC	Indicates if the command is GO_IRQ_STATE If the command is GO_IRQ_STATE, the period between command token and response token will not be limited. 0: The command is not GO_IRQ_STATE. 1: The command is GO_IRQ_STATE.
14	STOP	STOP	Indicates if the command is a stop transmission command 0: The command is not a stop transmission command. 1: The command is a stop transmission command.
13	RW	RW	Defines the command is a read command or write command The register bit is valid only when the command causes a transaction with data token. 0: The command is a read command. 1: The command is a write command.
12:11	DTYPE	DTYPE	Defines data token type for the command 00: No data token for the command 01: Single block transaction 10: Multiple block transaction, i.e. the command is a multiple block read or write command. 11: Stream operation. It can only be used when an multi-media card is applied.
10	IDRT	IDRT	Identification response time The register bit indicates if the command has a response with NID (i.e. 5 serial clock cycles as defined in SD Memory Card

Bit(s)	Mnemonic	Name	Description
9:7	RSPTYP	RSPTYP	<p>Specification Part 1 Physical Layer Specification version 1.0) response time. The register bit is valid only when the command has a response token. Thus the register bit must be set to 1 for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD).</p> <p>0: Otherwise 1: The command has a response with NID response time.</p> <p>Defines response type for the command</p> <p>For commands with R1 and R1b response, register SDC_CSTA (not SDC_STA) updates after response token is received. This register SDC_CSTA contains the status of the SD, and it can be used as a response interrupt source.</p> <p><i>Note: If CMD7 is used with all 0's RCA, then RSPTYP must be "000". Command "GO_TO_IDLE" also has RSPTYP="000".</i></p> <p>000: There is no response for the command, e.g. broadcast command without response and GO_INACTIVE_STATE command. 001: The command has R1 response. R1 response token is 48-bit. 010: The command has R2 response. R2 response token is 136-bit. 011: The command has R3 response. Even though R3 is 48-bit response, it does not contain CRC checksum. 100: The command has R4 response. R4 response token is 48-bit. (only for MMC) 101: The command has R5 response. R5 response token is 48-bit. (only for MMC) 110: The command has R6 response. R6 response token is 48-bit. 111: The command has R1b response. If the command has a response of R1b type, SD controller must monitor the data line 0 for card busy status from the bit time that is 2 or 4 serial clock cycles after the command end bit to check if the operation in SD memory card has finished. There are two cases for detection of card busy status. The first case is that the host stops the data transmission during an active write data transfer. The card will assert busy signal after the stop transmission command end bit followed by 4 serial clock cycles. The second case is that the card is in idle state or receiving a stop transmission command between data blocks when multiple block write command is in progress. The register bit will be valid only when the command has a response token.</p>
6	BREAK	BREAK	<p>Aborts pending MMC GO_IRQ_MODE command</p> <p>It is only valid for a pending GO_IRQ_MODE command waiting for MMC interrupt response.</p> <p>0: Other fields are valid. 1: Break a pending MMC GO_IRQ_MODE command in the controller. Other fields are invalid.</p>
5:0	CMD	CMD	<p>SD memory card command</p> <p>Total 6 bits.</p>

A0130028 SDC_ARG SD Memory Card Controller Argument Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ARG[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ARG[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains the argument of the SD memory card command.

Bit(s)	Mnemonic	Name	Description
31:0	ARG	ARG	Contains argument of the SD memory card command.

A013002C SDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WP											FEDATBUS Y	FECMBUS Y	BEDATBUS Y	BECMBUS Y	BESDCBUS Y
Type	RO											RO	RO	RO	RO	RO
Reset	0											0	0	0	0	0

Overview: The register contains various statuses of SD controller as the controller is configured as the host of SD memory card.

Bit(s)	Mnemonic	Name	Description
15	WP	WP	Detects the status of write protection switch on SD memory card The register bit shows the status of write protection switch on SD memory card. There is no default reset value. Pin WP (Write Protection) is only useful when the controller is configured for SD memory card. 1: Write protection switch on, i.e. memory card is desired to be write-protected. 0: Write protection switch off, i.e. memory card is writable.
4	FEDATBUS Y	FEDATBUSY	Indicates if there is any transmission going on DAT line on SD bus This bit indicates directly the CMD line at card clock domain. For those commands without data but still involving DAT line, the register bit is useless. For example, if an erase command is issued, checking if the register bit is 0 before issuing the next command with data will not guarantee that the controller is idle. In this case, use register bit BESDCBUSY. 0: No transmission is going on DAT line on SD bus. 1: There is transmission going on DAT line on SD bus.
3	FECMBUS Y	FECMBUSY	Indicates if there is any transmission going on CMD line on SD bus This bit indicates directly the CMD line at card clock domain. 0: No transmission is going on CMD line on SD bus. 1: There is transmission going on CMD line on SD bus.
2	BEDATBUS Y	BEDATBUSY	Indicates if there is any transmission going on DAT line on SD bus 0: Backend SDC controller gets the info that no transmission is going on DAT line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on DAT line on SD bus.
1	BECMBUS	BECMBUSY	Indicates if there is any transmission going on CMD line on SD bus

Bit(s)	Mnemonic	Name	Description
Y			bus This bit shows backend controller's CMD busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SDC controller gets the info that no transmission is going on CMD line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on CMD line on SD bus.
0	BESDCBUS	BESDCBUSY	Indicates if SD controller is busy, i.e. is there any transmission going on CMD or DAT line on SD bus This bit shows backend controller's SDC busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SD controller is idle. 1: Backend SD controller is busy.

A0130030 SDC_RESP0 SD Memory Card Controller Response Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[31:0][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[31:0][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[31:0]	RESP_31_0	

A0130034 SDC_RESP1 SD Memory Card Controller Response Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[63:32][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[63:32][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[63:32]	RESP_63_32	

A0130038 SDC_RESP2 SD Memory Card Controller Response Register 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[95:64][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[95:64][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[95:64]	RESP_95_64	

A013003C SDC_RESP3 SD Memory Card Controller Response Register 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[127:96][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[127:96][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, i.e. response of commands ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[127:96]	RESP_127_96	

A0130040 SDC_CMDSTA SD Memory Card Controller Command Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RSPC RCER R	CMDT O	CMDR DY
Type														RC	RC	RC
Reset														0	0	0

Overview: The register contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The

register can also be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
2	RSPRCER R	RSPRCERR	CRC error on CMD detected '1' indicates the SD controller detects a CRC error after reading a response from the CMD line. 0: Otherwise 1: SD controller detects a CRC error after reading a response from the CMD line.
1	CMDTO	CMDTO	Time-out on CMD detected '1' indicates the SD controller detects a time-out condition while waiting for a response on the CMD line. 0: Otherwise 1: SD controller detects a timeout condition while waiting for a response on the CMD line.
0	CMDRDY	CMDRDY	For command without response, the register bit will be 1 once the command is completed on SD bus For command with response, the register bit will be 1 whenever the command is issued onto the SD bus and its corresponding response is received without CRC error. 0: Otherwise 1: Command with/without response is finished successfully without CRC error.

A0130044 SDC DATSTA SD Memory Card Controller Data Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DATCRCERR								DATT O	BLKD ONE
Type							RC								RC	RC
Reset							0	0	0	0	0	0	0	0	0	0

Overview: The register contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register can be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
9:2	DATRCER R	DATRCERR	CRC error on DAT detected '1' indicates that the SD controller detects a CRC error for bit n after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. 0: Otherwise 1: SD controller detects a CRC error after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. Note: n is 7 ~ 1 for 8-bits mode. Each bit is read and cleared

Bit(s)	Mnemonic	Name	Description
1	DATTO	DATTO	<i>individually.</i> Time-out on DAT detected A '1' indicates that the SD controller detects a time-out condition while waiting for data token on the DAT line. 0: Otherwise 1: SD controller detects a time-out condition while waiting for data token on the DAT line.
0	BLKDONE	BLKDONE	Indicates the status of data block transfer 0: Otherwise 1: A data block is successfully transferred.

A0130048 SDC CSTA SD Memory Card Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSTA [31:0][31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSTA [31:0][15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: After commands with R1 and R1b respond this register containing the status of the SD card, it will be used as a response interrupt source. In all register fields, logic high indicates error, and logic low indicates no error. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
31:0	CSTA [31:0]	CSTA_31_0	CSTA31: OUT_OF_RANGE. The command's argument is out of the allowed range for this card. CSTA30: ADDRESS_ERROR. A misaligned address that does not match the block length is used in the command. CSTA29: BLOCK_LEN_ERROR. The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length. CSTA28: ERASE_SEQ_ERROR. An error in the sequence of erase commands occurs. CSTA27: ERASE_PARAM. An invalid selection of write-blocks for erase occurs. CSTA26: WP_VIOLATION. Attempt to program a write-protected block. CSTA25: Reserved. Return to 0. CSTA24: LOCK_UNLOCK_FAILED. Set when a sequence or password error is detected in lock/unlock card command or if there is an attempt to access a locked card. CSTA23: COM_CRC_ERROR. The CRC check of the previous command fails. CSTA22: ILLEGAL_COMMAND. Command not legal for the card state. CSTA21: CARD_ECC_FAILED. Card internal ECC is applied but fails to correct the data. CSTA20: CC_ERROR. Internal card controller error.

- CSTA19: ERROR. A general or unknown error occurs during the operation.
- CSTA18: UNDERRUN. The card cannot sustain data transfer in stream read mode.
- CSTA17: OVERRUN. The card cannot sustain data programming in stream write mode.
- CSTA16: CID/CSD_OVERWRITE. It can be either one of the following errors: 1) The CID register has been already written and cannot be overwritten; 2) The read-only section of the CSD does not match the card; 3) An attempt to reverse the copy (set as the original) or permanent WP (unprotected) bits is made.
- CSTA[15: 4]: Reserved. Return to 0.
- CSTA3: AKE_SEQ_ERROR. Error in the sequence of authentication process
- CSTA[2: 0]: Reserved. Return to 0.

A013004C SDC_IRQMASK0 SD Memory Card IRQ Mask Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [31:0][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [31:0][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. See the register descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from registers SDC_CMDSTA and SDC_DATSTA.

Bit(s)	Mnemonic	Name	Description
31:0	IRQMASK [31:0]	IRQMASK_31_0	

A0130050 SDC_IRQMASK1 SD Memory Card IRQ Mask Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [63:32][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [63:32][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt

sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then the interrupt source from register field OUT_OF_RANGE of register SDC_CSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.

Bit(s)	Mnemonic	Name	Description
31:0	IRQMASK [63:32]	IRQMASK_63_32	

A0130054 SDIO_CFG SDIO Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											DISSEL		INTCSEL	DSBSEL		INTEN
Type											RW		RW	RW		RW
Reset											0		0	0		0

Overview: The register is used to configure functions for SDIO.

Bit(s)	Mnemonic	Name	Description
5	DISSEL	DISSEL	Selects data block interrupt source 0: The host detects SDIO interrupt during interrupt period between two data blocks of multiple block data access. 1: The host ignores SDIO interrupt during interrupt period between two data blocks of multiple block data access.
3	INTCSEL	INTCSEL	Selects interrupt control 0: The host detects DAT1 low as SDIO interrupt. 1: The host detects DAT3/DAT2/DAT1/DAT0 4'b1101 as SDIO interrupt.
2	DSBSEL	DSBSEL	Selects data block start bit 0: Use data line 0 as start bit of data block. Other data lines are ignored. 1: Start bit of a data block is received only when all data line 0-3 become low.
0	INTEN	INTEN	Enables interrupt for SDIO 0: Disable 1: Enable

A0130058 SDIO_STA SDIO Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																IRQ
Type																RO
Reset																0

Overview: This register is used to identify if there is SDIO interrupt during the interrupt period on data line.

Bit(s)	Mnemonic	Name	Description
0	IRQ	IRQ	<p>SDIO interrupt exists on the data line.</p> <p>For example, when in the interrupt period or the 1-bit data line mode and DAT1/5 goes low from high, this bit will become 1 from 0. If DAT1/5 goes high from low, this bit will become 0 from 1.</p> <p>0: There is no SDIO interrupt existing on the data line. 1: There is SDIO interrupt existing on the data line.</p>

A0130080 CLK_RED CLK Latch Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			CMD_RED													
Type			RW													
Reset			0													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DAT_RED						CLKP_AD_R	CLK_LATC						
Type			RW						ED	H						
Reset			0						0	0						

Overview: The register is used to configure the MSDC sample data/response clock. Note that only when MSDC_IOCON[19] = 1 will the host latch response; otherwise MSDC FSM will handle the response from PAD directly.

Bit(s)	Mnemonic	Name	Description
29	CMD_RED	CMD_RED	<p>Determines the command response from card output is latched at falling edge or rising edge of internal clock</p> <p>Only effective when CLK_LATCH = 1</p> <p>0: Internal clock rising edge to latch response 1: Internal clock falling edge to latch response</p>
13	DAT_RED	DAT_RED	<p>Determines the input data from card output is latched at falling edge or rising edge of internal sample clock</p> <p>Only effective when CLK_LATCH = 1</p> <p>0: Internal clock rising edge to latch data 1: Internal clock falling edge to latch data</p>
7	CLKPAD_R ED	CLKPAD_RED	<p>Determines the input data from card is latched at falling edge or rising edge of the feedback clock from pad</p> <p>The suggested setting is 0 when SD serial clock is lower than 25MHz. The suggestion setting is 1 when SD serial clock is higher than 25MHz. The suggestion setting is 0 for MMC card no matter the serial clock rate is high speed or default speed. Only effective when CLK_LATCH = 0.</p> <p>0: Internal feedback clock rising edge to latch data/response 1: Internal feedback clock falling edge to latch data/response</p>
6	CLK_LATCH	CLK_LATCH	<p>Determines which clock to latch data from card</p> <p>The suggested setting is 1 if SCLKF in register field MSDC_CFG is 0x0. Otherwise, the suggested setting is 0.</p> <p>0: Internal feedback clock is used to latch data/response from card.</p>

Bit(s)	Mnemonic	Name	Description
			1: Internal clock is used to latch data/response from card.

A0130098 DAT_CHECKSUM MSDC Rx Data Checksum Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAT_CHECKSUM[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT_CHECKSUM[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to complete the checksum value of Rx read data

Bit(s)	Mnemonic	Name	Description
31:0	DAT_CHEC KSUM	DAT_CHECKSUM	The checksum algorithm is 32-bit XOR.

3.17 SD Memory Card Controller (MSDC1)

3.17.1 Introduction

The controller fully supports the SD memory card bus protocol as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 2.0.

