μPD71051
Serial Control Unit

Description

The μPD71051 serial control unit is a CMOS USART designed to provide serial data communications in microcomputer systems. The CPU uses it as a peripheral I/O device and programs it to communicate in synchronous or asynchronous serial data transmission protocols, including IBM bisync.

The USART receives serial data streams and converts them into parallel data characters for the CPU. While receiving serial data, the USART can also accept parallel data from the CPU, convert it to serial, and transmit the data. The USART signals the CPU when it has received or transmitted a character and requires service. The CPU may read complete USART status data at any time.

Features

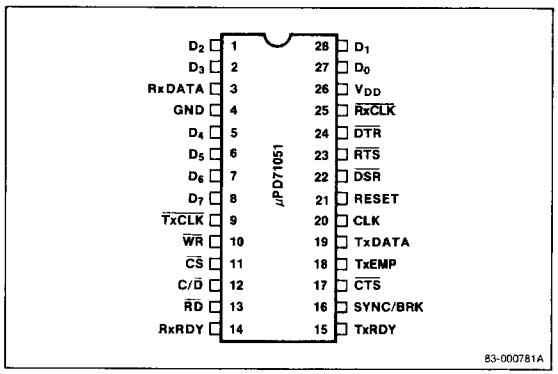
- ☐ Synchronous operation
  - One or two SYNC characters
  - Internal/external synchronization
  - Automatic SYNC character insertion
- ☐ Asynchronous operation
  - Clock rate: (baud rate)
  - x1, x16, or x64
  - Send stop bits: 1, 1.5, or 2 bits
  - Break transmission
  - Automatic break detection
  - Valid start bit detection
- ☐ Baud rate: DC - 240 kbit/s at x1 clock
- ☐ Full duplex, double-buffered transmitter/receiver
- ☐ Error detection: parity, overrun, and framing
- ☐ Five- to eight-bit characters
- ☐ Low-power standby mode
- ☐ Compatible with standard microcomputers
- ☐ Functionally equivalent to (except standby mode) and can replace the μPD8251AF
- ☐ CMOS technology
- ☐ Single +5 V ± 10% power supply
- ☐ Industrial temperature range −40 to +85 °C
- ☐ 28-pin plastic DIP or PLCC or 44-pin plastic QFP
- ☐ 8 MHz and 10 MHz

Ordering Information

Part Number	Clock (MHz)	Package
μPD71051C-8	8	28-pin plastic DIP
C-10	10	
GB-8	8	44-pin plastic QFP
GB-10	10	
L-8	8	28-pin PLCC
L-10	10	

Pin Configurations

28-Pin Plastic DIP

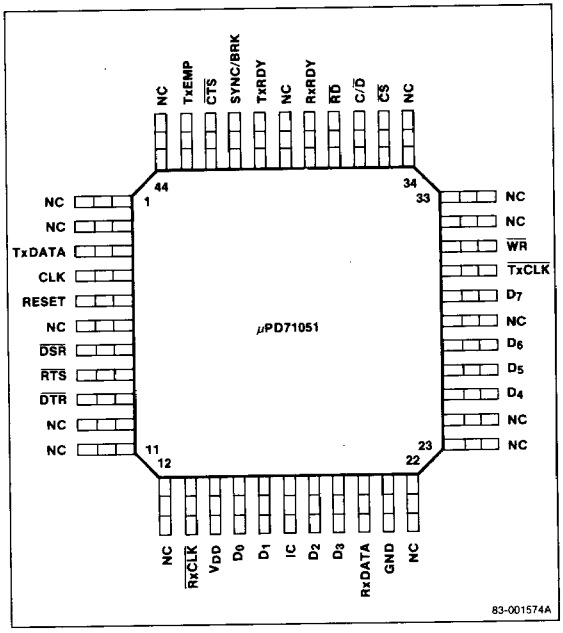


5c

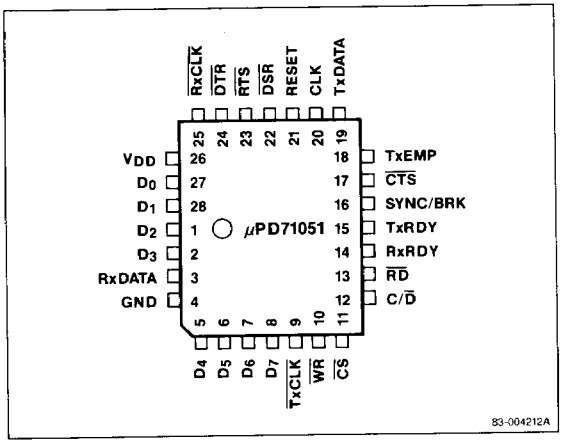
50011 (NECEL-000138)

Pin Configurations (cont)

44-Pin Plastic QFP



28-Pin Plastic Leaded Chip Carrier (PLCC)



Pin Identification

Symbol	Function
TxDATA	Transmit data output
CLK	Clock input
RESET	Reset input
$\overline{DSR}$	Data set ready input
$\overline{RTS}$	Request to send output
$\overline{DTR}$	Data terminal ready output
$\overline{RxCLK}$	Receiver clock input
V_{DD}	+5 V power supply
D_7-D_0	Data bus
IC	Internally connected (Do not connect any signal to an IC pin)
RxDATA	Receive data input
GND	Ground
$\overline{TxCLK}$	Transmitter clock input
$\overline{WR}$	Write strobe input
$\overline{CS}$	Chip select input
C/$\overline{D}$	Control or data input
$\overline{RD}$	Read strobe input
RxRDY	Receiver ready output
TxRDY	Transmitter ready output
SYNC/BRK	Synchronization/Break input/output
$\overline{CTS}$	Clear to send input
TxEMP	Transmitter empty output
NC	Not connected

Pin Functions

D_7-D_0 [Data Bus]

D_7-D_0 are an 8-bit, 3-state, bidirectional data bus. The bus transfers data by connecting to the CPU data bus.

RESET [Reset]

A high level to the RESET input resets the μPD71051 and puts it in an idle state. It performs no operations in the idle state. The μPD71051 enters standby mode when this signal falls from a high level to a low level. Standby mode is released when the CPU writes a mode byte to the μPD71051. The reset pulse width must be at least 6 t_{CYK} cycles and the clock must be enabled.

CLK [Clock]

This clock input produces internal timing for the μPD71051. The clock frequency should be at least 30 times the transmitter or receiver clock input frequency ($\overline{TxCLK}$, $\overline{RxCLK}$) in sync or async mode with the X1 clock. This assures stable operation. The clock frequency must be more than 4.5 times the $\overline{TxCLK}$ or $\overline{RxCLK}$ in async mode using x16 or x64 clock mode.

$\overline{CS}$ [Chip Select]

The $\overline{CS}$ input selects the μPD71051. The μPD71051 is selected by setting $\overline{CS} = 0$. When $\overline{CS} = 1$, the μPD71051 is not selected, the data bus (D_7-D_0) is in the high impedance state, and the $\overline{RD}$ and $\overline{WR}$ signals are ignored.

$\overline{RD}$ [Read Strobe]

The $\overline{RD}$ input is low when reading data or status information from the μPD71051.

$\overline{WR}$ [Write Strobe]

The $\overline{WR}$ input is low when writing data or a control byte to the μPD71051.

C/$\overline{D}$ [Control or Data]

The C/$\overline{D}$ input determines the data type when accessing the μPD71051. When C/$\overline{D}$ = 1, the data is a control byte (table 1) or status. When C/$\overline{D}$ = 0, the data is character data. This pin is normally connected to the least significant bit (A_0) of the CPU address bus.

Table 1. Control Signals and Operations

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	C/$\overline{D}$	μPD71051	CPU Operation
0	0	1	0	Receive data buffer ⇩ Data bus	Read receive data
0	0	1	1	Status register ⇩ Data bus	Read status
0	1	0	0	Data bus ⇩ Transmit data buffer	Write transmit data
0	1	0	1	Data bus ⇩ Control byte register	Write control byte
0	1	1	x	Data bus: High impedance	None
1	x	x	x	Data bus: High impedance	None

$\overline{DSR}$ [Data Set Ready]

$\overline{DSR}$ is a general-purpose input pin that can be used for modem control. The status of this pin can be determined by reading bit 7 of the status byte.

$\overline{DTR}$ [Data Terminal Ready]

$\overline{DTR}$ is a general-purpose output pin that can be used for modem control. The state of this pin can be controlled by writing bit 1 of the command byte. If bit 1 = 0, then $\overline{DTR} = 1$. If bit 1 = 1, then $\overline{DTR} = 0$.

$\overline{RTS}$ [Request to Send]

$\overline{RTS}$ is a general-purpose output pin that can be used for modem control. The status of this pin can be controlled by writing bit 5 of the command byte. If bit 5 = 1, then $\overline{RTS} = 0$. If bit 5 = 0, then $\overline{RTS} = 1$.

