SIEMENS

SIMATIC

ASPC 2 / Hardware

User Description

(Advanced PROFIBUS Controller according to IEC 61158)

Version: V2.4

Date: 05/ 2009

Liability Exclusion
We have tested the contents of this document regarding agreement with the hardware and software described. Nevertheless, there may be deviations, and we don't guarantee complete agreement. The data in the document is tested periodically, however. Required corrections are included in subsequent versions. We gratefully accept suggestions for improvement

Copyright

Subject to technical changes.

Releases

Release	Date	Changes
V 2.1	Sept. 2003	Description of the differences between the different manufacturers. Specifications of the different manufacturers in chap. 7.1, 7.4 and 1.14 Chap 10.1 Contact addresses
V 2.2	Dec 2005	Included order numbers
V 2.3	Apr 2009	Remove data of the old ASIC
V 2.4	May 2009	Using of the ASPC2R

Table of Contents

1 Introduction

Note:
Because of the complexity of the ASIC ASPC2, it is urgently recommened to use the software package offered; this will avoid errors in ASIC handling. Support for software generation and software testing beyond information over the phone is not possible.

1.1 General

For simple and fast digital data exchange between PLCs, Siemens offers to its users ASICs which, on the basis of PROFIBUS EN 50170, support or completely process data traffic between the individual automation stations.

In the **SPC2**, parts of Layer2 that execute the bus protocol are already integrated. For the remaining functions of Layer2 (interface service, management), an additional microprocessor is needed.

The **SPC3**, through the integration of the complete PROFIBUS DP protocol, relieves the processor considerably, and can be operated on the bus with 12Mbaud.

The **SPC4** makes the protocol types DP, FMS, and PA possible, and can be operated on the bus also with 12Mbaud.

The chips support the passive stations on the bus system, and filter out all outside messages, and faulty user messages.

However, in the field of automation, there are also simple devices such as switches, thermoelements, etc.. No microprocessor is needed to record their states.

For a low cost adaptation of these devices, an additional ASIC is available, called **LSPM2** (Lean Siemens Profibus Multiplexer). The ASICs process as slaves in the bus system. A master addresses the ASICs via Layer2 of the 7-layer model. After a faulty message was received, both autonomously generate the requested reply messages (according to EN 50 170, Part 3).

With the communication chip **ASPC2** (Advanced Serial PROFIBUS Controller), Layer1 and Layer2 is processed completely by PROFIBUS EN 50170. At the same time, the ASPC2 serves as a master for PROFIBUS DP and, via a segment coupler, also for PROFIBUS PA (process automation). All ASICs are available on the market, and can be ordered from Siemens branches.

1.2 Marketing the ASPC2 Software

For the ASIC ASPC2 as DP master, extensive software is available (approx. 64Kbyte). Obtaining the software is bound to a license contract.

Software Structure

The figure below provides a short overview of the software of the master package. A detailed description is provided in the documentation **ASPC2 / Software**.

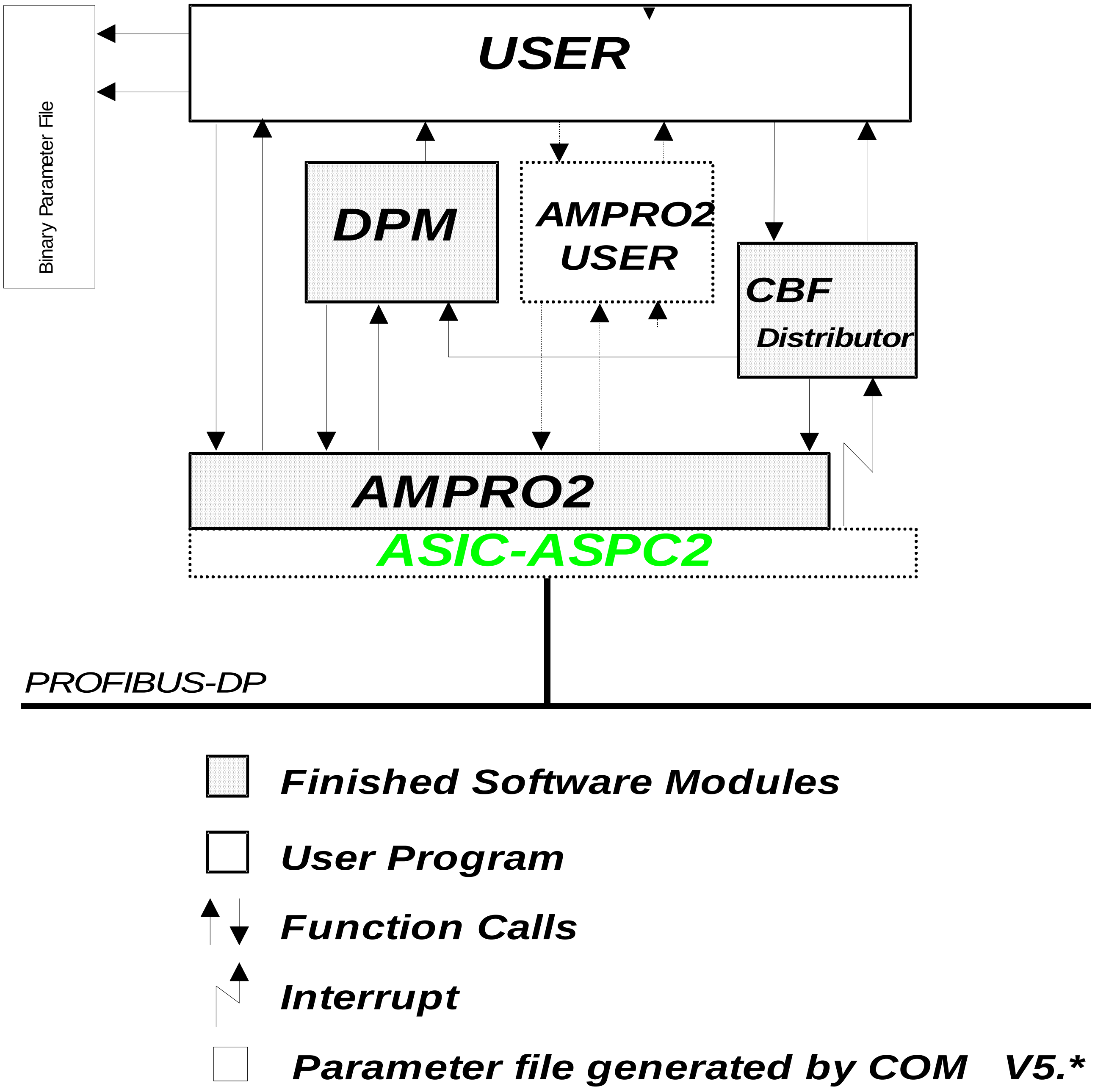


Figure 1.3-1: Software Structure

☞ **The software uses the structure of the binary parameter file of COM PROFIBUS**

1.3 Overview of the ASICs

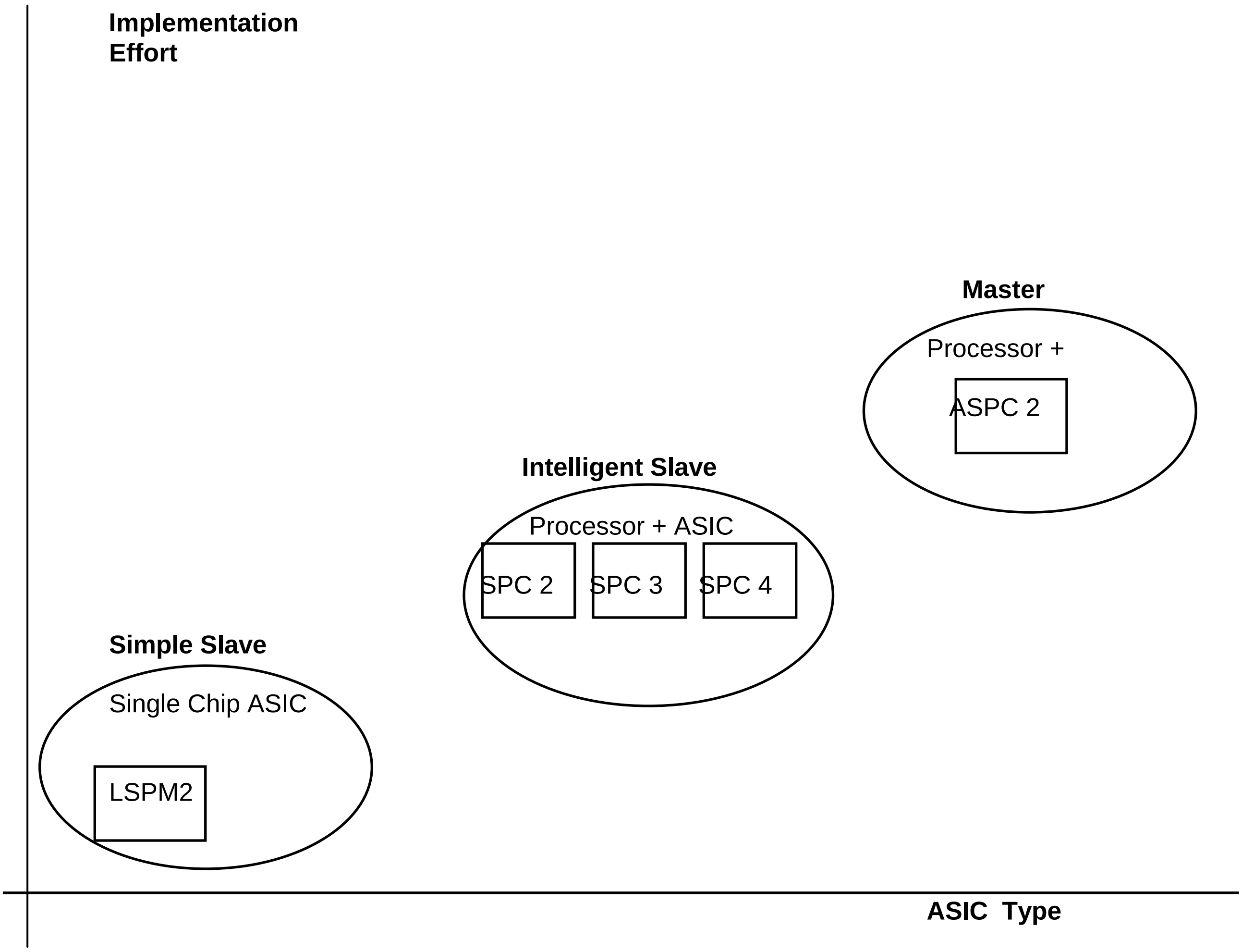


Figure 1.4-1: ASIC Overview

1.4 Application Field ASPC2

The ASPC2 is the consistent continued development of the ASIC line for PROFIBUS.

With the communication chip ASPC2, Layer1 and Layer2 is processed entirely by PROFIBUS EN 50 170. At the same time, the ASPC2 serves as master for PROFIBUS DP, and, via a segment coupler, also for PROFIBUS PA.

This highly integrated controller chip is used in production engineering as well as in process engineering.

By using the ASPC2, the following are considerably relieved of tasks concerning communication processing: PLCs, PCs, drive controls, process control systems, up to operator interface systems.

To interface secondary devices -such as controllers, final controlling elements, transducers, and distributed peripherals- the PROFIBUS ASICs are used for slave appplications (refer to Figure 1.4-1).

Special Features of the ASPC2 ASIC:

- PROFIBUS DP, PROFIBUS FMS, and PROFIBUS PA are supported with a single chip
- High net data throughput
- Support of DP communication for the fastest response time
- Complete token management and job processing
- Optimum interfacing to available processor types
- No general conditions imposed on the microprocessor with respect to time

Interface to Host

- Processor Interface
  8/16 bit can be set
  Intel/Motorola byte ordering can be set
- User Interface
  The ASPC2 can address 1 Mbyte externally as communication RAM
- Memory and microprocessor can be interfaced with the ASIC in the shared memory mode or in the dual port memory mode
- In the shared memory mode, equivalent operation of several ASPC2 on one microprocessor is possible

Supported Services

- Request-FDL-Status,
- Send Data with no Acknowledge (SDN) Broad/Multicast
- Send Data with Acknowledge (SDA)
- Send and Request Data with Reply (SRD)

Baudrates

- 9.6; 19.2; 45.45; 93.75; 187.5; 500, 750 kbit/s;
- 1.5; 3; 4; 6 and 12 Mbit/s

Response Time

- Short acknowledgement; for example, SDA starting at 1 msec (11 bit timing)
- Typical, for example SRD starting at 3 msec

Number of Stations

- 126 active/passive, mixed as required
- 64 service access points and one default SAP

Transmission Procedure according to

- PROFIBUS Standard EN 50 170 Volume 2

Ambient temperatures

- Operating temperature -40 ^{o}C ... + 85 ^{o}C
- Storage temperature -65 ^{o}C ... +150 ^{o}C
- Chip temperature in operation -40 ^{o}C ... +125 ^{o}C

Constructive Design

- P-MQFP 100 Case 14 x 20 mm^{2}
  or 17.2 x 23.2 mm^{2}

1.5 Notes

1.5.1 ASPC2 Revision C

This ASIC differs only little fom its predecessor so that, because of its advantages, it can directly replace Revision B, provided the following points are noted:

1.5.1.1 XWRL-XWRH Mode

This mode can no longer be parameterized via Mode Register 1, but is set by connecting Pin 33. The previous function as test output is applied to Pin 35.

☞ Pin 33 is provided with an integrated pull-up resistor, so that if unconnected, as in the case of Revision B, the XBHE/XWR mode is switched on after reset.
☞ When using the test mode, the pin change is to be taken into account.
☞ The bit in Mode Register 1 now has a different meaning.

1.5.1.2 Reset Input

This input is designed as CMOS Schmitt Trigger. At reset, the RTS output becomes immediately inactive.

1.5.1.3 CLK48 Input

This input is designed as CMOS Schmitt Trigger. The pulse duty factor is distorted, and can now amount to 80/20 or 20/80.

1.5.1.4 Bus Access

The ASIC has a faster data transfer cycle time and, with a setting in Mode Register 2, can be operated in the quick access mode. With this, a further enhancement is attained.

1.5.1.5 Guranteed Operating Range

Temperature Range		-40 C to +85 C
DC Supply Voltage	:	+5V ± 10 %

1.5.1.6 Version

The version can be read from the address 0BH.
☞ Revision B (value = 0), or Revision C (value = 1)

1.5.1.7 Lock Handling

In the dual port memory mode, the user can set the lock handling of the ASPC2.

1.5.1.8 EOI Inactive Time

The time between EOI and the next possible interrupt can be parameterized for1 µs, or 1 ms (another time base applies to the monitor mode).

1.5.1.9 User Timer Interrupt

With the delay timer, the user can implement a software timer that can be set to 2.1 sec, or 10 msec via Mode Register 2.

1.5.1.10 Blocked Mode

With this setting, input data that is less than the FIFO size can be held back in the FIFO until the entire message is received and was checked for correctness. This function depends no longer on the DP mode setting, but on a setting in Mode Register 2.

1.5.1.11 FIFO Size

The FIFO size can be set for 64 (as in Revision B), or 128 bytes in Mode Register 2.

1.5.1.12 Time between Two Tokens

In the pass token mode, there now is a wait of Tid1 between two successive token messages to itself.

1.5.1.13 NOP Request

Requests that are provided with this option are no longer sent (for example, FORCE PASS TOKEN request).

1.5.1.14 Data Length

If the L2 and L4 data length results in a value larger than 250D, the request is confirmed directly with 8EH, without sending a message.
If, at the slave, the received data length is less than the specified L4 length, the reply is now ‘RS’.

1.5.2 ASPC2 Revision D

The current shipping version of the ASIC ASPC2 is Revision D. For that reason, all values, tables, and documentation is laid out for Revision D.
The ASPC2 Revision C can be directly replaced with Revision D. However, a few points have to be noted.

1.5.2.1 Version

The version can be read from Address 0BH.
☞ Revision C (value = 1) or Revision D (value = 2)

1.5.2.2 GAP Error in the DP Mode

Under certain circumstances the wrong GAP area is processed in the DP mode. This error has been removed in Revision D.

1.5.2.3 Deadlock Error if HW Wiring is Faulty

If, through faulty HW wiring, the access of the ASPC2 to the request data block was prevented through ready delay (approx. 15 µs at 12 Mbaud), a deadlock could occur in Revision C. This deadlock has been removed in Revision D.

1.5.3 General Information

1.5.3.1 Daisy Chain

If several ASICs are chained with a daisy chain, all chips have to be supplied synchronously with the same clock pulse.

1.5.3.2 8 Bit Access

In the case of read accesses in the 8 bit mode, the output signal DTXR can remain on log. 1 if slow bus mode and dual port memory mode is set.

1.6 Overview of the FLC Interface

The ASPC2 is the **MAC** (Medium Access Controller) that carries out the bus protocol of Layer 2. The remaining functions of Layer2 (interface service, management) are handled by the **FLC software** (Field Link Controller). The FLC connects the user and the ASPC2 (MAC).

The **System Control Block, SCB** (refer to Figure 1.7) is used as the interface between the ASPC2 and the FLC. The SCB has to be established completely by the FLC. The FLC chains send requests in the **request lists**. Then, the ASPC2 processes these requests, removes them from the request lists, and enters them, depending on parameter assignment, either in the **CON IND lists (high or low)**, or **CON SEP or NOT OK lists (high or low)**. With interrupt, the FLC is informed of each request that was processed (confirmation).
In the receive mode, the FLC makes up to **64 service access points (SAP[0] … [63])** available to which the FLC loads receive requests (response data), and the ASPC2 stores the request data. An additional service access point is available as **default SAP**, if no DSAP address is specified in the call message. After a receive request has been processed successfully, it is removed from the corresponding **SAP list**, and chained in the **CON IND lists (high or low)** or **IND SEP lists (high or low)**. Here also, the FLC is informed with an interrupt (indication).
Send and receive requests are made available in **applicaton blocks**. These contain branching pointers, management data, the **request header including Layer4 data**, and the **response header. Request** and **response data** is available in separate buffers respectively.
The ASPC2 can address a 1Mbyte memory area directly. In that case, it is in the master mode. The handshake between the ASPC2 and an external bus master (for example, processor) is based on the **HOLDX/HOLDA signals**. For supporting dual port memory configurations, the ASPC2 has a bus arbiter which allocates the bus upon the request of an external bus partner. For parameter assignment and interrupt event handling, the ASPC2 is addressed by the processor in the peripheral mode by writing to diverse ASPC2 registers, or reading them. In addition, the processor has to make one or -in the case of CON/IND and error event handling- two interrupt inputs available.

The bus access is a hybrid method, **token passing**. Token management in the ASPC2 is completely autonomous. Fault modes and special operating modes are controlled by the chip.

A total of 127 stations, active, passive, or mixed as needed, can be connected to the bus. The ASPC2 supports the **following services**:

- Request-FDL Status
- Send Data with no Acknowledge, SDN (Broadcast/Multicast)
- Send Data with Acknowledge, SDA
- Send and Request Data with Reply, SRD

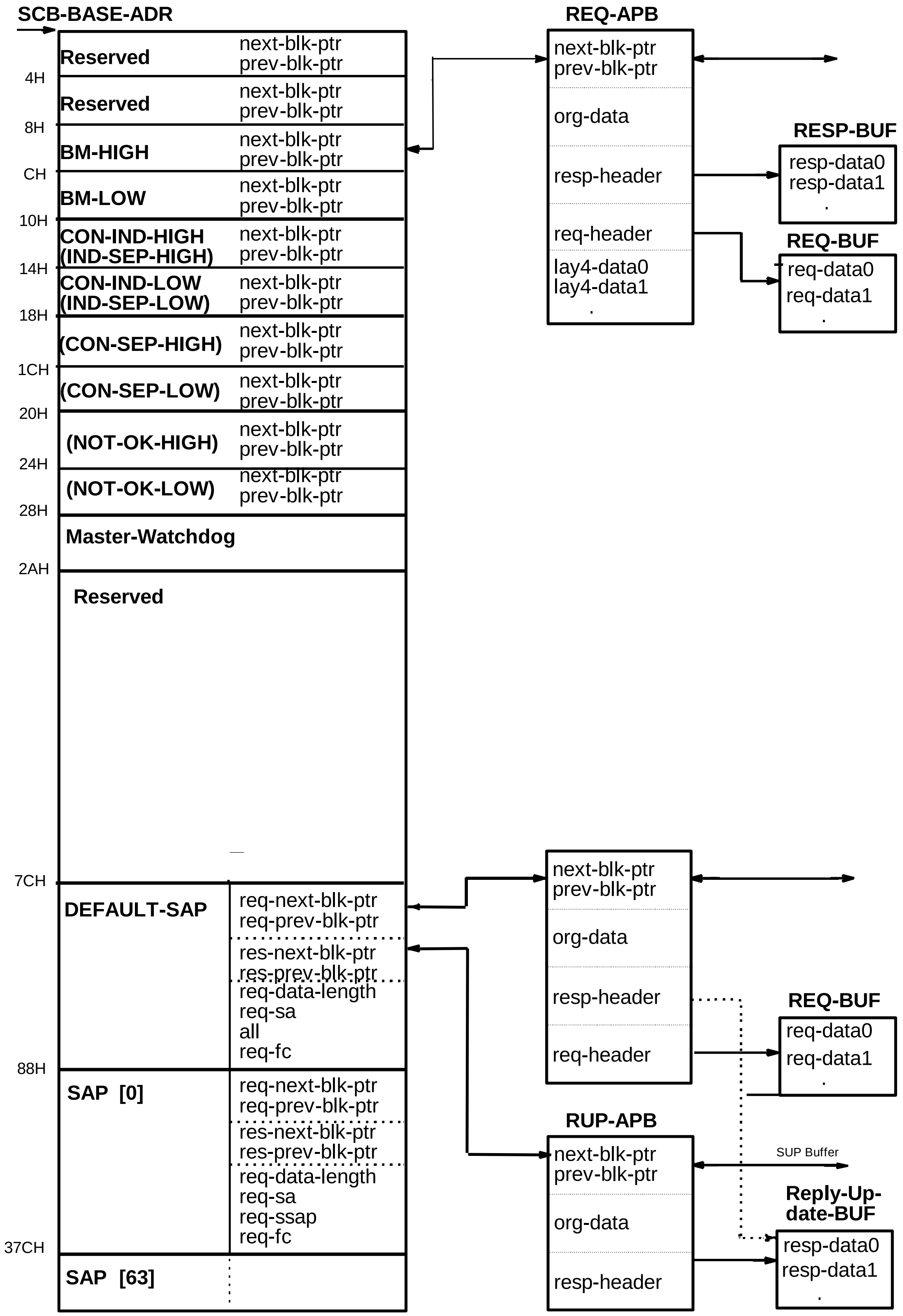


Figure 1.7-1: System Control Block (SCB) of the ASPC2 (Intel Format)

1.7 Function Overview

The ASPC2 has a multiplexed bus interface (8/16 bit bus width) which can be set to Intel/Motorola. In the master mode, it becomes active itself, and transfers data from/to the memory (byte by byte, or word by word). For the memory access, 1 to 5 wait states can be set. In addition, the **Bus Interface Unit (BIU)** can process asynchronously, with external ready. Furthermore, a fast bus mode is implemented (quick access mode). The addressing space consists of 1Mbyte. In the shared memory mode, several ASPC2s can be interconnected via a daisy chain. To support dual port memory configurations, the ASPC2 has a bus arbiter to which an external bus master can be connected (XREQ). In the peripheral mode, the ASPC2 is addressed by the processor directly via a little address window. This mode is used for parameter assignment and interrupt event handling.

In the **parameter register file**, and in the **mode registers**, protocol-specific parameters are to be transferred by the processor after switch-on (diverse idle timing, station addresses, baudrate, etc.). In the **status register**, the *MAC status* can be polled any time.

In the **interrupt controller**, various events are entered (such as error events, *MAC states,* etc.). These events can be enabled individually via a mask register. The acknowledgement is made via the acknowledge register. The ASPC2 has two innterrupt outputs: one for the confirmation/indication message only, and the other for error events.

The **Token Rotaton Timer** controls the token cycle time. The user parameterizes the token target rotation time (TTR), and is a multiple of 256 x tBit.

The **FIFO (64/256Byte)** is used as temporary buffer for the receive and send messages. It is controlled in the half duplex mode, and behaves according to the transfer direction in the SER bus. The ASIC transfers data between UART and FIFO, and the BIU between FIFO and the external memory. The SYS bus is obtained only if the content of the FIFO exceeds the quarter-full threshold, or undershoots the quarter-empty threshold. Thus, the data will be transferred only package by package.

In the **LAS RAM (128 x 4 bits)**, the list of the active stations is stored. Each active station has to carry this list in order to ensure faultless token traffic. In addition, the call sequence bits are stored in the LAS RAM for each bus station (they prevent message loss or message duplication).

In the **UART**, the parallel data stream is converted into a serial data stream, and vice versa. The transmission procedure is asynchronous with startbit, 8 data bits, parity bit, and stop bit. The baudrate has to be parameterized at initialization. Baudrates in the range of from 9.6 kBd to 12MBd are permissible (refer to Table 4.2-1).

The **idle timer** checks the idle phase (binary ‘1’) directly on the bus line, the Syn time that is necessary prior to a call for clear synchronization of the receivers. At the passive and the active station without a token, this timer is loaded with TSYN. At the active station with token, TID1 or TID2 is loaded, depending on the transmission service.

The **syni timer** monitors the transmission medium. If there is no receiver synchronization within the time TSYNI, there is a continuous malfunction. In this case, the MAC returns to the *not hold token mode*, and generates an error interrupt.

The **slot timer** monitors -at the active station with token after a call or token passing- whether the receiving station responds within the specified time TSLOP, or shows activity. If the timer expires, a repetition or a new message cycle is initiated. In the case of passive or active stations without token, the time TSLOT is used as count interval for the timeout timer.

The **timeout timer** monitors the bus activity of the active and passive stations. The TTIMEOUT is a multiple of the TSLOT in dependence on the station address. If the SER bus is idle during this time, there is a token loss which triggers (re)initialization at the active station.

1.8 Address Assignment for the Peripheral Mode Interface

For parameter assignment and event handling, the processor has to address the ASPC2 in the peripheral mode. For this, a small address window of 64 bytes is available (refer to Table 1.9-1). An ASPC2 register is selected only if the 'XCS signal' and the corresponding offset address is applied.

Read		Write		OFF-ADDR

High(A0=1,XBHE=0)
High(A0=0,XBHE=1)
	Low(A0=0.XBHE=1)
	Low(A0=1,XBHE=0)
		High(A0=1,XBHE=0)
		High(A0=0,XBHE=1)
			Low(A0=0.XBHE=1)
			Low(A0=1,XBHE=0)
				Intel
				Motorola

TTHOLD15..8	TTHOLD 7..0	TTR15..8	TTR7..0	00H
Delay Timer15..8	Delay Timer7..0	INT-MASK-REG 15..8	INT-MASK-REG 7..0	02H
INT-REQ-REG 15..8	INT-REQ-REG 7..0	INT-REQ-REG 15..8	INT-REQ-REG 7..0	04H
INT-REG 15..8	INT-REG 7..0	INT-ACK-REG 15..8	INT-ACK-REG 7..0	06H
Status-REG15..8	Status-REG 7..0	Mode-REG0 15..8	Mode-REG0 7..0	08H
Status-REG 31..24	Status-REG 23..16	Mode-REG1-Set 15..8	Mode-REG1-Set 7..0	0AH
-	LAS-REG 3..0	Mode-REG1-Res 15..8	Mode-REG1-Res 7..0	0CH
		SCB-BASE-LW 15..8	SCB-BASE-LW 7..0	0EH
		SCB-BASE-HW 31..24	SCB-BASE-HW 23..16	10H
		TSLOT-REG 13..8	TSLOT-REG 7..0	12H
		TID1-REG 9..8	TID1-REG 7..0	14H
		TID2-REG 9..8	TID2-REG 7..0	16H
		TRDY-REG 9..8	TRDY-REG 7..0	18H
		BR-REG 10..8	BR-REG 7..0	1AH
		SAP-MAX 7..0	TS-ADDR-REG 6..0	1CH
		Token-Err- 7..0	GUD-REG 7..0	1EH
		TQUI-REG 7..0	LAY4-HLen-REG 7..0	20H
		Resp-Err- 3..0	HSA 6..0	22H
		Reserved 2	Reserved 1	24H
		Mode-REG2 5..0	Retry Tok 3..0 Retry Msg 3..0	26H
		WAIT STATES 15..8	WAIT STATES 7..0	28H

Table 1.9-1: ASPC2 Internal Registers

Token Rotation Timer (Register):
The setpoint rotation time is parameterized (TTR). For system initialization, the token holding time (TTHOLD) can be read out.