Furthermore, the controller also partially supports the SDIO card specification version 2.0. However, the controller can only be configured as the host of the SD memory card. Hereafter, the controller is also abbreviated as the SD controller. The following are the main features of the controller.

- Interface with MCU by APB bus
- 16/32-bit access on APB bus
- 16/32-bit access for control registers
- 32-bit access for FIFO
- Built-in 32 bytes FIFO buffers for transmit and receive, FIFO is shared for transmit and receive
- Built-in CRC circuit
- CRC generation can be disabled
- DMA supported
- Interrupt capabilities
- Data rate up to 48 Mbps in serial mode, 48x4 Mbps in parallel model, the module is targeted at 48 MHz operating clock
- Serial clock rate on SD bus is programmable
- Card detection capabilities during sleep mode
- Controllability of power for memory card
- Does not support SPI mode for SD memory card

- Does not support multiple SD memory cards

3.17.2 Overview

3.17.2.1 Pin Assignment

The following lists pins required for the SD memory card. Table 51 shows how the pins are shared. Note that all I/O pads have embedded both pull-up and pull-down resistors because they are shared by the SD memory card. The pull-down resistors for these pins can be used for power saving. If optimal pull-up or pull-down resistors are required on the system board, all embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers. The VDDPD pin is used for power saving. Power for the SD memory card can be shut down by programming the corresponding control register. The WP (Write Protection) pin is used to detect the status of the Write Protection Switch on the SD memory card.

Table 51. Sharing of pins for SD memory card controller

No.	Name	Type	MMC	SD	Description
1	SD_CLK	O	CLK	CLK	Clock
2	SD_DAT3	I/O/PP		CD/DAT3	Data Line [Bit 3]
3	SD_DAT0	I/O/PP	DAT0	DAT0	Data Line [Bit 0]
4	SD_DAT1	I/O/PP		DAT1	Data Line [Bit 1]
5	SD_DAT2	I/O/PP		DAT2	Data Line [Bit 2]
6	SD_CMD	I/O/PP	CMD	CMD	Command Or Bus State
7	SD_PWRON	O			VDD ON/OFF
8	SD_WP	I			Write Protection Switch in SD
9	SD_INS	I	VSS2	VSS2	Card Detection

3.17.2.2 Card Detection

For SD memory card, detection of card insertion/removal by hardware is supported, and a dedicated pin “INS” is used to perform card insertion and removal for SD. The pin “INS” will be connected to the pin “VSS2” of a SD connector (see **Figure 45**).

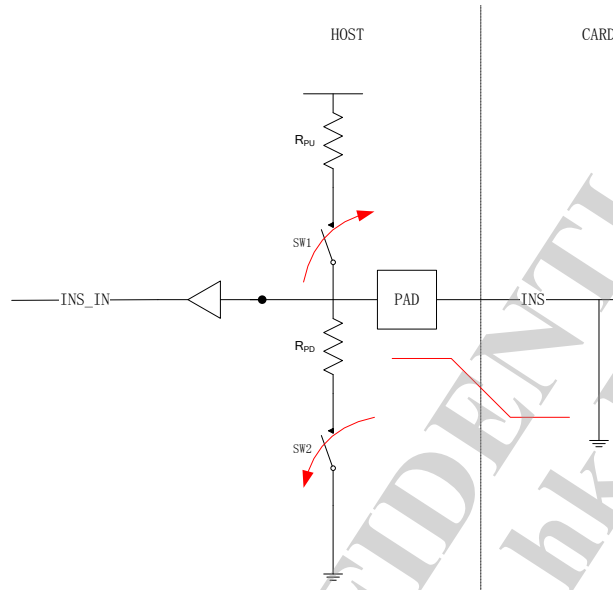


Figure 46. Card detection for SD memory card

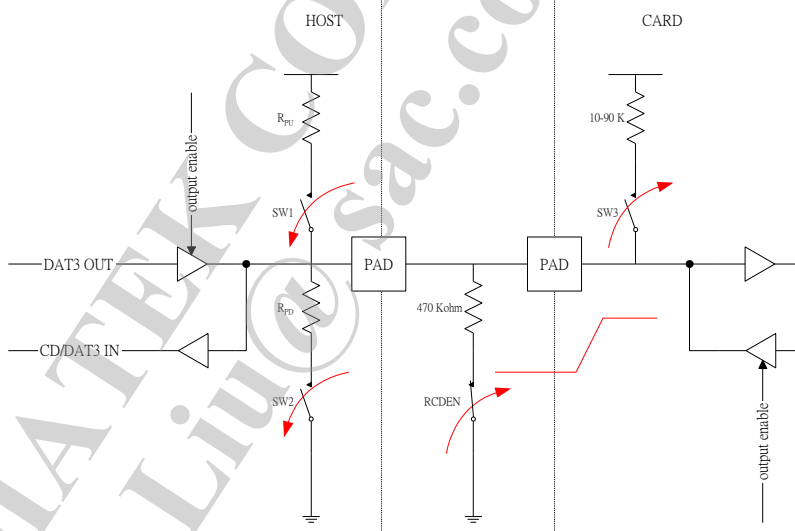


Figure 47. Card detection for SD memory card (Scheme 2)

3.17.3 Register Definition

Module name: MSDC1 base address: (+A0270000h)

Address	Name	Width	Register function
A0270000	MSDC_CFG	32	SD memory card controller configuration register For general configuration of the SD controller. <i>Note: MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.</i>
A0270004	MSDC_STA	32	SD memory card controller status register

Address	Name	Width	Register function
			Contains the status of FIFO, interrupts and data requests.
A0270008	<u>MSDC_INT</u>	32	SD memory card controller interrupt register Contains the status of interrupts. Note that the register still shows the status of interrupt even though the interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting up register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.
A027000C	<u>MSDC_PS</u>	32	SD memory card pin status register Used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull down when the system is powered on. The scheme of card detection for MS is the same as that for SD. For detecting card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.
A0270010	<u>MSDC_DAT</u>	32	SD memory card controller data register Reads/Writes data from/to FIFO inside SD controller. Data access unit: 32 bits
A0270014	<u>MSDC_IOCON</u>	32	SD memory card controller IO control register Specifies output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current will depend on the PAD type selected for the chip.
A0270018	<u>MSDC_IOCON1</u>	32	SD memory card controller IO control register 1
A0270020	<u>SDC_CFG</u>	32	SD memory card controller configuration register Configures the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. <i>Note: SDC_CFG[31:16] can be accessed by 16-bit APB bus access.</i>
A0270024	<u>SDC_CMD</u>	32	SD memory card controller command register

Address	Name	Width	Register function
			Defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.
A0270028	<u>SDC_ARG</u>	32	SD memory card controller argument register Contains argument of the SD memory card command.
A027002C	<u>SDC_STA</u>	32	SD memory card controller status register Contains various statuses of SD controller as the controller is configured as the host of SD memory card.
A0270030	<u>SDC_RESP0</u>	32	SD memory card controller response register 0 Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0270034	<u>SDC_RESP1</u>	32	SD memory card controller response register 1 Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0270038	<u>SDC_RESP2</u>	32	SD memory card controller response register 2 Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A027003C	<u>SDC_RESP3</u>	32	SD memory card controller response register 3 Contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For responses of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.
A0270040	<u>SDC_CMDSTA</u>	32	SD memory card controller command status register Contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register will also be used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0270044	<u>SDC_DATSTA</u>	32	SD memory card controller data status register Contains the status of SD controller during data

Address	Name	Width	Register function
			transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register is also used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0270048	<u>SDC_CSTA</u>	32	SD memory card status register After commands with R1 and R1b response this register containing the status of the SD card, it will be used as the response interrupt source. In all register fields, logic high indicates error, and logic low indicates there is no error. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A027004C	<u>SDC_IRQMASK0</u>	32	SD memory card IRQ mask register 0 Contains parts of SD memory card interrupt mask register. See the descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CMDSTA and SDC_DATSTA.
A0270050	<u>SDC_IRQMASK1</u>	32	SD memory card IRQ mask register 1 Contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then interrupt source from register field OUT_OF_RANGE of register SDC_CSTA will be masked. '0' in some bit does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.
A0270054	<u>SDIO_CFG</u>	32	SDIO configuration register Configures functions for SDIO.
A0270058	<u>SDIO_STA</u>	32	SDIO status register Identifies if there is SDIO interrupt during the interrupt period on data line.
A0270080	<u>CLK_RED</u>	32	CLK latch configuration register Configures the MSDC sample data/response clock.

Address	Name	Width	Register function
			<i>Note: When MSDC_IOCEN[19] = 1, the host will latch response; otherwise MSDC FSM will handle the response from PAD directly.</i>
A0270098	DAT_CHECKSUM	32	MSDC Rx data checksum register Completes the checksum value of Rx read data

A0270000 MSDC_CFG SD Memory Card Controller Configuration Register 04000020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FIFOTH						VDDPD	RCDEN	DIRQEN	PINEN	DMAEN	INTEN
Type					RW						RW	RW	RW	RW	RW	RW
Reset					0	1	0	0			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCLKF								SCLKON	CRED	STDBY	CLKSRC	NOCRC	RST	MSDC	
Type	RW								RW	RW	RW	RW	RW	W1C	RW	
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Overview: For general configuration of the SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
27:24	FIFOTH	FIFOTH	FIFO threshold The register field determines when to issue a DMA request. For write transactions, DMA requests will be asserted if the number of free entries in FIFO are bigger than or equal to the value in the register field. For read transactions, DMA requests will be asserted if the number of valid entries in FIFO are bigger than or equal to the value in the register field. The register field must be set according to the setting of data transfer count in DMA burst mode. If single mode for DMA transfer is used, the register field shall be set to 0b0001. 0000: Invalid 0001: Threshold value is 1. 0010: Threshold value is 2. 0011~0111: ... 1000: Threshold value is 8. Others: Invalid
21	VDDPD	VDDPD	Controls output pin VDDPD used for power saving Output pin VDDPD controls the power for memory card. 0: Output pin VDDPD outputs logic low. The power for memory card will be turned off. 1: Output pin VDDPD outputs logic high. The power for memory card will be turned on.
20	RCDEN	RCDEN	Controls output pin RCDEN used for card identification process when the controller is for SD memory card Its output controls the pull-down resistor on the system board to connect to or disconnect from signal CD/DAT3. 0: The output pin RCDEN outputs logic low. 1: The output pin RCDEN outputs logic high.
19	DIRQEN	DIRQEN	Enables data request interrupt The register bit is used to control if data request is used as an

Bit(s)	Mnemonic	Name	Description
			interrupt source. 0: Data request is not used as an interrupt source. 1: Data request is used as an interrupt source.
18	PINEN	PINEN	Enables pin interrupt The register bit is used to control if the pin for card detection is used as an interrupt source. 0: The pin for card detection is not used as an interrupt source. 1: The pin for card detection is used as an interrupt source.
17	DMAEN	DMAEN	Enables DMA <i>Note: If DMA capability is disabled, the application software must poll the status of register MSDC_STA to check on any data transfer request. If DMA is desired, the register bit must be set up before command register is written.</i> 0: DMA request induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: DMA request induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
16	INTEN	INTEN	Enables interrupt <i>Note: If interrupt capability is disabled, the application software must poll the status of register MSDC_STA to check on any interrupt request.</i> 0: Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: Interrupt induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
15:8	SCLKF	SCLKF	Controls clock frequency of serial clock on SD bus and denotes clock frequency of SD bus serial clock as fslave and clock frequency of the SD controller as fhost which is 98.3 or 96.2 MHz <i>Note: The allowed maximum frequency of fslave is 49.15MHz.</i> While changing the clock rate, "1T clock period before change + 1T clock period after change" is required for HW signal to re-synchronize. 0000000b: fslave = $(1/2)*fhost$ 0000001b: fslave = $[1/(4*1)]*fhost$ 0000010b: fslave = $[1/(4*2)]*fhost$ 0000011b: fslave = $[1/(4*3)]*fhost$ 0000100b~11111110b: ... 1111111b: fslave = $[1/(4*255)]*fhost$
7	SCLKON	SCLKON	Serial clock always on For debugging. 0: Serial clock not always on 1: Serial clock always on
6	CRED	CRED	Rising edge data The register bit is used to determine the serial data input is latched at the falling edge or rising edge of the serial clock. The default setting is at the rising edge. If the serial data have bad timing, set the register bit to 1. When the memory card has bad timing on returned read data, set the register bit to 1. 0: Serial data input is latched at the rising edge of serial clock. 1: Serial data input is latched at the falling edge of serial clock.

Bit(s)	Mnemonic	Name	Description
5	STDBY	STDBY	<p>Standby mode</p> <p>If the module is powered down, operating clock to the module will be stopped. At the same time, the clock to card detection circuitry will also be stopped. If detection on memory card insertion and removal is desired, write 1 to the register bit. If interrupt for detection on memory card insertion and removal is enabled, the interrupt will take place whenever the memory is inserted or removed.</p> <p>0: Standby mode is disabled. 1: Standby mode is enabled.</p>
4:3	CLKSRC	CLKSRC	<p>Specifies which clock is used as source clock of memory card</p> <p>00 : MPLL/5.5MHz clock 01 : MPLL/7MHz clock (this divider is default off, before switch controller source clock to this clock, you should switch controller source clock to 26Mhz first, and then enable divider MPLL/7, after a moment you can switch controller source clock to MPLL/ 7 to void source clock glitch. For detail setting please refer to configsys document) 10 : MPLL/8MHz clock 11 : MPLL/10MHz clock</p> <p>For phone 00 : 94.5MHz clock Need to keep BT_APP_DIV_EN= 1'b0 in CLK_CONDA[15]. 01 : 74.3MHz clock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 : 65MHz clock 11 : Forbidden</p> <p>For BT app. 00 : Forbidden 01 : 89.1MHz clock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 : 78MHz clock 11 : 62.4MHz clock NOTE: Need to set POWERFUL_DIV_EN2 = 1'b1 first in CLK_CONDA[9].</p>
2	NOCRC	NOCRC	<p>Disable CRC</p> <p>'1' indicates data transfer without CRC is desired. For write data block, the data are transmitted without CRC. For read data block, CRC will not be checked. For testing purpose.</p> <p>0: Data transfer with CRC is desired. 1: Data transfer without CRC is desired.</p>
1	RST	RST	<p>Software reset</p> <p>Writing 1 to the register bit will cause internal synchronous reset of SD controller but will not reset register settings, RST should only be set when RST equal to 0.</p> <p>0: Read 0 stands for the reset process is finished. 1: Write 1 to reset SD controller.</p>
0	MSDC	MSDC	<p>Configures the controller as SD memory card mode</p> <p>CLK/CMD/DAT line is pulled low when SD memory card mode is disable.</p> <p>0: Disable SD memory card 1: Enable SD memory card</p>

A0270004 MSDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	BUSY	FIFOC LR							FIFOCNT	INT	DRQ	BE	BF
Type	R	W1C							RO	RO	RO	RO	RO
Reset	0	0							0	0	0	0	0

Overview: The register contains the status of FIFO, interrupts and data requests.