$\overline{CTS}$ [Clear to Send]

The $\overline{CTS}$ input controls data transmission. The μPD71051 is able to transmit serial data when $\overline{CTS} = 0$ and the command byte sets TxEN = 1. If $\overline{CTS}$ is set equal to 1 during transmission, the sending operation stops after sending all currently written data and the TxDATA pin goes high.

TxDATA [Transmit Data]

The μPD71051 sends serial data over the TxDATA output.

TxRDY [Transmitter Ready]

The TxRDY output tells the CPU that the transmit data buffer in the μPD71051 is empty; that is, that new transmit data can be written. This signal is masked by the TxEN bit of the command byte and by the $\overline{CTS}$ input. It can be used as an interrupt signal to request data from the CPU.

The status of TxRDY can be determined by reading bit 0 of the status byte. This allows the μPD71051 to be polled. Note that TxRDY of the status byte is not masked by $\overline{CTS}$ or TxEN.

TxRDY is cleared to 0 by the falling edge of $\overline{WR}$ when the CPU writes transmit data to the μPD71051. Data in the transmit data buffer that has not been sent is destroyed if transmit data is written while TxRDY = 0.

5c

TxEMP [Transmitter Empty]

The μPD71051 reduces CPU overhead by using a double buffer; the transmit data buffer (second buffer) and the transmit buffer (first buffer) in the transmitter. When the CPU writes transmit data to the transmit data buffer (second buffer), the μPD71051 sends data by transferring the contents of the second buffer to the first buffer, after transmitting the contents of the first buffer.

This empties the second buffer and TxRDY is set to 1. The TxEMP output becomes 1 when the contents of the first buffer are sent and the second buffer is empty. Thus, TxEMP = 1 shows that both buffers are empty. In half-duplex operation, you can determine when to change from sending to receiving by testing TxEMP = 1.

When TxEMP = 1 occurs in async mode, the TxDATA pin goes high. When the CPU writes transmit data, TxEMP is set to 0 and data transmission resumes.

When TxEMP = 1 occurs in sync mode, the μPD71051 loads SYNC characters from the SYNC character register and sends them through the TxDATA pin. TxEMP is set to 0 and resumes sending data after sending (one or two) SYNC characters and the CPU writes new transmit data to the μPD71051.

$\overline{\text{TxCLK}}$ [Transmitter Clock]

The $\overline{\text{TxCLK}}$ input is the reference clock input that determines the transmission rate. Data is transmitted at the same rate as $\overline{\text{TxCLK}}$ in sync mode. In async mode, set $\overline{\text{TxCLK}}$ to 1, 16, or 64 times the transmission rate. Serial data from TxDATA is sent at the falling edge of $\overline{\text{TxCLK}}$.

For example, a rate of 19200 baud in sync mode means that $\overline{\text{TxCLK}}$ is 19.2 kHz. A rate of 2400 baud in async mode can represent a $\overline{\text{TxCLK}}$ of:

x1 clock = 2.4 kHz
x16 clock = 38.4 kHz
x64 clock = 153.6 kHz

RxDATA [Receive Data]

The μPD71051 receives serial data through the RxDATA input.

RxRDY [Receiver Ready]

The RxRDY output becomes 1 when the μPD71051 receives one character of data and transfers that data to the receive data buffer; that is, when the receive data can be read. This signal can be used as an interrupt signal for a data read request to the CPU. You can determine the status of RxRDY by reading bit 1 of the status byte and use the μPD71051 in a polling application. RxRDY becomes 0 when the CPU reads the receive data.

Unless the CPU reads the receive data (after RxRDY = 1 is set) before the next single character is received and transferred to the receive buffer, an overrun error occurs, and the OVE status bit is set. The unread data in the receive data buffer is overwritten by newly transferred data and lost.

RxRDY is set to 0 in the receive disable state. This state is set by changing the RxEN bit to 0 through the command byte. After RxEN is set to 1 (making receiving possible), RxRDY becomes 1 whenever new characters are received and transferred to the receive data buffer.

SYNC/BRK [Synchronization/Break]

The SYNC pin detects synchronization characters in sync mode. The SYNC mode byte selects internal or external SYNC detection. The SYNC pin becomes an output when internal synchronization is set, and an input when external synchronization is set.

The SYNC output goes high when the μPD71051 detects a SYNC character in internal synchronization. When two SYNC characters are used, SYNC goes high when the last bit of the two consecutive SYNC characters is detected. You can read the status of the SYNC signal in bit 6 of the status byte. Both the SYNC pin and status are set to 0 by a read status operation.

In external synchronization, in order for the external circuit to detect synchronization, a high level of at least one period of $\overline{\text{RxCLK}}$ must be input to the SYNC pin. When the μPD71051 detects the high level, it begins to receive data, starting at the rising edge of the next $\overline{\text{RxCLK}}$. The high level input may be removed when synchronization is released.

The BRK output is used only in async mode and shows the detection of a break state. BRK goes high when a low level signal is input to the RxDATA pin for two character bit lengths (including the start, stop, and parity bits). As with SYNC, you can read the status of BRK in bit 6 of the status byte. BRK is not cleared by the read operation.

The set BRK signal is cleared when the RxDATA pin returns to high level, or when the μPD71051 is reset by hardware or software. The SYNC/BRK pin goes low on reset, regardless of previous mode. Figure 1 shows the break state and BRK signal.

$\overline{\text{RxCLK}}$ [Receiver Clock]

$\overline{\text{RxCLK}}$ is a reference clock input that controls the receive data rate. In sync mode, the receiving rate is the same as $\overline{\text{RxCLK}}$. In async mode, $\overline{\text{RxCLK}}$ can be 1, 16, or 64 times the receive rate. Serial data from RxDATA is input by the rising edge of $\overline{\text{RxCLK}}$.

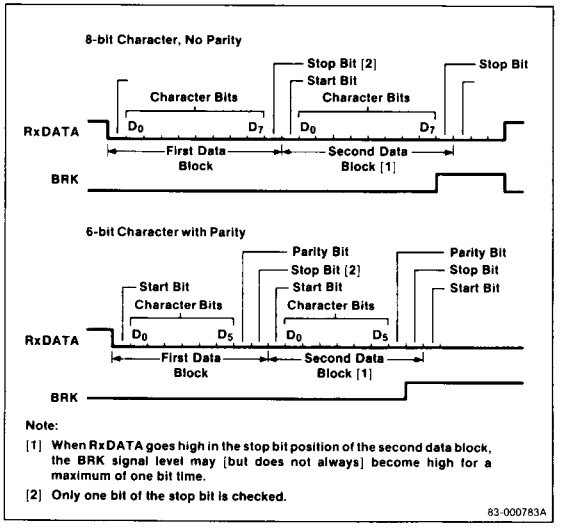
V_{DD} [Power]

+5 V power supply.

GND [Ground]

Ground.

Figure 1. Break Status and Break Signal



Note:

[1] When RxDATA goes high in the stop bit position of the second data block, the BRK signal level may [but does not always] become high for a maximum of one bit time.

[2] Only one bit of the stop bit is checked.

83-000783A

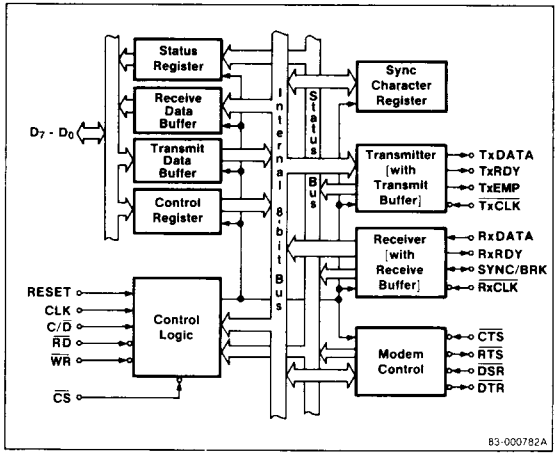
μPD71051 Functions

The μPD71051 is a CMOS serial control (USART) unit that provides serial communications in microcomputer systems. The CPU handles the μPD71051 as an ordinary I/O device.

The μPD71051 can operate in synchronous or asynchronous systems. In sync mode, the character bit length, number of sync characters, and sync detection mode must be designated. In async mode, the communication rate, character bit length, stop bit length, etc., must be designated. The parity bit may be designated in either mode.

The μPD71051 converts parallel data received from the CPU into serial transmitted data (from the TxDATA pin), and converts serial input data (from the RxDATA pin) into parallel data so that the CPU can read it (receiving operation).

Block Diagram



The CPU can read the current status of the μPD71051 and can process data after checking the status, after checking for transfer errors, and μPD71051 data buffer status.