Interrupt Controller Register:
The meaning of these registers will be explained in Chapter 4.3.

Status Register:
The meaning of this register is described in Chapter 4.2.1.4.

LAS Register:
Via this register, the processor can read out the LAS RAM. Each access increments the internally generated LAS RAM address (Chapter 4.7).

Mode Register:
The mode registers 0, 1, and 2 are used for parameterizing individual bits. The meaning is described in Chapter 4.2.1.

SCB BASE HW/LW
The 20/32 bit address of the system control block is parameterized (Chapter 4.2.1).

Slot Timer Register
The wait for receive time TDL is parameterized (Chapter 4.2.1).

Idle Timer Register
The follwing timing is parameterized (Chapter 4.2.1):

- TID1 (valid after acknowledgement, response, or token messages)
- TID2 (valid after a call message that is not acknowledged)
- TRDY (ready time, valid prior to sending a reply message)

Baudrate Register
The scaling factor is parameterized (Chapter 4.2.1).

TS Address Register
The station address is parameterized (Chapter 4.2.1).

GUD Register
The GAP updating time TGUD is parameterized (Chapter 4.2.1).

Token Error Limit Register
The number of unvalidated token messages per 256 token rotations is parameterized, before the ASPC2 enters the *listen token mode* (Chapter 4.2.1).

SAP MAX Register
The highest SAP list number generated in the SCB is parameterized (Chapter 4.2.1).

LAY4 Hlen Register
Two different Layer4 header lengths are parameterized (Chapter 4.2.1).

TQUI Register
The modulator fading time TQUI is parameterized (Chapter 4.2.1). In addition, the delay time between XENBUF and XREQRDY is set (Chapter 4.1.2).

HSA Register
The highest active address is parameterized (Chapter 4.2.1).

Response Error Limit Register
The number of faulty response messages is parameterized, according to which double token is assumed (Chapter 4.2.1).

Reserved Register
In the reserved register, the early setting of RTS for TxD is parameterized (Chapter 4.2.1).

Retry Register
The number of message and token repetitions is parameterized (Chapter 4.2.1).

Wait States Register
The wait states for each 256K memory segment are parameterized (Chapter 4.2.1).

Delay Timer
The meaning of this timer is described in Chapter 4.5.

1.9 Pin Description

The ASPC2 has an R-PQFP G100 pin case with the following signals:

11 Inputs:

Signal Name	Quantity	Function	Source
CLK*	1	Clock Input (48MHz)	System Support
RESET*	1	Hardware Reset	CPU, Port
XCS	1	Chip Select	System Support
XRDY	1	Asynchronous Ready	System Support
X/HOLDAIN	1	Hold Acknowledge In	CPU or previous AMPlus
XREQ	1	Bus Request from ext. master	System Support
RxD	1	Serial receive channel	RS 485 Receiver
XCTS	1	Clear to Send	FSK Modem
XB8/B16**	1	System bus configuration	Sold. Jumper
XINT/MOT***	1	System bus configuration	Sold. Jumper
XWRL-MODE**	1	80C165-Interface: XWRH, XWRL	Sold. Jumper

* CMOS Schmitt Trigger Input ** CMOS input with integrated pull-up
*** CMOS input with integrated pull-down

Table 1.10-1: Input Pins of the ASPC2 (without test inputs)

35 Outputs

Signal Name	Quantity	Function	Destination
X/INT-CI	1	Interrupt Confirmation/Indication	CPU or Inter. Contr.
X/INT-EVENT	1	Interrupt Event Register	CPU or Inter. Contr.
DT/XR	1	Data Transmit Receive	System Support
HOLD	1	HOLD Request, Tristate Output	CPU
X/HOLDAOUT	1	Hold Acknowledge Out	Next AMPlus
XENBUF	1	Enable of ext. buffers at X/REQ	System Support
XREQRDY	1	Ready for ext. bus master	CPU or System Support
TxD	1	Serial send channel	RS 485 Sender
RTS	1	Request to Send	RS 485 Sender
AB19..6	14	Address Bus	Memory, System Support
DIA9..0	10	Diagnostic Port	System Support
XCLK2	1	24MHz Clock	System Support
XHTOK	1	Hold Token Indication	Led

Table 1.10-2: Output Pins of the ASPC2

25 Bidirectional Signals:

Signal Name	Quantity	Function	Source/Destination
AB5..0	6	Address Bus	CPU, Memory, Sys. Support
DB15..0*	16	Data Bus	CPU, Memory
XBHE*	1	Byte High Enable	CPU, Memory
XRD	1	Read	CPU, Memory
XWR	1	Write	CPU, Memory

*** TTL input with integrated pull-up**

Table 1.10-3: Bidirectional Pins of the ASPC2

Other Signals :	VDD Pins	10
	VSS Pins	17
	Test Pins	2
Total:	Inputs:	11
	Outputs:	35
	Bidirect.:	25
	Other:	29
	All Pins:	100

1.10 Pin Assignment

The ASPC2 has a 100 pin P-MQFP EIAJ case:

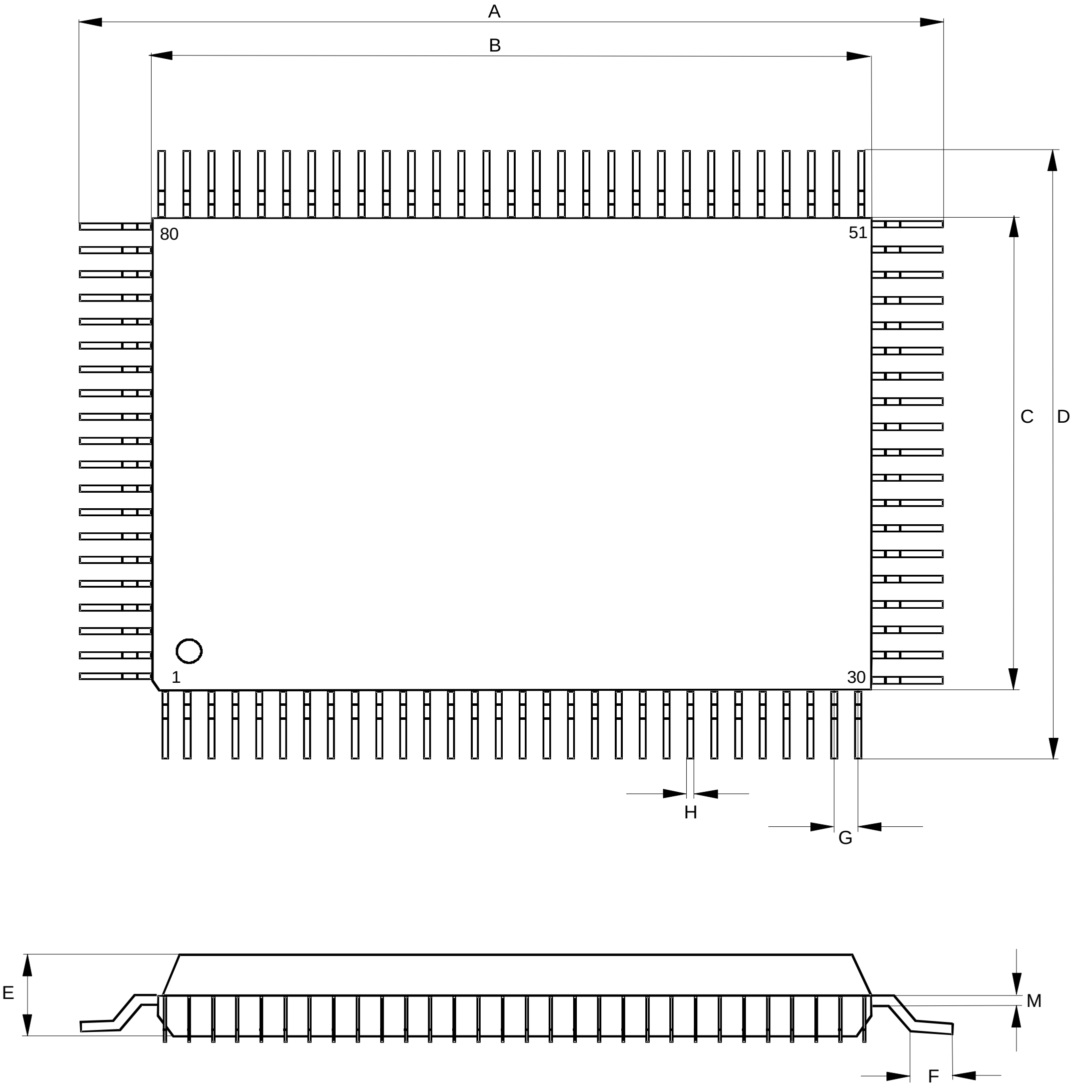
01: XRD	T	26: XREQ	T	51: AB11		76: DIA5	
02: DT/XR		27: XREQRDY		52: AB10		77: VSS	
03: VSS2		28: XENBUF		53: VSS		78: DIA4	
04: VDD3		29: VSS2		54: VDD3		79: DIA3	
05: XBHE/XWRH	TPU	30: XINT/MOT	CPD	55: AB9		80: VSS2	
06: HOLD		31: XTEST0	C	56: AB8		81: DIA2	
07: DB7	TPU	32: XTEST1	C	57: AB7		82: DIA1	
08: DB6	TPU	33: XWRL_MODE	CPU	58: AB6		83: DIA0	
09: VDD		34: XCTS	C	59: VDD		84: X/INT-EV	
10: VSS		35: DIA9		60: VSS		85: X/INT-CI	
11: DB5	TPU	36: XB8/B16	CPU	61: AB5	T	86: RTS	
12: DB4	TPU	37: XWR/XWRL	T	62: AB4	T	87: TXD	
13: DB3	TPU	38: AB19		63: AB3	T	88: DB15	TPU
14: DB2	TPU	39: AB18		64: AB2	T	89: DB14	TPU
15: VDD		40: VDD		65: VDD		90: VDD	
16: VSS		41: VSS		66: VSS		91: VSS	
17: VSS3		42: AB17		67: VSS3		92: VSS3	
18: DB1	TPU	43: AB16		68: AB1	T	93: DB13	TPU
19: DB0	TPU	44: AB15		69: AB0	T	94: DB12	TPU
20: X/HOLDAOUT		45: AB14		70: XCLK2		95: DB11	TPU
21: X/HOLDAIN	T	46: VSS		71: CLK	CS	96: DB10	TPU
22: RESET	CS	47: VDD		72: XHTOK		97: VSS	
23: RXD	C	48: VSS2		73: DIA8		98: VDD	
24: XRDY	T	49: AB13		74: DIA7		99: DB9	TPU
25: XCS	T	50: AB12		75: DIA6		100: DB8	TPU

Table 1.11-1: Pin Assignment

VDD: Output pads and internals cells
VDD3: Input pads
VSS: Output pads
VSS2: Internal cells
VSS3: Input pads
T: TTL level
TPU: TTL level with pull-up
C: CMOS input
CPU: CMOS input with pull-up
CPD: CMOS input with pull down
CS: CMOS Schmitt Trigger input

1.11 Housing (P-MQFP100)

P-MOFP100 Case



A 23.20 + - 0.25 mm
B 20.00 + - 0.20 mm
C 14.00 + - 0.20 mm
D 17.20 + - 0.25 mm
E 2.70 + - 0.20 mm
F 0.88 + - 0.15 mm
G 0.65 mm
H 0.30 + - 0.08 mm
L 0.25 mm min
M 0.16 mm typ

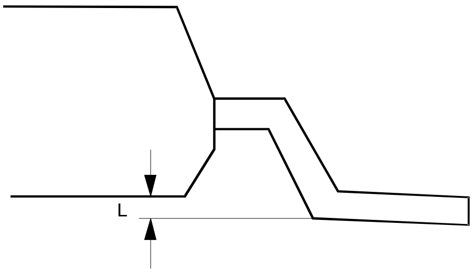


Figure 1.12-1: Mechanics

1.12 Notes on Processing

Always, **EGB protective steps** have to be taken regarding all electronic components.

The ASPC2 is a **cracking sensitive device** that has to be handled accordingly.

2 FLC Interface

2.1 System Control Block

The system control block SCB (refer to Figure 2.1.1-1) is the interface between the ASPC2 and the FLC.

The following lists are chained in the SCB:

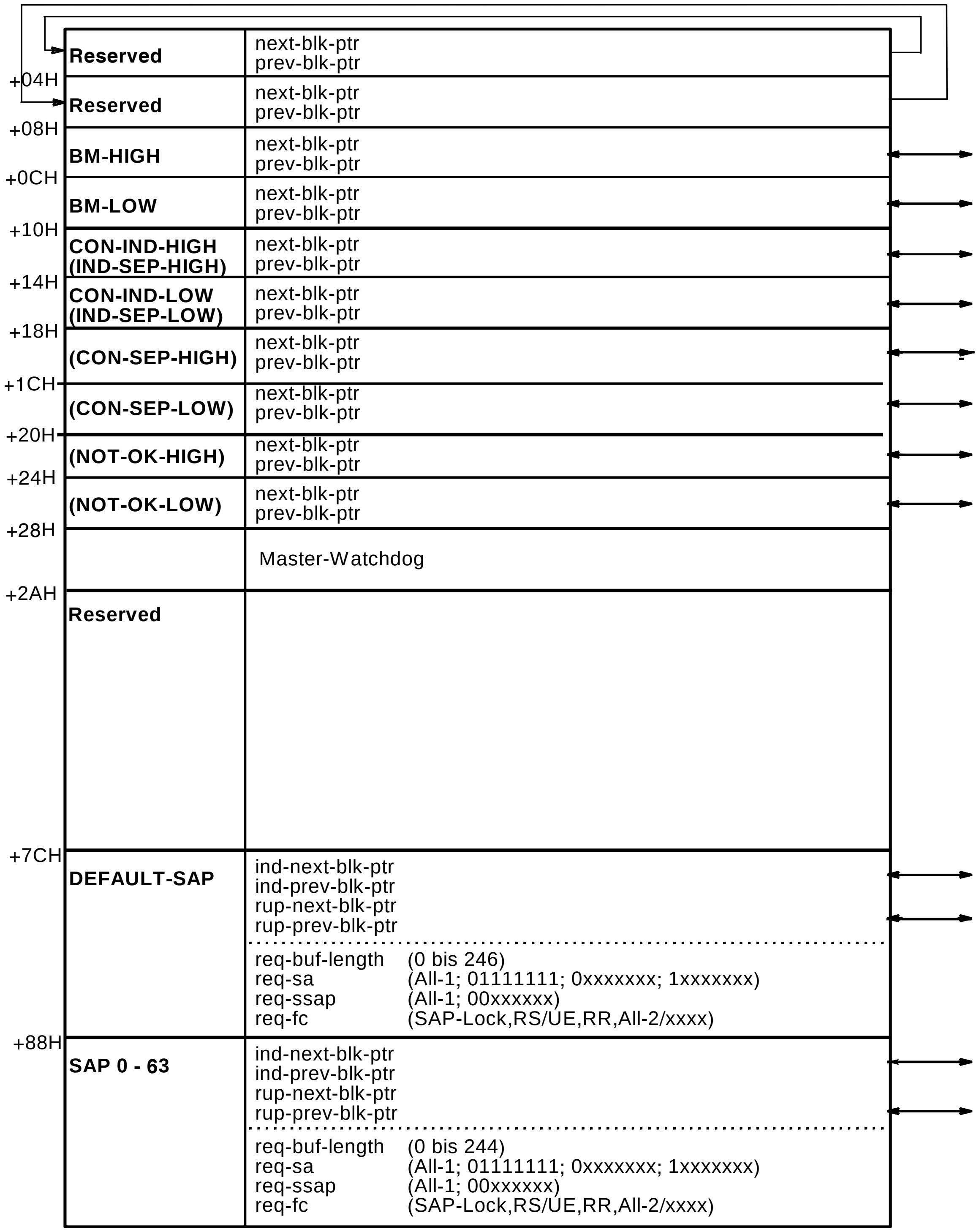
- Background Message High List (BMH)
- Background Message Low List (BML)
- Confirmation Indication High List (CON-IND-HIGH)
  or separate Indication High List (IND-SEP-HIGH)
- Confirmation Indication Low List (CON-IND-LOW)
  or separate Indication Low List (IND-SEP-LOW)
- Separate Confirmation High List (CON-SEP-HIGH)
- Separate Confirmation Low List (CON-SEP-LOW)
- NOT-OKAY High List (NOT-OK-HIGH)
- NOT-OKAY Low List (NOT-OK-LOW)
- DEFAULT SAP List, SAP Lists (SAP[0..63])

Each request is entered in an application block (REQ APB) (refer to Chapter 2.2.1). Several such REQ APBs together make up a chained list, also called queue (refer to Chapter 2.3.1).

2.1.1 Request Lists (BMH, BML)

The ASPC2 manages the following background message lists (BMH, BML) in which the FLC enters send requests. The ASPC2 processes the BMH list (high priority) and the BML list (low priority). For each new request selection, the ASPC2 always goes first to the list with the highest priority; if it is empty, it branches to the next, lower priority etc.. Regardless of the token holding time, at least one high priority request is processed per token rotation. The two request priorities of PROFIBUS are mapped onto the background lists BMH and BML.

All high priority requests are confirmed in a high priority confirmation list (CON-IND-/CON-SEP-/NOT- OK-HIGH), and all low priority requests in a low priority confirmatrion list (CON-IND-/CON-SEP-/ NOT-OK-LOW) (see below). The priority is not determined from the entry in the corresponding request lists, but from the coding of the PRIO bit in Opcode 7..0 of the application block (refer to Chapter 2.3.1). The GAP messages are attached only to the BML.



	Reserved	next-blk-ptr prev-blk-ptr	
+04H	Reserved	next-blk-ptr prev-blk-ptr	
+08H	BM-HIGH	next-blk-ptr prev-blk-ptr	
+0CH	BM-LOW	next-blk-ptr prev-blk-ptr	
+10H	CON-IND-HIGH (IND-SEP-HIGH)	next-blk-ptr prev-blk-ptr	
+14H	CON-IND-LOW (IND-SEP-LOW)	next-blk-ptr prev-blk-ptr	
+18H	(CON-SEP-HIGH)	next-blk-ptr prev-blk-ptr	
+1CH	(CON-SEP-LOW)	next-blk-ptr prev-blk-ptr	
+20H	(NOT-OK-HIGH)	next-blk-ptr prev-blk-ptr	
+24H	(NOT-OK-LOW)	next-blk-ptr prev-blk-ptr	
+28H		Master-Watchdog	
+2AH	Reserved		
+7CH	DEFAULT-SAP	ind-next-blk-ptr ind-prev-blk-ptr rup-next-blk-ptr rup-prev-blk-ptr	
		req-buf-length req-sa req-ssap req-fc	(0 bis 246) (All-1; 01111111; 0xxxxxxx; 1xxxxxxx) (All-1; 00xxxxxx) (SAP-Lock,RS/UE,RR,All-2/xxxx)
+88H	SAP 0 - 63	ind-next-blk-ptr ind-prev-blk-ptr rup-next-blk-ptr rup-prev-blk-ptr	
		req-buf-length req-sa req-ssap req-fc	(0 bis 244) (All-1; 01111111; 0xxxxxxx; 1xxxxxxx) (All-1; 00xxxxxx) (SAP-Lock,RS/UE,RR,All-2/xxxx)

All-1 = 11111111; All-2 = xxxx0000;

Figure 2.1.1-1: SCB of the ASPC2

The ASPC2 now processes successively the REQ APBs from the request lists. It fetches the request message from the request segment of the application block, and sends it to the remote station. If the response is faultless, it enters -if needed- the response data and the status in the response segment of the application block. If the remote station does not respond, or if its response was received faulty, the ASPC2 generates a repetition if needed, etc. If the last repetition is unsuccessful also, the request will be confirmed with ‘NA (No Access)’.

If “SEP LIST = 0” (Mode Register 0), the ASPC2 removes a request that was processed (REQ APB) from the corresponding request list and enters it in the CON IND HIGH or CON IND LOW (refer to Chapter 2.3.1). In addition, it generates an interrupt for the FLC for the executed **confirmation (INT CI)**. If “SEP LIST = 1” is set, the ASPC2 enters the faultless send requests in the CON SEP HIGH/LOW, and the faulty requests (for example “NA”) in the NOT OK HIGH/LOW with the associated **event interrupts (INT EV) CON SEP LIST** or **NOT OK LIST** (refer to Chapter
4.3).
If the respective request lists are empty, the ASPC2 sets “REQ EMPTY = 1” in Mode Register 1. It now does not access the request lists until the FLC has chained a new REQ APB in the lists. Each time the FLC chaines an REQ APB in the request lists, it has to set “REQ EMPTY = 0” in Mode Register 1. All request lists can be locked at the ASPC2. If “LOCK REQUEST = 1” is set in Mode Register 1, no requests will be sent. The ASPC2 immediately passes the token.

The SCB is only an anchor for the BMH and BML. The **‘next block pointer (next blk ptr)** points to the first REQ APB of the list (refer to Chapter 2.3.1). If there is no REQ APB in the list, the ‘next blk ptr” points to itself.

Watchdog for Master Monitoring
A watchdog function is located on the 28.H word of the SCB which monitors the master. When a new request is executed, the ASPC2 decrements this cell. When the value ‘0000H’ is reached, no more requests are processed, and “REQ EMPTY = 1” is set. In this case, however, the counter stops in the SCB on ‘0001H’. The user has to load this cell cyclically to a start value, which has to be < ‘FF7FH’, however. If the value is higher (>= ‘FF7FH’), the watchdog is turned off.

2.1.2 CON IND HIGH, CON IND LOW or IND SEP HIGH, IND SEP LOW

If “SEP LIST = 0”, all processed applications blocks get into the confirmation-indication lists, the high prioity APBs in the CON IND HIGH, and the low priority APBs in the CON IND LOW. REQ APBs from the request lists are entered as **confirmation**, and IND APBs and RUP APBs from the SAP lists (refer to Chapter 2.1.5) as **indicaton** (refer to Chapter 2.2.1). Each entry is signalled to the FLC with an interrupt (**INT CI**) (refer to Chapter 4.3).

In the mode “SEP LIST = 1”, these lists represent pure indication lists. Here, the ASPC2 enters only the processed IND APBs and RUP APBs of the SAP lists. Each entry is signalled to the FLC by interrupt (**INT CI**).

Here also, the SCB is used only as an anchor for CON IND HIGH/LOW or IND SEP HIGH/LOW. The ‘next blk ptr’ points to the first APB in the list. If there is no APB in the list, it points to itself.

2.1.3 CON SEP HIGH, CON SEP LOW

These lists have a meaning only for “SEP LIST = 1” or collect requests. If “SEP LIST = 1”, all faultlessly executed send requests are entered in these lists (confirmation); that is, all REQ APBs that were acknowledged positive by the remote station, or the blocks that don’t require acknowledgement (SDN). Each entry is signalled to the FLC with the event interrupt **CON SEP LIST** (refer to Chapter
4.3). In the case of **collect request**, marked in the opcode field of the REQ APB (refer to Chapter 2.2.1), all send requests associated with the collect service are entered in the CON SEP HIGH/LOW.
Here also, the SCB is used only as an anchor for CON SEP HIGH/LOW. The ‘next blk ptr’ points to the first REQ APB in the list. If there is no REQ APB in the list, it points to itself.

2.1.4 NOT OK HIGH, NOT OK LOW

These lists have only one meaning in the case of “SEP LIST = 1” and repeat requests. All request blocks that the remote station acknowledges negative or not at all are stored in these lists by the ASPC2 (refer to Chapter 2.2.1). Each entry is signalled to the FLC with the event interrupt **NOT OK LIST** (refer to Chapter 4.3).
Here also, the SCB is used only as an anchor for NOT OK HIGH/LOW. The ‘next blk ptr’ points to the first REQ APB in the list. If there is no REQ APB in the list, it points to itself.

2.1.5 SAP Lists (SM SAP, DEFAULT SAP, SAP [0..63])

In the FLC, a data transmission service is processed via a **service access point (SAP)**. If “FULL SAP = 0”, there are up to 64 SAPs **SAP [0..63]** at each station at the same time (refer to Mode Register 0). For reasons of message efficiency, the transmission of SAPs can be omitted; then, the data transmission services have to be processed via the **default SAP**. The stations can communicate with each other via any SAP.
Each service access point (DEFAULT SAP, SAP [0..63] has two separate lists where the FLC makes receive resources (indication application blocks, IND APBs) available in the one list, and reply update application blocks (RUP APBs) in the other (refer to Figure 2.1.5-1). The ASPC2 enters the receive messages in the IND APB list, and fetches the response data from the RUP APB list.

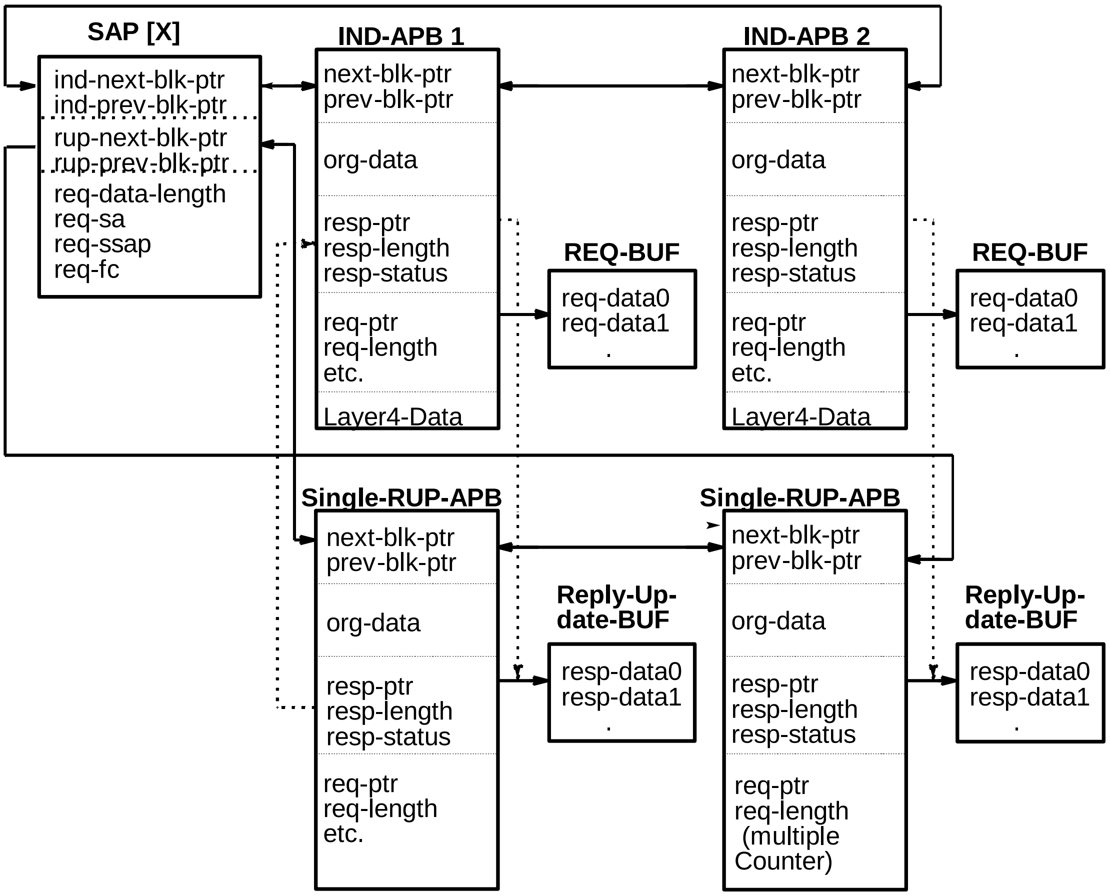


Figure 2.1.5-1: SAP Structure ASPC2

In the RUP APB list, for example, several single reply update buffers can be chained one after the other via one Single RUP APB respectively.