Bit(s)	Mnemonic	Name	Description
15	BUSY	BUSY	<p>Status of the controller</p> <p>If the controller is in busy state, the register bit will be 1; otherwise 0. 0: The controller is in busy state. 1: The controller is in idle state.</p>
14	FIFOC LR	FIFOC LR	<p>Clears FIFO</p> <p>Writing 1 to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.</p> <p>0: Read 0 stands for the FIFO clear process is finished. 1: Write 1 to clear the content of FIFO clear and reset the status of FIFO controller.</p>
7:4	FIFOCNT	FIFOCNT	<p>FIFO count</p> <p>The register field shows how many valid entries are there in FIFO.</p> <p>0000: 0 valid entry in FIFO 0001: 1 valid entry in FIFO 0010: 2 valid entries in FIFO 0011~0111: ... 1000: 8 valid entries in FIFO</p>
3	INT	INT	<p>Indicates if there is any interrupt existing</p> <p>When there is interrupt existing, the register bit will still be active even if register bit INTEN in register MSDC_CFG is disabled. The SD controller can interrupt MCU by issuing interrupt request to the interrupt controller, or the software/application will poll the register endlessly to check if there is any interrupt request existing in the SD controller. When register bit INTEN in register MSDC_CFG is disabled, the second method is used. For read commands, it is possible that time-out error takes place. The software can read the status register to check if the time-out error takes place without OS time tick support or data request asserted.</p> <p><i>Note: The register bit will be cleared when register MSDC_INT is read.</i></p> <p>0: No interrupt request existing. 1: Interrupt request exists.</p>
2	DRQ	DRQ	<p>Indicates if any data transfer is required</p> <p>When a data transfer is required, the register bit will still be active even if register bit DIRQEN in register MSDC_CFG is disabled. Data transfer can be achieved by DMA channel alleviating MCU loading, or by polling the register bit to check if any data transfer is requested. When register bit DIRQEN in register MSDC_CFG is disabled, the second method is used.</p> <p>0: No DMA request existing. 1: DMA request exists.</p>
1	BE	BE	<p>Indicates if FIFO in SD controller is empty</p> <p>0: FIFO in SD controller is not empty. 1: FIFO in SD controller is empty.</p>
0	BF	BF	<p>Indicates if FIFO in SD controller is full</p> <p>0: FIFO in SD controller is not full. 1: FIFO in SD controller is full.</p>

A0270008 MSDC_INT SD Memory Card Controller Interrupt Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SDIOI RQ	SDR1 BIRQ		SDMC IRQ	SDDA TIRQ	SDCM DIRQ	PINIR Q	DIRQ
Type									RC	RC		RC	RC	RC	RC	RC
Reset									0	0		0	0	0	0	0

Overview: The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.

Bit(s)	Mnemonic	Name	Description
7	SDIOIRQ	SDIOIRQ	<p>SDIO interrupt</p> <p>The register bit indicates if there is any interrupt for SDIO existing. Whenever an interrupt for SDIO exists, the register bit will be set to 1 if the interrupt is enabled. It will be reset when the register is read.</p> <p>0: No SDIO interrupt 1: Interrupt for SDIO exists.</p>
6	SDR1BIRQ	SDR1BIRQ	<p>SD R1b response interrupt</p> <p>The register bit will be active when a SD command with R1b response is finished and the DAT0 line is transitioned from busy to idle state. Single block write commands with R1b response will cause interrupt when the command is completed either successfully or with CRC error. However, multi-block write commands with R1b response do not cause interrupt because multi-block write commands are always stopped by STOP_TRANS commands. STOP_TRANS commands (with R1b response) behind multi-block write commands will cause interrupt. Single block read command with R1b response will cause interrupt when the command is completed, but multi-block read commands do not.</p> <p><i>Note: STOP_TRANS commands (with R1b response) behind multi-block read commands will cause interrupt.</i></p> <p>0: No interrupt for SD R1b response. 1: Interrupt for SD R1b response exists.</p>
4	SDMCIRQ	SDMCIRQ	<p>SD memory card interrupt</p> <p>The register bit indicates if there is any interrupt for SD memory card existing. Whenever an interrupt for SD memory card exists, i.e. any bit in register SDC_CSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read.</p> <p><i>Note: This bit will not trigger MSDC hardware interrupt.</i></p> <p>0: No SD memory card interrupt 1: SD memory card interrupt exists.</p>
3	SDDATIRQ	SDDATIRQ	<p>SD bus DAT interrupt</p> <p>The register bit indicates if there is any interrupt for SD DAT line existing. Whenever interrupt for SD DAT line exists, i.e. any bit in register SDC_DATSTA is active, the register bit will be set to 1 if</p>

Bit(s)	Mnemonic	Name	Description
2	SDCMDIRQ	SDCMDIRQ	<p>interrupt is enabled. It will be reset when the register is read.</p> <p>0: No SD DAT line interrupt 1: SD DAT line interrupt exists.</p> <p>SD bus CMD interrupt</p> <p>The register bit indicates if there is any interrupt for SD CMD line existing. Whenever interrupt for SD CMD line exists, i.e. any bit in the register SDC_CMDSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read.</p> <p>0: No SD CMD line interrupt 1: SD CMD line interrupt exists.</p>
1	PINIRQ	PINIRQ	<p>Pin change interrupt</p> <p>The register bit indicates if there is any interrupt for memory card insertion/removal existing. Whenever the memory card is inserted or removed and card detection interrupt is enabled, i.e. register bit PINEN in register MSDC_CFG is set to 1, the register bit will be set to 1. It will be reset when the register is read.</p> <p>0: Otherwise 1: Card is inserted or removed.</p>
0	DIRQ	DIRQ	<p>Data request interrupt</p> <p>The register bit indicates if there is any interrupt for data request existing. Whenever data request exists and data request as an interrupt source is enabled, i.e. register bit DIRQEN in register MSDC_CFG is set to 1, the register bit will be active. It will be reset when being read. For software, data requests can be recognized by polling register bit DRQ or by data request interrupt. Data request interrupts will be generated every FIFOTHD data transfers.</p> <p>0: No data request interrupt 1: Data request interrupt occurs.</p>

A027000C MSDC_PS SD Memory Card Pin Status Register 0000008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								CMD	DAT							
Type								RO	RO							
Reset								0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CDDEBOUNCE											PINCHG	PINO	POEN0	PIENO	CDEN
Type	RW											RC	RO	RW	RW	RW
Reset	0	0	0	0								0	1	0	0	0

Overview: The register is used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull-down when the system is powered on. The scheme of card detection for MS is the same as that for SD. To detect card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.

Bit(s)	Mnemonic	Name	Description
24	CMD	CMD	Memory card/SDIO card/MMC card command lines
23:16	DAT	DAT	Memory card/SDIO card/MMC card data lines
15:12	CDDEBOUN	CDDEBOUNCE	Specifies the time interval for card detection de-bounce

Bit(s)	Mnemonic	Name	Description
	CE		Default value: 0. It means the de-bounce interval is 32-cycle time at 32kHz. The interval can extend one cycle time at 32kHz by increasing the counter by 1.
4	PINCHG	PINCHG	Pin change The register bit indicates the status of card insertion/removal. If the memory card is inserted or removed, the register bit will be set to 1 no matter pin change interrupt is enabled or not. It will be cleared when the register is read. 0: Otherwise 1: Card is inserted or removed.
3	PINO	PINO	Shows the value of input pin for card detection 0: The value of input pin for card detection is logic low. 1: The value of input pin for card detection is logic high.
2	POENO	POENO	Controls output of input pin for card detection 0: Output of input pin for card detection is disabled. 1: Output of input pin for card detection is enabled.
1	PIENO	PIENO	Controls input pin for card detection 0: Input pin for card detection is disabled. 1: Input pin for card detection is enabled.
0	CDEN	CDEN	Enables card detection The register bit is used to enable or disable card detection. 0: Card detection is disabled. 1: Card detection is enabled.

A0270010 MSDC_DAT SD Memory Card Controller Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to read/write data from/to FIFO inside the SD controller. Data access unit: 32 bits.

Bit(s)	Mnemonic	Name	Description
31:0	DATA	DATA	Reads/Writes data from/to FIFO inside SD controller Data access unit: 32 bits

A0270014 MSDC_IOCON SD Memory Card Controller IO Control Register 010000C3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							SAMPLEDLY		FIXDLY		SAMPON	CRCDS	CMDS	INTLH		DSW
Type							RW		RW		RW	RW	RW	RW		RW
Reset							0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDRE					HIGH SPEED	DMABURST	SRCFG1	SRCFG0	ODCCFG1				ODCCFG0		

Type	RW					D										
Reset	0					0	0	0	1	1	0	0	0	0	1	1

Overview: The register specifies the output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current depends on the PAD type selected for the chip.

Bit(s)	Mnemonic	Name	Description
25:24	SAMPLEDL Y	SAMPLEDLY	Used for SW to select the turn-around delay cycle between write data end bit and CRC status for SD card 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
23:22	FIXDLY	FIXDLY	Used for SW to select the delay cycle after clock fix high for the host controller to SD card 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
21	SAMPON	SAMPON	Data sample enabling always on The bit is suggested to be set to 1 when the feedback clock is used and to 0 when the multiple phase clock is used. 0: Data sample enabling not always on 1: Data sample enabling always on
20	CRCDIS	CRCDIS	Switches off data CRC check for SD read data 0: CRC check is on. 1: CRC check is off.
19	CMDSEL	CMDSEL	Determines whether the host should delay 1-T to latch response from card 0: Host latches response without 1-T delay 1: Host latches response with 1-T delay.
18:17	INTLH	INTLH	Selects latch timing for SDIO multi-block read interrupt 00: Host latches INT at the second backend clock after the end bit of the current data block from card is received. (Default) 01: Host latches INT at the first backend clock after the end bit of the current data block from card is received. 10: Host latches INT at the second backend clock after the end bit of the current data block from card is received. 11: Host latches INT at the third backend clock after the end bit of the current data block from card is received.
16	DSW	DSW	Determines whether the host should latch data with 1-T delay or not For SD card, this bit is suggested to be 0. For MSPRO cards, it is suggested to be 1. 0: Host latches the data with 1-T delay. 1: Host latches the data without 1-T delay.
15	CMDRE	CMDRE	Determines whether the host should latch response token (sent from card on CMD line) at rising edge or falling edge of serial clock (T.B.D this bit is un-useful) 0: Host latches response at rising edge of serial clock.

Bit(s)	Mnemonic	Name	Description
10	HIGH_SPEE D	HIGH_SPEED	1: Host latches response at falling edge of serial clock. For high-speed mode when internal sample clock is used High-speed mode means the SD/MMC serial bus clock rate is bigger than 25MHz. The default speed mode means that the SD/MMC serial bus clock rate is bigger than 25MHz. 0: Default speed 1: High speed
9:8	DMABURST	DMABURST	Used for SW to select burst type when data are transferred by DMA <i>Note: Only single mode can support non-4N bytes data transfer in read operation.</i> 00: Single mode 01: 4-beat incrementing burst 10: 8-beat incrementing burst 11: Reserved
7	SRCFG1	SRCFG1	Output driving capability for pins DAT0, DAT1, DAT2 and DAT3 0: Fast slew rate 1: Slow slew rate
6	SRCFG0	SRCFG0	Output driving capability for pins CMD/BS and SCLK 0: Fast slew rate 1: Slow slew rate
5:3	ODCCFG1	ODCCFG1	Output driving capability for pins DAT0, DAT1, DAT2 and DAT3 000: 4mA 001: 8mA 010: 12mA 011: 16mA
2:0	ODCCFG0	ODCCFG0	Output driving capability for pins CMD/BS and SCLK 000: 4mA 001: 8mA 010: 12mA 011: 16mA

A0270018 MSDC_IOCON1 SD Memory Card Controller IO Control Register 1 00022022

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		PRCFG_CK	PRVAL_CK			PRCFG_CM	PRVAL_CM			PRCFG_DA	PRVAL_DA					
Type		RW	RW			RW	RW			RW	RW					
Reset		0	1 0			0	0 0			0	1 0					

Bit(s)	Mnemonic	Name	Description
14	PRCFG_CK	PRCFG_CK	Pull-up/down register configuration for pin CK Default value: 0 0: Pull-up resistor in the I/O pad of pin CK is enabled. 1: Pull-down resistor in the I/O pad of pin CK is enabled.
13:12	PRVAL_CK	PRVAL_CK	Pull-up/down register value for pin CLK Default value: 10

Bit(s)	Mnemonic	Name	Description
10	PRCFG_CM	PRCFG_CM	00: Pull-up/down resistor in the I/O pad of pin CLK are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CLK value is 23.5k. Pull-up/down register configuration for pin CM Default value is 0. 0: Pull-up resistor in the I/O pad of pin CM is enabled. 1: Pull-down resistor in the I/O pad of pin CM is enabled.
9:8	PRVAL_CM	PRVAL_CM	Pull-up/down register value for pin CMD/BS Default value: 00 00: Pull-up/down resistor in the I/O pad of pin CMD/BS are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 23.5k.
6	PRCFG_DA	PRCFG_DA	Pull-up/down register configuration for pin DAT0, DAT1, DAT2 and DAT3 Default value: 0 0: Pull-up resistor in the I/O pad of pin DAT is enabled. 1: Pull-down resistor in the I/O pad of pin DAT is enabled.
5:4	PRVAL_DA	PRVAL_DA	Pull-up/down register value for pin DAT0, DAT1, DAT2 and DAT3 Default value: 10 00: Pull-up/down resistor in the I/O pad of pin DAT are all disabled. 01: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 10: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 11: Pull-up/down resistor in the I/O pad of pin DAT value is 23.5k.