The μPD71051 can be reset under hardware or software control to a standby mode that consumes less power and removes the device from system operation. In this mode, the μPD71051's previous operating mode is released and it waits for a mode byte to set the mode. The μPD71051 leaves standby mode and shifts to a designated operating mode when the CPU writes a mode byte to it.

5c

Status Register

The status register allows the CPU to read the status of the μPD71051 except in standby mode. This register indicates status and allows the CPU to manage data reading, writing, and error handling during operations.

Receive Data Buffer

When the receiver has converted the serial data input from the RxDATA pin into parallel data, the converted data is stored in the receive data buffer. The CPU can then read it. Data for one character entering the receive buffer is transferred to the receive data buffer and RxRDY becomes 1, requesting that the CPU read the data.

Transmit Data Buffer

The transmit data buffer holds the parallel data from the CPU that the transmitter will convert to serial data and output from the TxDATA pin. When the CPU writes transmit data to the μPD71051, the μPD71051 stores data in the transmit data buffer. The transmit data buffer transfers the data to the transmitter, which sends the data from the TxDATA pin.

Control Register

This register stores the mode and the command bytes.

Control Logic

The control logic sends control signals to the internal blocks and controls the operation of the μPD71051 based on internal and external signals.

Synchronous Character Register

This register stores one or two SYNC characters used in sync mode. During transmission, the SYNC characters stored in this register are output from the TxDATA pin when the CPU does not send a new character and TxEMP status is set. During receiving, synchronization is established when the characters received and the SYNC characters stored in this register are the same.

Transmitter

The contents of the transmit data buffer are transferred to the transmitter, converted from parallel to serial, and output from the TxDATA pin. The transmitter adds start, stop, and parity bits.

Receiver

The receiver converts serial data input from the RxDATA pin into parallel data and transfers the parallel data to the receive data buffer, allowing the CPU to read it.

The receiver detects SYNC characters and checks parity bits in sync mode. It detects the start and stop bits, and checks parity in the async mode.

In async mode, receiving does not begin (the start bit is not detected) until one effective stop bit (high level) is input to the RxDATA pin and Receive Enable (RxEN = 1) is set after setting up the mode.

Modem Control

This block controls the $\overline{CTS}$, $\overline{RTS}$, $\overline{DSR}$, and $\overline{DTR}$ modem interface pins. The $\overline{RTS}$, $\overline{DSR}$, and $\overline{DTR}$ pins can also be used as general-purpose I/O pins.

Absolute Maximum Ratings

$T_A = +25°C$

Power supply voltage, V_{DD}	−0.5 to +7.0 V
Input voltage, V_I	−0.5 to V_{DD} + 0.3 V
Output voltage, V_O	−0.5 to V_{DD} + 0.3 V
Operating temperature, T_{OPT}	−40°C to +85°C
Storage temperature, T_{STG}	−65°C to +150°C
Power dissipation, PD_{MAX}	1.0 W

Comment: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Capacitance

$T_A = +25°C$, $V_{DD} = 0$ V

Parameter	Symbol	Limits Min	Limits Max	Unit	Test Conditions
Input capacitance	C_I		10	pF	fc = 1MHz Unmeasured pins returned to 0 V
I/O capacitance	C_{IO}		20	pF	

DC Characteristics

$T_A = -40°C$ to $+85°C$, $V_{DD} = +5$ V ± 10%

Parameter	Symbol	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
Input voltage high	V_{IH}	2.2		V_{DD}+0.3	V	
Input voltage low	V_{IL}	−0.5		0.8	V	
Output voltage high	V_{OH}	0.7 x V_{DD}			V	$I_{OH} = -400\ \mu A$
Output voltage low	V_{OL}			0.4	V	$I_{OL} = 2.5$ mA
Input leakage current high	I_{LIH}			10	μA	$V_I = V_{DD}$
Input leakage current low	I_{LIL}			−10	μA	$V_I = 0$ V
Output leakage current high	I_{LOH}			10	μA	$V_O = V_{DD}$
Output leakage current low	I_{LOL}			−10	μA	$V_O = 0$ V
Supply current μPD71051	I_{DD1}			10	mA	Normal mode
	I_{DD2}		50	100	μA	Stand-by mode
μPD71051-10	I_{DD1}			10	mA	Normal mode
	I_{DD2}		2	50	μA	Stand-by mode

AC Characteristics

$T_A = -40°C$ to $+85°C$, $V_{DD} = 5\ V \pm 10\%$

Parameter	Symbol	8 MHz Limits Min	8 MHz Limits Max	10 MHz Limits Min	10 MHz Limits Max	Unit	Test Conditions
Read Cycle							
Address setup to $\overline{RD}\downarrow$	t_{SAR}	0		0		ns	$\overline{CS}$, $C/\overline{D}$
Address hold from $\overline{RD}\uparrow$	t_{HRA}	0		0		ns	$\overline{CS}$, $C/\overline{D}$
$\overline{RD}$ low level width	t_{RRL}	150		95		ns	
Data delay from $\overline{RD}\downarrow$	t_{DRD}		120		85	ns	$C_L = 150$ pF
Data float from $\overline{RD}\uparrow$	t_{FRD}	10	80	10	65	ns	
Port ($\overline{DSR}$, $\overline{CTS}$) set-up to $\overline{RD}\downarrow$	t_{SPR}	20		20		t_{CYK}	
Write Cycle							
Address setup to $\overline{WR}\downarrow$	t_{SAW}	0		0		ns	$\overline{CS}$, $C/\overline{D}$
Address hold from $\overline{WR}\uparrow$	t_{HWA}	0		0		ns	$\overline{CS}$, $C/\overline{D}$
$\overline{WR}$ low level width	t_{WWL}	150		95		ns	
Data setup to $\overline{WR}\uparrow$	t_{SDW}	80		80		ns	
Data hold from $\overline{WR}\uparrow$	t_{HWD}	0		0		t_{CYK}	
Port ($\overline{DTR}$, $\overline{RTS}$), delay from $\overline{WR}\uparrow$	t_{DWP}		8		8	t_{CYK}	
Write recovery time	t_{RV}	6		6		t_{CYK}	Mode initialize
		8		8		t_{CYK}	Async mode
		16		16		t_{CYK}	Sync mode
Serial Transfer Timing							
CLK cycle time	t_{CYK}	125	DC	100	DC	ns	
CLK high level width	t_{KKH}	50		35		ns	
CLK low level width	t_{KKL}	35		25		ns	
CLK rise time	t_{KR}	5	20	5	20	ns	
CLK fall time	t_{KF}	5	20	5	20	ns	
TxDATA delay from $\overline{TxCLK}$	t_{DTKTD}		0.5		0.5	μs	
Transmitter input clock pulse width low level	t_{TKTKL}	12		12		t_{CYK}	1xBR (Note 1)
		1		1		t_{CYK}	16x, 64xBR
Transmitter input clock pulse width high level	t_{TKTKH}	15		15		t_{CYK}	1xBR
		3		3		t_{CYK}	16x, 64xBR
Transmitter input clock frequency	f_{TK} (Note 2)	DC	240	DC	300	kHZ	1xBR
		DC	1536	DC	1920	kHz	16xBR
		DC	1536	DC	1920	kHz	64xBR
Receiver input clock pulse width low level	t_{RKRKL}	12		12		t_{CYK}	1xBR
		1		1		t_{CYK}	16x, 64xBR
Receiver input clock pulse width high level	t_{RKRKH}	15		15		t_{CYK}	1xBR
		3		3		t_{CYK}	16x, 64xBR

5c

AC Characteristics (cont)

Parameter	Symbol	8 MHz Limits Min	8 MHz Limits Max	10 MHz Limits Min	10 MHz Limits Max	Unit	Test Conditions
Serial Transfer Timing (cont)							
Receiver input clock frequency	f_{RK} (Note 2)	DC	240	DC	300	kHz	1xBR
		DC	1536	DC	1920	kHz	16xBR
		DC	1536	DC	1920	kHz	64xBR
RxDATA set-up to Sampling pulse	t_{SRDSP}	1		1		μs	
RxDATA hold from sampling pulse	t_{HSPRD}	1		1		μs	
TxEMP delay time (TxDATA)	t_{DTXEP}		20		20	t_{CYK}	
TxRDY delay time (TxRDY↑)	t_{DTXR}		8		8	t_{CYK}	
TxRDY delay time (TxRDY↓)	t_{DWTXR}		200		100	ns	
RxRDY delay time (RxRDY↑)	t_{DRXR}		26		26	t_{CYK}	
RxRDY delay time (RxRDY↓)	t_{DRRXR}		200		100	ns	
SYNC output delay time (for internal sync)	t_{DRKSY}		26		26	t_{CYK}	
SYNC input set-up time (for external sync)	t_{SSYRK}	18		18		t_{CYK}	
RESET pulse width		6		6		t_{CYK}	

Notes:

(1) BR = Baud rate

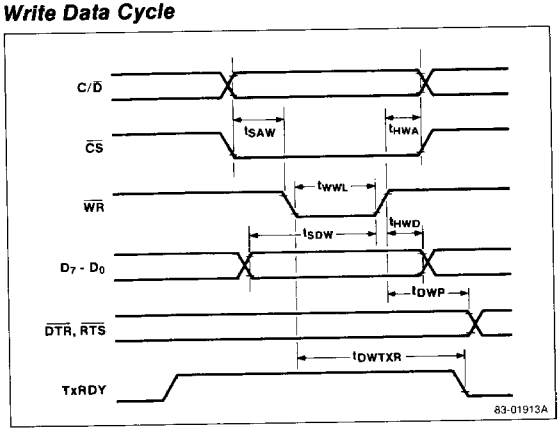
(2) 1xBR: f_{TK} or $f_{RK} \leq 1/30\, t_{CLK}$, 16x, 64xBR: f_{TK} or $f_{RK} \leq 1/4.5\, t_{CLK}$

(3) System CLK is needed during reset operation

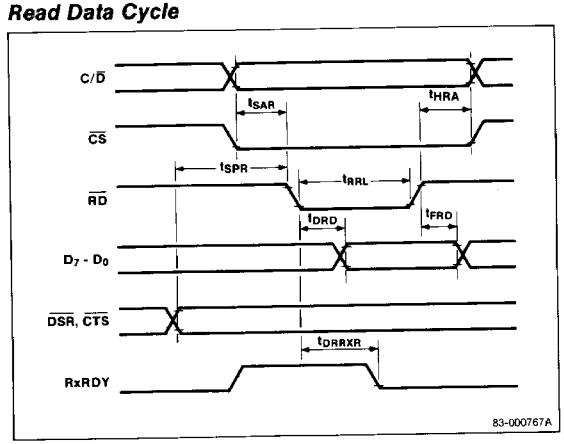
(4) Status update can have a maximum delay of 28 t_{CYK} from the event effecting the status.

Timing Waveforms

Write Data Cycle

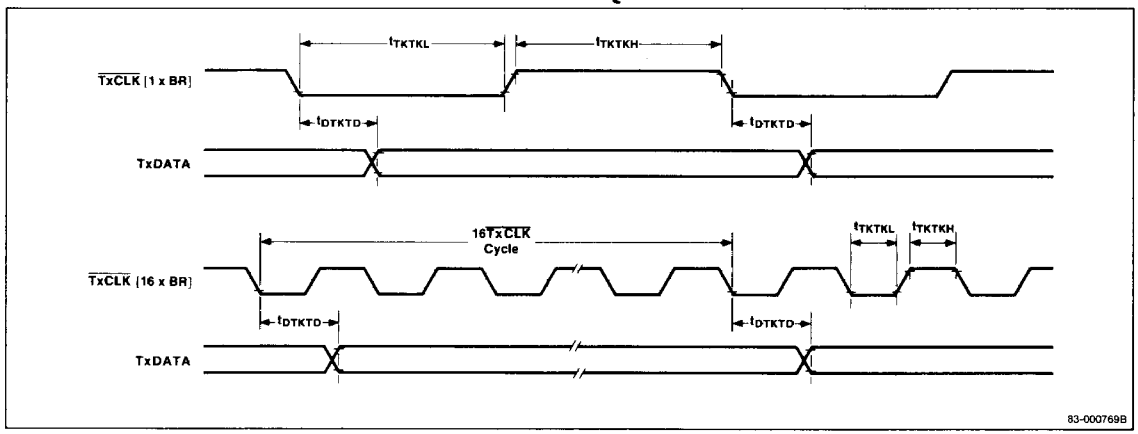


Read Data Cycle

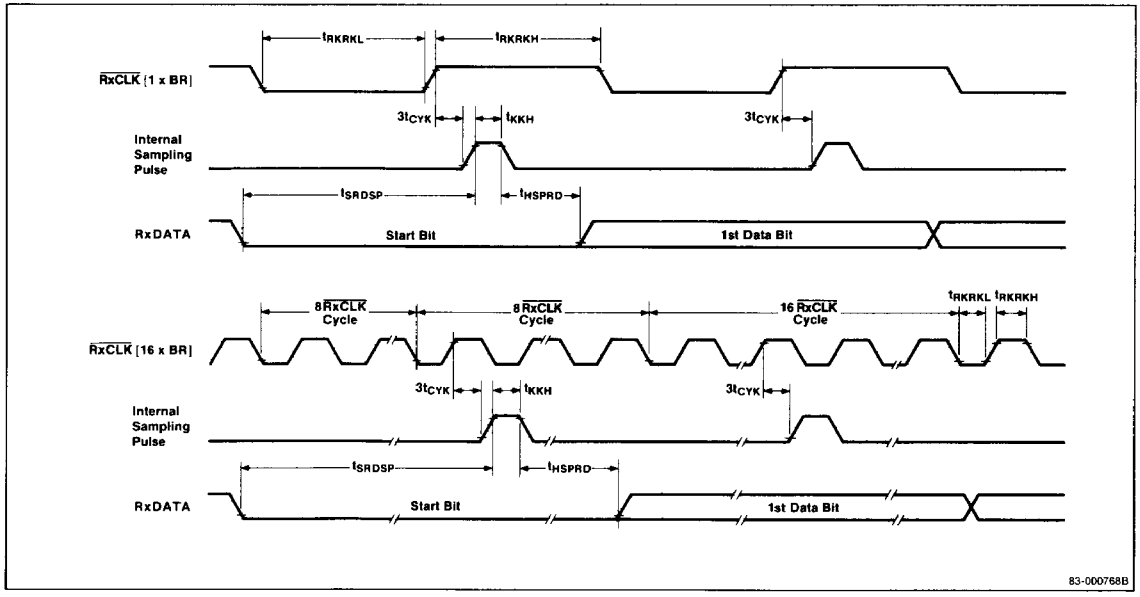


Timing Waveforms (cont)

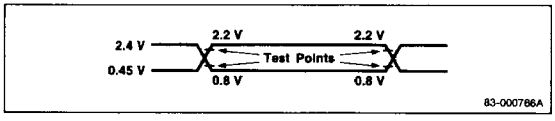
Transmitter Clock and TxDATA



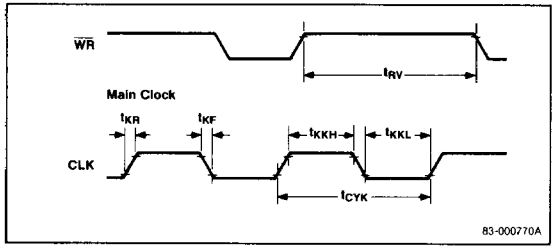
Receiver Clock and RxDATA Timing



AC Test Input



Write Recovery Time



5c

Connecting the μPD71051 to the System

The CPU uses the μPD71051 as an I/O device by allocating two I/O addresses, set by the value of C/$\overline{D}$. One I/O address is allocated when the level of C/$\overline{D}$ is low and becomes a port to the transmit and receive data register. The other I/O address is allocated when C/$\overline{D}$ is high and becomes a port to the mode, command, and status registers. Generally, the least significant bit (A_0) of the CPU address bus is connected to C/$\overline{D}$ to get a continuous I/O address. This is shown in figure 2.

Pins TxRDY and RxRDY are connected to the CPU or, when interrupts are used, to the interrupt pin of the interrupt controller.

Operating the μPD71051

Start with a hardware reset (set the RESET pin high) after powering on the μPD71051. This puts the μPD71051 into standby mode and it waits for a mode byte. In async mode, the μPD71051 is ready for a command byte after the mode byte; the mode byte sets the communication protocol to the async mode. In sync mode, the μPD71051 waits for one or two SYNC characters to be sent after the mode byte; set C/$\overline{D}$ = 1. A command byte may be sent after the SYNC characters are written. Figure 3 shows this operation sequence.

In both modes, it is possible to write transmit data, read receive data, read status, and write more command bytes after the first command byte is written. The μPD71051 performs a reset, enters standby mode, and returns to a state where it waits for a mode byte when the command byte performs a software reset.