If the ASPC2 receives a request message, first “MAC Lock Flag = 1” under “Type = indication” is set in the first IND APB of the list (refer to Table 2.1.5-1). In addition, “resp length, resp status = 0” is loaded. If there is a reply update buffer at an SRD, the ASPC2 sets “MAC Locked = 1” and “Type = Indication” in the opcode field of the first RUP APB of the list. Then, the request message is transferred to the IND APB, and the reply update buffer is transmitted. If there is no RUP APB, it responds with ‘SC’ after the request message was cleared. If a reply update buffer was transmitted, the ASPC2 copies, in the case of a valid indication (after the faultless receipt of a new call message header), ‘resp ptr’, resp length, and resp status’ from the RUP APB to the IND APB. In the case of a single RUP APB, “MAC Locked = 0” , “Inv.Indication Flag = 0” is then set in the opcode, and the block is entered in the CON IND list (according to the priority of the request message). Then, “MAC Locked = 0”, “Inv.Indication Flag = 0” , “Reply Flag/Indication Mode Data = 0” is set in the opcode of the IND APB, the PRIO bit is assigned according to the request priority, the block is chained in the CON IND list, and the CON IND interrupt is triggered. In the case of an indication, the difference between a RUP APB and an IND APB in the opcode (Bit(3), can be recognized. In the case of a RUP APB, the ASPC2 does not reset this flag (reply flag). However, the FLC had to set “Reply Flag = 1” when making it available. In the case of the IND APB, the ASPC2 clears this bit position (indication mode data flag).

	IND-APB Opcode = Opcode..	Single RUP-APB Opcode = Opcode..
MAC-Locked set (Receive)	or 82H	or 82H
MAC-Locked reset (Indication)		
Ind.	and 66H (or PRIO-High)	and 6FH (no PRIO)
Ind. invalid (Inv.Ind.Flag = 0)	and 7CH (stays in SAP)	and 7DH (stays in SAP)
Ind. invalid (Inv.Ind.Flag = 1)	and 76H (after CON-IND)	and 7FH (after CON-IND)

Table 2.1.5-1: Opcode Handling

If the indication is invalid, the ASPC2 sets “MAC Locked = 0” and “Type = 00B” in the opcode of both application blocks if “Inv.Indication Flag = 0”, whereby the blocks remain chained in the SAP list. If “Inv.Indication Flag =1” is set, “MAC Locked = 0” is set in both blocks; at the IND APB “Reply Flag/Indication Mode Data = 0” is set; then, first the RUP APB and after that the IND APB, or an APB of either is chained in the low priority CON IND list, and the CON IND interrupt is generated.

So that not all SAP lists have to be set up in the SCB, the highest SAP list number is parameterized for the ASPC2 (SAP MAX, refer to Chapter 4.2.1). If the ASPC2 receives a message to an SAP that is not available, it responds with *No Service Activated* (SD1 response). In each SAP list, the FLC can store validation data about the following:

- Request buffer length (req-buf-length)
- Request station (req-sa)
- Request service access point (req-ssap)
- Request service (req-fc)

This data is valid for all receive messages to the respective SAP. Thus, among other things, private connections can be implemented between 2 stations. If not only certain request stations, request SAPs, and request transmission services are to be allowed, the FLC has to enter the coding “all” in the corresponding fields.

When receiving a request message (call), the ASPC2 validates the corresponding message header characters with the values assigned to it from the SAP list. The following responses are possible:

req-buf-length: This value specifies the length of the request buffer in the SAP list. If the Layer2 net data length of the request message is larger than the available buffer length, the ASPC2 rejects this message with *No Resource* (SD1 response).

req-sa: If the FLC preassigns "All-1 = FFH" to this field, this validation is omitted. With the coding "req-sa = 7FH", the SAP can be deactivated. The ASPC2 rejects all request messages to this SAP and responds with *No Service Activated*. Otherwise, it compares the SA field of the request message with the preselected req-sa. If they are not the same, it rejects the message and responds with *No Service Activated* . It is to be noted that at the receipt of a remote SAP, the 8th bit (address extension bit) of the req-sa has to have been set to log. ‘1‘, or of a remote default SAP to log.'0' . In the SM SAP lists, the setting always has to be on log.'0' .

req-ssap: If this field is preassigned "All-1 = FFH", validation is omitted. Otherwise, the ASPC2 compares the SSAP field of the request message with the preselected req-ssap. If they are not the same, it also responds with *No Service Activated* . It is to be noted that in all SAP lists, the 2 upper bits of the req-ssap are set to log.'0'. For validation of a certain remote station under default SAP, 'All-1 = FFH" is to be entered in this field. The validation Default SAP of any station is not possible.

req-fc: This is a tripartite field. The access value is on the lower 4 bits req-fc(3..0). If "req-fc(3..0) = 0H (All-2)" is preassigned, validation is omitted. Otherwise, the ASPC 2 validates the function code from the FC field (FC3..0) of the request message with the preassigned req-fc(3..0) . If there is no hit, the ASPC 2 responds with *No Service Activated*.

SAP locked	Res	Event Ind		Access Value			
		RS/ UE	RR	3	2	1	0
7							0

Via the most significant bit, the FLC can temporarily lock the SAP (req-fc(7) = 1, SAP locked). The ASPC2 then rejects all request messages, and responds with *User Error* (SD1 response).
req-fc(6) is always to be set to 0.
The ASPC2 uses the remaining bits as event indication if a request message had to be rejected for the reasons mentioned above. The ASPC2 sets the corresponding bit to log.'1'. After evaluation, the FLC has to reset the bit to log.'0'. The individual bits have the following meaning:

RR = No Ressource (see below)
UE = User Error
RS = No Service Activated

Service	Access3..0	Service	Access3..0
All	0H		8H
SDN-Low	1H	SRD-Low	9H
SDN-High	2H	SRD-High	AH
SDN-Low/High	3H	SRD-Low/High	BH
-	4H		
SDA-Low	5H		
SDA-High	6H		
SDA-Low/High	7H		

Table 2.1.5-2: Coding Access Value

Note: SAP validation takes place according to the following sequence:

❍ > SAP-MAX:	resp-status RS
❍ req-sa:	resp-status RS
❍ req-ssap:	resp-status RS
❍ SAP-locked:	resp-status UE
❍ req-fc:	resp-status RS
❍ no Ind Resource:	resp-status RR
❍ L4+L2 data < parameterized L4:	resp-status RS
❍ req-buf-length:	resp-status RR

Here also, the SCB is used as anchor for the DEFAULT SAP list and SAP [0..63] lists. The ‘next blk ptr’ points to the first application block of the list. If the list is empty, it points to itself.

Important: The basic address of the SCB has to be word-aligned, because the ASPC2 can only perform word transfers to even addresses!

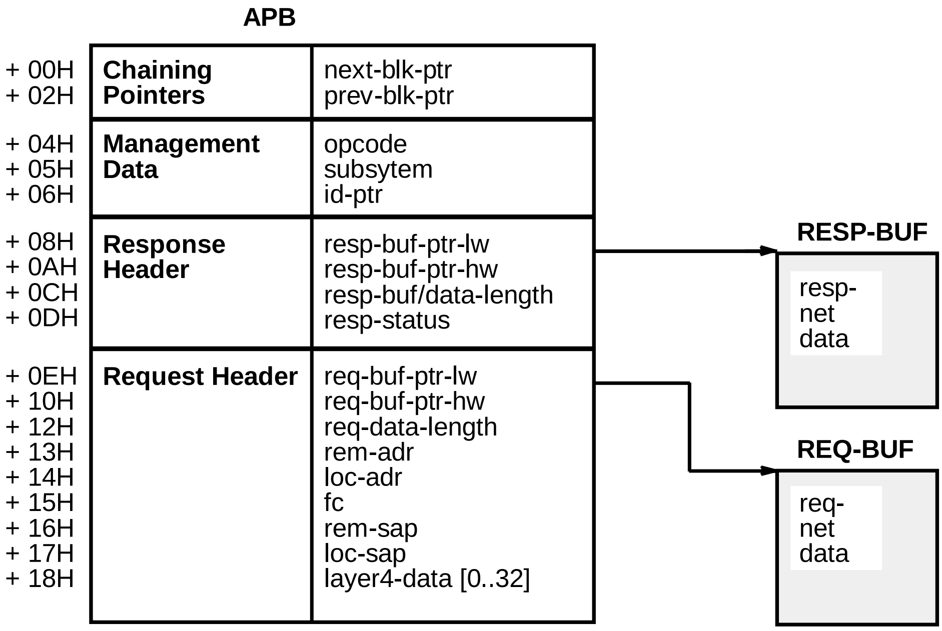
2.2 Structure of the Application Block

2.2.1 Appplication Block

The structure of the application blocks (APB) for the request and the SAP lists is identical (refer to Figure 2.2.1-1).
In the application blocks of the request lists (REQ-APB), the FLC preprocesses the request message, and the ASPC2 stores the response message. In the case of the application blocks of the SAP lists (IND APB, RUP APB), it is exactly the other way around.

The following data packages are included in the application block:

- Chaining pointer
- Management data
- Request header
- Request buffer (Request Net Data)
- Response header
- Response buffer (Response Net Data)



APB

+ 00H + 02H	**Chaining Pointers**	next-blk-ptr prev-blk-ptr
+ 04H + 05H + 06H	**Management Data**	opcode subsytem id-ptr
+ 08H + 0AH + 0CH + 0DH	**Response Header**	resp-buf-ptr-lw resp-buf-ptr-hw resp-buf/data-length resp-status
+ 0EH + 10H + 12H + 13H + 14H + 15H + 16H + 17H + 18H	**Request Header**	req-buf-ptr-lw req-buf-ptr-hw req-data-length rem-adr loc-adr fc rem-sap loc-sap layer4-data [0..32]

Figure 2.2.1-1: Structure of the Application Block

Chaining Pointers:
The chaining pointers are used for setting up the list (refer to Chapter 2.3.1).

Management Data:
The management data characterizes the APB. It contains the following information:

opcode: This field specifies the momentary association of the APBs (for example, being processed by the MAC). When being rechained in the confirmation-indication lists, the ASPC2 enters the corresponding ID for confirmation or indication in the opcode, and sets the priority bit.

Opcode

MAC-Locked	LAY4-Head-Len		Collect (1)	Collect (0)	Typ		PRIO
	1	0	Inv. Ind. Flag	Ind.Mo. Data Reply-Flag	1	0	
7							0

PRIO: Confirmation: Is not changed by the ASPC 2.

Indication: Is set by the ASPC 2 in accordance with the priority of the request message.

Type: 10 = Request

11 = Confirmation

01 = Indication

Collect(1..0): In the case of the Collect service, several requests are bound together as a unit, and in addition, the function Force Pass Token is coded (refer to Chapter 3.2.2). All requests are entered in the CON-SEP lists, but only at the last request of a Collect service, the CON-SEP-LIST interrupt is generated. The encoding of the collect requests is shown in the table below:

Collect (1..0)		Collect Reques
0	0	no
1	1	Yes, first-penultimate
0	1	Yes, last
1	0	Yes, last

Collect (1..0)		Collect-Auftrag	Eintrag in Con-SEP-Liste	CON-SEP-LIST-Interrupt	Force-Pass-Token
0	0	nein	nein(bei SEP-List=0)	nein (bei SEP-List=0)	nein
1	1	ja, erste- vorletzte	ja (Typ: 10B) nein (Typ: 00B)	nein	nein
0	1	ja, letzte	ja (Typ: 10B) nein (Typ: 00B)	ja	nein
1	0	ja, letzte	ja (Typ: 10B) nein (Typ: 00B)	ja	ja

Indication Mode Data: This flag has to be set in the IND-APB if there is to be no indication during empty polling. At the indication, the ASPC 2 resets this flag.

0 = All request messages that were received faultlessly are indicated.
1 = The empty polling check is made only in the case of SDA and SRD without RUP-APB in the SAP list. There is an indication only if data was transmitted in the call message.

Reply Flag: The FLC has to set this flag in the RUP APB. At the indication, the ASPC2 resets this flag.

0 = No RUP APB

1 = RUP APB

Invalid Indication Flag: In the case of the SAP resources, the user can have the ASPC2 remove a locked IND-APB or RUP-APB ("MAC-Locked = 1") from the SAP lists after the indication (invalid or valid).

0 = If there is a valid indication, the IND/RUP-APB is put in the corresponding CON-IND list.

If there is an invalid indication, the IND/RUP-APB remains in the corresponding SAP list.

1 = If there is a valid indication, the IND/RUP-APB is put in the corresponding CON-IND list. The ASPC2 resets the invalid indication flag.

If there is an invalid indication of an IND-APB, the ASPC 2 chaines the IND-APB in the low priority CON-IND list. The invalid indication flag remains set.

LAY4 Head Len: Is not changed by the ASPC 2 !
00 = No Layer4 data in the request header
01 = Layer4 data consists of 2 bytes
10 = Layer4 data consists of the par. length
(L4-Hl1$_{3..0}$: 32 bytes max.)
11 = Layer4 data consists of par. length
(L4-Hl2$_{3..0}$: 32 bytes max.)
If a responder receives a request message with a shorther data length than its parameterized Layer4 data length, this message is rejected with *No Service Activated*.

MAC Locked: 0 = Not being processed by the ASPC 2 (MAC).
1 = Being processed by the ASPC 2 (MAC) .

subsystem: The ASPC2 does not touch the 8Bit field.

id-ptr: The ASPC2 does not touch the 16Bit field.

Request Header:
In the request header, all important message header characters and branching to the request buffer is stored. It conatins in detail:

req-buf-ptr-lw/hw: This 20/32 bit pointer points to the request buffer. The pointer represents a linear or segment offset address (refer to XLIN/SEGOFF in Mode Register 0) , and is always to be made available by the FLC. The value req-length in the SAP lists specifies the length of the request buffer (refer to Chapter 2.1).

req-data-length: This value specifies the length of the net data entered in the request buffer (0 to 244 bytes with SAP expansion; 0 to 246 bytes for SM SAPs or default SAPs).

rem-addr: Is the station address of the remote station. Overall, 127 stations (Address 0 to 126) can be connected to the bus. Station address 127 is used as broadcast/multicast address, with which all/several stations are addressed. **If there is a remote SAP extension, the extension bit in rem-addr has to be set to log.'1' (8th bit, MSB).**
The remote station that is to carry on data traffic with the respective service access point of the local station can be specified in the SAP lists under req-sa (refer to Chapter 2.1). In the IND-APB, the ASPC 2 writes the received SA field to this cell (with or without extension bit).

loc-addr: Is the address of the local station. This value is not to be specified in the REQ-APB. The ASPC 2 generates this field from the parameterized station address (TS-ADDR-REG). For this, the following functions are defined in the loc-adr field of the des REQ-APBs:

Extension Bit	req_buf_ exec	resp_buf_ exec	res_	res_	don't care		
7	6	5	4	3	2	1	0

Extension Bit: A local SAP extension follows.

resp_buf_exec: Response data pointer has already been read once (the ASPC2 generates it, refer to Chapter 3.2.2).

res: 0 is to be entered in these bits.

In the IND-APB, ther ASPC 2 writes the received DA field to this cell (with or without extension bit).

fc: This value specifies the function code of the request message. The following codes are possible:

Function	Code
Send Data with Acknowledge low	00xy0011B
Send Data with no Acknowledge low	00xy0100B
Send Data with Acknowledge high	00xy0101B
Send Data with no Acknowledge high	00xy0110B
Request FDL Status with Reply	00xy1001B
Send and Request Data low	00xy1100B
Send and Request Data high	00xy1101B
NOP**)	1---1---B

xy = Consistency Control at the master x => WRCONS, y => RDCONS

**) The NOP request does not appear on the SER-BUS. It is used for controlling local events (such as force pass token).

In IND-APB, the ASPC 2 enters the received FC field (without FCB/FCV) in fc. The 'WRCONS' set by the user is saved. In the SAP lists, the request call service can be specified under req-fc; with it, the respective service access point of the local station may be addressed (refer to Chapter 2.1).

rem-sap: Is the SAP of the remote station. This field is valid only if the extension bit is set in rem-addr.

In the IND-APB, the ASPC 2 writes the received SSAP field to this cell. In the case of a default SAP, 'FFH' is entered here.

In the field req-ssap of the SAP lists, the SAP of the remote station may be specified with which the respective SAP of the local station would like to carry on data traffic. (refer to Chapter 2.1).

loc-sap: Specifies the service access point of the local station. This field is valid only if the extension bit is set in loc-addr.

In the IND-APB, the ASPC 2 writes the received DSAP field to this cell. In the case of a default SAP, 'FFH' is entered here.

layer4 data[0..31]: The length of the Layer4 data can be parameterized specific to the application. Overall, there are three different possibilities: in addition to the fixed length of 2 bytes, two different lengths up to 32 bytes can be parameterized for the ASPC2 (L4-HL1$_{3..0}$, L4-Hl2$_{3..0}$; refer LAY4-HLen-REG, Chapter 4.2.1). The length is selected separately for each REQ-APB and IND-APB in the opcode field "LAY4 Head-Len".

Request Buffer:
The request buffer is removed from the application block, and contains the net data of the request message.

Response Header:
In the response header, the response status and branching to the response buffer is stored. Meaning:

resp-buf-ptr-lw/hw: This 20/32 bit pointer points to the remotely located response buffer. The pointer represents a linear or segment offset address (refer to XLIN/SEGOFF in Mode Register 0), and is always to be made available by the FLC.

resp-buf/data-length: In the REQ-APB, the FLC enters here the length of the response buffer. At confirmation, the ASPC2 overwrites this value with the length of the entered net data in the response buffer (0 to 246 bytes).

In the RUP-APB, the FLC specifies in this cell the length of the net data entered in the response buffer (0 to 246 bytes). If 'resp-buf/data length = 0', no response data is available.

resp-status: In this field, the responder status is stored. In the case of the RUP-APB, the FLC has to make the status available if there is response data. The following codes are permissible:

Function	Code
'OK' ACKnowledgement pos. (generates SC, however	00000000B
"DL' Response FDL/FMA1/2-Data low (& Send Data OK)	00001000B
'DH' Response FDL/FMA1/2-Data high (& Send Data OK)	00001010B

At confirmation, the ASPC2 enters the resp-status in the REQ APB. The table below shows the permissible codes. In the mode "SEP-LIST = 1", the REQ-APBs with "resp-status = ACKnowledgement positive, Response FDL/FMA1/2-

Data low/high, Short character" are chained in the CON-SEP-HIGH/LOW, and the remaining ones in NOT-OK-HIGH/LOW. In addition, the resp-status shows whether there was a repetition. If there was a repetition, the 6th bit position is log.'1'; otherwise log.'0'.

Function	resp-status
'OK' ACKnowledgement positive	xyxx0000B
'UE' SAP locked	xyxx0001B
'RR' No Resource for Send Data	xyxx0010B
'RS' No Service activated	xyxx0011B
'DL' Response FDL/FMA1/2-Data low	xyxx1000B
(& Send Data OK)	xyxx1010B
'DH' Response FDL/FMA1/2-Data high	1y000000B
(& Send Data OK)	1y011111B
'SOK' Short charakter	1y101111B
'NA' no Access	1y111111B
'NADT' Double Token detect	1y001111B
'NAB' Response Buffer too small	10001110B
'NAINVRL' Invalid Request Length	

x-xx = StationType

y = 0 → No repetition

y = 1 → Repetiton

Function	StationType (x-xx)
passive	0-00B
Ready for the logic ring	0-10B
ative	0-11B

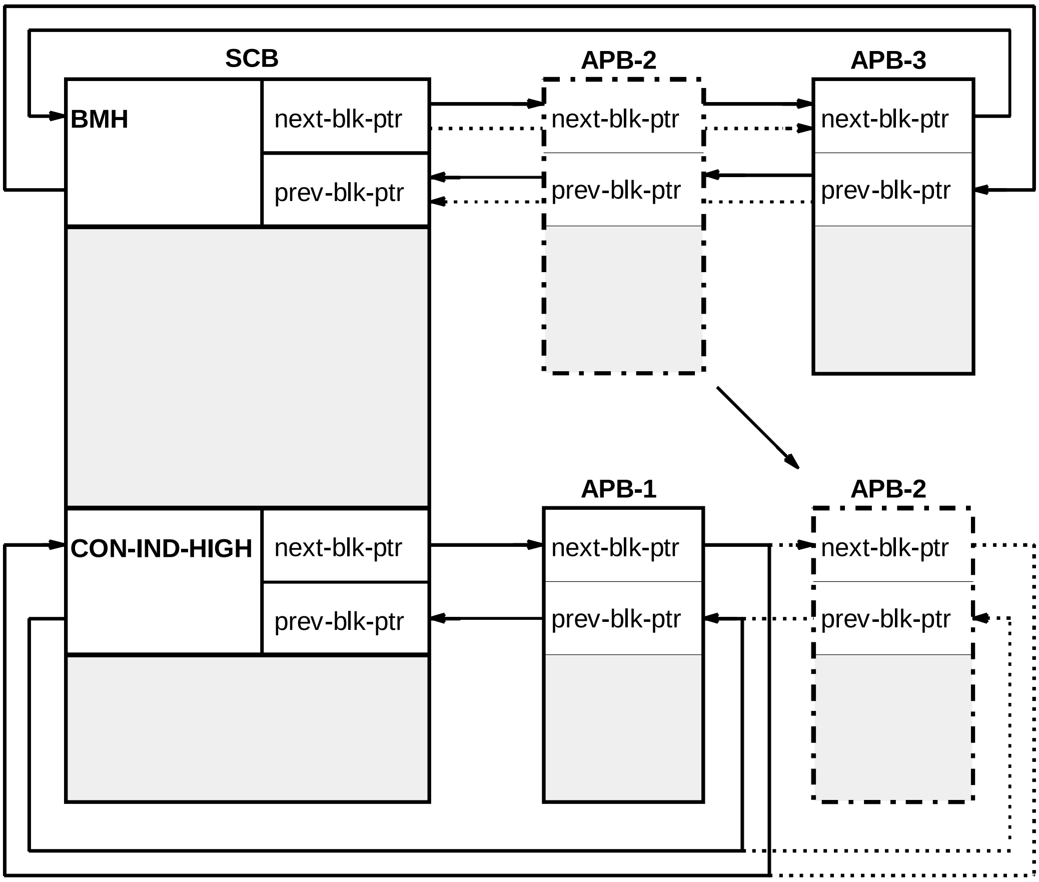
The ASPC2 takes the resp-status from the received FC field. In addition to the function code, it includes the station type of the remote station (x-xx). The station type includes the codings mentioned above:

Response Buffer:
The response buffers is removed from the RUP-APB, and contains the pure net data of the response message.
Important: All basic addresses of the application blocks, request buffers, and response buffers have to be word-aligned, because the ASPC2 can only execute word transfers to even addresses!

2.3 Structure of the Lists

2.3.1 Request and SAP Lists

The structure of the request and SAP lists is identical (refer to Figure 2.3.1-1). A list consists of one or several application blocks (APBs) which are chained in a type of queue in the list. The first entry gets to the head of the queue, the next one is chained to it, etc. The FLC can chain additional blocks into the list any time. However, it has to take note that the ASPC2 is not accessing the corresponding list at this time. For this, “BUS LOCK = 1” is to be set during chaining time in the Mode Register (refer to Chapter 4.2.1). As long as ‘BUS LOCK’ is active, the ASPC2 will not access locked areas.



Unchain from BMH:	next-blk-ptr (BMH) = next-blk-ptr (APB-2) prev-blk-ptr (APB-3) = prev-blk-ptr (APB-2)
Chain in CON-IND-HIGH:	next-blk-ptr (APB-2) = next-blk-ptr (APB-1) prev-blk-ptr (APB-2) = prev-blk-ptr (CON-IND-HIGH) next-blk-ptr (APB-1) = APB-2-BASE-ADR prev-blk-ptr (CON-IND-HIGH) = APB-2-BASE-ADR

Figure 2.3.1-1: List Structure, Rechaining

The lists are set up via double chaining the APBs. The **'next-block-pointer (next-blk-ptr)'** points to the next APB, and the **'previous-block-pointer (prev-blk-ptr)'** points to the previous APB in the list. **Both pointers consist of 16 bits and are an offset to the linear SCB-BASE address (20 Bit) if "XLIN/OFF = 0", or to the Segment-SCB-Base address (16 bits) if "XLIN/SEG = 1" was parameterized.** The base address of the APBs is to be entered respectively at the pointers. The SCB is used only as an anchor for the lists in that the 'next-blk-ptr' points to the first and the 'prev-blk-ptr' points to the last APB in the list. If the list is empty, 'next- and prev-blk-ptr' point to themselves respectively..
This double chaining has the advantage that the FLC can remove entered blocks relatively easily.
In the case of a confirmation, the ASPC2 removes the processed APB from the BMH, for example (refer to Figure 2.3.1-1), and attaches it to the CON-IND-HIGH. For unchaining, two pointers are to be reloaded, and for chaining 4 pointers. In the case of an indication, the sequence is identical.

2.4 ASPC2 PROFIBUS Services

2.4.1 Send Data with No Acknowledge (SDN)

This service permits the FLC of an active station to send data to a single remote station, to several (multicast) or simultaneously to all remote stations (broadcast). The FLC receives a confirmation regarding the completion of the transmission, but not regarding the correct receipt of the data. At the remote station, the indication is made after the faultless receipt of the message (INT-CI); faulty messages are filtered. If the FLC did not make a corresponding SAP or application block available, there is no indication, and in the req-fc of the corresponding SAP list, the responsible event bit is set (refer to Chapter 2.1). This also happens in the case of stations that don't accept the multicast.

Indication List: Depending on request priority, in the CON-IND-HIGH/LOW.
Confirmation-Status: 'OK'

2.4.2 Send Data with Acknowledge (SDA)

With this service, the FLC of an active station can send data to a single remote station. The remote station confirms the faultless receipt of the data with a short acknowledgement. The acknowledgement will be negative if the FLC of the remote station did not make a corresponding SAP or IND APB available. If there is an error in transmission, the ASPC2 of the local station repeats the call message (refer to Chapter 2.2.1). The number of repetitions (Retry M3..0) is parameterized in the retry register (refer to Chapter 4.2.1). After the last unsuccessful retry, it enters 'NA' in the responder status (refer to Chapter 2.2.1). The local station receives a confirmation after the faultless receipt of the acknowledgement, or after the last retry. At the remote station, the indication is carried out with the next new call message.

Indication List: Depending on request priority, in the CON-IND-HIGH/LOW.

Confirmation Status: 'SOK', 'UE', 'RR', 'RS', 'NA', 'NADT'

2.4.3 Send and Request Data with Reply (SRD)

This service permits the FLC of an active station to send data to a single remote station, and at the same time request data from it which the FLC of the remote station made available beforehand. The service also permits a local station to request data from a remote station without sending data to it.

The local station either receives the requested data, or an indication (short acknoweldgement) that the data was not available, or an indication (negative acknowledgement) that the FLC of the remote station has made no corresponding SAP or IND APB available. If there is a transmission error, the call is repeated (refer to Chapter 2.2.1). The FLC of the local station receives a confirmation after the faultless receipt of the response, or after the last repetition.
At the remote station, the indication is carried out with the next new call message.

Indication List: Depending on the request priority, in the CON-IND-HIGH/LOW.

Confirmation Status: 'SOK', 'DL', 'DH', 'UE', 'RR', 'RS', 'NA', 'NADT', 'NAB'

Note: The ASPC2 itself does not send SD3 messages, but is able to receive them.

2.4.4 Request FDL Status with Reply

This service is available to the user for establishing the life list. The life list is a current list of all stations that can be reached on the bus at the moment. The FLC has to address all station addresses with this service, and store the response status in the life list. There is no repetition regarding this service.
If there is no response or a faulty response, the ASPC2 enters 'NA' in the response status (refer to Capter 2.2.1). If the station is present, it responds with OK, including its station type (passive, ready for the logic ring, active). The ASPC2 confirms the receipt of the response to the FLC with an interrupt (confirmation).

Indication List: No indication

Confirmation Status: 'OK', 'NA', 'NADT'

3 Bus Access Protocol

3.1 Token Management

Regarding the ASPC2, it is a controlled bus access with a hybrid access method: decentrally according to the principle of **token passing**, and subordinated centrally according to the **master-slave** principle. The access control is in each active station. The passive stations are neutral regarding the bus access; that is, they are not performing any autonomous send activity, but only on request.
Communication is always initiated by the active station which receives the token. The token is passed from active station to active station in a numerically ascending station address sequence with the token message (refer to Figure 3.1-1). The station with the highest address then passes the token to the station with the lowest address to close the logical token ring. Each active station knows its predecessor (PS, from which it receives the token) and its successor (NS, to which it passes the token). The PS and NS addresses are determined autonomously by each station for the first time after the operating parameters are initialized, and later updated dynamically.