A0270020 SDC_CFG SD Memory Card Controller Configuration Register 00008000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIOC								WDOD				SDIO		MDLE N	SIEN
Type	RW								RW				RW		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BSYDLY				BLKLEN											
Type	RW				RW											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to configure the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
31:24	DIOC	DIOC	Data time-out counter The period from finish of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit of 65,536 serial clock. See the register field descriptions of register

Bit(s)	Mnemonic	Name	Description
			bit RDINT for reference. 00000000: Extend 65,536 more serial clock cycles 00000001: Extend 65,536x2 more serial clock cycles 00000010: Extend 65,536x3 more serial clock cycles 00000011~11111110: ... 11111111: Extend 65,536x 256 more serial clock cycles
23:20	WDOD	WDOD	Write data output delay The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
19	SDIO	SDIO	Enables SDIO 0: Disable SDIO mode 1: Enable SDIO mode
17	MDLEN	MDLEN	Enables multiple data line The register can be enabled only when SD memory card is applied and detected by software application. It is the responsibility of the application to program the bit correctly when an multi-media card is applied. If an multi-media card is applied and 4-bit data line is enabled, the 4 bits will be output every serial clock. Therefore, data integrity will fail. 0: Disable 4-bit data line 1: Enable 4-bit data line
16	SIEN	SIEN	Enables serial interface It should be enabled as soon as possible before any command. 0: Disable serial interface for SD 1: Enable serial interface for SD
15:12	BSYDLY	BSYDLY	Only valid for the commands with R1b response If the command has a response of R1b type, the SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if the operation in SD memory card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, the controller will abandon the detection. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
11:0	BLKLEN	BLKLEN	Block length The register field is used to define the length of one block in unit of byte in a data transaction. The maximum value of block length is 2,048 bytes. 000000000000: Reserved 000000000001: Block length is 1 byte. 000000000010: Block length is 2 bytes. 000000000011~011111111110: ...

Bit(s)	Mnemonic	Name	Description
			011111111111: Block length is 2,047 bytes. 100000000000: Block length is 2,048 bytes.

A0270024 SDC_CMD SD Memory Card Controller Command Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																CMDFAIL
Type																RW
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTC	STOP	RW	DTYPE	IDRT		RSPTYP	BREAK		CMD						
Type	RW	RW	RW	RW	RW		RW	RW		RW						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto the SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.

Bit(s)	Mnemonic	Name	Description
16	CMDFAIL	CMDFAIL	If 4-bit SDIO mode is enabled and when CMD/DAT error occurs, set this bit to select whether to "wait stop command" or "wait data state machine idle". 0: Wait stop command 1: Wait data state machine idle
15	INTC	INTC	Indicates if the command is GO_IRQ_STATE If the command is GO_IRQ_STATE, the period between command token and response token will not be limited. 0: The command is not GO_IRQ_STATE. 1: The command is GO_IRQ_STATE.
14	STOP	STOP	Indicates if the command is a stop transmission command 0: The command is not a stop transmission command. 1: The command is a stop transmission command.
13	RW	RW	Defines the command is a read command or write command The register bit is valid only when the command causes a transaction with data token. 0: The command is a read command. 1: The command is a write command.
12:11	DTYPE	DTYPE	Defines data token type for the command 00: No data token for the command 01: Single block transaction 10: Multiple block transaction, i.e. the command is a multiple block read or write command. 11: Stream operation. It can only be used when an multi-media card is applied.
10	IDRT	IDRT	Identification response time The register bit indicates if the command has a response with NID (i.e. 5 serial clock cycles as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 1.0)

Bit(s)	Mnemonic	Name	Description
9:7	RSPTYP	RSPTYP	<p>response time. The register bit is valid only when the command has a response token. Thus the register bit must be set to 1 for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD).</p> <p>0: Otherwise 1: The command has a response with NID response time.</p> <p>Defines response type for the command</p> <p>For commands with R1 and R1b response, register SDC_CSTA (not SDC_STA) updates after response token is received. This register SDC_CSTA contains the status of the SD, and it can be used as a response interrupt source.</p> <p><i>Note: If CMD7 is used with all 0's RCA, then RSPTYP must be "000". Command "GO_TO_IDLE" also has RSPTYP="000".</i></p> <p>000: There is no response for the command, e.g. broadcast command without response and GO_INACTIVE_STATE command. 001: The command has R1 response. R1 response token is 48-bit. 010: The command has R2 response. R2 response token is 136-bit. 011: The command has R3 response. Even though R3 is 48-bit response, it does not contain CRC checksum. 100: The command has R4 response. R4 response token is 48-bit. (only for MMC) 101: The command has R5 response. R5 response token is 48-bit. (only for MMC) 110: The command has R6 response. R6 response token is 48-bit. 111: The command has R1b response. If the command has a response of R1b type, SD controller must monitor the data line 0 for card busy status from the bit time that is 2 or 4 serial clock cycles after the command end bit to check if the operation in SD memory card has finished. There are two cases for detection of card busy status. The first case is that the host stops the data transmission during an active write data transfer. The card will assert busy signal after the stop transmission command end bit followed by 4 serial clock cycles. The second case is that the card is in idle state or receiving a stop transmission command between data blocks when multiple block write command is in progress. The register bit will be valid only when the command has a response token.</p>
6	BREAK	BREAK	<p>Aborts pending MMC GO_IRQ_MODE command</p> <p>It is only valid for a pending GO_IRQ_MODE command waiting for MMC interrupt response.</p> <p>0: Other fields are valid. 1: Break a pending MMC GO_IRQ_MODE command in the controller. Other fields are invalid.</p>
5:0	CMD	CMD	<p>SD memory card command</p> <p>Total 6 bits.</p>

A0270028 SDC_ARG SD Memory Card Controller Argument Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ARG[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ARG[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains the argument of the SD memory card command.

Bit(s)	Mnemonic	Name	Description
31:0	ARG	ARG	Contains argument of the SD memory card command.

A027002C SDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WP											FEDATBUS	FECMDBUS	BEDATBUS	BECMDBUS	BESDCBUS
Type	RO											Y	Y	Y	Y	Y
Reset	0											0	0	0	0	0

Overview: The register contains various statuses of SD controller as the controller is configured as the host of SD memory card.

Bit(s)	Mnemonic	Name	Description
15	WP	WP	Detects the status of write protection switch on SD memory card The register bit shows the status of write protection switch on SD memory card. There is no default reset value. Pin WP (Write Protection) is only useful when the controller is configured for SD memory card. 1: Write protection switch on, i.e. memory card is desired to be write-protected. 0: Write protection switch off, i.e. memory card is writable.
4	FEDATBUS	FEDATBUSY	Indicates if there is any transmission going on DAT line on SD bus This bit indicates directly the CMD line at card clock domain. For those commands without data but still involving DAT line, the register bit is useless. For example, if an erase command is issued, checking if the register bit is 0 before issuing the next command with data will not guarantee that the controller is idle. In this case, use register bit BESDCBUSY. 0: No transmission is going on DAT line on SD bus. 1: There is transmission going on DAT line on SD bus.
3	FECMDBUS	FECMDBUSY	Indicates if there is any transmission going on CMD line on SD bus This bit indicates directly the CMD line at card clock domain. 0: No transmission is going on CMD line on SD bus. 1: There is transmission going on CMD line on SD bus.
2	BEDATBUS	BEDATBUSY	Indicates if there is any transmission going on DAT line on SD bus 0: Backend SDC controller gets the info that no transmission is going on DAT line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on DAT line on SD bus.
1	BECMDBUS	BECMDBUSY	Indicates if there is any transmission going on CMD line on SD bus

Bit(s)	Mnemonic	Name	Description
Y			bus This bit shows backend controller's CMD busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SDC controller gets the info that no transmission is going on CMD line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on CMD line on SD bus.
0	BESDCBUS	BESDCBUSY	Indicates if SD controller is busy, i.e. is there any transmission going on CMD or DAT line on SD bus This bit shows backend controller's SDC busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SD controller is idle. 1: Backend SD controller is busy.

A0270030 SDC_RESP0 SD Memory Card Controller Response Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[31:0][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[31:0][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[31:0]	RESP_31_0	

A0270034 SDC_RESP1 SD Memory Card Controller Response Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[63:32][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[63:32][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[63:32]	RESP_63_32	

A0270038 SDC_RESP2 SD Memory Card Controller Response Register 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[95:64][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[95:64][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[95:64]	RESP_95_64	

A027003C SDC_RESP3 SD Memory Card Controller Response Register 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[127:96][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[127:96][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, i.e. response of commands ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.

Bit(s)	Mnemonic	Name	Description
31:0	RESP[127:96]	RESP_127_96	

A0270040 SDC_CMDSTA SD Memory Card Controller Command Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RSPC RCER R	CMDT O	CMDR DY
Type														RC	RC	RC
Reset														0	0	0

Overview: The register contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The

register can also be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
2	RSPRCER R	RSPRCERR	CRC error on CMD detected '1' indicates the SD controller detects a CRC error after reading a response from the CMD line. 0: Otherwise 1: SD controller detects a CRC error after reading a response from the CMD line.
1	CMDTO	CMDTO	Time-out on CMD detected '1' indicates the SD controller detects a time-out condition while waiting for a response on the CMD line. 0: Otherwise 1: SD controller detects a timeout condition while waiting for a response on the CMD line.
0	CMDRDY	CMDRDY	For command without response, the register bit will be 1 once the command is completed on SD bus For command with response, the register bit will be 1 whenever the command is issued onto the SD bus and its corresponding response is received without CRC error. 0: Otherwise 1: Command with/without response is finished successfully without CRC error.

A0270044 SDC DATSTA SD Memory Card Controller Data Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DATCRCERR								DATT O	BLKD ONE
Type							RC								RC	RC
Reset							0	0	0	0	0	0	0	0	0	0

Overview: The register contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register can be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
9:2	DATRCER R	DATRCERR	CRC error on DAT detected '1' indicates that the SD controller detects a CRC error for bit n after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. 0: Otherwise 1: SD controller detects a CRC error after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. Note: n is 7 ~ 1 for 8-bits mode. Each bit is read and cleared

Bit(s)	Mnemonic	Name	Description
1	DATTO	DATTO	<i>individually.</i> Time-out on DAT detected A '1' indicates that the SD controller detects a time-out condition while waiting for data token on the DAT line. 0: Otherwise 1: SD controller detects a time-out condition while waiting for data token on the DAT line.
0	BLKDONE	BLKDONE	Indicates the status of data block transfer 0: Otherwise 1: A data block is successfully transferred.

A0270048 SDC CSTA SD Memory Card Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSTA [31:0][31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSTA [31:0][15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: After commands with R1 and R1b respond this register containing the status of the SD card, it will be used as a response interrupt source. In all register fields, logic high indicates error, and logic low indicates no error. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
31:0	CSTA [31:0]	CSTA_31_0	CSTA31: OUT_OF_RANGE. The command's argument is out of the allowed range for this card. CSTA30: ADDRESS_ERROR. A misaligned address that does not match the block length is used in the command. CSTA29: BLOCK_LEN_ERROR. The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length. CSTA28: ERASE_SEQ_ERROR. An error in the sequence of erase commands occurs. CSTA27: ERASE_PARAM. An invalid selection of write-blocks for erase occurs. CSTA26: WP_VIOLATION. Attempt to program a write-protected block. CSTA25: Reserved. Return to 0. CSTA24: LOCK_UNLOCK_FAILED. Set when a sequence or password error is detected in lock/unlock card command or if there is an attempt to access a locked card. CSTA23: COM_CRC_ERROR. The CRC check of the previous command fails. CSTA22: ILLEGAL_COMMAND. Command not legal for the card state. CSTA21: CARD_ECC_FAILED. Card internal ECC is applied but fails to correct the data. CSTA20: CC_ERROR. Internal card controller error.

- CSTA19: ERROR. A general or unknown error occurs during the operation.
- CSTA18: UNDERRUN. The card cannot sustain data transfer in stream read mode.
- CSTA17: OVERRUN. The card cannot sustain data programming in stream write mode.
- CSTA16: CID/CSD_OVERWRITE. It can be either one of the following errors: 1) The CID register has been already written and cannot be overwritten; 2) The read-only section of the CSD does not match the card; 3) An attempt to reverse the copy (set as the original) or permanent WP (unprotected) bits is made.
- CSTA[15: 4]: Reserved. Return to 0.
- CSTA3: AKE_SEQ_ERROR. Error in the sequence of authentication process
- CSTA[2: 0]: Reserved. Return to 0.

A027004C SDC_IRQMASK0 SD Memory Card IRQ Mask Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [31:0][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [31:0][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. See the register descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from registers SDC_CMDSTA and SDC_DATSTA.

Bit(s)	Mnemonic	Name	Description
31:0	IRQMASK [31:0]	IRQMASK_31_0	

A0270050 SDC_IRQMASK1 SD Memory Card IRQ Mask Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [63:32][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [63:32][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt

sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then the interrupt source from register field OUT_OF_RANGE of register SDC_CSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.

Bit(s)	Mnemonic	Name	Description
31:0	IRQMASK [63:32]	IRQMASK_63_32	

A0270054 SDIO_CFG SDIO Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											DISSEL		INTCSEL	DSBSEL		INTEN
Type											RW		RW	RW		RW
Reset											0		0	0		0

Overview: The register is used to configure functions for SDIO.

Bit(s)	Mnemonic	Name	Description
5	DISSEL	DISSEL	Selects data block interrupt source 0: The host detects SDIO interrupt during interrupt period between two data blocks of multiple block data access. 1: The host ignores SDIO interrupt during interrupt period between two data blocks of multiple block data access.
3	INTCSEL	INTCSEL	Selects interrupt control 0: The host detects DAT1 low as SDIO interrupt. 1: The host detects DAT3/DAT2/DAT1/DAT0 4'b1101 as SDIO interrupt.
2	DSBSEL	DSBSEL	Selects data block start bit 0: Use data line 0 as start bit of data block. Other data lines are ignored. 1: Start bit of a data block is received only when all data line 0-3 become low.
0	INTEN	INTEN	Enables interrupt for SDIO 0: Disable 1: Enable

A0270058 SDIO_STA SDIO Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																IRQ
Type																RO
Reset																0

Overview: This register is used to identify if there is SDIO interrupt during the interrupt period on data line.