Figure 2. System Connection

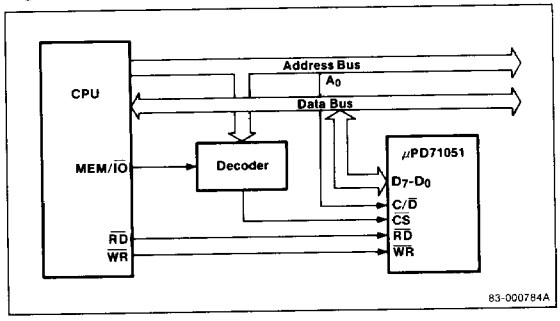
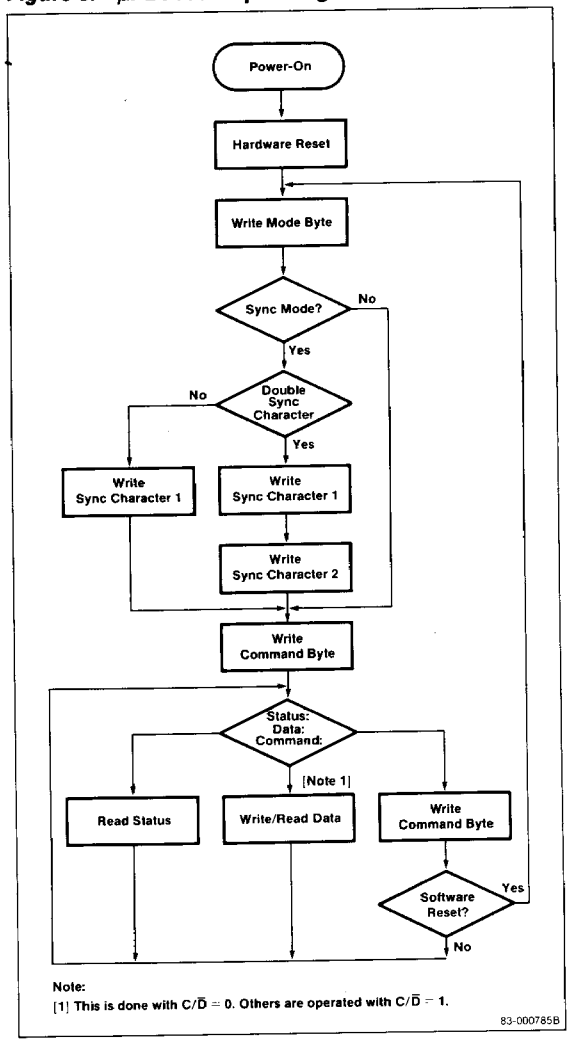


Figure 3. μPD71051 Operating Procedure



Mode Register

When the μPD71051 is in standby mode, writing a mode byte to it will release standby mode. Figure 4 shows the mode byte format for designating async mode. Figure 5 shows the mode byte format for designating sync mode. Bits 0 and 1 must be 00 to designate sync mode. Async mode is designated by all other combinations of bits 0 and 1.

The P1, P0 and L1, L0 bits are common to both modes. Bits P1 and P0 (parity) control the generation and checking (sending and receiving) functions. These parity bit functions do not operate when P0 = 0. When P1, P0 = 01, the μPD71051 generates and checks odd parity. When P1, P0 = 11, it generates and checks even parity.

Bits L1 and L0 set the number of bits per character (n). Additional bits such as parity bits are not included in this number. Given n bits, the μPD71051 receives the lower n bits of the 8-bit data written by the CPU. The upper bits (8 -n) of data that the CPU reads from the μPD71051 are set to zero.

The ST1, ST0 and B1, B0 bits are used in async mode. The ST1 and ST0 bits determine the number of stop bits added by the μPD71051 during transmission.

The B1 and B0 bits determine the relationship between the baud rates for sending and receiving, and the clocks $\overline{TxCLK}$ and $\overline{RxCLK}$. B1 and B0 select a multiplication rate of 1, 16, or 64 for the frequency of the sending and receiving clock relative to the baud rate. Multiplication by 1 is not normally used in async mode. Note that the data and clock must be synchronized on the sending and receiving sides when multiplication by 1 is used.

The SSC and EXSYNC bits are used in sync mode. The SSC bit determines the number of SYNC characters. SSC = 1 designates one SYNC character. SSC = 0 designates two SYNC characters. The number of SYNC characters determined by the SSC bit are written to the μPD71051 immediately after writing the mode byte.

The EXSYNC bit determines whether sync detection during receiving operations is internal or external. EXSYNC = 1 selects external sync detection and EXSYNC = 0 selects internal sync detection.

Figure 4. Mode Byte for Setting Asynchronous Mode

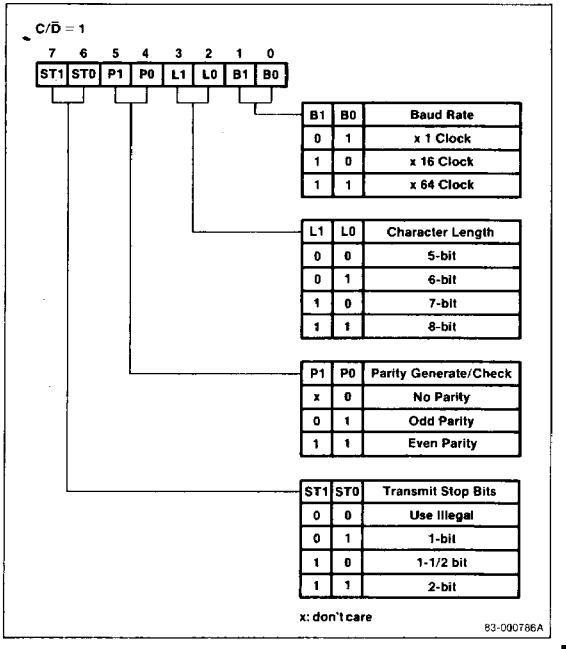
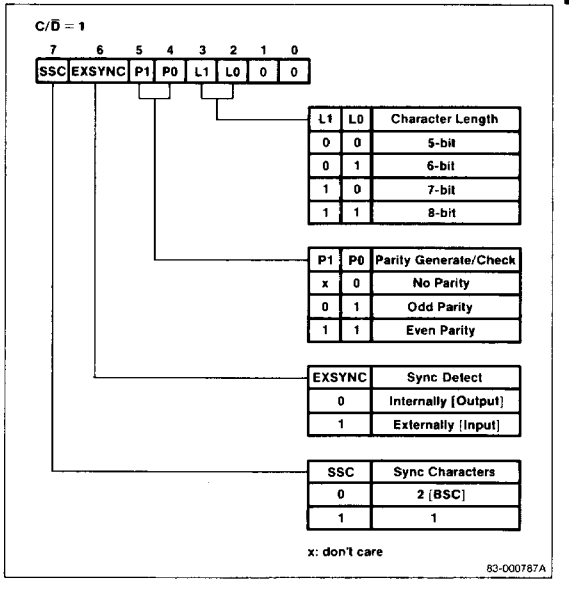


Figure 5. Mode byte for Setting Synchronous Mode



5c

Command Register

Commands are issued to the μPD71051 by the CPU by command bytes that control the sending and receiving operations of the μPD71051. A command byte is sent after the mode byte (in sync mode, a command byte may only be sent after writing SYNC characters) and the CPU must set C/$\overline{D}$ = 1. Figure 6 shows the command byte format.

Bit EH is set to 1 when entering hunt phase to synchronize in sync mode. Bit RxEN should also be set to 1 at that time. Data reception begins when SYNC characters are detected and synchronization is achieved, thus releasing hunt phase.

When bit SRES is set to 1, a software reset is executed, and the μPD71051 goes into standby mode and waits for a mode byte.

Bit RTS controls the $\overline{RTS}$ output pin. $\overline{RTS}$ is low when the RTS bit = 1 , and goes high when RTS = 0.

Setting bit ECL to 1 clears the error flags (PE, OVE, and FE) in the status register. Set ECL to 1 when entering the hunt phase or enabling the receiver.

Bit SBRK sends a break. When SBRK = 1, the data currently being sent is destroyed and the TxDATA pin goes low. Set SBRK = 0 to release a break. Break also works when TxEN = 0 (send disable).

Bit RxEN enables and disables the receiver. RxEN = 1 enables the receiver and RxEN = 0 disables the receiver. Synchronization is lost if RxEN = 0 during sync mode.

Bit DTR controls the $\overline{DTR}$ output pin. $\overline{DTR}$ goes low when the DTR bit = 1 and goes high when the DTR bit = 0.

The TxEN bit enables and disables the transmitter. TxEN = 1 enables the transmitter and TxEN = 0 disables the transmitter. When TxEN = 0, sending stops and the TxDATA pin goes high (mark status) after all the currently written data is sent.