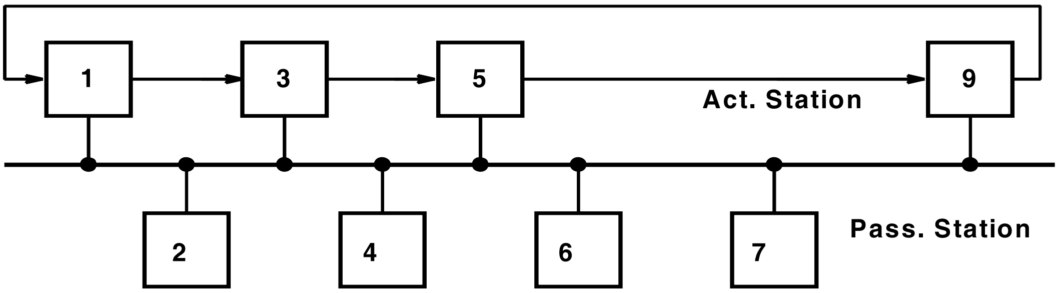


Figure 3.1-1: Logical Token Ring

The ASIC has control over the following fault modes and special operating modes:

- Token loss
- Multiple token
- Error at token passing
- Multiple assignment of station addresses
- Station with defective sender/receiver
- Adding and removing stations during operation

If an active station receives the token, measuring of the token rotation time starts. At the next token transfer, time measuring ends for the expired cycle with the result of the actual token rotation time TRR (real rotation time). At the same time, measuring of the following rotation time starts.
To adhere to a system response time that is necessary because of the application area, the token setpoint rotation time TTR (target rotation time) of the token in the logical ring has to be assigned (refer to Chapter 4.2.1). Each active station (ASPC2) can, regardless of the TRR, always process **one** high-priority message sequence per token receipt. To process additional high priority or low priority message cycles, the TRR has to be less than the TTR (available token holding time) at processing time.

3.2 Layer 2 Setting

3.2.1 PROFIBUS Mode

In Mode Register 0, Layer2 is set with 'RES'.

RES = 0
- PROFIBUS Layer2
- The GAP area ends at the HSA
- If 'req-sa validation' or 'req-ssap validation' is invalid, the ASPC2 responds with 'No service activated (RS)'.

3.2.2 PROFIBUS DP Support

To better support the DP master, the ASPC2 is equipped with a few additional functions.

3.2.2.1 NOP Request

NOP requests are local services that don't appear on the SER BUS. The user utilizes these services to synchronize his SW in references to the status of the repeat list. The ASPC2 confirms the NOP request immediately with 'OK'. The repeat behavior is the same as that for the productive services. However, the field 'resp-status-exp(3..0) = OK' has to be set.

For the REQ APB, the NOP request is encoded in the field 'fc' on the 7th and 3rd bit position with a 'log. 1' (refer to Chapter 2.2.1). An example of a NOP request is the force pass token instruction which is encoded in the opcode. This REQ APB causes the MAC to pass the token immediately whithout transmitting this request on the SER BUS.

3.2.2.2 DP Mode Parameterization

Min Slave Interval, GAP Request
Instead of the Target Token Rotation Time, the Min Slave Interval is monitored in the DP mode. For this, the token rotation timer is used. After receiving the token message, the ASPC2 checks whether the Min Salve Interval has expired. It it hasn't, the token is passed. This continues until the token rotation timer has reached the parameterized Min Slave Interval (here: TTR: Target Rotation Time). Then, the Min Slave Interval is restarted by the token rotation timer being cleared. The maximum permissible value for the Min Slave Interval is ' TTR = FF7EH'.

Then, the ASPC2 first sends a GAP request. The normal GAP sequence is switched off in this case. This ensures that in the *check token pass phase* of the predecessor (monitoring of bus activity), no TSLOT is expiring. If, after that, there is an access to consistent data areas at the master, the timeout is always monitored at all other bus stations. The access time to consistent data blocks has to be shorter than the shortest timeout (6 x TSLOT). After the GAP message, the ASPC2 then processes the first request.

3.2.2.3 Consistency Support for the Master

Depending on the FIFO size (64/256 bytes, refer to Mode Register 2), the ASPC2 can support a Layer2 data consistency up to 58/250 bytes. For this, 'Blocked Mode = 1' is to be parameterized in Mode Register 2.

When sending (master), it fetches the net data with a lock cycle to the internal FIFO. For the shared memory mode, a lock cycle is a hold sequence, and in the dual port memory mode, an ASPC2 lock sequence. The send process starts only if the ASPC2 has received access to the Layer2 data (at least 1Byte/Word entered in FIFO).
At receipt, the ASPC2 holds the net data until the entire message was received faultlessly. Then, in a lock cycle, all data is transferred to the external memory. For larger data packages than the specified consistency length, it transfers the net data as soon as the FIFO threshold (quarter full) is exceeded.
The ASPC2 has no consistency support for Layer4 data!

So that the ASPC2 can transfer consistent data (250 bytes max.) directly to/from an external mapping memory, it makes two outputs ‘RDCONS/WRCONS’ available to support an external consistency control unit. These signals are activated by the ASPC2 when the Layer2 data is fetched from the memory (RDCONS) or written (WRCONS). So that the user can mix consistent and non-consistent APBs, the control signals in the FC field of the APB are activated:

REQ-APB fc:	FC7	don't care	WRCONS	RDCONS	FC3..0			
	7	6	5	4	3	2	1	0

IND-APB fc:	FC7	FC6	WRCONS	Res	FC3..0			
	7	6	5	4	3	2	1	0

RUP-APB fc:	don't care			RDCONS	don't care			
	7	6	5	4	3	2	1	0

RDCONS: If the bit =1, Pin DIA0 (Pin 83) is switched to log. ‘1’ during the transfer from the buffer to the ASPC2 FIFO of the request/response L2 data associated with this REQ/RUP APB. The signal is activated prior to the first address output to this area, and deactivated after the last access. If this sequence is interrupted in the dual memory mode by an external access XREQ, the signals remains activated!

WRCONS: If the bit =1, Pin DIA1 (Pin 82) is switched to log. ‘1’ during the transfer from the ASPC2 FIFO to the buffer of the request/response L2 data associated with this REQ/IND APB. The signal is activated prior to the first address output to this area, and deactivated after the last access. If this sequence is interrupted in the dual memory mode by an external access XREQ, the signals remains activated!

RES The res bit is to be set to 0.

So that ‘RDCONS’ and ‘WRCONS’ can be wired to the pins mentioned above, “DEBUG MODE = 00B or 10B” has to be parameterized (refer to Chapter 5.1). Consistency support is independent of Layer parameter assignment.

The timing RDCONS, WRCONS in reference to the subsequent bus access is described in Chapter 7.5.2.1).

4 Function of the Individual Blocks

4.1 Processor Bus Interface

4.1.1 Bus Interface that can be Set

The ASPC2 has a 8/16 bit interface that can be set. The setting is made via Pin XB8/B16 (XB8/B16 = 0 Chapter 8 Bit Bus Interface; XB8/B16 = 1 Chapter 16 Bit Bus Interface). The input is provided with an internal pull-up resistor. If unwired, a 16 bit interface is set.
Via Pin XINT/MOT, it can be operated in the Intel bus format as well as in the Motorola bus format. The input is provided with an internal pull-down resistor. If unwired, an Intel bus format is set. In the 8Bit Intel mode, only the lower data bus byte is connected (DB7..9). The higher data bus byte is wired permanently to Input, and is provided with internal pull-up resistors (refer to Figure 4.1.1-1). The pin XBHE is also on Input and does not have to be wired because it is terminated with an internal pull-up resistor.

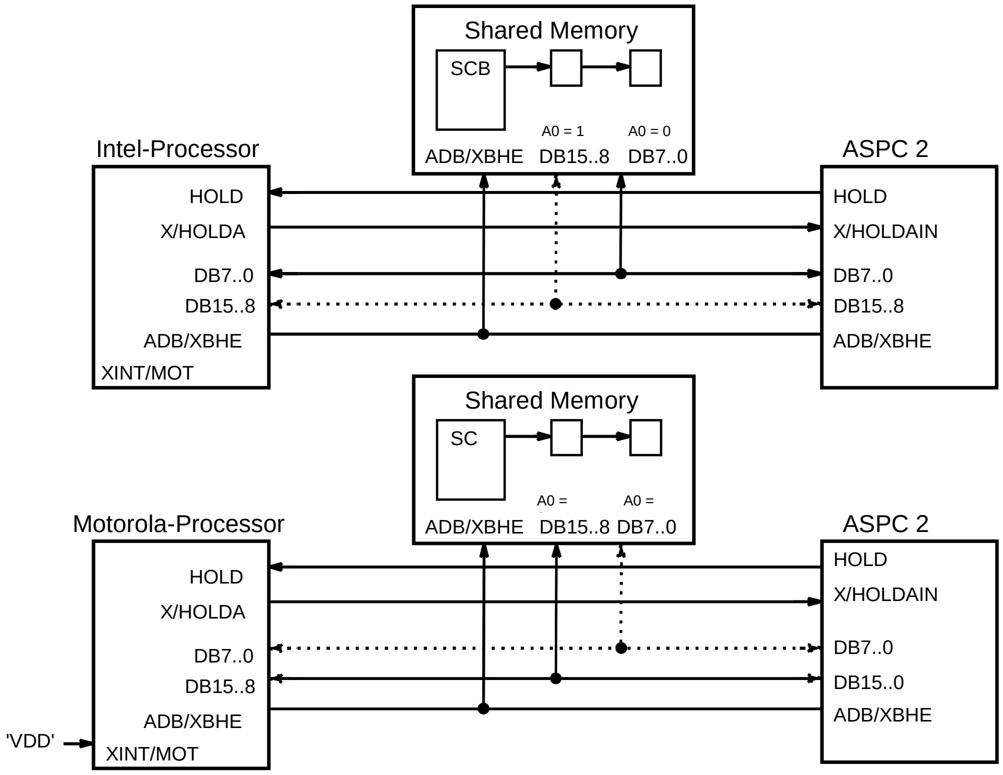


Figure 4.1.1-1: Connection Intel/Motorola Processor

When using Motorola processors, XINT/MOD has to be applied to ‘VDD’. The ASPC2 is to be connected in the 16Bit mode as in the Intel Mode (refer to Figure 4.1.1-1). The XBHE signal is to be generated externally from the Motorola control signals. The following convention is to be adhered to:
In the 8Bit Motorola mode, only the higher data bus byte is used (DB15..7). The lower data bus byte is permanently wired to Input, and provided with internal pull-up resistors. Pin XBHE is also on Input, and does not have to be wired.

Access Type for Motorola	XBHE	A0
higher Byte	1	0
lower Byte	0	1
Word	0	0

In the case of Motorola, all byte positions within a word are exchanged for processor accesses to the ASPC2 (refer to figure 4.1.1-2). The same applies to the structure of the SCB and the application blocks. However, it does not apply to the Layer4 data in the application block, and the remotely located response data blocks. The FLC sets up these data areas as byte arrays. When accessing these areas (DATACCESS=1), the ASPC2 exchanges the higher with the lower byte of an aligned data word. Figure 4.1.1-2 shows the access mechanisms.

Attention: The FLC has to store the long word address pointers (for example, resp-buf-ptr-lw/hw) word-exchanged in the application blocks.

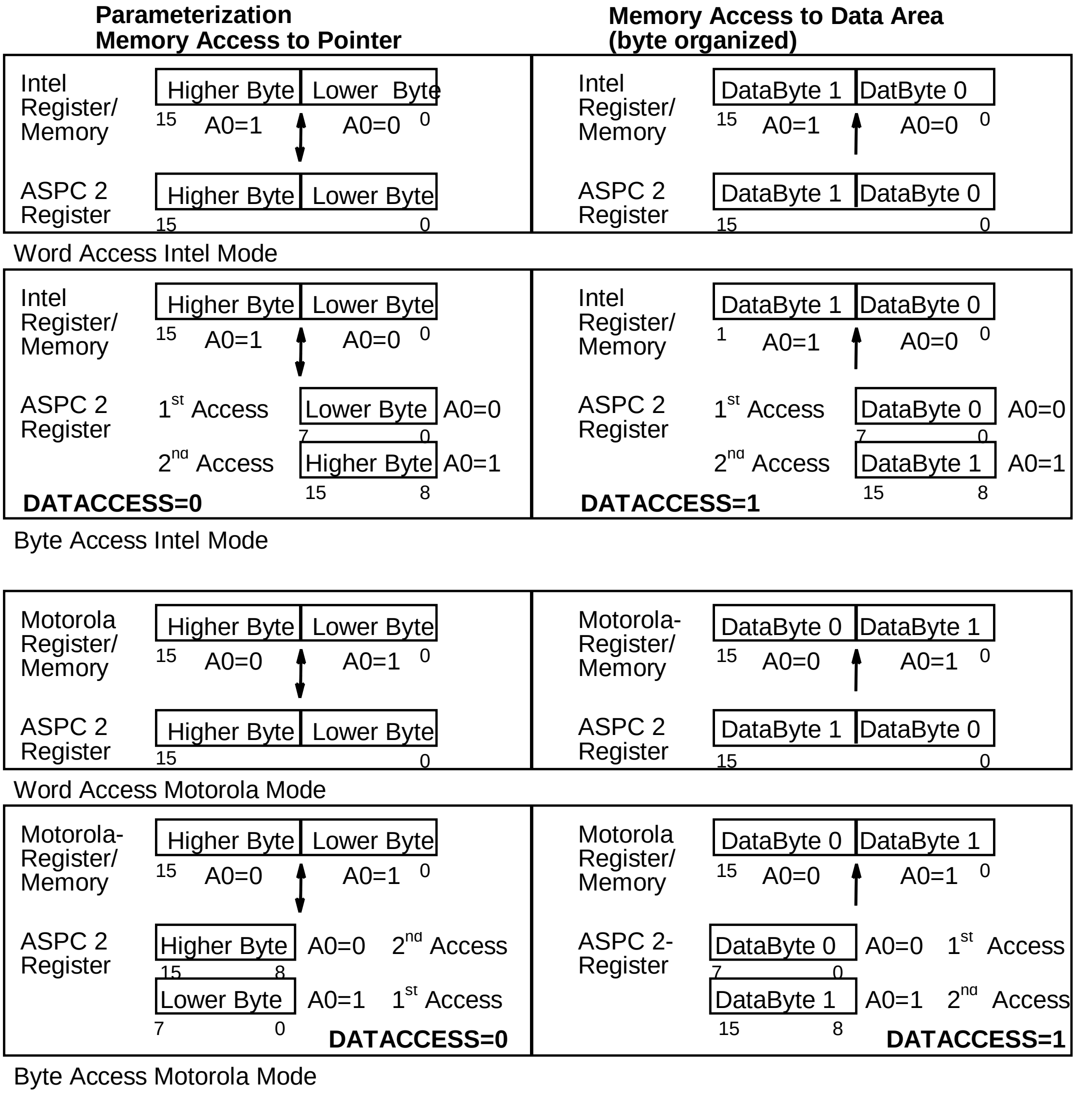


Figure 4.1.1-2: Access Mechanisms in the Intel/Motorola Mode

All ASPC2 accesses to the external memory are word accesses. In the 8Bit mode, these word accesses are dissolved into two successive byte accesses. The sequence of the byte accesses is designated with 1st and 2nd access in Figure 4.1.1-2.

4.1.1.1 XWRL/XWRH Mode

To save external hardware in the case of 80C185 applications, the two pins XWR (Pin 37) and XBHE (Pin 5) can be reparameterized to XWRL (Pin 37) and XWRH (Pin 5) (XWRL/XWRH mode). They are switched via Pin 33 (XWRL MODE). This input is provided with an integrated pull-up. If there is no wiring, the XBHE/XWR mode sets itself. This ensures compatibility with Revision A/B versions (here, the test output NTREE is on this pin).

XWRL_MODE = '0':XWRL/XWRH mode switched on

XWRL_MODE = '1':XBHE/XWR mode switched on

4.1.2 Master Mode

4.1.2.1 SYS Bus Access

The ASPC2 has a multiplexed bus interface with a parameterizable 8/16Bit data bus width, and can directly address a 1Mbyte memory area. However, the SCB and the application blocks have to be located within any 64k segment; the data blocks, on the other hand, may be distributed in the 1Mbyte area.
The ASPC2 can process with 20 bit linear as well as with 32 bit segment/offset address pointers (setting in Mode Register 0: XLIN/SEGOFF).
In the linear mode, the SCB base address consists of a 20 bit address that the user parameterizes for the ASPC2 in the SCB BASE LW/HW register (refer to Chapter 4.2.1). Internally, it processes with 16 bit offset addresses (for example, application block pointers). In the case of accesses to the SCB and the application blocks, the physical 20 bit addresses is calculated from the linear addition of the SCB base address and the offset address. In the case of accesses to the request and response data blocks, the base address is switched in the ASPC2 to the 20 bit buffer base address that was fetched from the application block. The memory access is peformed word by word to even memory addresses.
In the SEGOFF mode, the SCB base address is defined via a 16 bit segment address and a 16 bit offset address. All near pointers to the application blocks are 16 bit offset addresses in reference to the 16 bit SCB segment address; that is, the SCB and all application blocks have to be located within the same segment. For addressing, the offset pointer is loaded to a 16bit Alu register. Adding up the additional offset (for example, within the SCB or the application block) is not to cause an overflow. The data buffers are addressed via 32bit far pointers (segment/offset). The memory access is carried out word by word to even memory addresses.

The bus interface unit (BIU) processes with half the system clock pulse rate (24 MHz in the case of 48MHz system clock pulse), either synchronously -with the possibility to insert 1 to 5 wait states via an internal wait state generator- or asynchronously, through the additional synchronization of an external Ready signal (XRDY) (refer to Chapter 4.2.1: Mode Register 0). The wait states can be set separately for each 256kByte memory segment (refer to Chapter 4.2.1). The XRDY delay is not to reach just any size; otherwise, there may be a FIFO

overrun/underrun. **In the case of a transmission rate of 12 MBd, the ASPC2 has to get an access to the external memory on the average every 1.8 µs when fetching the Layer2/4 data**. With 1 wait state being parameterized, the ASPC2/Revision C/D needs external RAM with an **access time of 70ns**. The write/read cycle time for a data word (Layer 2 and 4) is at 167ns for this setting. The exact timing diagrams are shown in Chapter 7.5.2.1).

4.1.2.2 Quick Access Mode

In Revision C/D, a faster bus mode is implemented in addition, the ‘Quick Access Mode’ (refer to Mode Register 2). In this mode, all external write and read operations (organizational data, message header data, Layer2 to Layer 4 data)are accelerated by a 24 MHz clock pulse. However, this is possible only by compressing the bus timing. In the case of read operations, the DT/XR signal gets the phase position as the XRD signal. This results in a bus cycle time at data transfer (Layer 2 and 4) of 125ns (1 WS parameterized). With this setting, the ASPCs needs external RAMs with an **access time of 48ns**. However, in the case of this access time, the external RAMs have to be always selected, otherwise, even faster RAMs have to be used. For each additional wait state, this time increases by 41.6ns. In Chapter 7.5.2.1, the corresponding timing diagrams are shown.

4.1.2.3 Shared Memory

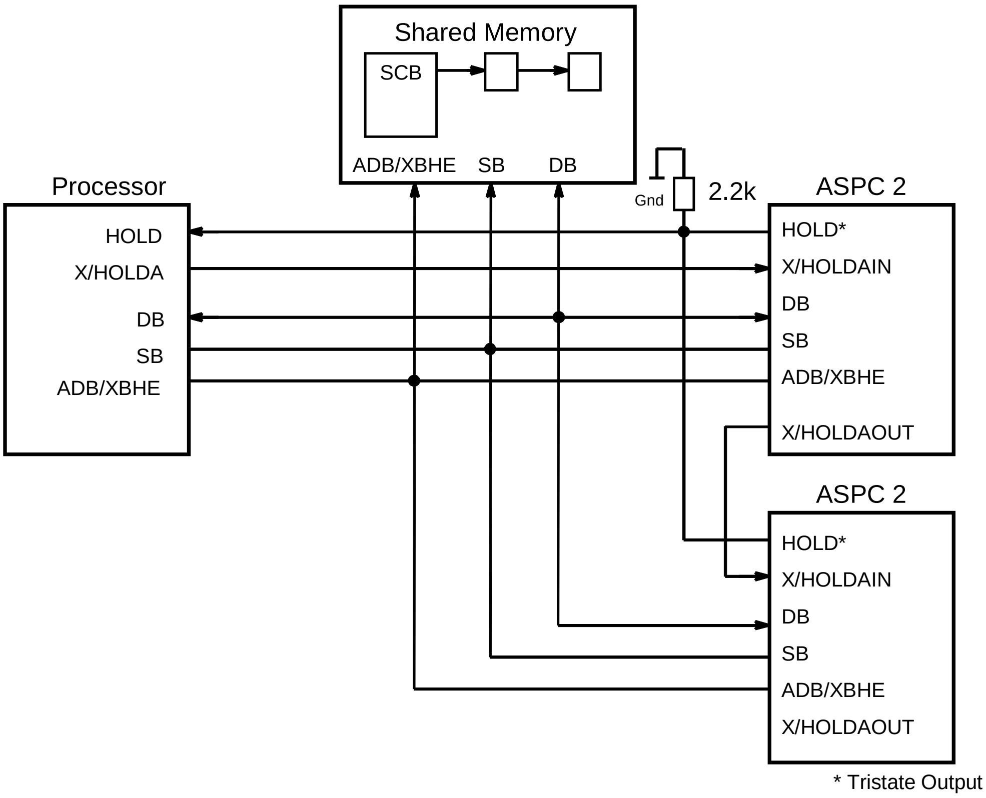


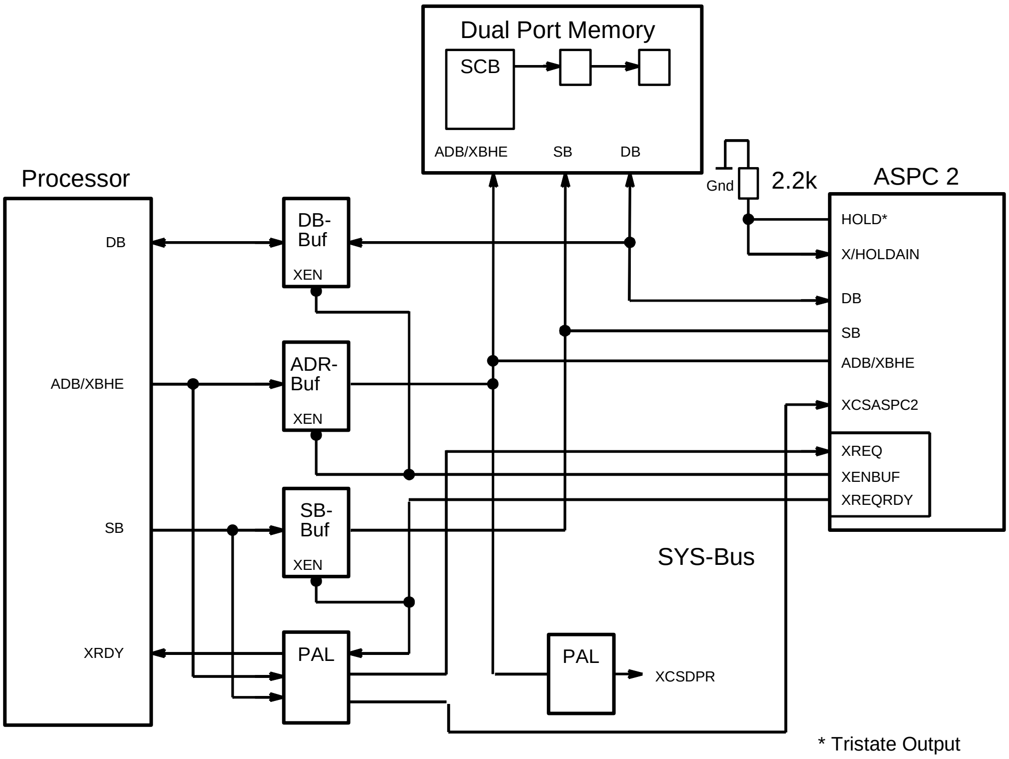
Figure 4.1.2.3-1: Master Mode, Shared Memory

By parameterizing “XSHMEM/DPMEM = 0” (Mode Register 0), communication between the FLC (processor) and the ASPC2 is carried out via a shared memory where the SCB and all lists are stored (refer to Figure 4.1.2.3-1). The ASPC2 accesses this memory in the master mode. For this, the SYS bus has to be obtained. For this handshake, the signals HOLD-X/HOLDA are used, whereby the polarity of X/HOLDA can be parameterized in Mode Register 0. After the HW reset, X/HOLDA is high active. With “HOLD=active”, the ASPC2 requests the SYS bus and waits until the processor assigns it with “X/HOLDA=active”. Then, the ASPC2 switches its SYS bus drivers from Tristate to Active, and starts with the memory accesses. **This hold phase can’t be interrupted from the outside**. After having finished, the ASPC2 switches the drivers back to Tristate, and sets “HOLD=inactive”. The processor then seizes the initiative again and acknowledges with “X/HOLDA=inactive (refer to Chapter 7.5.2.1).
Several ASPC2 can be wired to a SYS bus via a daisy chain. As an additonal output, the signal X/HOLDAOUT is available. With it, the series-connected ASPC2 is granted bus access if the processor reads out X/HOLDA, and the preceding ASPC2s have not requested the bus. The priority in the chain is fixed; the first has the highest, and the last the lowest priority. All HOLD outputs are to be connected together via a wired Or-line, and are to be terminated with a pull- down resistor.
When the application blocks are chained/unchained in/from the lists, the processor has to prevent the ASPC2 from accessing the lists. For this, “BUS LOCK=1” is to be set in Mode Register 1. During this time, the bus does not attempt to obtain the SYS bus. However, the BUS LOCK time is not to become randomly long, otherwise there may be a FIFO overrun/underrun. In Chapter 7, different timing for the load on the processor interface is specified.

4.1.2.4 Dual Port Memory

In the case of HW environments whose processors have no hold function, or applications that don't want to halt the processor, this shared memory has to be laid out as a dual port memory. For this, "XSHMEM/DPMEM = 1"is to be parameterized. For support, the ASPC2 provides an integrated bus arbiter. Via this bus arbiter, an external bus master (processor) receives the access to the SYS bus (refer to Figure 4.1.2.4.1). For each processor access to the ASPC2 or the dual port memory, an external bus request has to be made to the ASPC2. For this, the XREQ pin has to be switched active low. At bus allocation, the ASPC2 first switches the output XENBUF active, via which the external data- and address bus drivers are switched through to the SYS bus, and then the output XREQRDY, via which the external signal bus driver (read or write signal) is enabled, and the XRDY signal is passed to the processor.The exact timing of bus allocation is shown in Chapter 7.5.2.1 (timing diagrams). The time between XENBUF active and XREQRDY active can be parameterized either to 1 clock cycle (T48) or 3 clock cycles (T48) via "QREQRDY" (TQUI register). **Cascading several ASPC2 blocks is not possible in this mode!**

In the shared memory mode, the processor (XREQ) can't interrupt the hold sequence. In the dual port memory mode, this is possible. The ASPC2 interrupts its accesses to the dual port memory under the following conditions:



4.1.2.4-2: Master Mode Dual Port Memory

a) An access that was just started is concluded; that is, in the 8bit mode, both bytes are transferred completely to/from the memory before giving up the bus.

b) The ASPS2 carries out critical sequences (unchaining, chaining, request- and SAP list selection up to locking the applicaton block 'MAC Locked = 1', SAP validation, etc.) under lock (ASPC Lock = 1). The FLC sets 'BUS LOCK = 1' exactly like that if it performs rechaining processes, changes in the SAP, etc.. As long as only one side is in a lock sequence, the other side can interrupt any time. If both sides are in the lock mode, there is no interruption until one cancels the lock. The one who first locks the bus will succeed. If they arrive at the same time, the external access is prioritized.

c) The FLC sets an external lock with ‘BUS LOCK = 1’. However, external lock sequences contain only dual port RAM accesses (XCSASPC2 = 1). Thus, during an external lock phase, accesses to internal ASPC2 registers are possible, even if the ASPC2 itself is performing an internal lock sequence. These accesses are identified with ‘XCSASPC2 = 0’ while ‘BUS LOCK = 1’ is set. The timing of XCSASPC2 in reference to XREQ is shown in Chapter 7.5.2.2 (1 T48 max. after XREQ). The longest lock sequence of ASPC2 amounts to 11.3 µsec (16bit mode). If, at the start of this sequence, an external ‘BUS LOCK’ occurs, there may be a Ready delay of 15 µsec maximum at the first extenal dual port RAM access (refer to Chapter 7.5.2.1). In the case of the PC buses (ISA, EISA), however, a Ready delay of only 2.5 µsec maximum is permitted. So that this requirement can be met, the FLC has to poll the cell ‘ASPC Lock’ in the status register after it has set “BUS LOCK = 1”, and wait until “ASPC Lock = 0” is set (Chapter 4.2.1). At this time, the arbiter allocated the external lock request, and can no longer be interrupted by an internal lock request. For a 16bit system, the Ready delay time is now at 350ns maximum.

d) If a separate XREQ request should be made for each FLC access to the dual port RAM, FLC- and ASPC2 accesses to the dual port RAM can be nested as required (except for lock sequences). However, XREQ may stay active also over several FLC accesses, but the ASPC2 will be retarded in that case. The XREQ active time may not be of random length; otherwise, there may be a FIFO overrun/underun.
At a baudrate of 12Mbaud, the ASPC2 has to receive an access to the dual port RAM every 1.8 µsec on the average when fetching the Layer 2/4 data. In Chapter 6, different timing for the load on the processor interface is specified.