Bit(s)	Mnemonic	Name	Description
0	IRQ	IRQ	<p>SDIO interrupt exists on the data line.</p> <p>For example, when in the interrupt period or the 1-bit data line mode and DAT1/5 goes low from high, this bit will become 1 from 0. If DAT1/5 goes high from low, this bit will become 0 from 1.</p> <p>0: There is no SDIO interrupt existing on the data line. 1: There is SDIO interrupt existing on the data line.</p>

A0270080 CLK_RED CLK Latch Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			CMD_RED													
Type			RW													
Reset			0													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DAT_RED						CLKP_AD_R	CLK_LATC						
Type			RW						ED	H						
Reset			0						0	0						

Overview: The register is used to configure the MSDC sample data/response clock. Note that only when MSDC_IOCON[19] = 1 will the host latch response; otherwise MSDC FSM will handle the response from PAD directly.

Bit(s)	Mnemonic	Name	Description
29	CMD_RED	CMD_RED	<p>Determines the command response from card output is latched at falling edge or rising edge of internal clock</p> <p>Only effective when CLK_LATCH = 1</p> <p>0: Internal clock rising edge to latch response 1: Internal clock falling edge to latch response</p>
13	DAT_RED	DAT_RED	<p>Determines the input data from card output is latched at falling edge or rising edge of internal sample clock</p> <p>Only effective when CLK_LATCH = 1</p> <p>0: Internal clock rising edge to latch data 1: Internal clock falling edge to latch data</p>
7	CLKPAD_R ED	CLKPAD_RED	<p>Determines the input data from card is latched at falling edge or rising edge of the feedback clock from pad</p> <p>The suggested setting is 0 when SD serial clock is lower than 25MHz. The suggestion setting is 1 when SD serial clock is higher than 25MHz. The suggestion setting is 0 for MMC card no matter the serial clock rate is high speed or default speed. Only effective when CLK_LATCH = 0.</p> <p>0: Internal feedback clock rising edge to latch data/response 1: Internal feedback clock falling edge to latch data/response</p>
6	CLK_LATCH	CLK_LATCH	<p>Determines which clock to latch data from card</p> <p>The suggested setting is 1 if SCLKF in register field MSDC_CFG is 0x0. Otherwise, the suggested setting is 0.</p> <p>0: Internal feedback clock is used to latch data/response from card.</p>

Bit(s)	Mnemonic	Name	Description
			1: Internal clock is used to latch data/response from card.

A0270098 DAT_CHECKSUM MSDC Rx Data Checksum Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAT_CHECKSUM[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT_CHECKSUM[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to complete the checksum value of Rx read data

Bit(s)	Mnemonic	Name	Description
31:0	DAT_CHEC KSUM	DAT_CHECKSUM	The checksum algorithm is 32-bit XOR.

3.18 BTIF

3.18.1 General Description

Bluetooth Interface (BTIF) is designed in SOC (BT+GSM) as the UART interface between the BT chip and baseband chip. As in the UART design, BTIF is an APB slave which transmits or receives data by MCU access or through DMA/VFIFO.

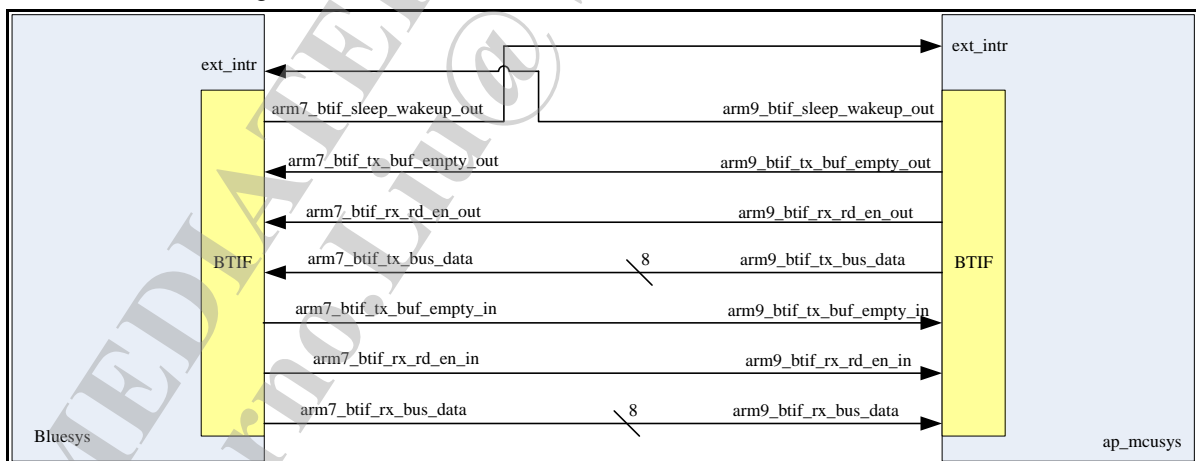


Figure 48. Interface connection between BT and baseband system

3.18.2 Register Definition

BTIF+0000h Rx Buffer Register **BTIF_RBR**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												RBR[7:0]				
Type												RO				

RBR Rx buffer register. A read-only register. The received data can be read by accessing this register. This register is valid only when BTIF_FAKELCR[7] (0x0C) is 0.

BTIF+0000h Tx Holding Register **BTIF_THR**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												THR[7:0]				
Type												WO				

THR Tx holding register. A write-only register. The data to be transmitted are written to this register and sent to the Bluetooth via BTIF. This register is valid only when BTIF_FAKELCR[7] (0x0C) is 0.

BTIF+0004h Interrupt Enable Register **BTIF_IER**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															TXEEN	RXFEN
Type															W/R	W/R
Reset															0	0

This register is valid only when BTIF_FAKELCR[7] is 0.

TXEEN Enables Tx empty interrupt. When set to 1, an interrupt will be generated if the Tx holding register is empty.

0 No interrupt will be generated if the Tx holding register is empty.

1 An interrupt will be generated if the Tx holding register is empty

RXFEN Enables Rx full interrupt. When set to 1, an interrupt will be generated if the Rx buffer contains data.

0 No interrupt will be generated if the Rx buffer contains data.

1 An interrupt will be generated if the Rx buffer contains data.

BTIF+0008h Interrupt Identification Register **BTIF_IIR**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													ID2	ID1	ID0	NINT
Type													RO	RO	RO	RO
Reset													0	0	0	1

This register is valid only when BTIF_FAKELCR (0x0C) is not 0xBF.

IIR Identifies if there are pending interrupts. The following table lists the IIR[5:0] codes associated with the possible interrupts:

Table 52. IIR[5:0] codes associated with the possible interrupts

IIR[3:0]	Priority level	Interrupt	Source
0001	-	No pending interrupt	
0100	1	Rx data received	Rx data received
1100	2	Rx data time-out	Time-out on character in Rx buffer
0010	3	Tx holding register empty	Tx holding register empty.

Rx data received interrupt

A Rx received interrupt (IIR[3:0] = 0x04) is generated when RXFEN (IER[0]) is set and Rx data are placed in the Rx buffer register. The interrupt is cleared by reading the Rx buffer register.

Rx data time-out interrupt

The Rx data time-out interrupt will be generated if all of the following conditions are applied:

1. Rx buffer is empty.
2. The most recent character is received longer than (RTOCNT*bclk period*4).
3. RXFEN (IER[0]) is set to 1.

The time-out timer is restarted upon receipt of a new byte from the Rx shift register. This interrupt is only valid while VFIFO is used. This register is cleared by reading the VFIFO status register (0x4C).

Tx holding register empty

A Tx holding register empty interrupt (IIR[3:0] = 0x02) is generated when TXEEN (IER[1]) is set and no data are placed in the Tx holding register. This interrupt is cleared by writing data into BTIF_THR (0x00).

BTIF+0008h FIFO_CTRL

**BTIF_FIFOC
RL**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														CLRT	CLRR	
Type														WO	WO	
Reset														0	0	

This register is valid only when BTIF_FAKELCR (0x0C) is not 0xBF.

CLRT Clears transmit FIFO. This bit is self-clearing.
0 Leave Tx FIFO intact.
1 Clear all the bytes in Tx FIFO.

CLRR Clears receive FIFO. This bit is self-clearing.
0 Leave Rx FIFO intact.
1 Clear all the bytes in Rx FIFO.

BTIF+000Ch FAKE LCR

**BTIF_FAKE
CR**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										FAKELCR[7:0]						
Type										R/W						
Reset										0	0	0	0	0	0	0

FAKELCR This register is added to synchronize the software control method of UART. When FAKELCR[7] is 1, RBR(0x00), THR(0x00) and IER(0x04) will not be readable/writable. When FAKELCR is 0xBF, RBR(0x00), THR(0x00), IER(0x04), IIR(0x08) and LSR(0x14) will not be readable/writable.

BTIF+0014h Line Status Register

BTIF_LSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										TEMT	THRE					DR
Type										RO	RO					RO
Reset										1	1					0

LSR Line status register. Readable when LCR != 0xBF.

TEMT Tx holding register is empty.
0 Empty conditions are not met.
1 This bit is set when the Tx holding register is empty.

- THRE** Indicates if Tx FIFO is reduced to its trigger level
- 0** Reset whenever the contents of Tx FIFO are more than its trigger level (FIFOs are enabled)
 - 1** Set whenever the contents of Tx FIFO are reduced to its trigger level (FIFOs are enabled)
- DR** Data Ready
- 0** Cleared by reading the Rx buffer.
 - 1** Set by the Rx buffer becoming full.

BTIF+0048h Sleep Enable Register

BTIF_SLEEP_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																R/W
Reset																0

- SLEEP_EN** For sleep mode issue
- 0** Does not deal with sleep mode indication signal
 - 1** Activate flow control according to software initial settings when the chip enters the sleep mode. Release hardware flow when the chip wakes up.

BTIFn+004Ch DMA Enable Register

BTIF_DMA_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														TO_CNT_UTORST	TX_DMA_EN	RX_DMA_EN
Type														R/W	R/W	R/W
Reset														0	0	0

- RX_DMA_EN** RX_DMA mechanism enabling signal
- 0** Does not use DMA in Rx.
 - 1** Use DMA in Rx. When this register is enabled, the flow control is based on the DMA threshold and generates a time-out interrupt

- TX_DMA_EN** TX_DMA mechanism enabling signal
- 0** Does not use DMA in Tx.
 - 1** Use DMA in Tx. When this register is enabled, the flow control is based on the DMA threshold and generates a time-out interrupt for DMA.

TO_CNT_AUTORST Time-out counter auto reset register

0 After Rx time-out takes place, the software shall reset the interrupt by reading BTIF 0x4C.

1 The time-out counter will be auto reset.

BTIF+0054h Rx Time-out Count

BTIF_RTOCNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																RTOCNT[7:0]
Type																R/W
Reset																0x40

RTOCNT Used for Rx time-out interrupt. The Rx time-out interrupt will be generated when:

1. RXFEN (0x04[0]) is set to 1.
2. Rx buffer is empty.
3. The most recent character is received longer than (RTOCNT*bclk period*4).

BTIF+0060h TRX_TRIGGER_LEVEL

BTIF_TRI_LVL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BTIF_LOOP		RX_TRI_LVL					TX_TRI_LVL
Type									R/W		R/W					R/W
Reset									0x0		0x5					0x0a

TX_TRI_LVL Used for Tx FIFO trigger threshold. THRE(0x14[5]) will be set if the data in the TXFIFO are less than TX_TRI_LVL.

RX_TRI_LVL Used for Rx FIFO trigger threshold. A Rx trigger interrupt (IIR(0x08)) = 4) might be set if the data in the RXFIFO are more than RX_TRI_LVL. The output flow control signal will also be set if the data in the RXFIFO are more than RX_TRI_LVL.

BTIF_LOOP Enables BTIF loop back mode. The data output from Tx will be received by Rx.

BTIF+0064h SLEEP_WAKEUP

BTIF_WAK

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_WAKE

Type																		WO
Reset																		1

SLEEP_WAKE ARM9 side btif_sleep_wakeup_in_b is connected to eint[16] (ARM9 has eint[19:0]).
ARM7 side btif_sleep_wakeup_in_b is connected to eint[0] (ARM7 has eint[3:0])

BTIF+0068h ASYNC_WAIT_TIME

BTIF_WAT_TIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											WAT_TIME_2			WAT_TIME_1		
Type											R/W			R/W		
Reset											0x2			0x2		

ASYNC_WAIT_TIME Sets up waiting time of RX read-out.

WAT_TIME_1 The first level of wait time.

WAT_TIME_2 The second level of wait time.

Notes: The value of WAT_TIME_1/ WAT_TIME_1 cannot be smaller than 0x2.

BTIF+006C h NEW_HANDSHAKE

BTIF_HANDSHAKE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RTO_EXT	HIGH_SPEED_EN	HANDSHAKE_EN
Type														R/W	R/W	R/W
Reset														0	0	1

NEW_HANDSHAKE The default value of handshake is 0. The function of handshake is disabled.
The function of BTIF has limitation. The ratio of bclk cannot be bigger than 2; otherwise, two system data transmissions will be wrong. If the value of handshake is 1, the ratio of bclk will be free.

HANDSHAKE_EN Enables handshake mode.

high_speed_en Enables high speed mode. Reserved.

RTO_EXT Extends the value of RX time-out counter (16**rto_time*).

4 GPS

4.1 RF Part

4.1.1 LNA/Mixer

Upon receiving RF input signal in through either multi-GNSS antenna to internal LNA or external antenna and LNA, the mixer down converts the amplified signal (GPS/Galileo=1575.42MHz, Beidou=1561.098-MHz, GLONASS=1601.71-MHz). The current chip provides 2 configurations to choose from, which are high-gain LNA and low-gain LNA. The high-gain LNA is used for low-cost solution without external LNA. The low-gain LNA offers high linearity to allow high external LNA gain, with much worsen noise figure performance. In the application with external LNA, the external LNA gain ranging from 15 to 20 dB is recommended. The down-conversion mixer is single-ended passive mixer with current mode interface between the mixer and multi-modes low pass filter.

4.1.2 VCO/Synthesizer

The entire frequency synthesizer includes crystal oscillator, VCO, divider, phase frequency detector (PFD), charge pump (CP) and loop filter which are all integrated on the MT2503D chip. Upon power-on, VCO is auto-calibrated to its required sub-band. The synthesizer adopts fractional-N sigma-delta PLL topology, which supports 12.6 to 40MHz reference clock frequencies.

4.1.3 LPF

The current-mode LPF supports multiple modes for different GNSS combinations. The LPF also provides 26dB gain-control range, with approximately 2dB per step.

4.1.4 ADC

The differential IF signal is being quantized by a high performance ADC. The sampling clock can be provided from divided clock from LO.