Status Register

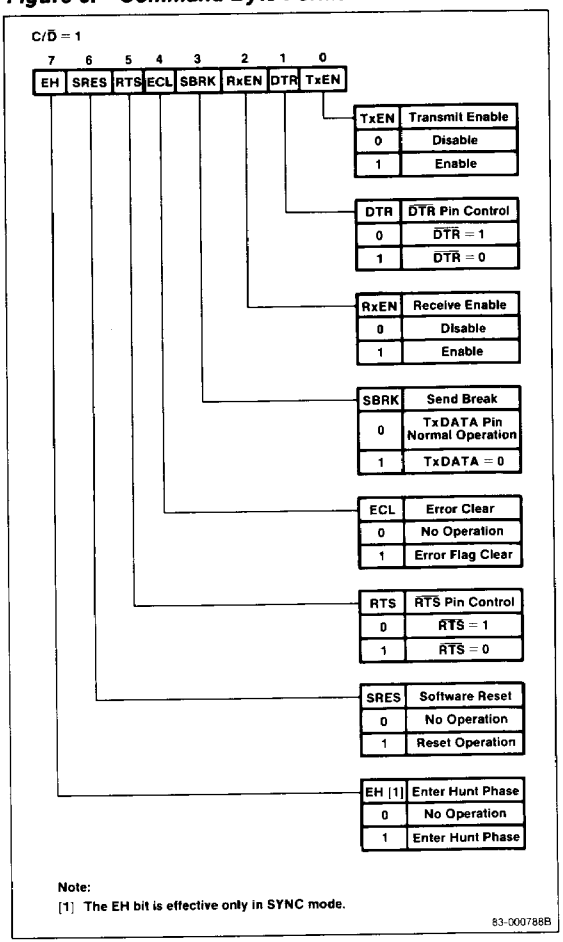
The CPU can read the status of the μPD71051 at any time except when the μPD71051 is in standby mode. Status can be read after setting C/$\overline{D}$ = 1 and $\overline{RD}$ = 0. Status is not updated while being read. Status updating is delayed at least 28 clock periods after an event that affects the status. Figure 7 shows the format of the status register.

The TxEMP and RxRDY bits have the same meaning as the pins of the same name. The SYNC/BRK bit generally has the same meaning as the SYNC/BRK pin. In external synchronization mode, the status of this bit does not always coincide with the pin. In this case, the SYNC pin becomes an input and the status bit goes to 1 when a rising edge is detected at the input. The status bit remains at 1 until it is read, even when the input level at the SYNC pin goes low. The status bit becomes 1 when a SYNC character is input with the RxDATA input, even when the pin is at a low level.

The DSR bit shows the status of the $\overline{DSR}$ input pin. The status bit is 1 when the $\overline{DSR}$ pin is low.

The FE bit (framing error) becomes 1 when less than one stop bit is detected at the end of each data block during asynchronous receiving. Figure 8 shows how a framing error can happen.

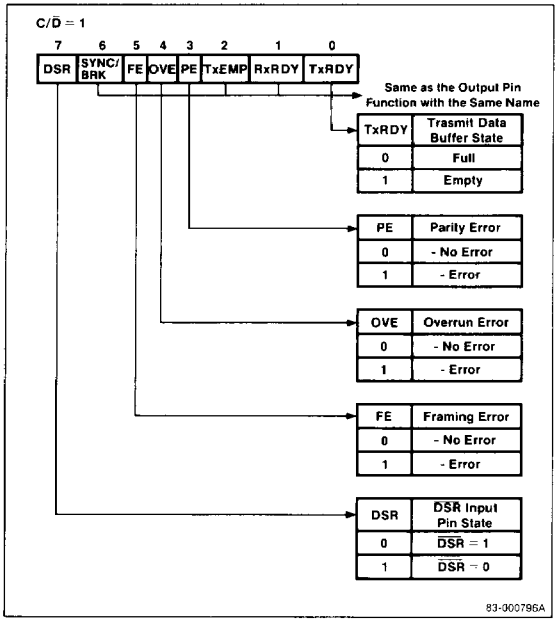
Figure 6. Command Byte Format



The OVE bit (overrun error) becomes 1 when the CPU delays reading the received data and two new data bytes have been received. In this case, the first data byte received is overwritten and lost in the receive data buffer. Figure 9 shows how an overrun can happen.

The PE bit (parity error) becomes 1 when a parity error occurs in a receive state.

Figure 7. Status Register Format



Framing, overrun, and parity errors do not disable the μPD71051's operations. All three error flags are cleared to 0 by a command byte that sets the ECL bit to 1.

The TxRDY bit becomes 1 when the transmit data buffer is empty. The TxRDY output pin becomes 1 when the transmit data buffer is empty, the $\overline{CTS}$ pin is low, and TxEN = 1. That is, bit TxRDY = Transmit Data Buffer Empty, pin TxRDY = (Transmit Data Buffer Empty)•($\overline{CTS}$ = 0)•(TxEN = 1).

Figure 8. Framing Error

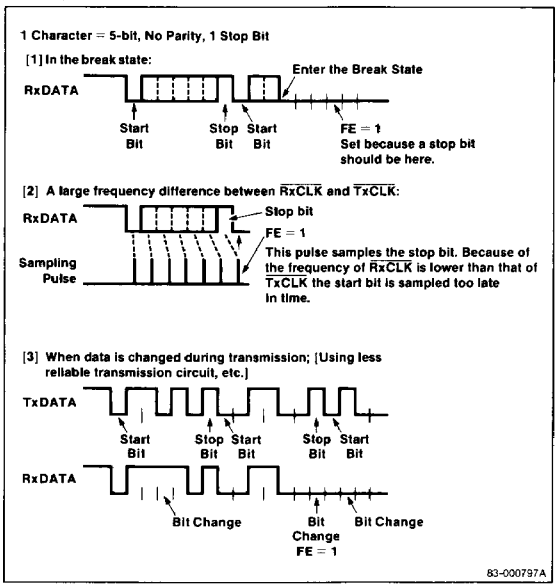
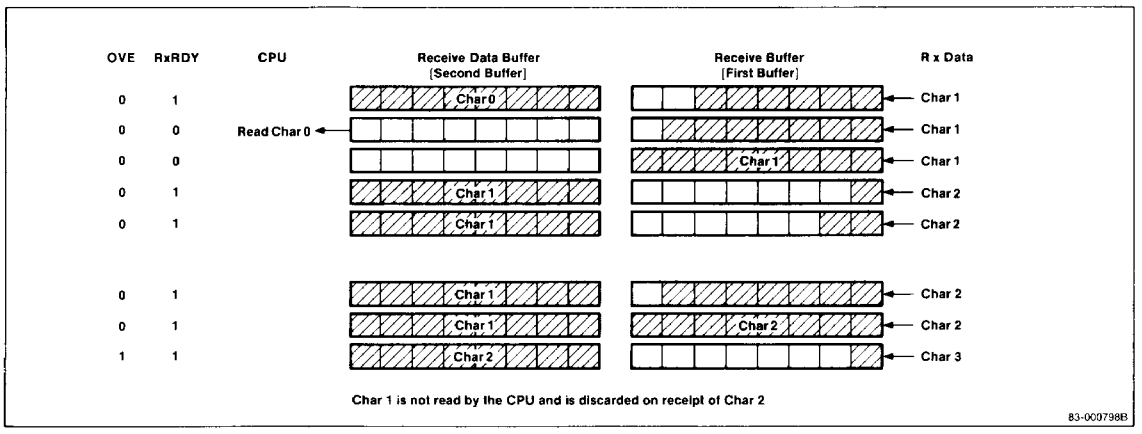


Figure 9. Overrun Error



5c

Sending in Asynchronous Mode

The TxDATA pin is typically in the high state (marking) when data is not being sent. When the CPU writes transmit data to the μPD71051, the μPD71051 transfers the transmit data from the transmit data buffer to the send buffer and sends the data from the TxDATA pin after adding one start bit (low level) and a programmed stop bit. If parity is used, a parity bit is inserted between the character and the stop bit. Figure 10 shows the data format for async mode characters. Serial data is sent by the falling edge of the signal that divided $\overline{\text{TxCLK}}$ (1/1, 1/16, or 1/64).

When bit SBRK is set to 1, the TxDATA pin goes low (break status), regardless of whether data is being sent. Figure 11 is a fragment of a typical program to send data in the async mode. Figure 12 shows the output from pin TxDATA.