4.1.3 Peripheral Mode

For parameterization and interrupt event handling, the processor has to address the ASPC2 in the peripheral mode (refer to Figure 4.1.3-1). Via a small address window (64 bytes), different internal registers can be written to or read (for address assignment, refer to Table 1.9-1). The access can be made word by word or byte by byte (Intel/Motorola format; controlled via the bus signals XBHE and AB0). The timing diagrams are shown in Chapter 7.5.2.2.

Figure 4.1.2.3-1 shows the principle of the processor access to the ASPC2 in the shared memory mode. In dual port memory operation, the processor first has to make an XREQ request at the access.

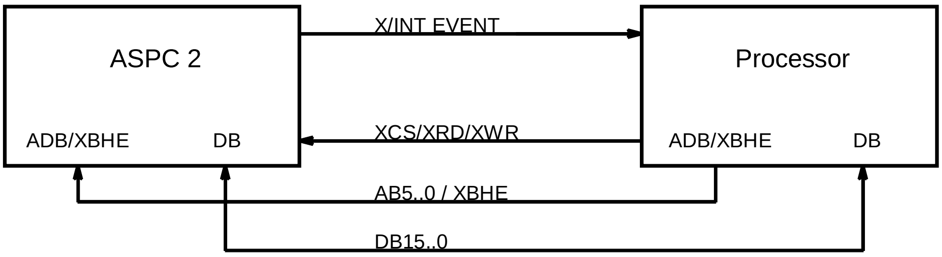


Figure 4.1.3-1: Peripheral Mode

4.1.4 SYS Bus Interface

Signal Names	Input = I
	Output = O
		Type	Comment

DB15..0	I/O	Tristate	High resistance at reset
AB19..6	O	Tristate	Master Mode
AB5..0	I/O	Tristate	Master/Peripheral Mode
XBHE/XWRH	I/O	Tristate	Byte high enable
XWR/XWRL	I/O	Tristate	
XRD	I/O	Tristate	
XCS	I		Base address, peripheral mode
DT/XR	O	Tristate	Data Transmit/ Read
X/INT-CI	O	non Tristate	Polarity parameterizable
X/INT-EVENT	O	non Tristate	Polarity parameterizable
HOLD	O	Tristate	
X/HOLDAIN	I		Polarity parameterizable
X/HOLDAOUT	O	non Tristate	Polarity parameterizable
XREQ	I		Ext. bus request
XENBUF	O	non Tristate	Enable ext. driver
XREQRDY	O	non Tristate	Ready for ext. master
XRDY	I		Asynchron.
CLK	I		48 MHz
RESET	I		Min. 4 clock cyc.
XB8/B16	I		Bus configuration
XINT/MOT	I		Bus configuration
XWRL-MODE	I		Bus configuration

Table 4.1.4-1: SYS Bus Signals of ASPC2

The reset is connected asynchronously and switched off synchronously. Thus, the RTS output will become inactive immediately at reset. During the reset phase (at least 4 T48 long), the data bus outputs are high resistance.

4.2 Parameter Register File

4.2.1 Operating Modes

In the parameter register file, the FLC transfers operating data to the ASPC2. Parameterization is performed only in the *offline state* (for example, after switch-on). Only after all parameters are loaded is the ASPC2 to leave *offline* (START ASPC2 = 1, Mode Register 1). Some control bits (for example, BUS LOCK) have to be continuously changed during operation, however. These are combined in a special register (Mode Register 1), and can be set or cleared independent of each other. **After reset, all registers in the parameter register file are cleared (except for IMR, reserved register, and status register).**

Token Target Rotation Timer Register (writable);

T TR7	T TR6	T TR5	T TR4	T TR3	T TR2	T TR1	T TR0
7							0

T TR15	T TR14	T TR13	T TR12	T TR11	T TR10	T TR9	T TR8
15							8

The token setpoint rotation time (TTR, 16 bit) is to be parameterized. One unit corresponds to 256 * tBit. (The bit time tBit is the time that passes when sending a bit. It correponds to the reciprocal value of the baudrate.)

Slot Time Register (writable):

T SL7	T SL6	T SL5	T SL4	T SL3	T SL2	T SL1	T SL0
7							0

		T SL13	T SL12	T SL11	T SL10	T SL9	T SL8
		13					8

TSL = max. (TSL1, TSL2) is to be parameterized. The wait-for-receive time TSL has a width of. 14 bits maximum, and is to be specified in transmission bit steps. The lowest permissible value is 37.

TIDLE 1 Register (writable):

TID1 7	TID1 6	TID1 5	TID1 4	TID1 3	TID1 2	TID1 1	TID1 0
7							0

						TID19	TID18
						9	8

TID1 becomes effective after an acknowledgement-, response-, or token message. The time has a width of 10 bits maximum, and is to be specified in transmission bit steps. The lowest permissible value is 35.

TIDLE 2 Register (writable):

TID2$_7$	TID2$_6$	TID2$_5$	TID2$_4$	TID2$_3$	TID2$_2$	TID2$_1$	TID2$_0$
7							0

TID2$_9$	TID2$_8$
9	8

TID2 becomes effective after a call message without acknowledgement (SDN). The time has a width of 10 bits maximum, and is to be specified in transmission bit steps. The lowest permissible value is 35.

TRDY Register (writable):

TRDY$_7$	TRDY$_6$	TRDY$_5$	TRDY$_4$	TRDY$_3$	TRDY$_2$	TRDY$_1$	TRDY$_0$
7							0

TRDY$_9$	TRDY$_8$
9	8

The time TRDY has to pass as idle time on the bus prior to sending a response message The time has a width of 10 bits maximum, and is to be specified in transmission bit steps. The lowest permissible value is 11. If the values are lower, the ASPC 2 takes 11 bit steps.

TS-ADR Register (writable):

TS $_6$	TS $_5$	TS $_4$	TS $_3$	TS $_2$	TS $_1$	TS $_0$
6						0

The station address may be between 0 and 126. Address 127 is reserved as Broadcast/Multicast.

TGUD Register (writable):

TGUD$_7$	TGUD$_6$	TGUD$_5$	TGUD$_4$	TGUD$_3$	TGUD$_2$	TGUD$_1$	TGUD$_0$
7							0

The GAP update time TGUD determines the cycle for GAP updating. It has a width of 8 bits, and is to be specified in the number of token rotations.

Token Error Limit Register (writable):

TEL $_7$	TEL $_6$	TEL $_5$	TEL $_4$	TEL $_3$	TEL $_2$	TEL$_1$	TEL$_0$
7							0

Token Error Limit (8 bit) specifies the permissible number of unvalidated token messages per 256 token rotations. When it is reached, the MAC enters *Listen Token*.

Baudrate Register (writable):

BR 7	BR 6	BR 5	BR 4	BR 3	BR 2	BR 1	BR 0
7							0

BR 10	BR 9	BR 8
10		8

In the baudrate register, the division factor for the baudrate generator is parameterized. The devision factor G is calculated according to the following formula:

$$G = \frac{CLK}{BR \times 4} - 1 \qquad \text{without RxD filter}$$

$$G = \frac{CLK}{BR \times 16} - 1 \qquad \text{with RxD filter}$$

C

LK = Clock Supply (48 MHz) BR = Baudrate G = Division Factor

The table below shows the division factors for the individual baudrates. Without RXD filter, baudrates of 9.6kBd to 12MBd, and with RxD filter of 9.6kBd to 3MBd can be set. The clock frequency is to be selected such that the residual error is at < 0.3%. The maximum clock frequency is 48 MHz.

Clock	Baudrate (BR)	Div. Factor (G) (w/out RxD Filter)	Div. Factor (G) (with RxD Filter)
48 MHz	12 MBd	0	-
	6 MBd	1	-
	4 MBd	2	-
	3 MBd	3	0
	1,5 MBd	7	1
	750 kBd	15	3
	500 kBd	23	5
	187.5 kBd	63	15
	93.75 kBd	127	31
	45.45 kBd	263	65
	19.2 kBd	624	155
	9.6 kBd	1249	311

Table 4.2.1-1: Parameterizing the Baudrate

SAP-MAX Register (writable):

SAP-M 7	SAP-M 6	SAP-M 5	SAP-M 4	SAP-M 3	SAP-M 2	SAP-M1	SAP-M0
7							0

SAP MAX specifies the highest SAP list number. It may be between 0 and 63 liegen (refer to Chapter 2.1)

LAY4-HLen Register (writable):

L4-Hl2 3	L4-Hl2 2	L4-Hl2 1	L4-Hl2 0	L4-Hl1 3	L4-Hl1 2	L4-Hl1 1	L4-Hl1 0
7							0

In LAY4 Hlen, two different Layer4 header lengths are specified (L4-Hl13..0, L4-Hl23..0). The parameterized values specify -depending on the opcode field 'LAY4-Head-Len' in the respective application block- the Layer4 header in the number of words that is appended to the request header in the application block. The ASPC 2 assigns 16 words (32 bytes) to the value ‘0’ (refer to Chapter 2.2.1).

TQUI Register (writable):

QREQRDY	TQUI 6	TQUI 5	TQUI 4	TQUI 3	TQUI 2	TQUI 1	TQUI 0
7							0

The modulator fading time (also in the case of self-controlled repeaters) has a width of 7 bits maximum, and is parameterized in transmission bit steps.

QREQRDY: With QREQRDY, the delay time between 'XENBUF active' and 'XREQRDY active' is set in the dual port RAM mode (refer to Chapter 7.5.2.1).

0 = The delay time is 3 clock cycles (T48).

1 = The delay time is 1 clock cycle (T48).

HSA Register (writable):

HSA 6	HSA 5	HSA 4	HSA 3	HSA 2	HSA 1	HSA 0
6						0

The highest active address specifies the highest address of an active station. It can be set optionally from 1 to 126 D at PROFIBUS, and from 1 to 119 D at PA/ISP. At PROFIBUS, the GAP cycle is limited to this HSA. In the case of PA/ISP, the temporary stations (120 - 124 D) are located in the GAP area in addition.

Response Error Limit Register (writable):

Resp-Error-Limit 3	Resp-Error-Limit 2	Resp-Error-Limit 1	Resp-Error-Limit 0
3			0

In the response error limit register, the number of faulty response (1 to 15) can be parameterized on 16 successive request messages (except for SDN messages) after the ASPC2 signals a double token error and enters *Not Hold Token*.

Reserved Register (writable);

RES1 7 RES2 7	RES1 6 RES2 6	RES1 5 RES2 5	RES1 4 RES2 4	RES1 3 RES2 3	RES1 2 RES2 2	RES1 1 RES2 1	RES1 0 RES2 0
7							0

After reset, 'Reserved2' is cleared, and 'Reserved1=01H' is set.

RTS Setup (Reserved Register1, writable)

Die setup time of the RTS signal for the TxD signal can be parameterized. In the register 'Reserved1' , the parameter 'RTS_Pre_Value' is transferred. After reset, the value is '01H'. The setup time is calculated as follows:

RTS_Setup = (5 + (2 ✸ RTS_Pre_Value)) ✸T48

Retry Register (writable):

Retry-T 3	Retry-T 2	Retry-T 1	Retry-T 0	Retry-M 3	Retry-M 2	Retry-M1	Retry-M 0
7							0

In the RETRY register, the number of message repetitions (RETRY-M3..0) and the number of token repetitions (Retry-T3..0) is specified. A maximum of 15 repetitions can be parameterized.

SCB BASE LW/HW Register (writable)

Linear Address Pointers:

SCBL 7	SCBL 6	SCBL5	SCBL4	SCBL3 SCBH19	SCBL 2 SCBH18	SCBL1 SCBH17	SCBL 0 SCBH16
7							0

SCBL 15	SCBL 14	SCBL13	SCBL12	SCBL11	SCBL10	SCBL9	SCBL8
15							8

The SCB BASE address (20 Bit) specifies the basic address of the system control block. All 16 bit application block pointers represent an offset to the SCB-BASE address.

Segment/Offset Address Pointer:

SCBOFF7 SCBSEG23	SCBOFF6 SCBSEG22	SCBOFF5 SCBSEG21	SCBOFF4 SCBSEG20	SCBOFF3 SCBSEG19	SCBOFF2 SCBSEG18	SCBOFF1 SCBSEG17	SCBOFF0 SCBSEG16
23/7							0

SCBOFF15 SCBSEG31	SCBOFF14 SCBSEG30	SCBOFF13 SCBSEG29	SCBOFF12 SCBSEG28	SCBOFF11 SCBSEG27	SCBOFF10 SCBSEG26	SCBOFF9 SCBSEG25	SCBOFF8 SCBSEG24
31/15							24/8

The SCB-SEG/OFF address (32bits) specifies the base address of the system control block. All 16 bit application block pointers represent an offset to the SCB-SEG address.

4.2.1.1 Mode Register 0

HOLDA-POL	INT-POL	SEP-INT	SEP-LIST	RES	PAS/ACT-Tn	RES	RES
7							0

RxD-Filter	XSHMEM /DPMEM	DP-MODE	RES	XLIN/ SEGOFF	DEBUG-MODE		RES
					1	0	
15							8

In Mode Register 0, the fixed parameters are transferred that have to be loaded only once, after reset. The individual bits have the following meaning:

RES: Here, always 0 has to be entered.

PAS/ACT Sta: Station type
0 = The ASPC2 is a passive station.
1 = The ASPC2 is an active station.

SEP-LIST: Confirmation-Indication lists (refer to Chapter 2.1)
0 = Joint confirmation-indication list.
1 = Separate confirmation-indication lists: IND-SEP-LIST, CON-SEP-LIST, NOT-OK-LIST. In the case of collect requests, this bit is irrelevant.

SEP-INT: Number of interrupt outputs (refer to Chapter 4.3).
0 = All interrupts are on the INT-EV pin.
1 = Two interrupt outputs are available:
INT-CI: Confirmation/Indication (SEP-LIST=0), Indication (SEP-LIST=1)
INT-EV: the remaining interrupts

INT-POL: Polarity of the interrupt outputs (refer to Chapter 4.3)
0 = The interrupt outputs are low-active
1 = The interrupt outputs are high active

HOLDA-POL: Polarity of the X/HOLDAIN-X/HOLDAOUT signal (refer to 4.1.2)
0 = The X/HOLDA signals are high active
1 = The X/HOLDA signals are low-active

DEBUG-MODE1..0: Selection of debug function (refer to Chapter 5.1)
00 = WRCONS and RDCONS are on the diagnostic port (DIA1..0)
10 = WRCONS and RDCONS are on the diagnostic port (DIA1..0)

XLIN/SEGOFF: Setting of the addressing mode(refer to Chapter 4.1.2)
0 = Linear address pointer
1 = Segment/Offset address pointer

DP-MODE: DP-MODE setting (refer to Chapter 3.2.2)
0 = No DP MODE
1 = DP MODE

XSHMEM/DPMEM: Setting of the external communication memory (refer to 4.1.2)

0 = Communication memory as shared memory between ASPC2 and FLC (processor). An external interruption in the Hold mode (ASPC2 active) is not possible. Several ASPC2 can be connected in daisy chain.

1 = Communication memory as dual port memory between ASPC2 and FLC (processor). External interruption in the Hold mode is possible (ASPC2 active). A daisy chain can be set up.

RxD Filter: Enable RxD receive filter

0 = No RxD receive filter connected

1 = RxD receive filter connected

4.2.1.2 Mode Register 1

GO-OFFLINE	EN-CLR-CHG	CLEAR-GAP-STORED	REQ-EMPTY	EOI	READ-LAS	BUS-LOCK	START-ASPC 2
7							0

RES	RES	RES	RES	Del-Tim-Cntr (1)	Del-Tim-Cntr (0)	RES	LOCK-REQUEST
15							8

In Mode Register 1, the processor sets various control bits for the ASPC2. These control bits change during operation, and can be set or cleared independent of each other. Different addresses are used for setting and clearing. Log. ‘1‘ is to be written to the bit position that is to be set or cleared. The individual bits have the following meaning:

RES: Here, 0 is always to be entered.

START-ASPC 2: Exits the *Offline mode*

0 = Not possible by the processor, only internally.

1 = The ASPC2 branches *Offline* (in addition, the idle and Syni timer is started. If 'GO-OFFLINE = 1' is set, this bit is reset) .

BUS-LOCK: Locks the bus accesses (refer to Chapter 4.1.2)

0 = No BUS-LOCK is set.

1 = BUS-LOCK is set: In the shared memory mode, the HOLD signal of the ASPC2 can’t be activated. In the dual port memory mode, the ASPC2 can’t access under ASPC lock.

READ-LAS: Enable for reading the LAS-RAM (refer to Chapter 4.7)

0 = The LAS access is disabled.

0->1= The LAS pointer is reset, and reading the LAS is enabled.

1 = After each reading of the LAS register, the internal LAS pointer is incremented.

EOI: End of Interrupt (refer to Chaper 4.3)

0 = No End of Interrupt

1 = End of Interrupt; the ASPC2 switches the interrupt outputs inactive, and resets EOI to log.'0'.

REQ-EMPTY: Request lists empty (refer to Chapter 2.1)

0 = Always, if the FLC adds an application block to the request lists, if it changes from the OM to the BM mode or vice versa, or after restart after the watchdog timer has expired.

1 = Not possible (only from ASPC2 if the request lists are empty)

CLEAR-GAP-STORED: Reset of a temporarily stored GAP request

0 = Not possible

1 = A request that was temporarily stored internally is cleared.

EN-CLR-CHG: Enable Clear Changed Bit (refer to Chapter 4.7)

0 = When reading out the LAS, the Changed bit is not reset.

1 = When reading out the LAS, the Changed bit is reset (can be set simultaneously with 'Read-LAS').

GO-OFFLINE: MAC-Reset

0 = Not possible by the processor, only internally.

1 = MAC-Reset; after completed confirmation or indication, the ASPC2 enters *Offline* and sets "START-ASPC 2 = 0". In addition, it generates the event interrupt 'MAC-RESET'.

LOCK-REQUEST: Locks the BML, BMH request lists

0 = The above request lists are enabled.

1 = The above request lists are disabled.

Del-Tim-Cntr1..0: Control of the Delay/User timer (parameter assignment via Mode REG1-Set)

00 = Not possible

01 = The delay timer is reset and started.

10 = The delay timer is stopped.

11 = The delay timer is reset.

4.2.1.3 *Mode Register 2*

Mode Register 2 is an expansion of Mode Register 0; that is, all parameters have to be transferred in the offline mode, and are not to be changed during operation. After reset, Mode Register 2 is cleared, and Revision C/D behaves like Revision A/B.

☒	☒	FIFO-128Byte-Mode	Blocked-Mode	Res	User-Timebase	EOI-Timebase	Quick-Access-Mode
7							0

Quick Access Mode: Fast bus access mode (refer to Chapter 4.1.2)

0 = The fast bus access mode is off.

1 = The fast bus access mode is on.

EOI-Timebase: Time base for the EOI pulse (refer to Chapter 4.3)

0 = The interrupt inactive time is 1 μs long.

1 = The interrupt inactive time is 1 ms long.

User Timebase: 2.1 sec / 10 ms User Timer (refer to Chapter 4.5)

0 = Clock approx. 2.1 sec.

1 = Clock approx. 10 ms.

Res: Always, 0 is to be entered.

Blocked Mode: Transfer from FIFO to external RAM (refer to Chapter 3.2.2).

0 = The received net data (Layer 2 and 4) is transferred immediately when the FIFO threshold is reached.

1 = At a data length of less than 58/250 bytes, the received net data (Layer 2 and 4) is transferred only if it has been received completely and faultlessly (including FCS, ED). This parameter assignment is necessary if consistent data is to be transferred.

FIFO-256Byte Mode: Parameterization of FIFO size

0 = The FIFO consists of 64 bytes.

1 = The FIFO consists of 256 bytes.

Wait State Register (writable):

256...512K				0...256K			
7	6	5	4	3	2	1	0

768...1024K				512...768K			
15	14	13	12	11	10	9	8

In the wait state register, the number of wait states for the master is parameterized. The wait states for the four 256 kByte segments can be set independently. For each segment, 1 to 5 wait states can be parameterized in the mode 'external ready not activated', and 0 or 1 wait state in the mode 'external ready activated' after the arrival of external ready. In addition, 1 to 4 internal wait states can be included. For this, the internal wait states as well as external ready have to have expired.

Bit1..0: Wait states in general (at ready activation before Ready) (refer to Chapter 7.5.2.1).

00 = 1 Wait State (no additional wait state possible)

01 = 2 Wait States

10 = 3 Wait States

11 = 4 Wait States

Bit2: Wait state that can be set in addition (at ready activation after Ready).

0 = 0 Wait State

1 = 1 Wait State

Bit3: Ready Activation

0 = OFF

1 = ON

Bit15..4: Have the same meaning for the other address areas.

4.2.1.4 Status Register

The status register mirrors the current MAC status, and can be read only.

Lock-Request	BUS-LOCK	REQ-EMPTY	Hold-Token	Not-Hold-Token	Listen-Token	RES	Offline
7							0

FIFO-Error	Rec-Frame-Overflow	CONF/IND Stored	GAP-Stored	ASPC-Lock	RES	Receive	Enable-Receiver
15							8

Station-Typ			Idle-Mux 1..0		Last-High	SYNI-XSLOT	Inv-Req-Len
2	1	0	1	0			
23							16

Release-Number							
7	6	5	4	3	2	1	0
31							24

RES: don't care.

Offline: *Offline Mode*
0 = The ASPC2 is not in *Offline.*
1 = The ASPC2 is in *Offline.*

Listen Token: *Listen Token Mode*
0 = The ASPC2 is not in *Listen Token.*
1 = The ASPC2 is in *Listen Token.*

Not-Hold-Token: *Not-Hold-Token Mode*
0 = The ASPC2 is another mode.
1 = The ASPC2 is in *Not-Hold-Token.*

Hold-Token: *Hold-Token-Zustand*
0 = The ASPC2 is in another mode.
1 = The ASPC2 is in *Hold Token.*

REQ-EMPTY: Request Lists empty (refer to Chapter 2.1)
0 = The request lists are not empty, or watchdog has not expired.
1 = The request lists are empty, or watchdog has expired.

BUS-LOCK: Lock of bus accesses (refer to Chapter 4.1.2)
0 = No BUS-LOCK is set.
1 = The processor has set a BUS-LOCK.

Lock Request: Lock BMH, BML (refer to Chapter 2.1)

0 = The request lists above are enabled.

1 = The request lists above are disabled. However, an APB may still be processed. Here, a corresponding acknowledgement has to be awaited (for example, Go-Hold/Not-Hold-Token interrupt).

Enable Receiver: Enable receiver

0 = The receiver is disabled.

1 = The receiver is enabled.

Receive: Transfer direction of the FIFO (refer to Chapter 4.6)

0 = The FIFO is on send direction.

1 = The FIFO is on receive direction. If "Enable Receiver = 1", a message is being received at the moment.

ASPC Lock: Status LOCK

0 = In the dual port memory mode, the FLC has set its Lock first if there is or is not a LOCK conflict. In the shared memory mode, ASPC lock is always '0'.

1 = The ASPC2 has set its lock first if there is a LOCK conflict.

GAP Stored: GAP Request stored

0 = No GAP request is temporarily stored.

1 = The GAP timer has expired, and the ASPC2 has temporarily stored a GAP request.

CONF/IND Stored: Confirmation or Indication temporarily stored

0 = No confirmation or indication temporarily stored.

1 = A confirmation or indication temporarily stored.

REC Frame Overflow: FIFO assigned (BUSLOCK time too long, or sender too fast (TID1))

0 = Reset with IAR(1) = 0 (refer to interrupt register).

1 = The ASPC2 again received a new message although the previous message has not been transferred completely to the external memory.

FIFO Error: FIFO overflow or FIFO empty (BUSLOCK time too long)

0 = Reset with IAR(1) = 0 (refer to interrupt register).

1 = Send: The FIFO has an underrun. The bus is/was locked too long (refer to Chapter 4.1.2).

Receive: The FIFO has an ovverun. The bus is/was locked too long (refer to Chapter 4.1.2).

Inv-Req-Len: Invalid Request Length-Bit

0 = Reset with IAR(1) = 0 (refer to interrupt register).

1 = In the request, an invalid length has been specified. The addition of the message header and the Layer4 data length resulted in a value of >=256. The request was sent without Layer4 data and with net data length = 0.

SYNIXSLOT: Status of theSyni/Slot Timer

0 = The timer is running as a Slot Timer.

1 = The timer is running as a Syni Timer.

Last High: Request-Priority d2f

0 = The application block processed at this time is low priority.

1 = The application block processed at this time is high priority.

Idle Mux1..0: Status Idle Multiplexer

00 = The Idle Mux is on TSYN.

01 = The Idle Mux is on TRDY.

10 = The Idle Mux is on TID1.

11 = The Idle Mux is onTID2.

Stn-Typ2..0: Station Type

000 = Passive station

010 = Active station ready for the log. ring

011 = Active station

Release Number: Release of the ASPC 2

00 = ASPC 2 / Revision A/B

01 = ASPC 2 / Revision C

02 = ASPC 2 / Revision D

4.3 Interrupt Controller

Via the interrupt controller, the processor is informed of various events. These are primarily confirmation/indication messages, and different error events. Overall, up to 16 events are stored in the interrupt controller that are applied to one or two interrupt outputs. The controller has no prioritization level, and does not supply an interrupt vector (not 8259A compatible).
It consists of an interrupt request register (IRR), an interrupt mask register (IMR), an interrupt register (IR), and an interrupt acknowledge register IAR). The structure is shown in Figure 4.3-1.