4.2 Digital Part

4.2.1 ARM7EJ-S

The ARM7EJ-S processor provides flexibility necessary for building Java-enabled, real-time embedded devices requiring small size, low-power and high performance. It builds on the features and benefits of the established ARM7TDMI core and is delivered in synthesizable form. ARM7EJ-S is supported by a wide variety of development tools and can run at speeding up to 158 MHz.

ARM7EJ-S includes a JTAG interface which provides a standard development and debugging interface. The interface can connect to a variety of off-the-shelf emulators. The emulators provide single-step, trap and access to all the internal registers of the digital part of MT2503D.

4.2.2 Cache

MT2503D provides cache to speed up program execution and reduce external flash access times. It supports up to 64 Kbits cache buffer and can be used as internal memory when it is not fully used.

4.2.3 Boot ROM

The embedded boot ROM provides a function of loading a set of user code through the host interface into SRAM. The host interface (UART/SPI/I2C) is decided by strap control.

4.2.4 Real Time Clock (RTC)

MT2503D provides very low leakage battery backed-up memory, which contains all the necessary multi-GNSS information for quick start-up and a small amount of user configuration variables. There is a built-in 1.1 volts LDO for RTC domain and it can be bypassed while an external LDO is applied. The RTC LDO is a voltage regulator having very low quiescent current. The small ceramic capacitor can be used as the output capacitor, and the stable operation region ranges from very light load (~=0) to about 3 mA.

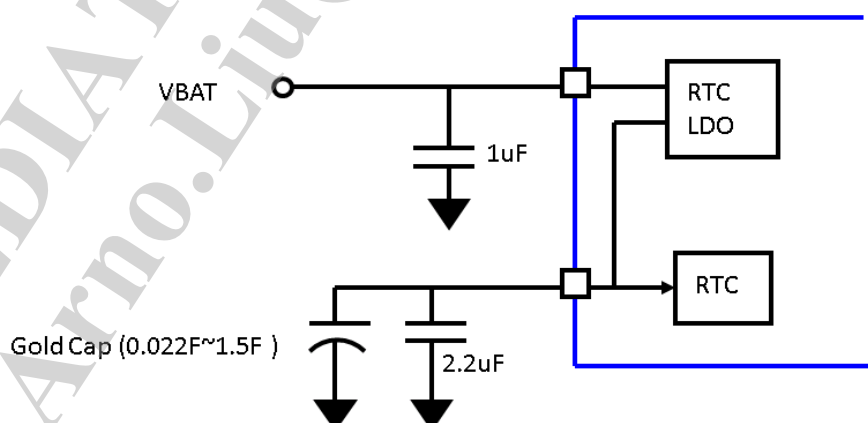


Figure 49. RTC with internal RTC LDO application circuit 1

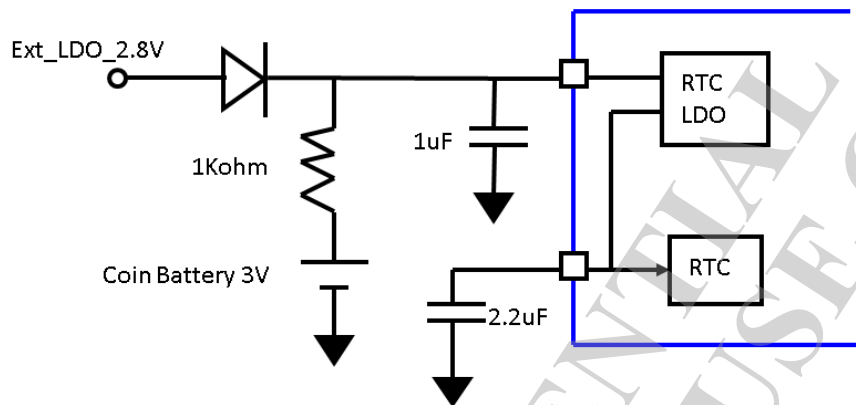


Figure 50. RTC with internal RTC LDO application circuit 2

4.2.5 SMPS

A built-in switching mode power supply provides 1.8 volts power supply for the digital 1.1 volts CLDO and RF input power. In the active mode, SMPS is operated in the PWM/PFM automatic mode. In the power saving mode, SMPS is operated with reduced switching frequency in the PFM mode. The recommended L/C value is 1 uH / 4.7 uF.

4.2.6 Timer function

The timer function supports a time tick generation of 31.25 ms resolution. With the 24-bit counter, the period of timer is from 31.25 ms to 524,287 s.

4.2.7 GPIO in RTC domain

The “32K_OUT” pin in RTC domain can output 32.768 KHz clock which can be used to support low clock rate operation mode for some applications or peripherals that need an external clock source. This pin can also be programmed to be an input pin to receive the signal from an external accelerator sensor IC to be the wake-up signal of MT2503D when it is in the low-power mode.

4.2.8 Low power detection

A low power detection circuit is implemented. Whenever the independent power source (AVDD11_RTC) becomes low voltage, the low power detection circuit will detect this condition and use an indicator signal (output high in normal condition and low in low-power condition) to reflect this condition.

4.2.9 Clock module

The clock module generates all internal clocks required by processor, correlator, internal memory, bus interface and so on. The referenced input clock is generated from the RF block. For system flexibility and maximum power saving, it supports various power management modes.

4.2.10 Reset controller

The built-in reset controller generates reset signals for all digital blocks. It has power-on reset feature and hardware trapping function. The power-on reset level is 2.7 ± 0.1 volts. The software reset function for different circuit blocks are also included for flexible applications.

In Figure 51, the voltage drop time T_{drop_vbat} and T_{drop_cldo} depend on the capacitance connection of their power net. But $T_{drop_vbat} > T_{drop_cldo}$ should be guaranteed for the correct operation of reset behavior during power off sequence. It is strongly recommend using external LDOs without output discharged function or make sure $T_{drop_vbat} > 100$ ms.

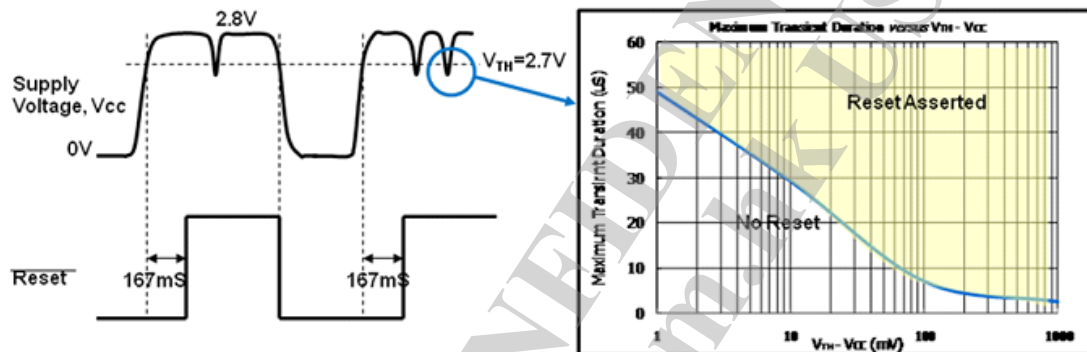


Figure 51. Power on reset diagram

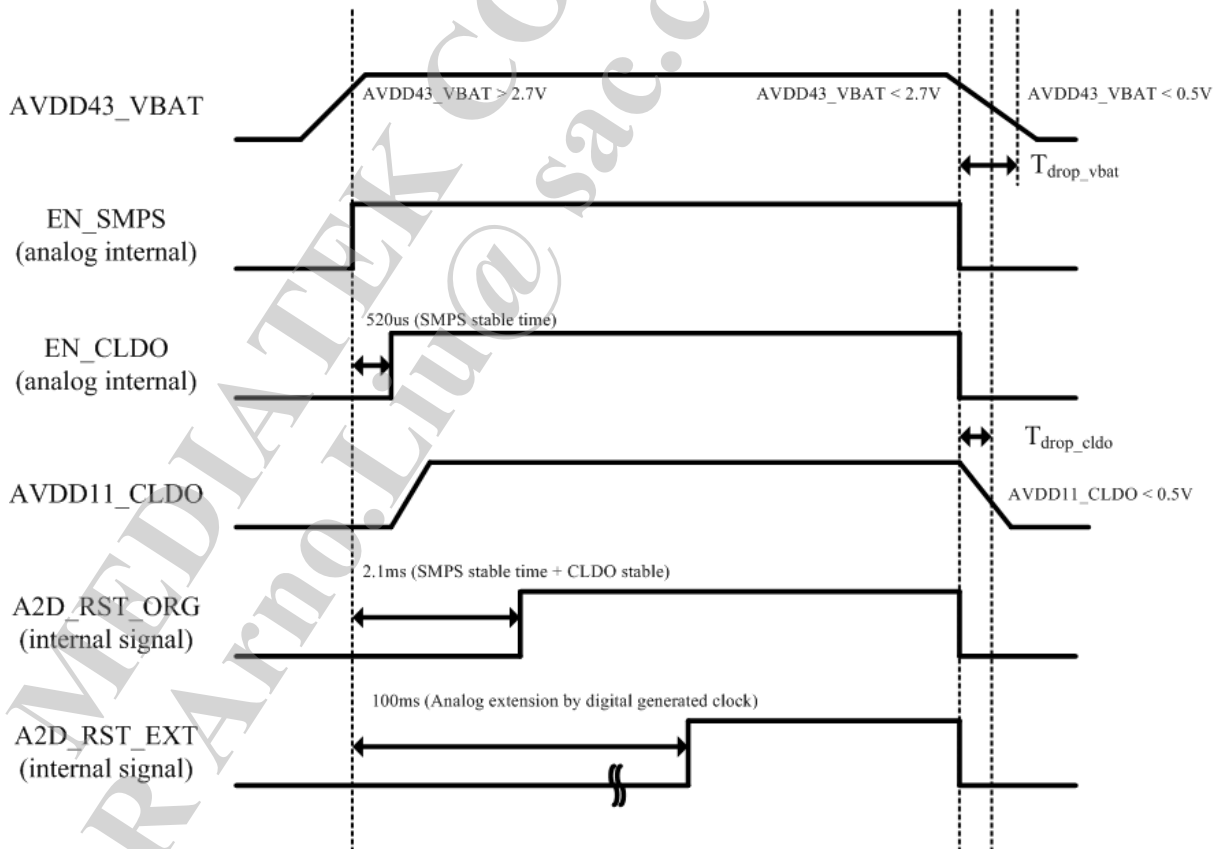


Figure 52. Power on/off reset behavior

4.2.11 Host interface

MT2503D supports 3 different host interfaces, which are UART, SPI, and I2C. The interface used as the host interface is determined by strap pins. Note that SPI and I2C support firmware update only for now.

4.2.11.1 UART

UART is the abbreviation of "Universal Asynchronous Receiver/Transmitter". MT2503D has 3 full duplex serial ports. It is used for serial data communication. A UART converts bytes of data to and from asynchronous start-stop bit streams represented as binary electrical impulses.

There are several functions in MT2503D related to UART communication, such as UART data transmission/receive and NMEA sentences input/output. In general, UART0 is as NMEA output and PMTK command input, UART1 as RTCM input. You can adjust the UART2 port as desired. The receiver (RX) and transmitter (TX) side of every port contains a 16-byte FIFO, but only UART0 has 256 bytes of URAM. The bit rates are selectable and range from 4.8 to 921.6 kbps. UART provides signal or message outputs.

4.2.11.2 SPI (by request)

The serial peripheral interface port manages the communication between digital BB and external devices. MT2503D supports both master and slave modes. Only 4 bytes of register in the master mode can be transferred. The slave has 4-byte-register mode or URAM mode. In the URAM mode, the transmitted and received data size is 256 bytes. The clock phase and clock polarity are selectable. MT2503D supports manual or automatic indicator for data transfer in the slave mode.

4.2.11.3 I2C (by request)

The I2C interface is mainly connected to external devices. MT2503D supports multi-master and slave modes. Both modes have 256-byte URAM mode and 8-byte FIFO mode for transmitting and receiving data. The multi-master mode supports 7-bit and 10-bit address modes up to 400 Kb/s fast mode and 3.4 Mb/s high-speed mode. In additions, MT2503D supports manual or automatic indicator for data transfer in the slave mode. Device addresses in the slave mode are programmable and support fast mode and high-speed mode data transmission and reception.

4.2.12 Interrupt control unit

The interrupt control unit manages all internal and external sources of interrupts, which include timer, watch-dog, all interfaces such as UART, I2C and SPI and external user interrupt pins. These interrupt sources can be wake-up events in the power saving mode.

4.2.13 Flash

An external SPI serial flash up to 128 Mb is supported. Specific MTK Flash Tool is also supported for downloading firmware into the internal flash.

4.2.14 GPIO unit

GPIO is the abbreviation of “General-Purpose Input/Output”. MT2503D supports a variety of peripherals through maximum 16 GPIO programmable ports. The unit manages all GPIO lines and supports a simple control interface. GPIO provides signal or message outputs.

4.2.15 PPS

The PPS (Pulse Per Second) signal is provided through designated output pin for many external applications. The pulse is not only limited to being active every second but also allowed to set up the required duration, frequency and active high/low by programming user-defined settings.

4.2.16 ECLK

ECLK is a clock input pin for introducing an external clock signal to MT2503D and obtaining the relation between the external clock and GPS local clock. With precise external clock input, the clock drift of the GPS local clock can be correctly estimated. Therefore, the doppler search range is narrowed down accordingly. The technology is beneficial to speeding up the satellite acquisition process. Particularly in the cold start case, due to limited priori information about the satellite’s location and local clock uncertainty, a receiver will execute a search in full frequency range. Consequently, a longer acquisition time is expected. However, the ECLK technology is able to reduce the frequency uncertainty so that the search process will be completed in a short time. Efficient acquisition and lower power consumption are attained by the ECLK technology.

4.2.17 SYNC

SYNC is a time stamp signal input pin for introducing an external timing to the GPS receiver and obtaining the relation between the external timing and the GPS receiver local timing. With precise external timing input and the established relation, the GPS time of week (TOW) can be correctly estimated in the GPS receiver. The technology is beneficial for time to first fix (TTFF), particularly in weak signal environments. In hot starts, with priori information about the GPS receiver’s location and satellite ephemeris data, the GPS receiver uses the correct GPS TOW to accurately predict the signal code chip/phase. Therefore, the code search range can be narrowed down accordingly. Hence, fast TTFF is achieved by the SYNC technology.