Figure 10. Asynchronous Mode Data Format

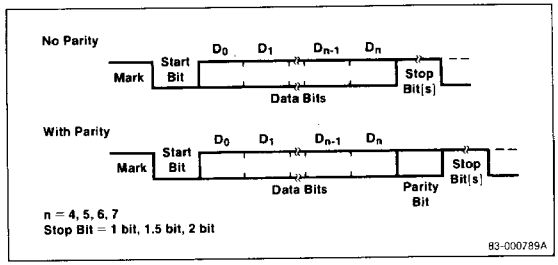
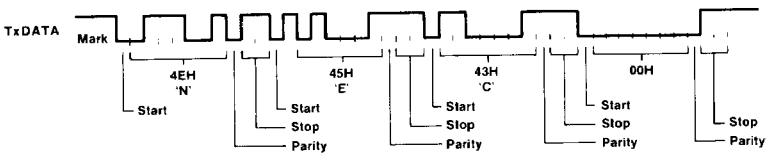


Figure 11. Asynchronous Transmitter Example

```
ASYNTX :    CALL    ASYNMOD              ;Set async mode
            MOV     AL, 00010001B        ;Command: clear error flag, transmit enable
            OUT     PCTRL,AL
            MOV     BW, OFFSET TXDADR    ;Transmit data area
TXSTART :   IN      AL, PCTRL
            TEST1   AL, 0                ;Read status
            BNE     TXSTART              ;Wait until TxRDY = 1
            MOV     AL, [BW]             ;Write transmit data
            OUT     PDATA, AL
            INC     BW                   ;Set next data address
            CMP     AL, 00H
            BNE     TXSTART              ;End if data = 0
            RET
TXDADR      DB      'NEC'                ;Transmit data 4EH, 45H, 43H, 00
            DB      0
ASYNMOD :   MOV     AL, 0                ;Writes control bytes three times
            OUT     PCTRL, AL            ;with 00H to unconditionally
            OUT     PCTRL, AL            ;accept the new command byte
            OUT     PCTRL, AL
            MOV     AL, 01000000B        ;Software reset
            OUT     PCTRL, AL
            MOV     AL, 11111010B        ;Write mode byte
            OUT     PCTRL, AL            ;Stop bit = 2 bits, even parity
            RET                          ;7 bits/character, x16 clock
```

Figure 12. TxDATA Pin Output



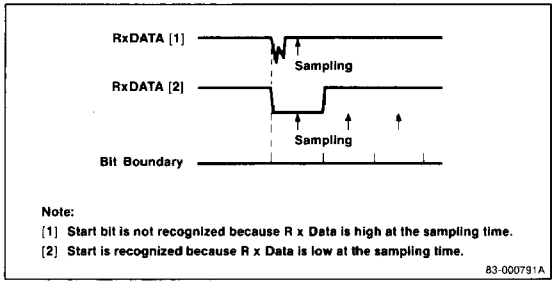
Receiving in Asynchronous Mode

The RxDATA pin is normally in the high state when data is not being received, as shown in figure 13. The μPD71051 detects the falling edge of a low level signal when a low level signal enters it.

The μPD71051 samples the level of the RxDATA input (only when x16 or x64 clock is selected) in a position 1/2 bit time after the falling edge of the RxDATA input to check whether this low level is a valid start bit. It is considered a valid start bit if a low level is detected at that time. If a low level is not detected, it is not regarded as a start bit and the μPD71051 continues testing for a valid start bit.

When a start bit is detected, the sampling points of the data bits, parity bit (when used), and stop bit are decided by a bit counter. The sampling is performed by the rising edge of the $\overline{\text{RxCLK}}$ when an X1 clock is used. When a x16 or x64 clock is used, it is sampled at the nominal middle of $\overline{\text{RxCLK}}$.

Figure 13. Start Bit Detection



Data for one character entering the receive buffer is transferred to the receive data buffer and causes RxRDY = 1, requesting that the CPU read the data. When the CPU reads the data, RxRDY becomes 0.

When a valid stop bit is detected, the μPD71051 waits for the start bit of the next data. If a low level is detected in the stop bit, a framing error flag is set; however, the receiving operation continues as if the correct high level had been detected. A parity error flag is set if a parity error is detected. An overrun error flag is set when the CPU does not read the data in time, and the next receiving data is transferred to the receive data buffer, overwriting the unread data. The μPD71051's sending and receiving operations are not affected by these errors.

If a low level is input to the RxDATA pin for more than two data blocks during a receive operation, the μPD71051 considers it a break state and the SYNC/BRK pin status becomes 1.

In async mode, the start bit is not detected until a high level of more than one bit is input to the RxDATA pin and the receiver is enabled. Figure 14 is a fragment of a typical program to receive the data sent in the previous async transmit example.

5c

Figure 14. Asynchronous Receiver Example

```
ASYNRX :    CALL    ASYNMOD                 ;Set ASYNC mode
            MOV     AL, 00010100B           ;Command: clear error flag, receive
                                            ;enable
            OUT     PCTRL,AL
            MOV     BW, OFFSET RXDADR       ;Data store area
RXSTART :   IN      AL, PCTRL
            TEST1   AL, 1                   ;Read status
            BNE     RXSTART                 ;Wait until RxRDY = 1
            IN      AL, PDATA               ;Read and store the receive data
            MOV     [BW], AL
            INC     BW                      ;Set next store address
            CMP     AL, 00H                 ;End if data = 0
            BNE     RXSTART
            RET
RXDADR      DB      256 DUP                 ;Reserve receive data area
```

Sending in Synchronous Mode

Following the establishment of sync mode and the enabling of the transmitter, the TxDATA pin stays high until the CPU writes the first character (normally, SYNC characters). When data is written, the TxDATA pin sends one bit for each falling edge of $\overline{TxCLK}$ if the $\overline{CTS}$ pin is low. Unlike async mode, start and stop bits are not used. However, a parity bit may be set. Figure 15 shows these data formats.

Once sending begins, the CPU must write data to the μPD71051 at the same rate as that of $\overline{TxCLK}$. If TxEMP goes to 1 because of a delay in writing by the CPU, the μPD71051 sends SYNC characters until the CPU writes data. TxEMP goes to 0 when data is written, and the data is sent as soon as transmission of SYNC characters stops.

Figure 15. Synchronous Mode Data Format

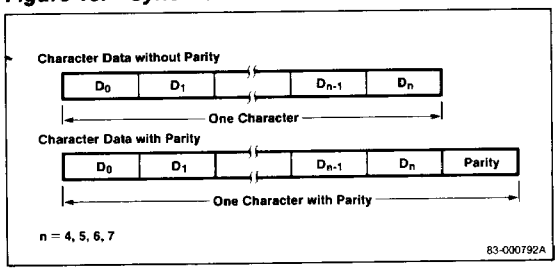


Figure 16. Synchronous Mode Transmit Timing

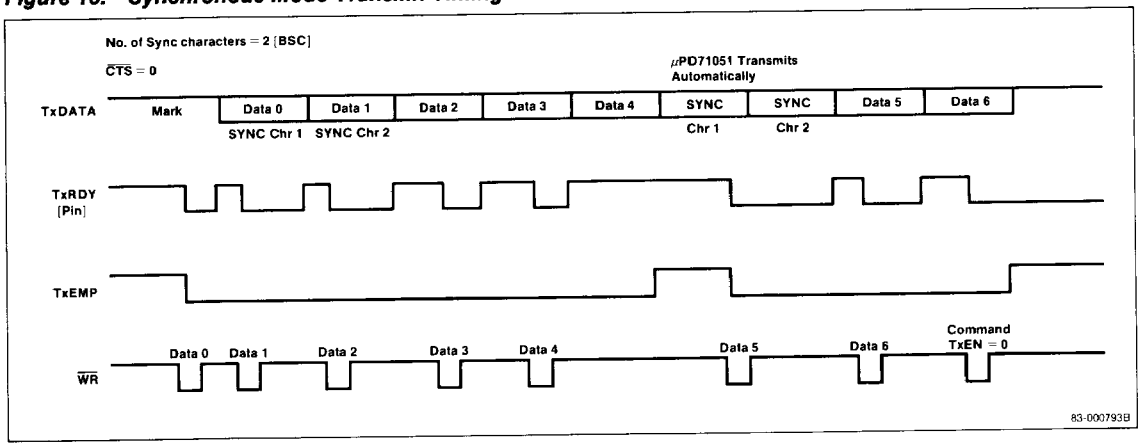
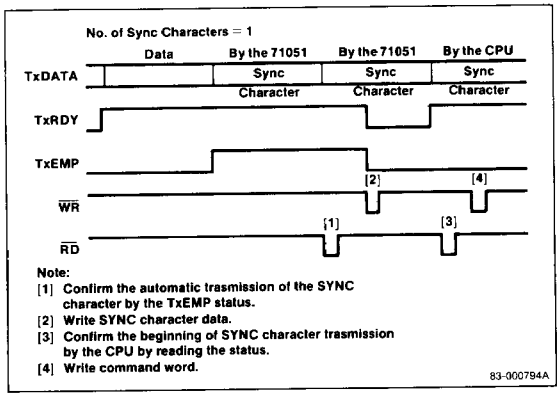


Figure 17. Issuing a Command During SYNC Character Transmission



Automatic transmission of SYNC characters begins after the CPU sends new data. SYNC characters are not automatically sent by enabling the transmitter. Figure 16 shows these timing sequences.

If a command is sent to the μPD71051 while SYNC characters are automatically being sent and TxEMP = 1, the μPD71051 may interpret the command as a data byte and transmit it as data. If a command must be sent under these conditions, the CPU should send a SYNC character to the μPD71051 and send the command while the SYNC character is being transmitted. This is shown in figure 17.

Figure 18 is a fragment of a typical program for sending in sync mode.