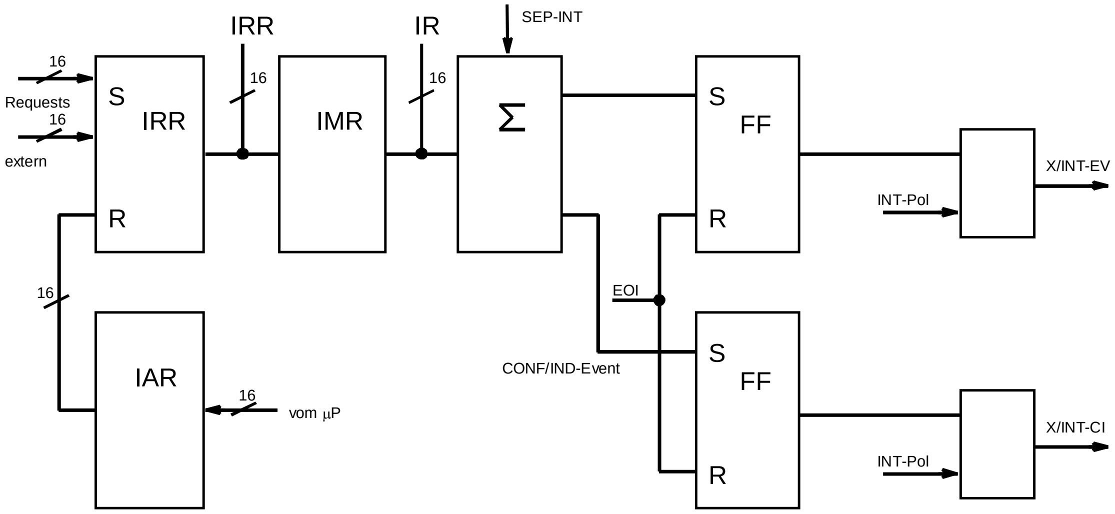


Figure 4.3-1: Interrupt Controller in the ASPC2

In the IRR, each event is stored. Via the IMR, individual events can be suppressed. The entry in the IRR is independent of the interrupt mask. The event signals that are not masked out generate the **X/INT-EV-Interrupt** via a Σ-network. In addition, a second interrupt output is available. Via the parameter "SEP-INT" in Mode Register 0, the COND/IND can be switched either to the joint X/INT-EV interrupt ("SEP-INT = 0") or to the separate **X/INT-CI interrupt** ("SEP-INT = 1").
In the case of the X/INT-EV, the processor has to read the interrupt register (IR) of the ASPC2, in order to determine what kind of interrupt request is pending. The interrupt register is the output of the IMR. For debugging, the processor can set any event in the IRR (only those bits are activated that are to be reset).

Each interrupt event that was processed by the processor has to be cleared via the IAR (including the COND/IND). For this, a log. '1' is to be written to the corresponding bit position. If at the same time a new event and an acknowledge from the previous event is pending at the IRR, the event remains stored. If the processor subsequently enables a mask, it has to be ensured that there is no past entry in the IRR. To make sure, the position in the IRR is to be cleared prior to enabling the maks. Before exiting the interrupt routine, the processor has to set **"End of Interrupt signal (EOI) = 1"** in Mode Register 1. With this edge change, both interrupt lines are switched inactive. If an event should still be stored, the corresponding interrupt output reenters Active after this EOI inactive time (refer to Chapter 7.5.2.2). This makes it possible to reenter the interrupt routine when using an edge-triggered interrupt input.
The EOI inactive time can be parameterized for 1µs or 1 ms (EOI time base, refer to Mode Register 2). To generate the 1ms time, the 32µs base of the delay timer is taken.

The polarity of the interrupt inputs can be parameterized (refer to Chapter 4.2.1.1). After the HW reset, the outputs are low-active.

Interrupt Request Register IRR (writable and readable):

LAS-Changed	Timeout	Syni-Error	Del-Tim-Overflow	Response-Error	Double-Token	Inv-Req-Len/ FIFO-Error/ Rec-Frame-Overflow	MAC-RESET
7							0

CON/IND	CON-SEP-LIST	NOT-OK-LIST	Go-Hold/ Not-Hold-Token	LAS-USELESS	Pass-Token-Error	TS-ADR-Error	HSA-Error
15							8

MAC-Reset: The MAC entered *Offline* upon request ("GO-OFFLINE = 1", refer to Mode Register 1).

Inv-Req-Len/FIFO-Error/Rec-Frame-Overflow:

Inv-Req-Len: An invalid net data length is specified in the request. The addition of the message header and the Layer4 data length resulted in a value of >= 256. The request was confirmed immediately 'NAINVRL' (refer to Chapter 2.2.1) and not sent. This event is stored in the status register.

FIFO Error: The FIFO either had an underrun (send) or an overrun (receive). This event is stored in the status register.

Rec-Frame Overflow: The ASPC2 received another message although it the previous message was not transferred completely to the external memory. This event is stored in the status register . However, when writing to this bit position in the IRR, only the Inv-Req-Len flag is set. The FIFO error and the Rec-Frame Overflow remain unaffected.

Double Token: The MAC detected a double token in the *Hold-Token* mode, and entered *Not-Hold-Token*.

Response Error An error was detected in the mode *Hold-Token* when the response was received.

Del-Tim Overflow: The delay timer (User Timer) has expired, and was restarted (refer to Chapter 4.5).

Syni-Error: The Syni timer expired. If the MAC is in *Hold-Token*, it enters *Not-Hold-Token*.

Timeout: In the ASPC2, the TTIMEOUT expired. In the case of passive stations, there is no further response. Active stations enter *Claim Token* and execute a (re)initialization.

LAS Changed: The status of the LAS has changed. The ASPC2 either entered or removed an active station. The corresponding change is identified in the LAS with "Changed = 1". In *Listen-Token*, this event is generated for each entry.

HSA-Error: A station outside the HSA was detected. The MAC enters *Offline,* if it is in *Listen-Token*; otherwise, the token error counter is incremented

TS-ADR Error: The station address already exists in the ring. The chip enters *Offline* or *Listen-Token*.

Pass-Token Error: When the token was passed, the own start delimiter was not read back at all, or faulty twice. The MAC enters *Offline* because there is a grave error (transmitter or receiver defective?)

LAS-USELESS: In the *Not-Hold-Token* mode, the token error counter has run into a limit. The MAC enters *Listen-Token* and reestablishes the LAS.

Go-Hold/Not-Hold Token: Every time the MAC enters *Not-Hold-Token* or *Hold-Token*, this interrupt is generated. This event arrives for the first time if the MAC has completed *Listen-Token* .

NOT-OK LIST: The ASPC2 executed a confirmation with faulty response status (refer to Chapter 2.1).

CON-SEP LIST: Der ASPC2 has executed a confirmation with faultless response status (refer to Chapter 2.1).

CON-IND: SEP-LIST=0: The ASPC2 has executed either a confirmation or an indication (refer to Chapter 2.1).

SEP-LIST=1: The ASPC2 has executed an indication (refer to Chapter 2.1).

The IR is cleared after reset.

Interrupt Register IR (readable):

For assignment, refer to interrupt request register

Interrupt Mask Register IMR (writable):

For assignment, refer to interrupt request register; after reset, all bits are set (mask is set).

Bit = 1: Mask is set and the interrupt is disabled.
Bit = 0: Mask is cleared, and the interrupt is enabled.

Interrupt Acknowledge Register IAR (writable):

For assignment, refer to interrupt request register; after reset, all bits are cleared.

Bit = 1: The IRR bit is cleared
Bit = 0: The IRR bit remains unchanged.

4.4 Serial Interface

4.4.1 Baudrate Generator

In the baudrate generator (10 bit scaler), baudrates between 9.6kBd and 12MBd can be generated (refer to Chapter 4.2.1). 48MHz is used as clock pulse supply. For the receiver, the baudrate generator (BRG) supplies the 4fold transmission clock pulse, and for the transmitter the single-fold transmission clock pulse. Token, idle, slot, syni, and timeout timers also receive the single-fold transmission clock pulse as their clock pulse.

In the *offline* mode (for example, after reset) the BRG is disabled. If it was parameterized, “START ASPC2= 1” has to be set in Mode Register 1. Thus, the BRG will start defined. It is not permissible to reparameterize the baudrate during operation.

4.4.2 Ser Bus Interface

Signal Name	Input = I
	Output = O
		Type	Comment

TxD	O	non Tristate	
RxD	I		
RTS	O	non Tristate	
XCTS	I		Send Enable

Figure 4.4.2-1: SER Bus Signals of the ASPC2

In the test mode, all outputs are switched high-resistance.

4.5 Delay Timer

In Mode Register 1 (refer to Chapter 4.2.1), the control of the delay timer for the user is implemented (DEL TIM CNTRL1..0). The user can either ‘reset and start’ the delay timer, ‘stop’ it, or ‘reset only’.

4.5.1 User Time Interrupt

The user timer interrupt is the ‘Delay Timer Overflow Interrupt’. If the user wants to implement a software timer, he has to start the delay timer (possible also in the *offline* mode). If ‘User Timebase = 0’, (refer to Mode Register 2), the delay timer covers its entire range and generates the interrupt every 2.1 sec. If ‘User Timebase = 1’, the delay timer is reset after 313 * T32μs, and thus the delay timer overflow interrupt is generated every 10ms.

4.6 FIFO

The ASPC2 has only one FIFO that is switched to the corresponding direction (receive or send direction) with the ‘receive control bit’. Its size is 64/256 bytes, and it is used as temporary buffer for the message characters, and thus for decoupling the SER Bus and the SYS bus. The FIFO consists of a dual port RAM cell with write and read pointers. It is addressed by both. Before one of them enters the first character in the FIFO, it switches the FIFO in the corresponding direction (receive/send) and clears the FIFO contents (reset FIFO). Figure 4.6-1 shows the structure of the FIFO.

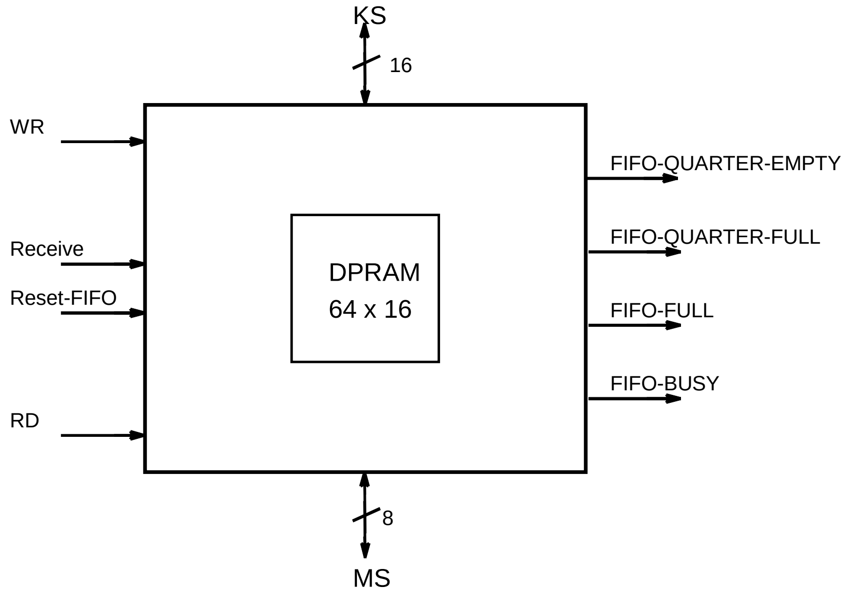


Figure 4.6-1: Structure of the FIFO

4.6.1 Size of the FIFO

The FIFO size can be set to 64 or 256 bytes (Mode Register 2: FIFO 256 byte mode).

4.6.2 Receive Mode

In the receive mode, the message characters are transferred from the UART to the FIFO, and then from the FIFO to the external memory. If an entry causes a FIFO overrun, receiving is aborted, and an event interrupt FIFO error is signalled. If the FIFO is a quarter full (16/64 bytes), it switches FIFO QUARTER FULL active. Then, the SYS bus is obtained, and the entire FIFO content is transferred to the memory.
In the case of the 64Byte FIFO, the transmission time of 48 characters (approx. 40µs at 12 Mbaud) remains for obtaining the bus, and in the case of the 256Byte FIFO, the transmission time of 96 characters (approx. 84µs at 12MBd); otherwise, there will be a FIFO overrun. This time dimensions the maximum external ‘BUS LOCK time’ (refer to Chapter 4.2.1). In the case of a daisy chain, the sum of all ‘BUS LOCK times’ is not to exceed the aforementioned 40/84µs at 12 MBd.

After the last character was entered in the FIFO, FIFO QUARTER FULL switches active also so that the remainder of the data is transferred to the memory.
If ‘Blocked Mode =1’ (refer to Mode Register 2), and if the data length is less than 58/250 bytes, the received net data (Layer2 and 4), in the case of the master, is transferred to the external memory only if it was received completely and faultlessly (including FCS, ED). This parameterization is necessary for data consistency (for example, the data is to reach the mapping memory directly).

4.6.3 Send Mode

In the send mode, the message characters are transferred from the external memory to the FIFO, and then from the FIFO to the UART. The FIFO status line FIFO QUARTER EMPTY is active if the FIFO is at least a quarter empty. Then, the SYS bus is obtained, and the FIFO is filled completely. As soon as the first Layer2 data character is entered in the FIFO, FIFO BUSY switches active. Then, the first message header character is fetched from the FIFO and written to the UART. If the transmit buffer in the UART is free again, the next character is fetched from the FIFO, etc. As soon as gapless transmission can no longer be ensured because there is no more data in the FIFO (FIFO underrun), the send process is cancelled, and an event interrupt FIFO error is generated. The same time conditions apply here to obtain the SYS bus as in the receive mode. If there is a violation, there will be a FIFO underrun.

4.7 LAS RAM

In the LAS RAM, the list of active stations is stored. The ASPC2 exclusively manages the LAS. The processor can only read part of the LAS (see below). The RAM is a 128x4 bit dual port memory cell; each SER bus station is assigned to one RAM address. The last address remains unassigned. In addition to the list of active stations (passive station: State = 0; active station: State = 1), the LAS RAM contains for each bus station the two call sequence bits (FCB, FCV), and a Changed bit which documents the change of the state bit. Figure 4.7-1 shows the structure of the LAS.

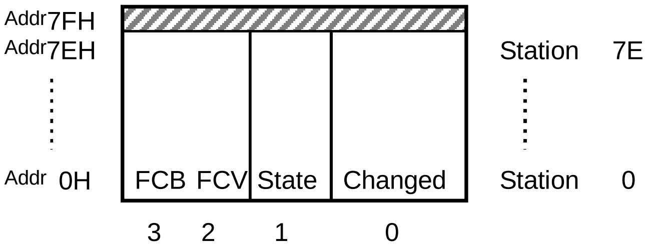


Figure 4.7-1: Content of the LAS RAM

The processor can read out the entire LAS via the LAS register (refer to Figure 4.7-1). For this, “READ LAS = 1” is to be set in Mode Register 1 (refer to Chapter 4.2.1). Then, the ASPC2 loads the content of Address 0H to the LAS register. After each access to the LAS register, the content of the next memory address is loaded to the register, etc.. For this, the ASPC2 needs a recovery time of 5 clock pulses after each LAS access (refer to Chapter 7.5.2.2). After the FLC has read the LAS, “READ LAS =0” has to be set. Depending on ‘EN CLR CHG’, the Changed bits are either not changed, or are reset by the ASPC2 automatically when an entry is copied from the LAS RAM to the LAS register.

0	0	0	0	Change	State	0	0
7							0

Figure 4.7-2: Assignment in the LAS Register

5 Test Support

5.1 Diagnostic Port

To facilitate commissioning, the ASPC2 has a diagnostic port (DIA9..0). With the parameter assignment ‘DEBUG MODE1..0’ in Mode Register 0 (refer to Chapter 4.2.1), the control signals ‘RDCONS’, ‘WRCONS’ can be selected at this port to support a DP master consistency control. Table 5.1-1 shows the assignment, depending on ‘DEBUG MODE1..0’:

DEBUG-MODE1..0	DIA1	DIA0
0 0	WRKONS	RDKONS

Table 5.1-1: Assignment Diagnostic Port

6 Specification of the Bus Timing

6.1 Load of the Processor Interface with Hold Accesses

The ASPC2 imposes no general conditions with respect to time regarding interrupt response timing of the processor; however, it does so regarding the timing for obtaining the bus. The ASPC2 accesses to the external memory may be delayed a maximum of 40/84 µs, otherwise, there will be a FIFO error (refer to Chapter 4.6: FIFO).
The table below shows the bus load timing for the individual channel programs It is based on synchronous bus timing with an internal wait state. To obtain the bus, a time of approx. 150ns is included. The baudrate is 12 Mbaud. The transfer of a word (2 data characters) needs 167ns for this timing (refer to Chapter 7.5.2.1 Timing Diagrams SYS Bus Interface, Master Mode). The timing applies to the 64 byte as well as the 256 byte FIFO (../..).

	Bus Accesses	Function	Bus Load in µs
Enter GAP	1.	Store last FDL Low Request	0,.9
Conf / Indication	1.	Execute confirmation	4.36
		Execute indication (IND-APB, RUP-APB)	8.71
Request	1.	Request lists are empty, no GAP	1.45
	1.	Request lists are empty, GAP	1.61
	1.	Request lists not empty, fill FIFO completely with request header/data + additional 4/10 req. data characters	9.74/15.9
	>= 2.	Reload data; approx. 18/66 char Prepare response (only for the last)	1.85/6.4 + 0.125
Response	1.	Transfer message header	0.775
	>= 2.	Transfer data; approx. 18/66 char	1.95/6.6
Indication	1.	SAP validation Request message header transfer	6.6
	2. - 6.	Transfer data; approx 18/66 char	1.95/6,4
	6.	Fill FIFO completely with response header data and additional 4/10 resp. data	8.69/15.84
	>= 7.	Reload res. data to FIFO ; approx. 18/66 char	1.85/6.4

Table 6.1-1: Bus Load through Hold Accesses

In the case of full SD2 message lengths and default timing for TID1 (35 tBit) and TRDY (11tBit), there is an average bus load of 15% for FMS applications in the *Hold Token* mode.

6.2 Dimensioning the Operating Parameters

If all SER bus components are to contain only ASPC2 chips or corresponding ASICs, the TRDY can be operated with the default value generated by the ASPC2 (refer to Chapter 4.2.1).When dimensioning the idle time TID2, the external 'BUS LOCK time', specified in the number of tBit steps, has to be taken into account for chaining and unchaining application blocks, in addition to the minimum value of TID1/TID2 = 35 tBIT. If a SER bus component is implemented with a SW emulation, it determines the idle timing for the ASPC2. Dimensioning the slot time TSL depends on the internal processing speed of the ASPC2. Depending on the baudrate and parameterized 'higest active address', the TSL has to be set to a time -also in the case of a pure ASPC2 bus- that is considerably above the minimum value of TSL = 37 tBit in the case of the higher baudrates (refer to Table 6.2-1).

Baudrate	HSA = 0FH	HSA = 1FH	HSA = 3FH	HSA = 7EH
12 MBd	150 + *	158 + *	174 + *	206 + *
6 MBd	75 + *	79 + *	87 + *	103 + *
3 MBd	38 + *	40 + *	44 + *	52 + *
1,5 MBd	19 + *	20 + *	22 + *	246 + *

* = BUS-LOCK-Time

Table 6.2-1: Dimensioning the Slot Time

The additive constant * corresponds to the 'BUS LOCK time' needed by the processor. When entering *Hold Token*, the ASPC2 can do up to three hold requests successively prior to becoming active on the SER bus. However, during this time, the station that has passed the token is in the *check token pass* mode in which it monitors token passing via the slot time. In the case of a dual port memory interface (refer to Chapter 4.1.2), double the 'BUS LOCK time' has to be taken into account if the FLC executes several chainig and unchaining processes successively. In the case of shared memory, the processor is stopped via the hold request, and is thus not able to set more than one 'BUS LOCK' during this *check token pass* phase.

7 Electrical Specification

The chapter below provides only the most important parameters. Additional features are described in the manual TGC2000/TEC2000 5V CMOS Arrays by TI.

7.1 Maximum Limits

Parameter	Designation	Limits	Unit
DC Supply Voltage	VDD	- 0.3 ... 6.0	V
Input Voltage	VI	- 0.3 ... VDD + 0.3	V
Output Voltage	VO	- 0.5 ... Vdd + 0.5	V
DC Input Diode Current	IIK	-10...10	mA
DC Output Diode Current	IOK	-20...20	mA
Storage Temperature	TST	-55...150	°C
Ambient Temperature	TA	-40...85	°C
Chip Temperature during Operation	TJ	-40...125	°C

7.2 Permissible Operating Values

Parameter	Designation	MIN.	MAX.	Unit
DC Supply Voltage (VSS=0V)	VDD	4.5	5.5	V
Input Voltage	VI	0	VDD	V
Output Voltage	VO	0	VDD	V
Ambient Temperature	TAI	-40	85	°C

Table 7.2-1: Permissible Operating Values

7.3 Guaranteed Operating Range for the Specified Parameters

Parameters		MIN.	MAX.	Unit
DC Supply Voltage (VSS=0V)		4.5	5.5	V
Ambient Temperature		-40	85	°C

Table 7.3-1: Guaranteed Operating Range for the Specified Parameters

7.4 DC Specification of the Pad Cells

Parameter		Name	MIN.	TYP.	MAX.	Unit
Input Voltage 0-Level	CMOS	VILC			0.3VDD (5)	V
	TTL	VILT			0.8	V
Input Voltage 1-Level	CMOS	VIHC	0.7VDD (4)			V
	TTL	VIHT	2.0			V
Pos. Threshold Schmitt-Trigger	CMOS	VT+C			0.8VDD	V
Neg. Threshold Schmitt-Trigger	CMOS	VT–C	0.2VDD			V
Output Voltage 0-Level		VOL			0.4 (1)	V
Output Voltage 1-Level		VOH	VDD-0.8		(1)	V
Input Leakage Curr.	0-Level	IIL	-1			µA
Input Leakage Curr.	1-Level	IIH			1	µA
Output Leakage Curr.	0-Level	IOZL	-10			µA
Output leakage Curr.	1-Level	IOZH			10	µA
Output Current 4mA Cell	0-Level	IOL	3,84 (1)			mA
Output Current 4mA Cell	1-Level	IOH	3,84 (1)			mA
Output Current 8mA Cell	0-Level	IOL	7,69 (1)			mA
Output Current 8mA Cell	1-Level	IOH	7,69 (1)			mA
Input Capacity		CIN	3.6		4.6	pF
Output Capacity		COUT	4		5	pF
I/O Capacity		CIOUT	4		5	pF
Output Current Pull-Up Resistor		IOPU	-5.5 mA <= IOPU <= 31.2 mA (2)			µA
Output Current Pull-Down Resistor		IOPD	80.7 mA <= IOPD <= 374 mA (3)			µA

(1) VOL,VOH,IOL,IOH are tested with VDD=4,5 V

(2) VIN= 0 V, VDD = 5,5 V (4) maximum permissible AC current

(3) VIN = VDD=5,5

Tabel 7.4.2: DC-Specification of the Pad Cells

7.5 AC Specifications

7.5.1 Driver Capability

The runtimes at the chip outputs always depend on the driver capacity of the pad cells, as well as on the assumed capacitive load. The capacitive load on which the timing data below is based is provided in Table 7.5.1-1. For specifying the maximum and minimum runtimes, the variations of temperature- and supply voltage range shown in Table 7.3-1 were also allowed for.

Signal Name	Direction	Driver Type	Capacity	Cap. Load
DB15..0	In/Out	Tristate	8mA	100pF
AB5..0	In/Out	Tristate	8mA	100pF
XBHE/XWRH	In/Out	Tristate	8mA	50pF
XRD	In/Out	Tristate	8mA	50pF
XWR/XWRL	In/Out	Tristate	8mA	50pF
AB19..6	Out	Tristate	8mA	100pF
RTS	Out	non-Trist.	8mA	50pF
TxD	Out	non-Trist.	8mA	50pF
XHTOK	Out	non-Trist.	8mA	50pF
DT/XR	Out	Tristate	4mA	50pF
X/INT-CI	Out	non-Trist.	4mA	50pF
X/INT-EVENT	Out	non-Trist.	4mA	50pF
HOLD	Out	Tristate	4mA	50pF
X/HOLDAOUT	Out	non-Trist.	8mA	50pF
XENBUF	Out	non-Trist.	4mA	50pF
XREQRDY	Out	non-Trist.	4mA	50pF
DIA9..0	Out	non-Trist.	4mA	50pF
XCLK2	Out	non-Trist.	4mA	50pF

Table 7.5.1-1: Identification Data of the Outputs

If, in reality, the capacitive load should deviate from the assumed values, there is a change of 1ns maximum per 25pF for the 4/8mA output level.

7.5.2 Timing Diagrams, Signal Runtimes

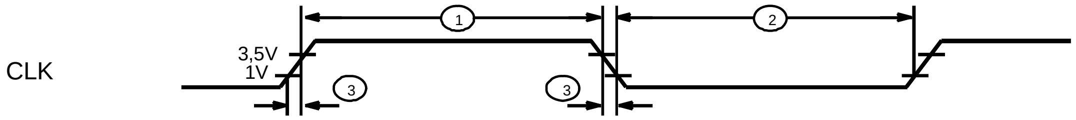
The signal runtimes are based on the capacitive loads specified in Table 7.5.2.1-1

7.5.2.1 SYS Bus Interface, Master Mode

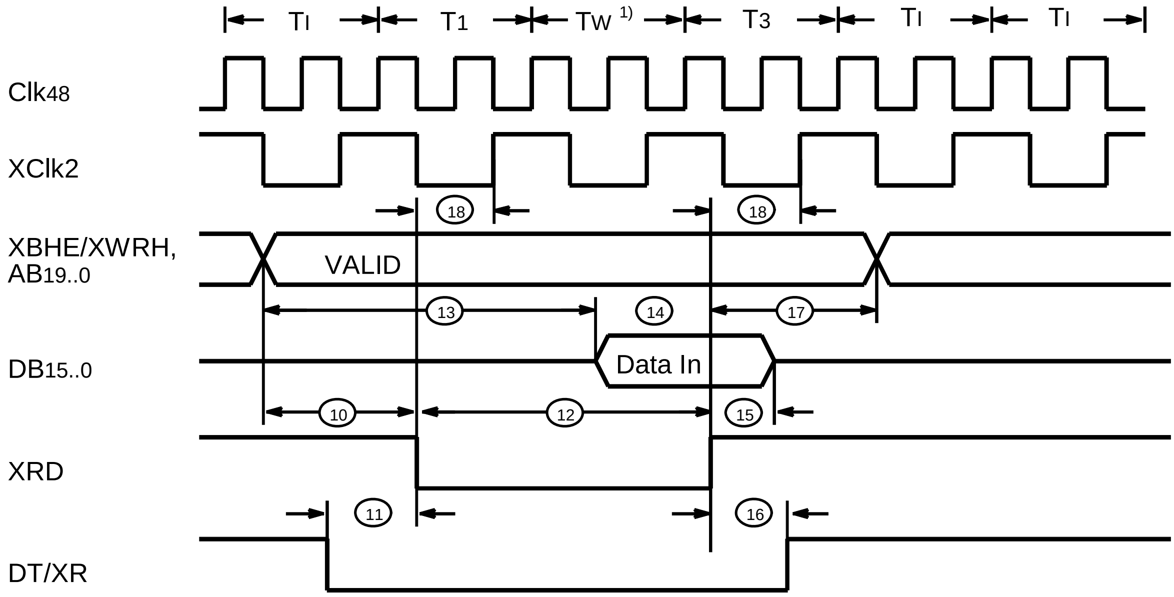
No.	Parameter	Min.	Max.	Unit
1	Clock High Time	5,2*	15,6*	ns
2	Clock Low Time	5,2**	15,6**	ns
3	Clock Rise- Fall Time		5,2	ns
	Quick Access Mode=0:			
10	Address to XRD ↓ Setup Time	27		ns
11	DT/XR Low to XRD ↓ Setup Time	19		ns
12	XRD Active Width	(2+2n)T_{48} [1]		ns
13	Address valid to Data valid		2(n-1)T_{48} + 100	ns
14	Data to XRD ↑ Setup Time (8/16Bit-Mode)	18		ns
15	Data to XRD ↑ Hold Time (8/16Bit-Mode)	3		ns
16	DT/XR High to XRD ↑ Hold Time	20		ns
17	Address to XRD ↑ Hold Time	2T_{48}		ns
18	XRD/XWR/XWRL ↓↑ XCLK2 ↑ Setup Time	0		ns
19	Address to XWR/XWRL ↓ Setup Time	27		ns
20	XWR/XWRL Active Width	(2+2n)T_{48} [1]		ns
21	Data Valid to XWR/XWRL ↑ Setup Time	(2+2n)T_{48} - 26		ns
22	Data to XWR/XWRL ↑ Hold Time	2T_{48}		ns
23	Address to XWR/XWRL ↑ Hold Time	2T_{48}		ns
	Quick Access Mode=1:			
25	Address to XRD ↓ Setup Time	15		ns
26	DT/XR Low to XRD ↓ Setup Time	0		ns
27	XRD Active Width	(1+2n)T_{48} [1]		ns
28	Address valid to Data valid		2(n-1)T_{48} + 65	ns
29	Data to XRD ↑ Setup Time (8/16Bit-Mode)	16		ns
30	DT/XR High to XRD ↑ Hold Time	0		ns
30a	Address to XRD ↑ Hold Time	1		ns
30b	Address to XWR ↑ Hold Time	1		ns
30c	Data to XWR ↑ Hold Time	1		ns
31	Address to XWR/XWRL ↓ Setup Time	13		ns
32	XWR/XWRL Active Width	(1+2n)T_{48} [1]		ns
33	Data Valid to XWR/XWRL ↑ Setup Time	(1+2n)T_{48} - 17 [4]		ns
35	XRDY to Clk_{48} ↑ Setup Time	5		ns
36	XRDY to Clk_{48} ↑ Hold Time	5		ns
37	XRDY Active Width	2T_{48} + 10		ns
38	XRDY ↓ to XRD/XWR ↑ Delay (kein zus. Waitstate)	2T_{48} [2]	4T_{48} [2]	ns
39	XRDY ↓ to XRD/XWR ↑ Delay (ein zus. Waitstate)	4T_{48} [2]	6T_{48} [2]	ns
40	Read- / Write Cycle Time (Quick Access = 0)	8T_{48} [3]		ns
41	XRD; XWR/XWRL Inactive Delay zwischen aufeinanderfolgenden Wort-Zugriffen (Quick Access = 0)	4T_{48}		ns
42	Read- / Write Cycle Time (Quick Access = 1)	4T_{48} [3]		ns
43	XRD; XWR/XWRL Inactive Delay zwischen aufeinanderfolgenden Wort-Zugriffen (Quick Access = 1)	15		ns
44	XRD; XWR/XWRL Inactive Delay zwischen den beiden aufeinanderfolgenden Zugriffen im 8Bit-Mode	15		ns