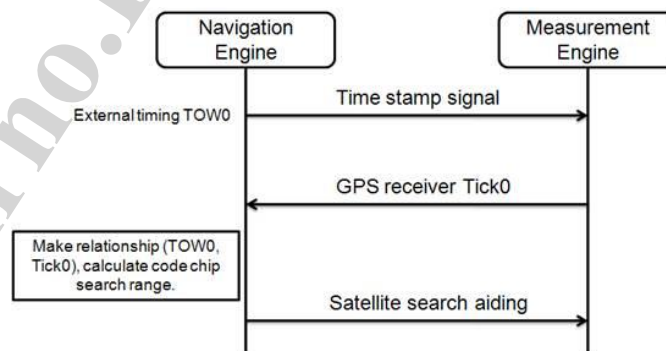


Figure 53. Flow diagram of SYNC function

4.2.18 Power scheme

Internal SMPS is used as the source power of the internal RF/BB LDO. It is also used as 1.8 volts I/O power, external TCXO/LNA voltage source via built-in TCXO switch. The internal SMPS can switch to the LDO mode to supply power to each of the about block

The minimum/maximum input voltage of AVDD43_VBAT and AVDD43_DCV is 2.8/4.3 volts.

The power-on reset voltage threshold of AVDD43_VBAT is 2.7 ± 0.1 volts. The maximum TLDO drop out voltage at half load (25 mA) is 0.2 volts. If one external LDO is used to provide power to MT2503D, the 3.3 volts external LDO will be recommended after taking TLDO drop-out into consideration.

The power efficiency in SMPS mode will be better than that in the internal LDO mode.

I/O supports 1.8 and 2.8 volts. The power comes from SMPS output for 1.8 volts application or TLDO output (AVDD28_TLDO) for 2.8 volts application.

The power for internal flash comes from AVDD28_TLDO.

TCXO power is from AVDD_TCXO_SW that can select either from AVDD28_TLDO (2.8V) or from AVDD28_CLDO (1.8V) by setting up power-on strap.

RTC LDO input power comes from backup battery or uses coin battery.

Here are 3 power schemes: low power (Figure 54), low cost (Figure 55) and external PMU (Figure 56).

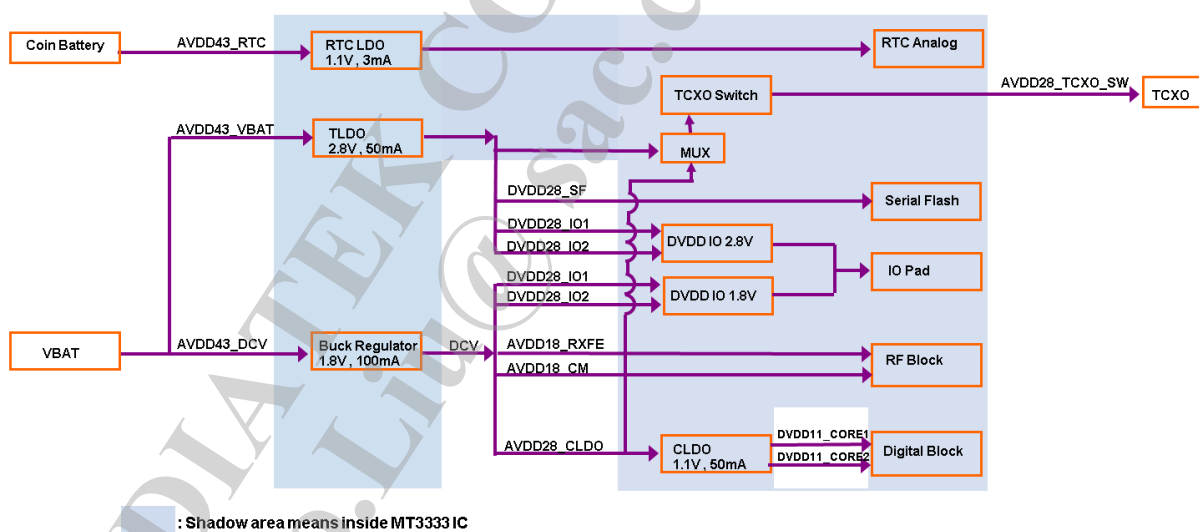


Figure 54. Power supply connection (low power)

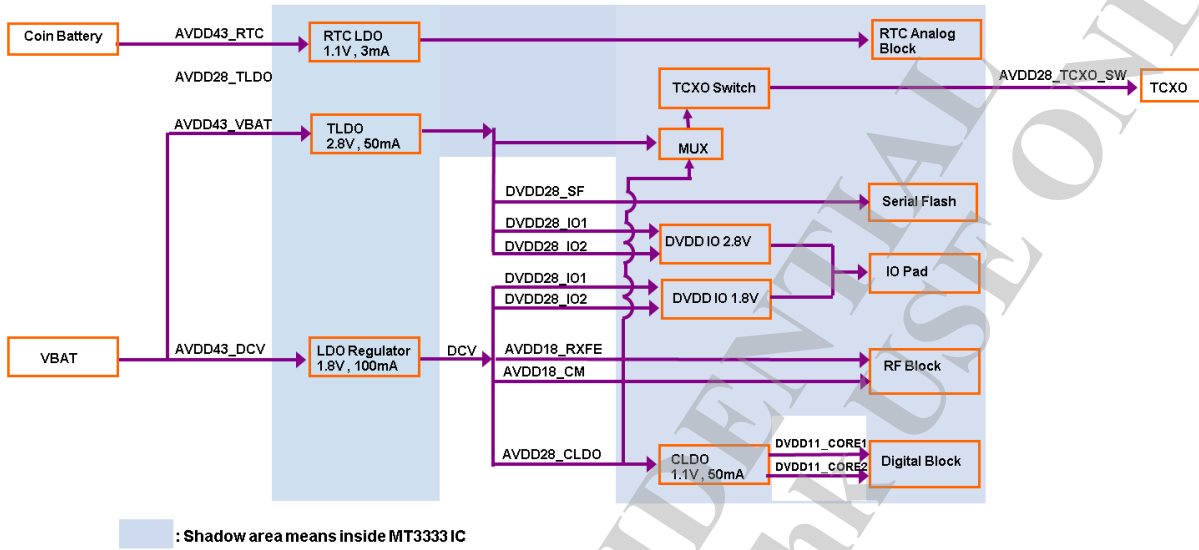


Figure 55. Power supply connection (low cost)

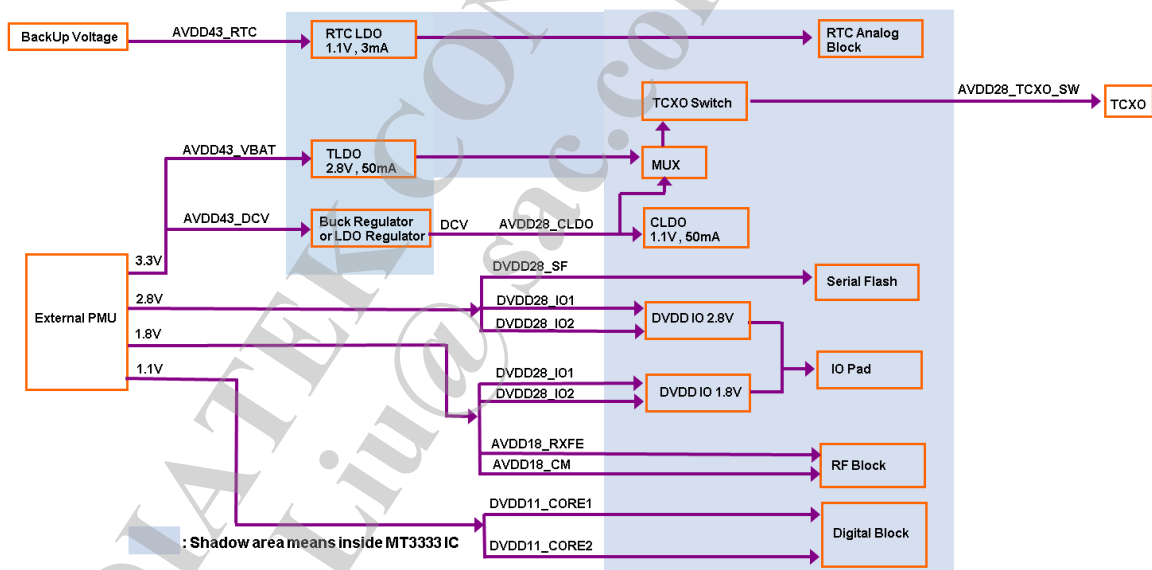


Figure 56. Power supply connection (external LDO)

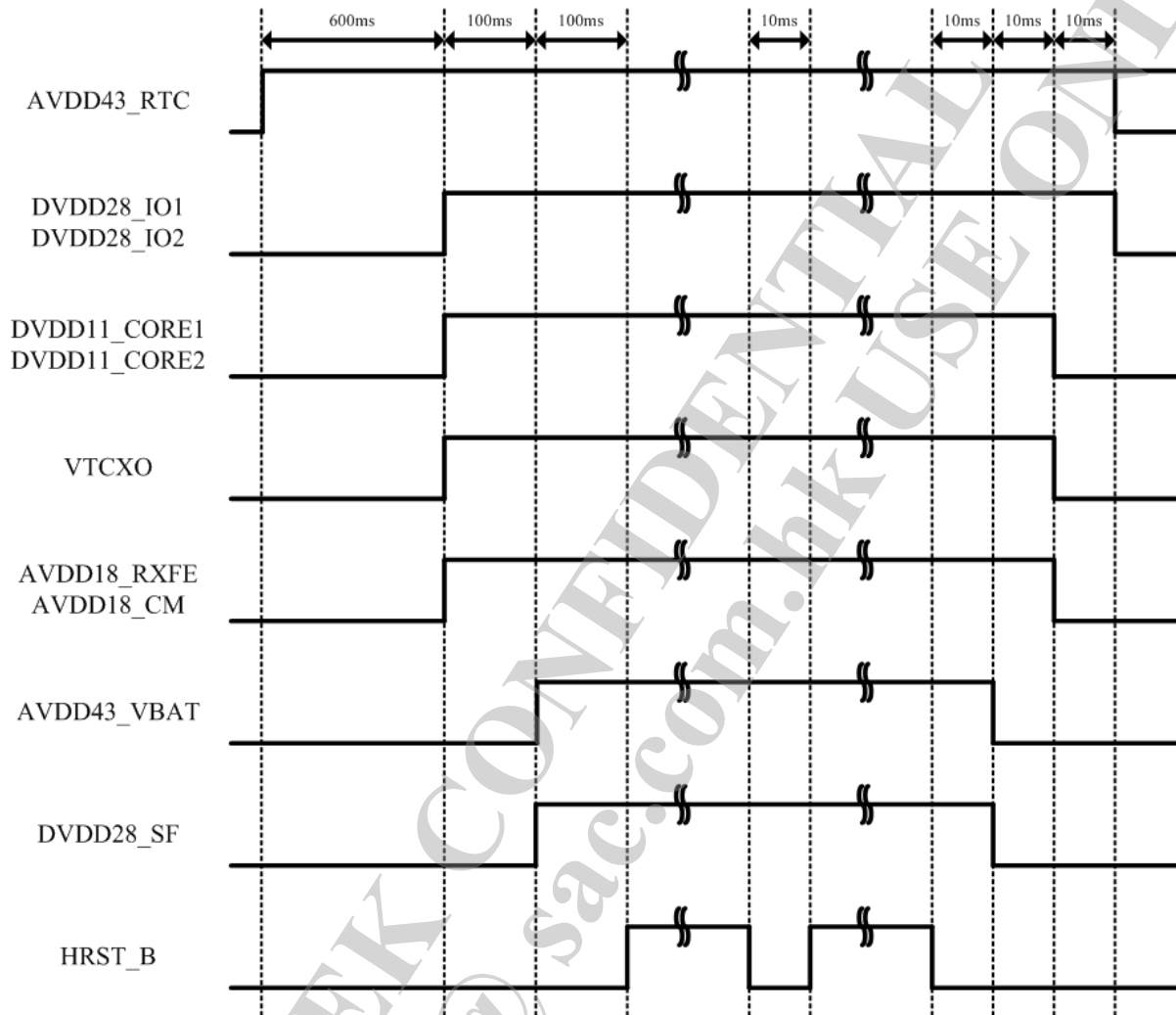


Figure 57. Power on/off sequence for external LDO mode

4.3 Electrical Characteristics

4.3.1 DC characteristics

4.3.1.1 Absolute maximum ratings

Symbol	Parameter	Rating	Unit
AVDD43_DCV	SMPS power supply	-0.3 ~ 4.3	V
AVDD43_VBAT	2.8 volts TLDO power supply	-0.3 ~ 4.3	V
AVDD28_CLDO	1.1 volts CLDO power supply	-0.3 ~ 3.6	V
DVDD28_SF	Embedded flash power supply	-0.3 ~ 3.6	V
DVDD28_I01	IO 2.8/1.8 volts power supply	-0.3 ~ 3.6	V

DVDD28_I02			
DVDD11_CORE1 DVDD11_CORE2	Baseband 1.1 volts power supply	-0.3 ~ 1.21	V
AVDD43_RTC	RTC 1.1 volts LDO power supply	-0.3 ~ 4.3	V
AVDD18_RXFE	1.8 volts supply for RF core circuits	-0.3 ~ 3.6	V
AVDD18_CM		-0.3 ~ 3.6	V
T _{STG}	Storage temperature	-50 ~ +125	°C
T _A	Operating temperature	-45 ~ +85	°C

4.3.1.2 Recommended operating conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
AVDD43_DCV	SMPS power supply	2.8	3.3	4.3	V
AVDD43_VBAT	2.8 volts TLDO power supply	2.8	3.3	4.3	V
DVDD11_CORE1 DVDD11_CORE2	1.1 volts baseband core power	0.99	1.1	1.21	V
DVDD28_I01	2.8 volts digital I/O power	2.52	2.8	3.08	V
DVDD28_I02	1.8 volts digital I/O power	1.62	1.8	1.98	V
DVDD28_SF	Embedded flash power supply	2.7	2.8	3.6	V
AVDD18_RXFE	1.35 volts supply for RF core circuits in bypass mode	1.3	1.35	1.98	V
	1.8 volts supply for RF core circuits in LDO mode	1.62	1.8	3.08	V
AVDD18_CM	1.35 volts supply for common RF block in bypass mode	1.3	1.35	1.98	V
	1.8V volts supply for common RF block in LDO mode	1.62	1.8	3.08	V
T _A T _j	Operating temperature	-40	25	85	°C
	Commercial junction operating temperature	0	25	115	°C
	Industry junction operating temperature	-40	25	125	°C