Figure 18. Synchronous Transmitter Example

```
SYNTX :   CALL   SYNMOD              ;Set sync mode
          MOV    AL, 00010001B       ;Command; clear error
          OUT    PCTRL, AL           ;flags, transmit enable
          MOV    BW, OFFSET TXDADR   ;Start location of data area TxDADR
          MOV    CL, LDLEN           ;Set number of bytes (LDLEN) to be transmitted
          MOV    CH, 00H
TXLEN :   IN     AL, PCTRL           ;Transmit the length byte
          TEST1  AL, 0
          BZ     TXLEN
          MOV    AL, LDLEN
          OUT    PDATA, AL
TXDATA :  IN     AL, PCTRL
          TEST1  AL, 0
          BZ     TXDATA              ;Transmit the number of
          MOV    AL, (BW)            ;bytes specified by LDLEN
          OUT    PDATA, AL
          INC    BW
          DBNZ   TXDATA
          MOV    AL, 00010000B       ;Command; clear error
          OUT    PCTRL, AL           ;flags, transmit disable
          RET
SYNC1     DB     ?                   ;SYNC character 1
SYNC2     DB     ?                   ;SYNC character 2
LDLEN     DB     ?                   ;transmit data count
TXDADR    DB     255 DUP (?)         ;transmit data
SYNMOD :  MOV    AL, 00H
          OUT    PCTRL, AL           ;Write control bytes
          OUT    PCTRL, AL           ;three times with 00H to
          OUT    PCTRL, AL           ;unconditionally accept the new
                                     ;command byte
          MOV    AL, 01000000B       ;Software reset
          OUT    PCTRL, AL
          MOV    AL, 00111100B       ;Write mode byte: 2 SYNC
          OUT    PCTRL, AL           ;characters, internal sync detect,
                                     ;even parity, 8 bits/character
          MOV    AL, SYNC1
          OUT    PCTRL, AL           ;Write SYNC characters
          MOV    AL, SYNC2
          OUT    PCTRL, AL
          RET
```

5c

Receiving in Synchronous Mode

In order to receive in sync mode, synchronization must be established with the sending side. The first command after setting sync mode and writing the SYNC character must be EH = 1, ECL = 1, and RxEN = 1. When hunt phase is entered all the bits in the receive buffer are set to 1. In internal synchronization, data on the RxDATA pin is input to the receive buffer for each rising edge of $\overline{\text{RxCLK}}$ and is compared with the SYNC character at the same time. Figure 19 shows this internal sync detection.

When the receive buffer and the SYNC character coincide, and parity is not used, the μPD71051 ends hunt phase and SYNC is set to 1 in the center of the last SYNC bit. When parity is used, SYNC becomes 1 in the center of the parity bit. Receiving starts with the bit which follows the bit when SYNC is set to 1.

In external sync detection, synchronization is achieved by setting the SYNC pin high from an external circuit for at least one period of $\overline{\text{RxCLK}}$. Hunt phase ends, and data reception can start. At this time, the SYNC status bit becomes 1, and goes to 0 when the status is read. The SYNC status bit is set to 1 when the SYNC input has a rising edge followed by a high level of more than one period of $\overline{\text{RxCLK}}$, even after synchronization is achieved.

The μPD71051 can regain lost synchronization anytime by issuing an enter hunt phase command.

After synchronization, the SYNC character is compared with each character regardless of whether internal or external synchronization is used. When the characters coincide, SYNC becomes 1, indicating that a SYNC character has been received. SYNC (SYNC status bit only in external detection) becomes 0 when the status is read.

Overrun and parity errors are checked the same way as in async mode, affecting only the status flag. Parity checking is not performed in the hunt phase. Figure 20 is a fragment of a typical program that receives the data sent by the previous sync transmit program example. Note that the frequencies of $\overline{\text{TxCLK}}$ on the transmitter and $\overline{\text{RxCLK}}$ on the receiver must be the same.

Figure 19. Internal Sync Detection Example

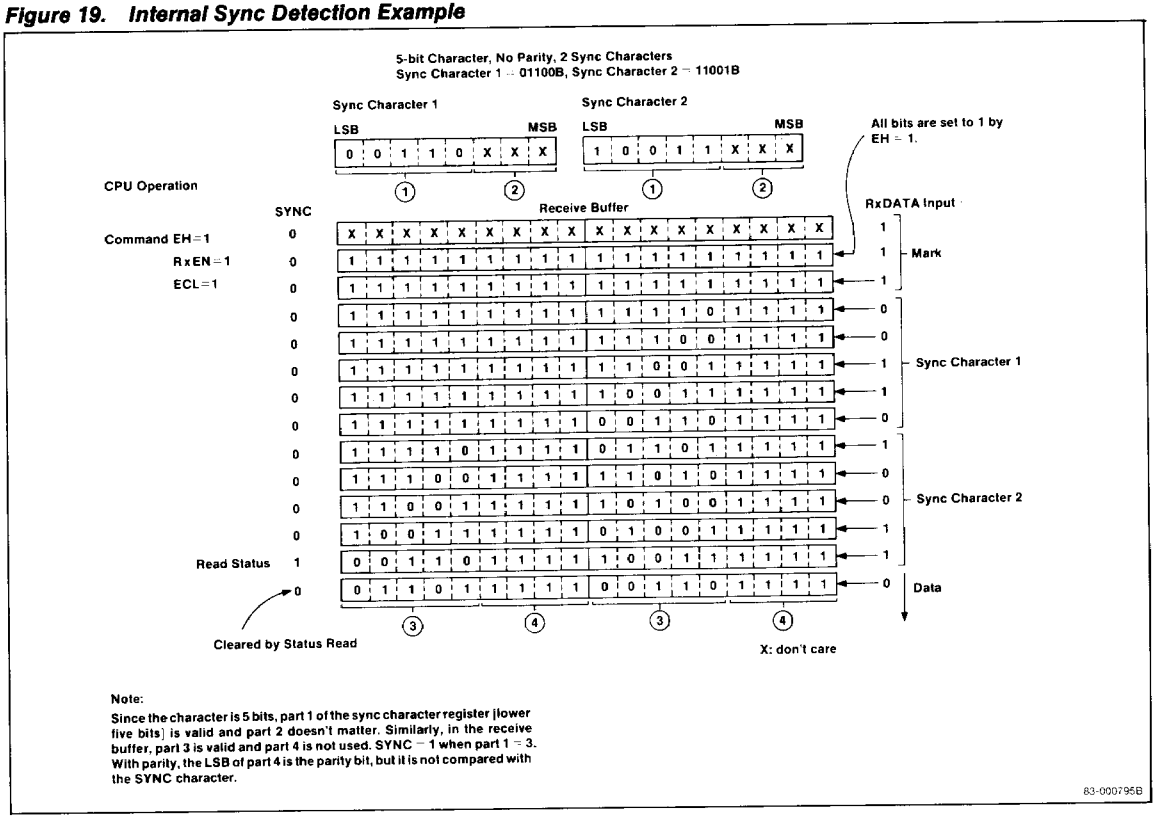


Figure 20. Synchronous Receiver Example

```
SYNRX :   CALL   SYNMOD               ;Set sync mode
          MOV    AL, 10010100B        ;Command: enter hunt
          OUT    PCTRL, AL            ;phase, clear error flags, receive enable
          MOV    BW, OFFSET RXDADR    ;Set receive data store address
RXLEN :   IN     AL, PCTRL
          TEST1  AL, 1
          BZ     RXLEN                ;Receive the number of
          IN     AL, DATA             ;receive data
          MOV    STLEN, AL            ;Set the number of
          MOV    CL, AL               ;receive data to both variable and
                                      ;counter
          MOV    CH, 00H
RXDATA :  IN     AL, PCTRL
          TEST1  AL, 1
          BZ     RXDATA               ;Receive and store the
          IN     AL, PDATA            ;number of data bytes
          MOV    [BW],AL              ;stated by the counter
          INC    BW
          DBNZ   RXDATA
          MOV    AL, 00000000B        ;Command: receive disable
          OUT    PCTRL, AL
          RET
STLEN     DB     ?                    ;Set number of receiver data
RXDADR    DB     256 DUP (0)          ;Reserve receive data area
```

5c

Standby Mode

The μPD71051 is a low-power CMOS device. In standby mode, it disables the external input clocks to the inside circuitry (CLK, $\overline{\text{TxCLK}}$, and $\overline{\text{RxCLK}}$), thereby consuming less power.

A hardware reset is one way to enter standby mode. The input of a high level to the RESET pin causes the μPD71051 to enter standby mode at the falling edge of the high level. A software reset command is the other way to enter standby mode. The only way to take the μPD71051 out of standby mode is to write a mode byte.

In standby mode, the TxRDY, TxEMP, RxRDY, and SYNC/BRK pins are at low level and the TxDATA, $\overline{\text{DTS}}$, and $\overline{\text{RTS}}$ pins are at high level.

Figure 21 shows the timing for standby mode. While the internal standby signal is high, the external clocks to the μPD71051 are ignored. If data (C/$\overline{\text{D}}$ = 0) is written to the μPD71051 in standby mode, the operations are undefined and unpredictable operation may result.

Figure 21. Standby Mode Timing