* treshold 3,5V ** treshold 1V 1) n = number of Waitstates 2) internal Waitstates expired
3) 1based on internal waitstate

Table 7.5.2.1-1: Specification of the Processor Bus Interface, Master Mode

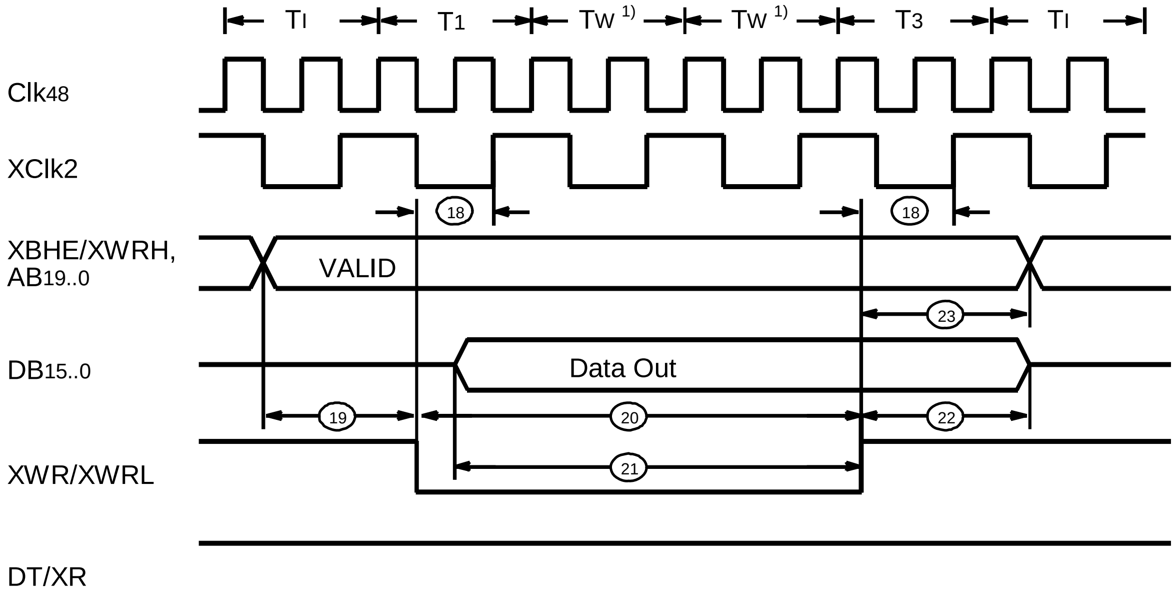


7.5.2.1-1: Clock-Timing



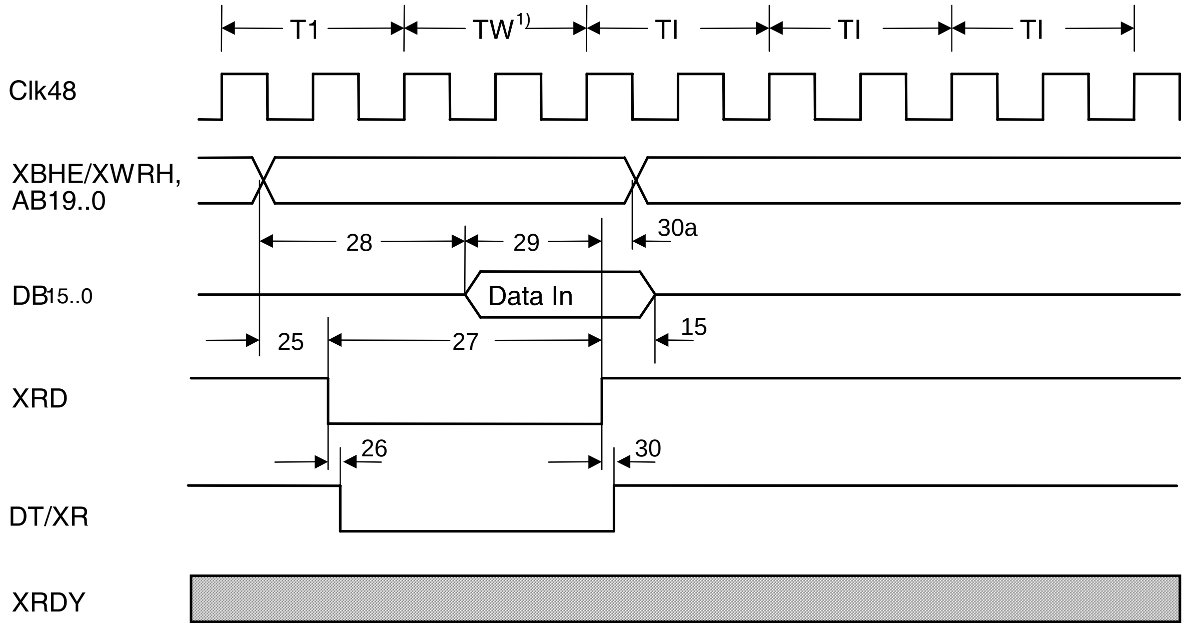
1) here 1 int. Waitstate, ext. Ready 'off'
2) XWR/XWRL = log.'1'

Figure 7.5.2.1-2: Master Mode (Quick Access=0), ASPC 2 Read Timing (without external Ready)



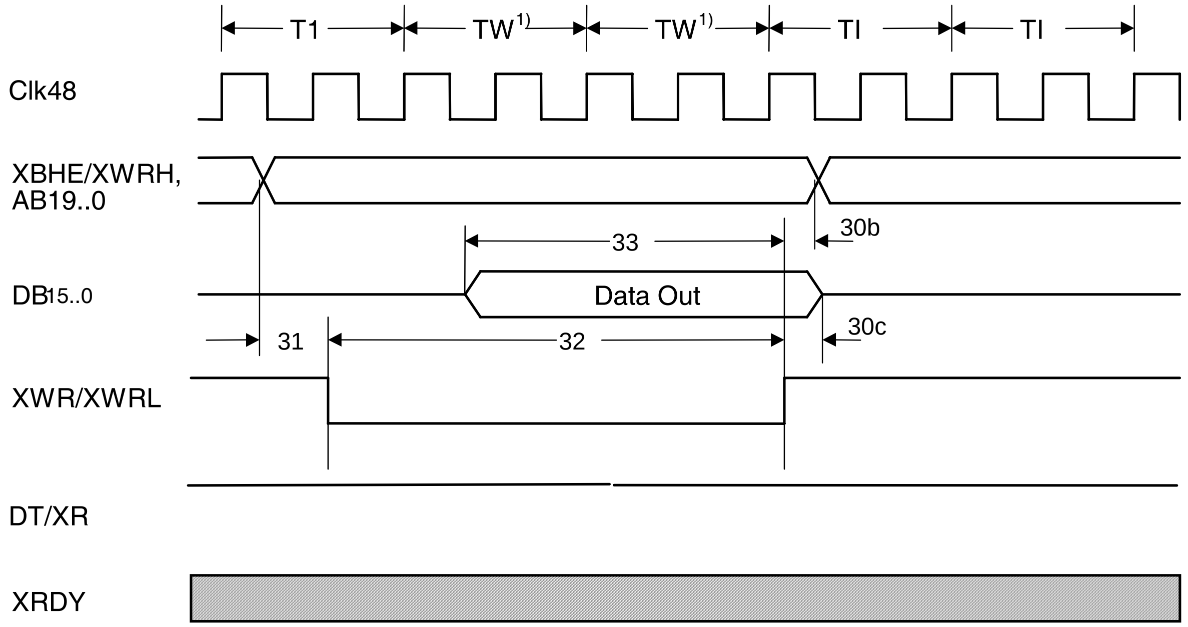
1) here, 2 int. wait states, ext.Ready 'off'
2) XRD = log.'1'

Figure 7.5.2.1-3: Master Mode (Quick Access=0), ASPC 2 Write Timing (without external Ready)



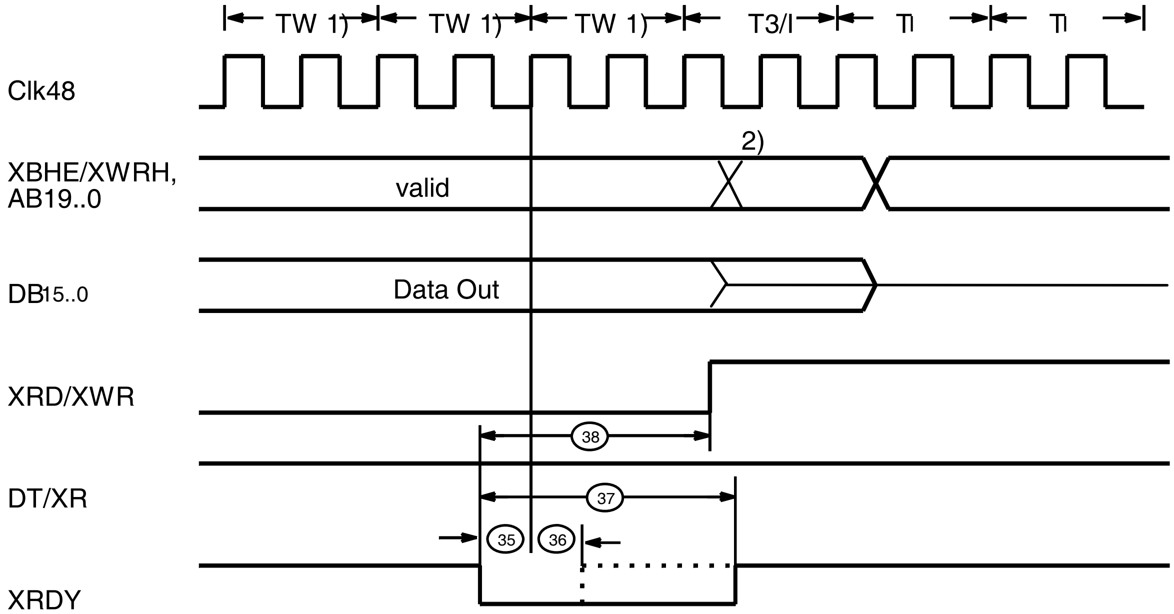
1) here 1 int. Waitstate, ext. Ready 'off'
2) XWR/XWRL = log.'1'

Figure 7.5.2.1-4 Master Mode (Quick-Access=1), ASPC 2 Read Timing (without external Ready)



1) here 2 int. Waitstate, ext. Ready 'off'
2) XRD = log.'1'

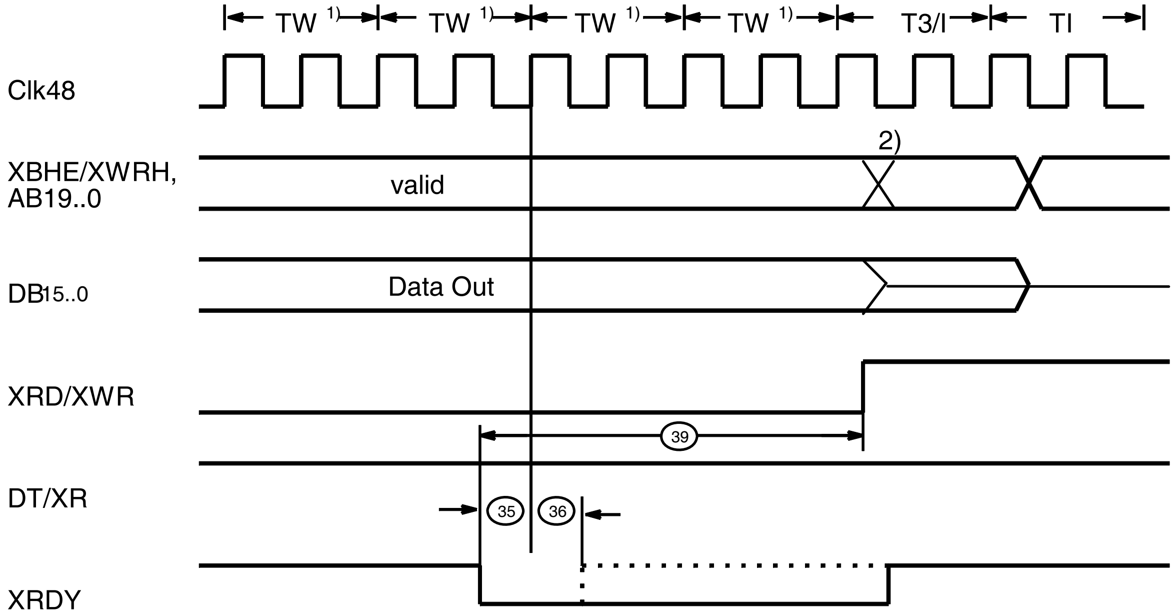
Figure 7.5.2.1-5: Master Mode (Quick Access=1), ASPC 2 Write Timing (without external Ready)



1) here int. Waitstates expired, ext. Ready `on'
2) valid for Quick-Access=1

Fig

Figure 7.5.2.1.6: Master-Mode, ext. Ready Timing (no additional wait state after Ready activation)



1) here internal waitstate expired, ext. Ready
Additional waitstate after activating ready

2) valid for Quick-Access=1

Figure 7.5.2.1-7: Master Mode, ext. Ready Timing (one additional wait state after Ready activation)

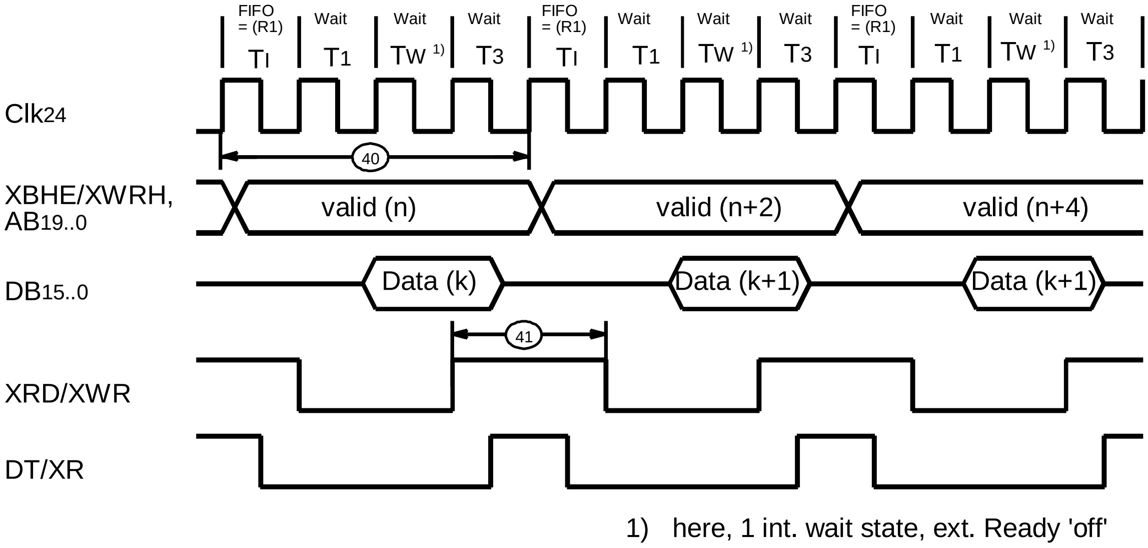


Figure 7.5.2.1-8: Master Mode (Quick Access=0), continuous accesses (here to data buffer)

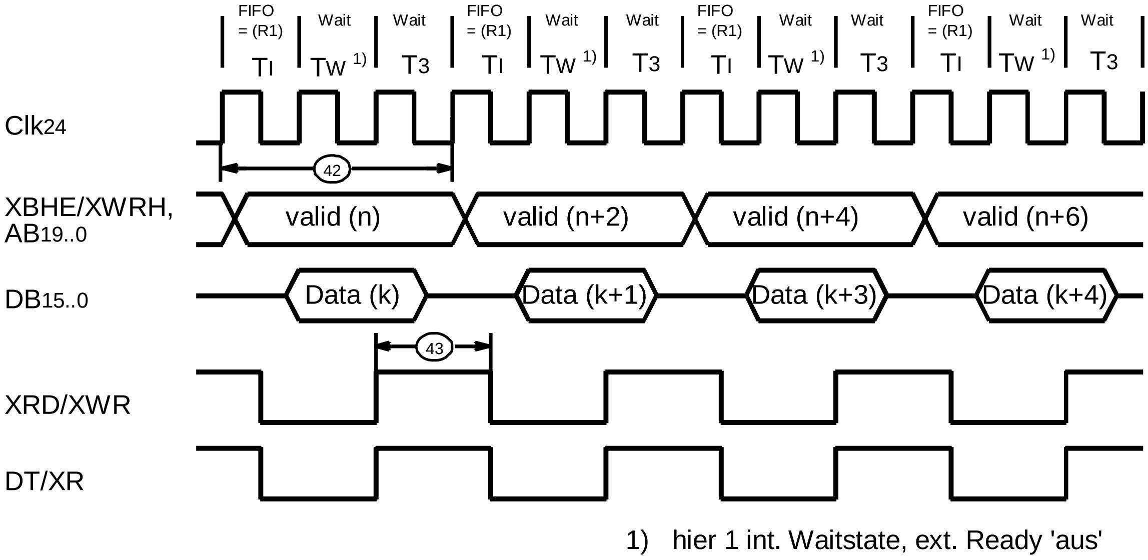


Figure 7.5.2.1-9: Master Mode (Quick Access=1), continuous accesses (here to data buffer)

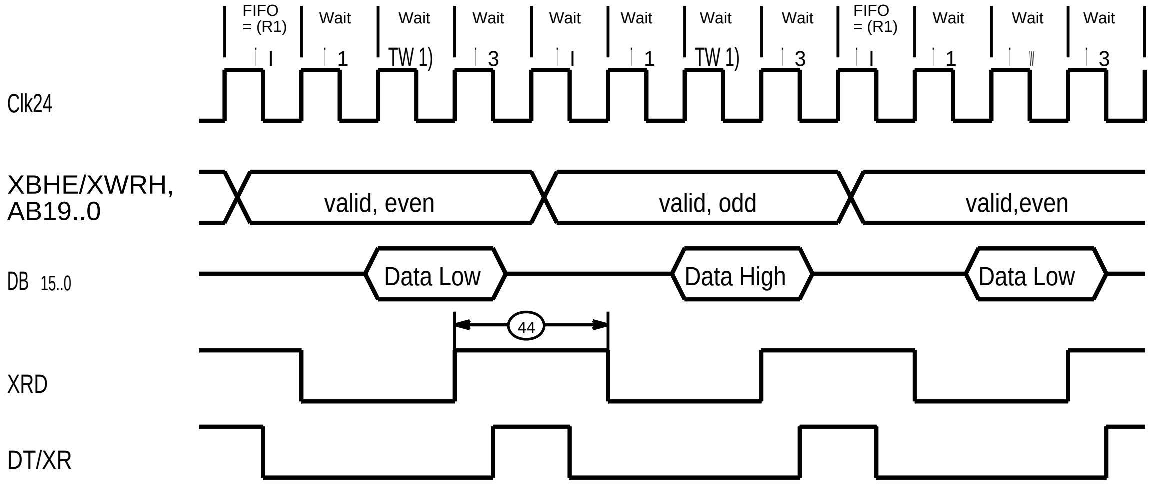


Figure 7.5.2.1-10: Master Mode (Quick Access=0), 8Bit Interface Read Timing, Intel Mode

No.	Parameter	Min.	Max.		Unit
			8Bit-Int.	16Bit-Int.	
50	X/HOLDAIN to Command/Address HiZ to Active Delay	2T48	6T48	6T48	ns
51	HOLD ↓ to Command/Address HiZ Delay	0			ns
52	X/HOLDAIN minimal Active Width	3T48			ns
53	min. Hold Inactive after X/HOLDAIN Inactive	2T48			ns
55	XREQ ↓ to XENBUF ↓ (BUSLOCK=0)	2T48	20T48	12T48	ns
	(BUSLOCK=1, ASPC-Lock=0)	2T48	20T48	12T48	ns
	(BUSLOCK=1, ASPC-Lock=1)	2T48	22.6 [1]	11.3 [1]	m
56	XENBUF↓ to XREQRDY↓(QREQRDY=0 in TQUI-Reg)	3T48			ns
	(QREQRDY=1 in TQUI-Reg)	1T48			ns
57 [2]	XREQ↑ to XENBUF/XREQRDY↓		1.5T48	1.5T48	ns
58	XREQ Inactive width	2T48 + 4			ns
60	RD/WRCONS ↑ to XRD/XWR ↓	2T48			ns
61	XRD/XWR ↑ to RD/WRCONS ↓	4T48			ns

1) Fill FIFO completely (128 bytes)
2) Spikes at the rising edge of XREQ also cause spikes on XENBUF and XREQRDY !

In general: each memory access of the ASPC 2 is based on 1 wait state.

Table 7.5.2.1-2: Specification of the Hold/XREQ Interface

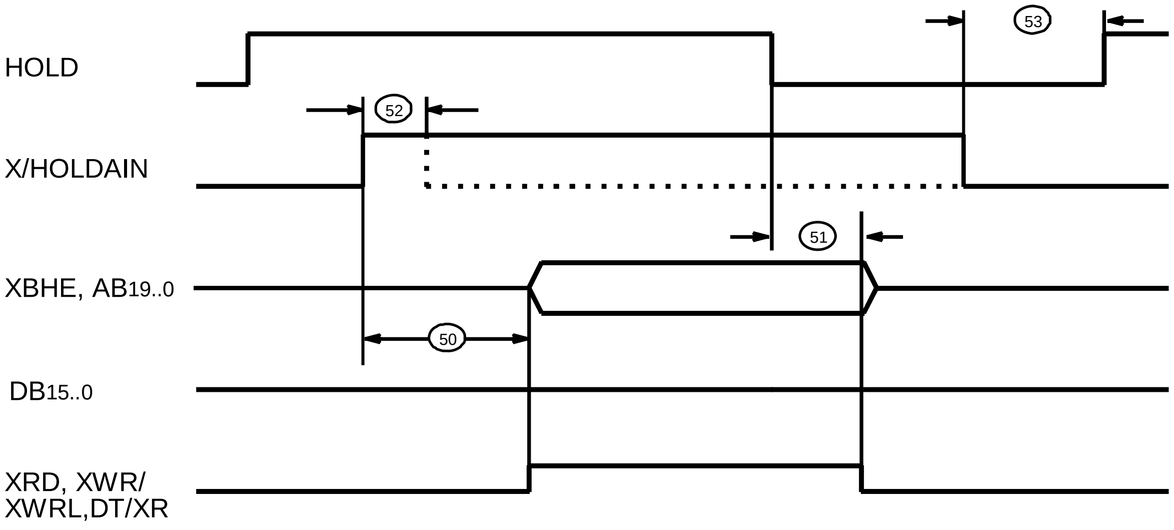


Figure 7.5.2.1-11: Master Mode Shared Memory (XSHM/DPM=0 in Mode Register 0), Hold Timing

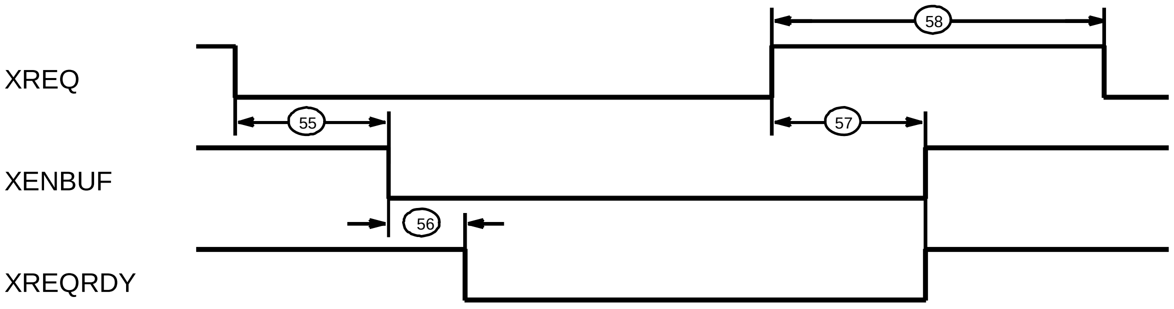


Figure 7.5.2.1-12: Master Mode Dual Port Memory (XSHM/DPM=1 in Mode-Register 0), XREQ Timing

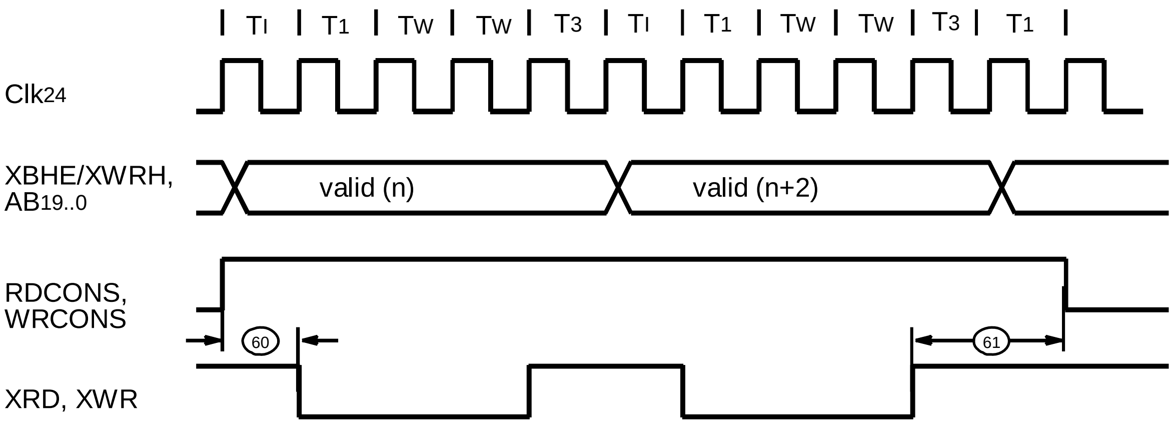


Figure 7.5.2.1-13: RDCONS- WRCONS Timing in reference to XRD/XWR

7.5.2.2 SYS Bus Interface , Peripheral Mode

No.	Parameter	Min.	Max.	Unit
70	Address Required Valid to Read Data Valid Delay		20	ns
71	XCS ↓ to Data Active Delay	3	15	ns
72	XRD ↓ to Data Active Delay	3	15	ns
73	XRD Active width	3T_{48} [1)]		ns
74	Output Hold from Address change	2		ns
75	XRD ↑ to Read Data HiZ	2	10	ns
76	XCS ↑ to Read Data HiZ	2	10	ns
77	XRD Inactive Delay	1T4		ns
		9T_{48} [2)]		ns
78	Address to XWR/XWRL ↓ Setup Time	0		ns
79	XCS Low to XWR/XWRL ↓ Setup Time	0		ns
80	XWR/XWRL Active width	4T_{48}		ns
81	Write Data Valid to XWR/XWRL ↑ Setup Time	4		ns
82	Write Data Valid to XWR/XWRL ↑ Hold Time	6		ns
83	Address to XWR/XWRL ↑ Hold Time	0		ns
84	XCS ↑ to XWR/XWRL ↑ Hold Time	4		ns
85	XWR/XWRL Inactive Delay	1T_{48} + 4		ns
86	XWR/XWRL ↑ to XRD ↓ Inactive Delay	1T_{48}		ns
		5T_{48} [3)]		ns
		9T_{48} [4)]		ns
90	XREQ ↓ to XCSASPC2 ↓		1T_{48}	ns
91	X/INT-Event Inactive Width (EOI-Timebase=0)	1000		ns
	(EOI-Timebase=1)	992	1025	ms
92	User Timer Clock Period	10302.6	10334.6	ms