4.3.1.3 General DC characteristics

Symbol	Parameter	Condition	Min.	Max.	Unit
I _{IL}	Input low current	No pull-up or down	-1	1	uA
I _{IH}	Input high current	No pull-up or down	-1	1	uA
I _{OZ}	Tri-state leakage current		-10	10	uA

4.3.1.4 DC electrical characteristics for 2.8 volts operation

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VDD	Supply voltage of core power		0.99	1.1	1.21	V
VDDIO	Supply voltage of IO power		2.52	2.8	3.08	V
V _{IL}	Input lower voltage	LVTTL	-0.3	-	0.25*VDDIO	V
V _{IH}	Input high voltage		0.75*VDDIO	-	VDDIO+0.3	V
V _{OL}	Output low voltage	VDDIO = min I _{OL} = -2 mA	-	-	0.15*VDDIO	V
V _{OH}	Output high voltage	VDDIO = min I _{OH} = -2 mA	0.85*VDDIO	-	-	V
R _{PU}	Input pull-up resistance	VDDIO = typ V _{input} = 0 V	40	85	190	KΩ
R _{PD}	Input pull-down resistance	VDDIO = typ V _{input} = 2.8 V	40	85	190	KΩ

4.3.1.5 DC electrical characteristics for 1.8 volts operation

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VDD	Supply voltage of core power		0.99	1.1	1.21	V
VDDIO	Supply voltage of IO power		1.62	1.8	1.98	V
V _{IL}	Input lower voltage	LVTTL	-0.3	-	0.25*VDDIO	V
V _{IH}	Input high voltage		0.75*VDDIO	-	VDDIO+0.3	V
V _{OL}	Output low voltage	VDDIO = min I _{OL} = -2 mA	-	-	0.15*VDDIO	V
V _{OH}	Output high voltage	VDDIO = min I _{OH} = -2 mA	0.85*VDDIO	-	-	V
R _{PU}	Input pull-up resistance	VDDIO = typ V _{input} = 0 V	70	150	320	KΩ
R _{PD}	Input pull-down resistance	VDDIO = typ V _{input} = 1.8 V	70	150	320	KΩ

4.3.1.6 DC electrical characteristics for 1.1 volts operation (for FORCE_ON and 32K_OUT)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
--------	-----------	-----------	------	------	------	------

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VDD	Supply voltage of core power		0.99	1.1	1.21	V
VDDIO	Supply voltage of IO power		0.99	1.1	1.21	V
V _{IL}	Input lower voltage	LVTTL	-0.3	-	0.25*VDDIO	V
V _{IH}	Input high voltage		0.75*VDDIO	-	VDDIO+0.3	V
V _{OL}	Output low voltage	VDDIO = min I _{OL} = -2 mA	-	-	0.15*VDDIO	V
V _{OH}	Output high voltage	VDDIO = min I _{OH} = -2 mA	0.85*VDDIO	-	-	V
R _{PU}	Input pull-up resistance	VDDIO = typ V _{input} = 0 V	130		560	KΩ
R _{PD}	Input pull-down resistance	VDDIO = typ V _{input} = 1.1 V	130		560	KΩ

4.3.2 Analog related characteristics

4.3.2.1 SMPS DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD43_DCV	SMPS input supply voltage	2.8	3.3	4.3	V	
DCV	SMPS output	1.74	1.84	1.94	V	
I _{cc}	SMPS output current	-	-	100	mA	
ΔV_PWM	Ripple of PWM mode	-	-	40	mV	With L=1uH, C=4.7uF
ΔV_PFM	Ripple of PFM mode	-	-	90	mV	With L=1uH, C=4.7uF

4.3.2.2 TCXO LDO DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD43_VBAT	TCXO LDO input supply voltage	2.8	3.3	4.3	V	Will change to bypass mode under 3.1 volts
AVDD28_TLDO	TCXO LDO output	2.71	2.8	2.89	V	
I _{cc}	LDO output current	-	-	50	mA	Not include external devices
	PSRR-30 KHz	35	-	-	dB	Co = 1 uF, ESR = 0.05, Iload = 25 mA
	Load regulation	-84	10	84	mV	

4.3.2.3 TCXO SWITCH DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
--------	-----------	------	------	------	------	------

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD_TCXO_SW	TCXO switch output voltage @ TCXO switch input = AVDD28_TLDO	2.66	-	-	V	
AVDD_TCXO_SW	TCXO switch output voltage @ TCXO switch input = AVDD28_CLDO	1.71	-	-	V	
I _{max}	TCXO SWITCH current limit	-	-	30	mA	

4.3.2.4 1.1 volts core LDO DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD28_CLDO	1.2 volts LDO input supply voltage	1.62	1.8	3.08	V	
AVDD11_CLDO	1.1 volts LDO output	1.05	1.12	1.2	V	
I _{cc}	LDO output current	-	-	50	mA	
	Load regulation	-	-	-	mV	

4.3.2.5 1.1 volts RTC LDO DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD43_RTC	RTC LDO input supply voltage	2	4	4.3	V	
AVDD11_RTC	RTC LDO output	0.99	1.1	1.21	V	
I _{cc}	LDO output current	-	-	3	mA	
I _{leak}	Leakage current	2.2	10	-	uA	Including LDO and RTC domain circuit

4.3.2.6 32 KHz crystal oscillator (XOSC32)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD11_RTC	Analog power supply	0.99	-	1.21	V	
Dcyc	Duty cycle	-	50	-	%	

4.3.3 RF related characteristics

4.3.3.1 DC electrical characteristics for RF part

Symbol	Parameter	Min.	Typ.	Max.	Unit
I _{cc} (GPS+GLONASS)	Total supply current:	-	8.9	-	mA

4.3.3.2 RX chain (GPS+GLONASS mode)

Parameter	Condition	Min.	Typ.	Max.	Unit
RF input frequency		-	1575.4	-	MHz
LO frequency		-	1588.6	--	MHz
LO leakage	Measured at balun matching network input at LNA high gain	-	-70	-	dBm
Input return loss	Differential input and external matched to 50Ω source using balun matching network for all gain	-10	-	-	dB
Gain (Av) (integrated average over Fc+-4M)	High current mode with max PGA gain	80	76	70	dB
	Low current mode with max PGA gain	-	64		dB
PGA Gain range		-	24	-	dB
PGA Gain step		-	2	-	dB
NF (integrated average over Fc+-4M)	High current mode with max PGA gain	-	2.2	-	dB

4.3.3.3 Crystal oscillator (XO)

Symbol	Parameter	Min.	Typ.	Max.	Unit
F _{tcxo}	TCXO oscillation frequency	12.6	16.368	40	MHz
V _{tcxo}	TCXO output swing	0.8	1.2	-	V _{pp}

4.4 Interface Characteristics

4.4.1 JTAG interface timing

Description	Sym bol	Min.	Max.	Unit	Note
TDI input setup to rising TCK	T1	0.35T	-	ns	1
TDI input hold from rising TCK	T2	0.15T	-	ns	1
TMS input setup to rising TCK	T1	0.35T	-	ns	1
TMS input hold from rising TCK	T2	0.15T	-	ns	1
Rising TCK to TDO valid	T3	-	0.5T	ns	1
TDO hold from rising TCK	T4	0	-	ns	1

Note: The maximal condition of JTAG clock cycle (TCK) is 50 MHz.

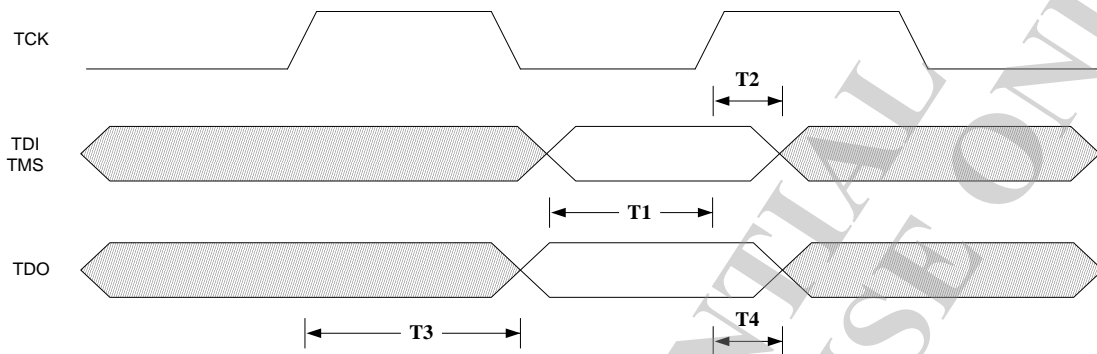


Figure 58. Timing diagram of JTAG interface

4.4.2 RS-232 interface timing

Baudrate required (bps)	Programmed baudrate (bps)	Baudrate error (%)	Baudrate error (%) ³
4,800	4,800.000	0.0000	0.002
9,600	9,600.000	0.0000	0.002
14,400	14,408.451	0.0587	0.0567
19,200	19,164.319	0.0587	0.0567
38,400	38,422.535	0.0587	0.0567
57,600	57,633.803	0.0587	0.0567
115,200	115,267.606	0.0587	0.0567
230,400	230,535.211	0.0587	0.0567
460,800	454,666.667	-1.3310	-1.3330
921,600	909,333.333	-1.3310	-1.3330

Notes:

1. UART baud-rate settings with UART_CLK frequency = 16.368 MHz (UART_CLK uses the reference clock of the system).
2. The baudrate error is optimized. Each baudrate needs to adjust counter to obtain the optimized error.

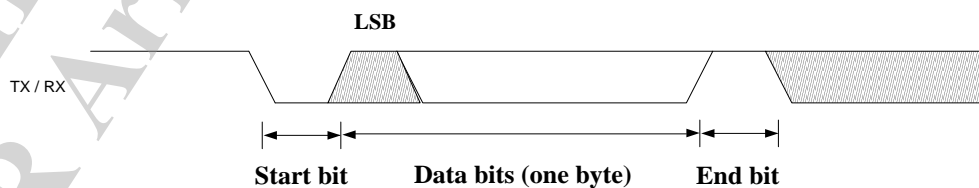


Figure 59. Timing diagram of RS-232 interface

4.4.3 SPI interface timing

Description	Symbol	Min.	Max.	Unit	Note
SCS# setup time	T1	0.5T	-	ns	1
SCS# hold time	T2	0.5T	-	ns	1
SO setup time	T3	0.5T - 3t	0.5T - 2t	ns	1, 2
SO hold time	T4	0.5T + 2t	0.5T + 3t	ns	1, 2
SIN setup time	T5	3t	-	ns	1, 2
SIN hold time	T6	10	-	ns	1

Notes:

- The condition of SPI clock cycle (T) is (SPI_IPLL/12) MHz ~ (rf_clk/1,020) MHz.
- t indicates the period of SPI controller clock, which is SPI_IPLL clock or rf_clk.

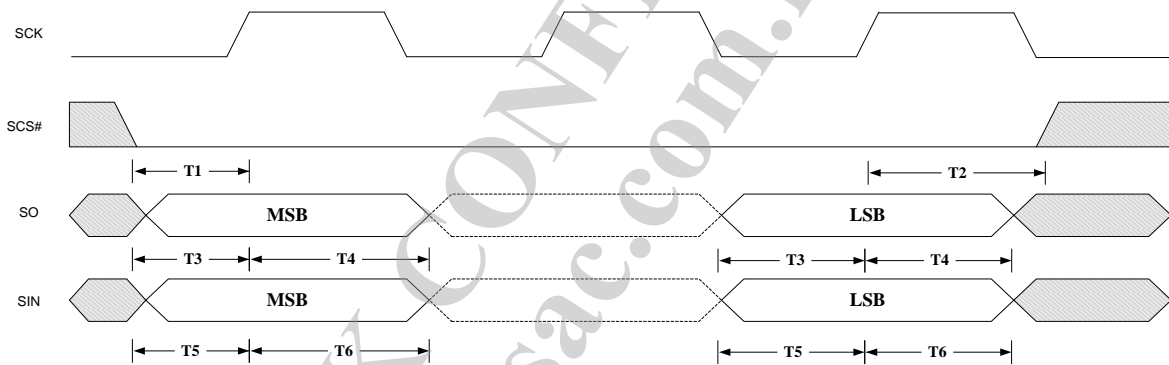


Figure 60. Timing diagram of SPI interface

4.4.4 I2C interface timing

Symbol	Period
T1	(MM_CNT_PHASE_VAL0+1)/TCXO_CLK
T2	(MM_CNT_PHASE_VAL1+1)/TCXO_CLK
T3	(MM_CNT_PHASE_VAL2+1)/TCXO_CLK
T4	(MM_CNT_PHASE_VAL3+1)/TCXO_CLK

Note: The condition of I2C clock cycle (I2C_CLK) is (TCXO_CLK/4) MHz ~ (TCXO_CLK/(MM_CNT+4)) MHz. The MM_CNT is sum of MM_CNT_PHASE_VAL0, MM_CNT_PHASE_VAL1, MM_CNT_PHASE_VAL2 and MM_CNT_PHASE_VAL3 in full speed mode.

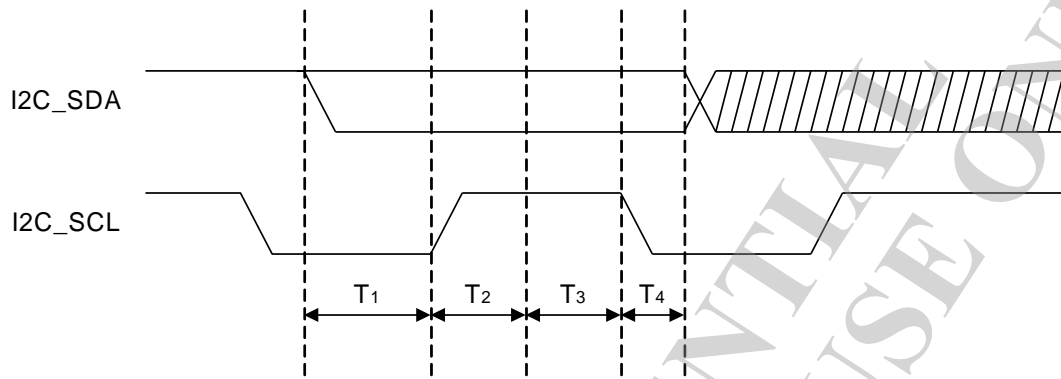


Figure 61. Timing diagram of HOST I2C interface