1) Determined by the LAS access
2) between two read accesses to the LAS
3) at Read to IRR/IR after Write to IRR/IAR
at Read to IR after Write to IMR
4) at Read to LAS after Write 'LASREAD=1' (Mode Reg. 1)

Table 7.5.2.2-1: Specification of the Processor Bus Interface, Peripheral Mode

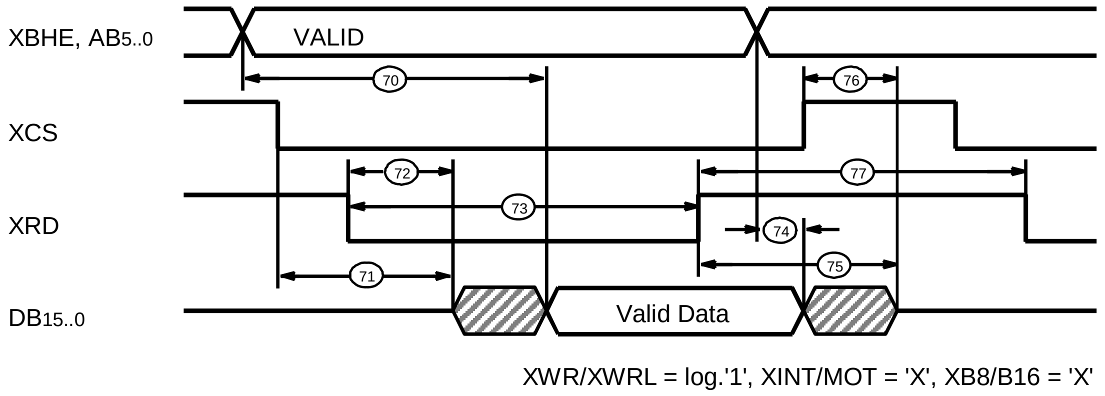


Figure 7.5.2.2-1: Peripheral Mode Shared Memory, Processor Read Access to the ASPC2 during ´BUSLOCK=1´

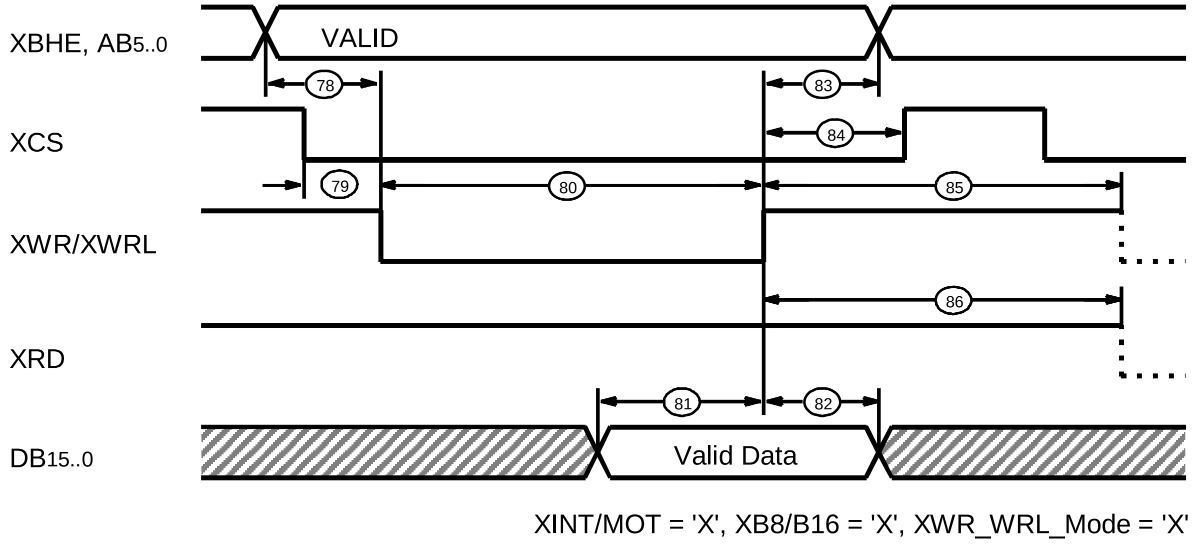


Figure 7.5.2.2-2 : Peripheral Mode Shared Memory, Prozessor Write Access to the ASPC2

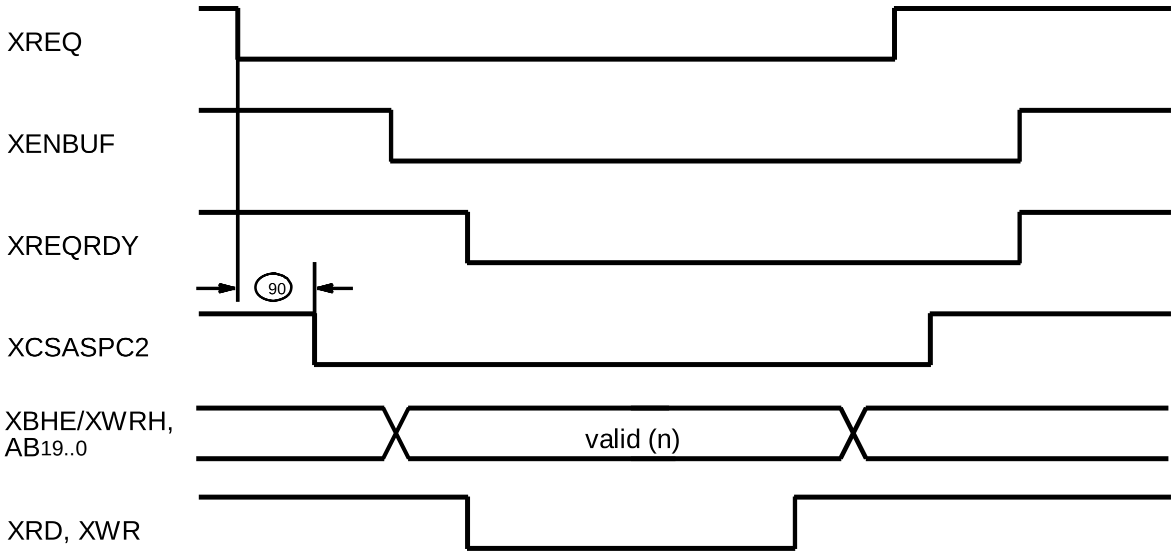


Figure 7.5.2.2-3: Peripheral Mode Dual Port Memory: Processor Access to the ASPC 2

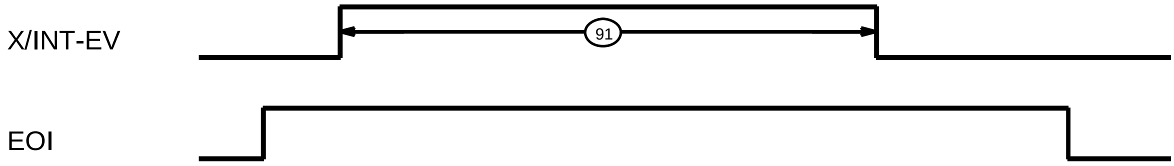


Figure 7.5.2.2-4: Peripheral Mode, Interrupt EOI Timing

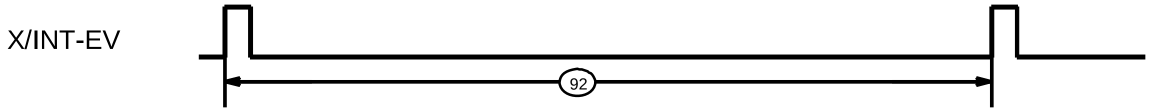


Figure 7.5.2.2-5: User Timer Clock Period

7.5.2.3 SER Bus Interface

No.	Parameter	Min.	Max.	Unit
95 96 97	RTS " to TxD Setup Time XCTS # to TxD Setup Time RTS # to TxD Hold Time	(2n+5)T48 (n≥1) [1] 3T48 4T48		ns ns ns

1) n = RTS_Pre_Value in Monitor Selector1

Table 7.5.2.3-1: Specification of the SER Bus Interface

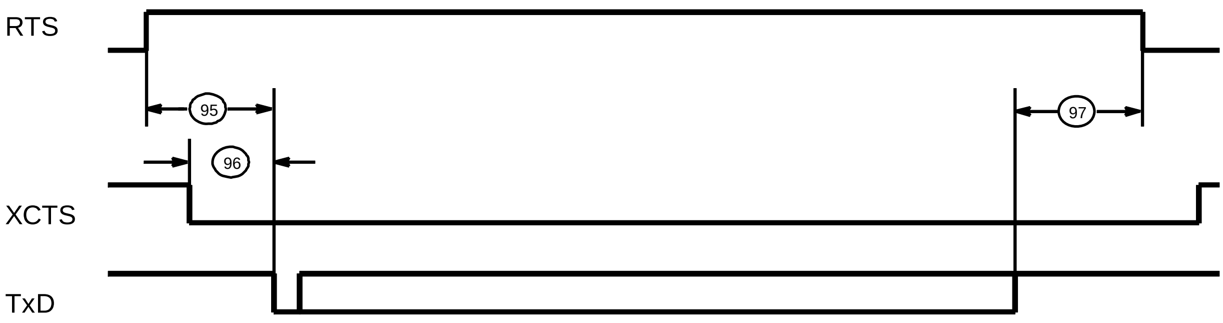


Figure 7.5.2.3-1: Transmit Timing:

8 Example

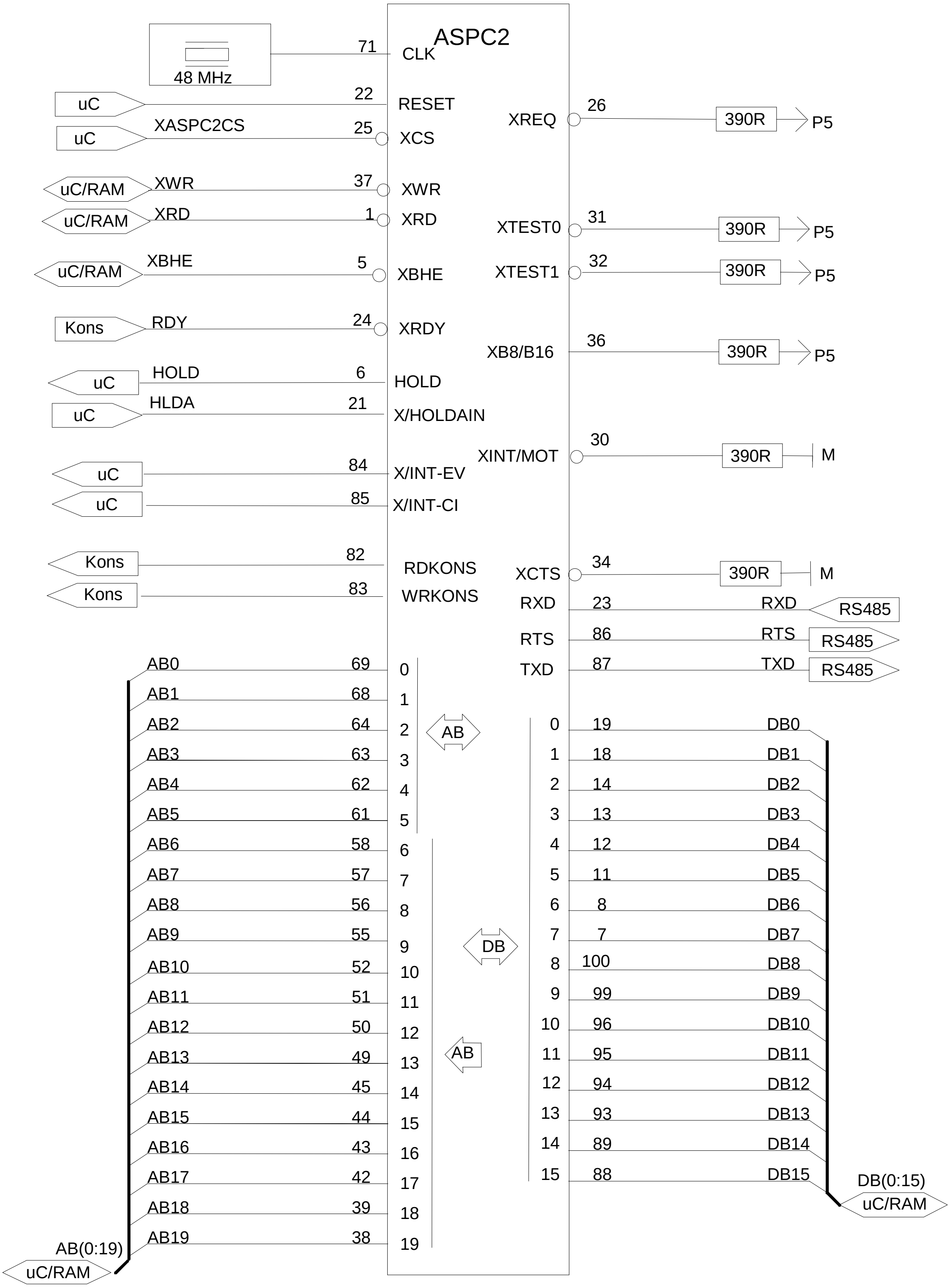


Figure 8-1: ASPC2 Wiring

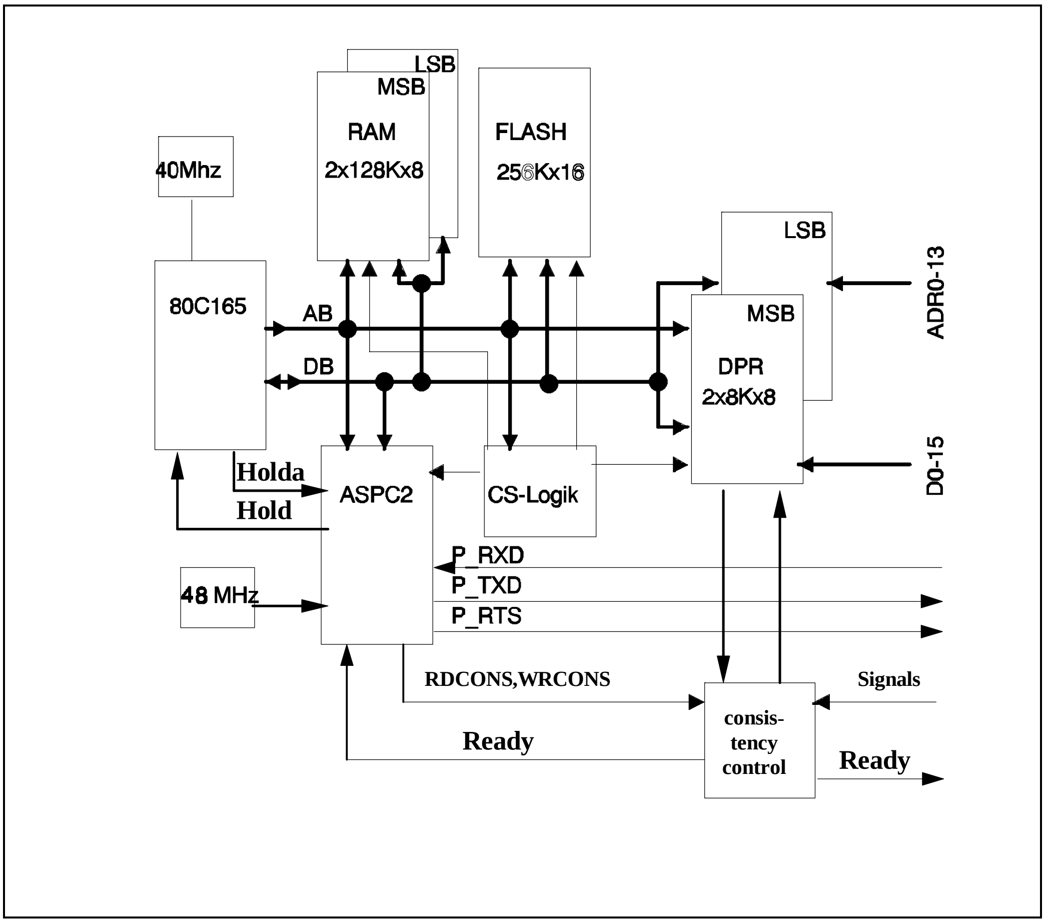


Figure 8-2: Example Overview

8.1 General

The ASPC2 and the 80C165 are connected via the ***Shared Memory Mode***.
The ASCP2 processes in the **16 Bit Intel Byte Ordering Mode**.
The **joint** interrupt line X/INT-EV is used.
The ASPC2 accesses the DPRAM in the **Ready delay mode**.
The consistency signals are applied to the pins DIA0 and DIA1.
The transfer in the FIFO is made in the **Blocked Mode**.

256 Kbyte RAM for system control-, application-, data blocks and variables.
512 Kbyte Flash for program and parameter binary file(128K for parameters).
16 Kbyte DPRAM for inputs, outputs, and diagnosis of slaves.

The Profibus signals can be wired according to Chapter 9.1.

8.2 Consistency Control

The consistency control controls the access to the DPRAM from the one side (CPU and ASPC2 (master)) as well as from the other side (host). Normally, the other side respectively is retarded only for a short time with ready delay by the DPRAM when the same address is accessed simultaneously.
If the data (inputs and outputs of the slaves up to FIFO size (58/250)) is to be transmitted consistently, the ASIC, on the master-side, can activate the RDCONS, WRCONS signals for the consistency control during access through entries in the request blocks for consistency control.

On the host-side, corresponding signals are needed, in order to set the consistency control for reading or writing.
Thus, through the control, the other side respectively can be locked out when required through hardware from accesssing entire memory areas by withholding Ready.

8.3 Comments

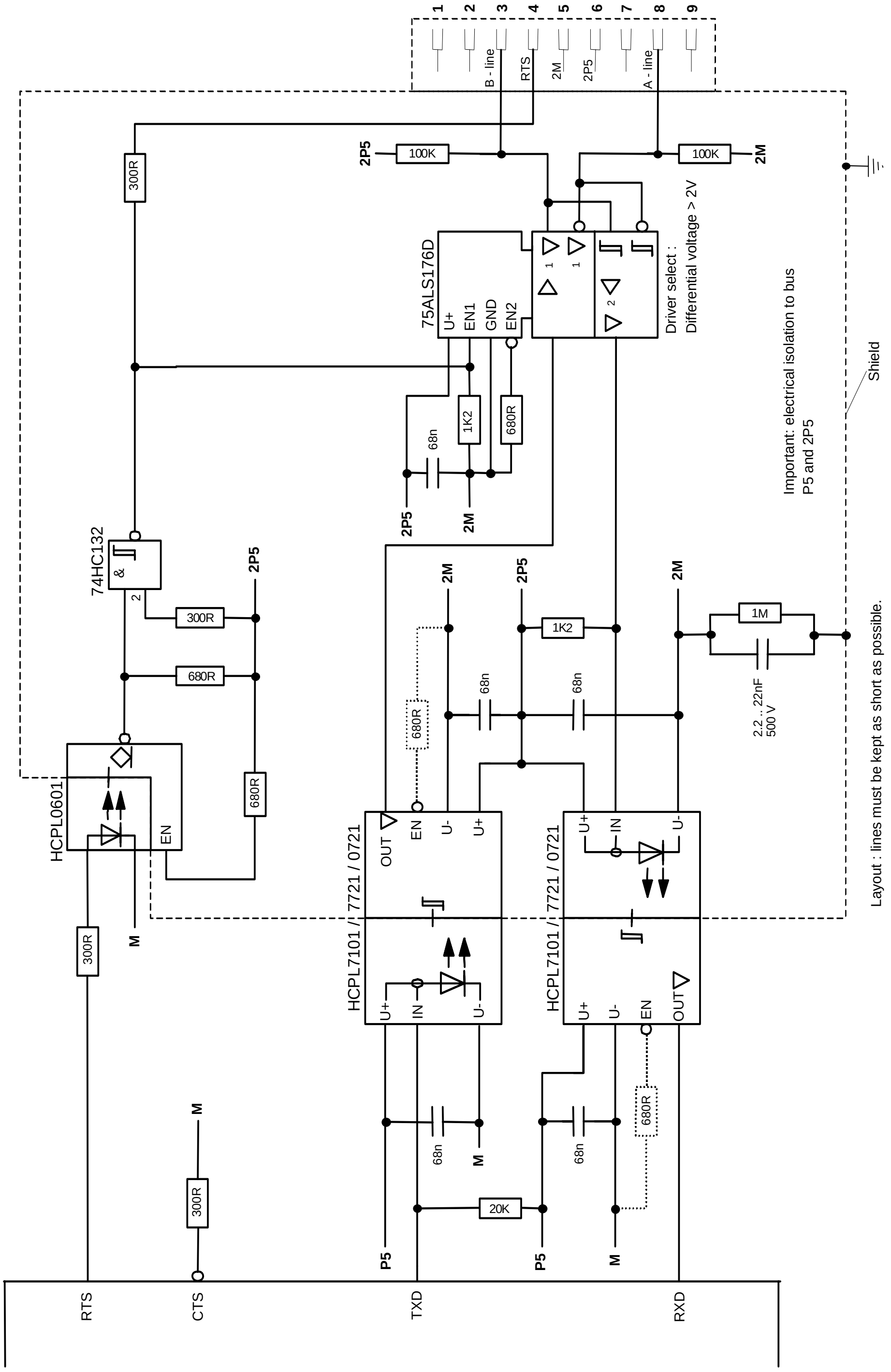
☞ **All data which the ASPC2 accesses has to be located in the 1MB range.**

- Parameter data of COM PROFIBUS V *.*
- DP RAM for inputs/outputs, and diagnoses of the slaves
- Layer2 data blocks for transmission
- System control block and application blocks have to be located in a 64Kbyte segment, in addition.

9 Profibus Interface

9.1 Wiring Example RS485 Interface

The data is transmitted in the operating mode RS485 (RS485 physics). The figure below shows the possible wiring for Profibus. The PROFIBUS interface is usually executed as a 9-pole SUB D female socket.



9.2 Pin Assignment

Pin 1 - free
Pin 2 - free
Pin 3 – B-Line
Pin 4 - Request to Send (RTS)
Pin 5 - Gound 5V **(M5)**
Pin 6 - Potential 5V **(floating P5)**
Pin 7 - free
Pin 8 – A-Line
Pin 9 - free

The line shield is to be connected to the socket housing.
The free pins are used optionally in the EN 50 170 and should, if the user uses them, correspond to this description.

ATTENTION:
The designations **A** and **B** of the lines on the socket correspond to the designations in the RS485 standard, and not to the pin designation of driver ICs.
The line length from the driver to the socket is to be kept as short as possible.

9.3 Connectors

Siemens offers connectors that make it possible to connect terminators.

When using the higher baudrates from 3 to 12 Mbaud, it is necessary to use special connectors. These connectors compensate for line influences regarding all possible line combinations.

Order No.	Notes	Baudrate	Color of the Connector Housing
6ES7 972 - 0BA00 - 0XA0	Without PG Conn.	Up to 1.5 Mbd	anthracite
6ES7 972 - 0BB00 - 0XA0	With PG Connection	Up to 1.5 Mbd	anthracite
6ES7 972 - 0BA10 - 0XA0	Without PG Conn.	> 1.5 Mbaud	anthracite
6ES7 972 - 0BB10 - 0XA0	With PG Connection	> 1.5 Mbaud	anthracite

9.4 Line Length

The maximum line lengths and spur line lengths are to be implemented according to the specifications in the EN 50170, and dependent on the baudrates. If these values are not sufficient, it is possible to use repeaters.

10 Appendix

10.1 Addresses

PROFIBUS Nutzer Organisation
PNO
Office
Mr. Dr. Peter Wenzel
Haid- und Neu- Strasse 7
76131 Karlsruhe/Germany
Tel.: (0721) 9658-590

Contact Persons at the Interface Center in Germany

Siemens AG
Deptl IA SE DE DP3
Mr. Putschky
Würzburgerstr.121
90766 Fürth/Germany

Email:
gerd.putschky@siemens.com

Tel.: (0911) 750 - 2078
Fax: (0911) 750 - 2100

Contact Persons at the Interface Center in the USA

PROFIBUS Interface Center
One Internet Plaza
PO Box 4991
Johnson City, TN 37602-4991

Fax : (423) - 262 - 2103

Your Partner:
Tel.: (423) - 262 - 2576

Email:
profibus.sea@siemens.com

10.2 List of Abbreviations

Most of the abbreviations used in the specification are defined in the terminology database. Here, those abbreviations are listed primarily that are not in the database.

APB	**Ap**plication **B**lock
ASPC2	**A**dvanced **S**iemens **P**ROFIBUS **C**ontroller **2**; AS**I**C
BM	**B**ackground **M**essages
BRCLK	**B**aud**r**ate **Cl**oc**k**
DP	**D**istributed **P**eriphery
ED	**E**nd **D**elimiter
FCS	**F**rame **C**heck **S**equence
FLC	**F**ieldbus **L**ink **C**ontrol
HW	**H**ard**w**are
IAR	**I**nterrupt **A**cknowledge **R**egister
IMR	Interrupt **M**ask **R**egister
IR	**I**nterrupt **R**egister
IRR	**I**nterrupt **R**equest **R**egister
L2	**L**ayer **2** of the **I**SO/ OS**I-**7 Layer Model
L4	**L**ayer **4** of the **I**SO/ OS**I-**7 Layer Model
LAS	**L**ist of **A**ctive **S**tations
LSB	**L**east **S**ignificant **B**it
MAC	**M**edium **A**ccess **C**ontroller
NA	**N**o **A**ccess
NS	**N**ext **S**tation
PA	**P**rocess **A**utomation
PRF	**P**arameter **R**egister **F**ile
PROFIBUS	**Pro**cess **Fi**eld**bus**
PS	**P**revious **S**tation
RTS	**R**equest **t**o **S**end
SAP	**S**ervice **A**ccess **P**oint
SCB	**S**ystem **C**ontrol **B**lock; memory area needed for the ASPC2
SC	**S**hort **C**haracter
SD	**S**tart **D**elimiter
SDA	**S**end **D**ata with **A**cknowledge
SDN	**S**end **D**ata with **n**o Acknowledge
SER**-**Bus	**ser**ial communication bus
SPC	**S**iemens **P**ROFIBUS **C**ontroller; AS**I**C
SPM	**S**iemens **P**ROFIBUS **M**ultiplexer; AS**I**C

SRD	**S**end and **R**equest **D**ata with Reply
SYS-Bus	parallel **Sys**tem Bus
Tn	Station
TRT	**T**oken **R**otation **T**imer
TTR	**T**arget **R**otation **T**ime
TS	**T**his **S**tation
USIF	**Us**er **I**nter**f**ace

11 Order Numbers

The ASPC2 can be ordered locally from your Siemens contact person. Please use the following order numbers, dependent on the number of units:

ASIC ASPC2 (ASPC2R)	6ES7 195-0AA05-0XA0	Evaluation pack; 6pcs. -lead free-
	6ES7 195-0AA15-0XA0	Single tray; 66pcs. -lead free-
	6ES7 195-0AA25-0XA0	Tray box; 660pcs. -lead free-
	6ES7 195-0AA35-0XA0	7 tray box; 4620pcs. -lead free-

The internet- address is: http://www.siemens.com/comdec

Here you have to select ”PROFIBUS Manuals” to get the newest documentation about the PROFIBUS ASIC’s.

When you select “Home” you can subscribe to the Newsletter for free, which inform you about the newest changes.