µPD7201/7201A
µPD72001
µPD7210
µPD72105

PRODUCT DESCRIPTION

μPD7201/7201A

μPD72001

μPD7210

μPD72105

PRODUCT DESCRIPTION

COMMUNICATIONS CONTROLLERS

μPD7201/7201A	MULTIPROTOCOL SERIAL COMMUNICATION CONTROLLER
μPD72001	ADVANCED MULTIPROTOCOL SERIAL CONTROLLER
μPD7210	GPIB CONTROLLER
μPD72105	LOCAL AREA NETWORK CONTROLLER

Table of Contents - uPD7201/7201A Product Decsription

Table of Contents - uPD7201/7201A Product Decsription

Table of Contents - uPD72001 Product Description

Table of Contents - uPD72001 Product Description

Table of Contents - uPD7210 Product Description

Table of Contents - uPD7210 Product Description

Table of Contents - uPD7210 Product Description

Table of Contents - uPD72105 Product Description

Table of Contents - uPD72105 Product Description

Section Page

μPD7201/7201A
MULTIPROTOCOL SERIAL COMMUNICATION CONTROLLER

Description

The μPD7201 and μPD7201A Multiprotocol Serial Communication Controllers (MPSCC) are microcomputer peripherals that serve as multi-function peripheral devices capable of supporting a variety of serial data communications. The basic function of the MPSCC is to interface a microcomputer system (CPU) with serial data processing devices such as modems. The μPD7201A is an improved version of the μPD7201. Added features are described in the text, and listed in Appendix A.

The MPSCC controls three protocols:

- Asynchronous (start-stop synchronous)
- Bit-oriented synchronous
- Byte-oriented synchronous

Asynchronous operation provides stop bit lengths of 1, 1½, or 2 bits. It also provides transmit and receive clock rates programmable to 1, 16, 32, and 64 times the input frequency.

Bit-oriented synchronous operations such as HDLC (see note 1) and SDLC (see note 2) include the following features:

- Zero insertion/deletion
- Flag transmission and removal detection
- Fractional character processing
- Secondary address/global address
- End of frame
- Abort transmission detection

Bit-oriented synchronous operation also has a 16-bit transmit length counter and register to control the number of transmit characters (7201A only).

Byte-oriented synchronous operation such as Monosync, External Sync, and Bisync (see note 3) operation includes program-selectable SYNC characters and SYNC character transmission/removal detection.

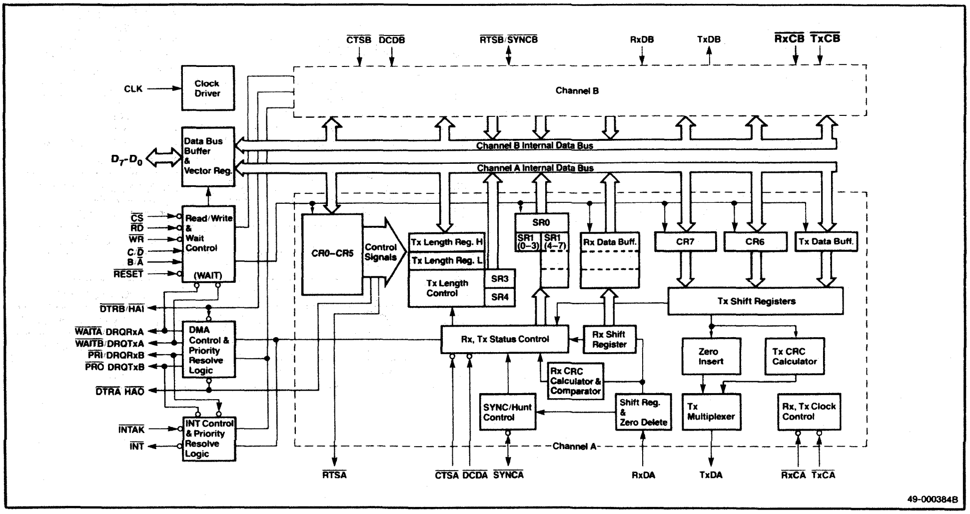
NOTES: 1. High-level Data Link Control (ISO)
2. Synchronous Data Link Control (IBM)
3. Binary Synchronous Communication (IBM)

Features

☐ μPD8085/8086 bus compatible
☐ High-speed transmit/receive operations of 1. 1 Mb/s at 5 MHZ system clock (7201A) or 880 kb/s at 4 MHz (7201)
  — Data buffering
  — 5- to 8-bit character length
  — Add/checking for odd, even, or no parity
  — Cyclic Redundancy Check (CRC) generation/checking (CCITT-0)
  — Error checking (parity, framing, Rx overrun, CRC)
☐ Modem control (two channels, four signals)
☐ 21/29 control, data, and status registers (7201/7201A)
☐ Various interrupt functions
☐ +5 V single power supply
☐ N-channel MOS
☐ 40-pin plastic DIP (μPD7201AC, μPD7201C)
☐ 40-pin ceramic DIP (μPD7201AD, μPD7201D)

Figure 1-1 shows the MPSCC functional block diagram.

Figure 1-1. MPSCC Block Diagram



Pin Identification

Figure 1-2 shows the pin configuration of the MPSCC. Table 1-1 gives the pin number, signal name, direction, and definition for each signal. The signal names and definitions of pins with two functions are separated by a slash (/).

Figure 1-2. Pin Configuration

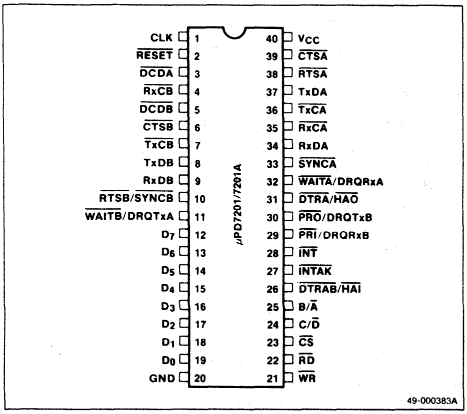


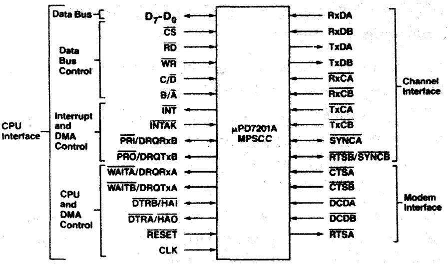
Table 1-1. MPSCC Pin Identification

No.	Symbol	Direction	Function
1	CLK	In	System clock
2	$\overline{\mathrm{RESET}}$	In	Reset
3	$\overline{\mathrm{DCDA}}$	In	Data carrier detect, channel A
4	$\overline{\mathrm{RxCB}}$	In	Receive clock, channel B
5	$\overline{\mathrm{DCDB}}$	In	Data carrier detect, channel B
6	$\overline{\mathrm{CTSB}}$	In	Clear to send, channel B
7	$\overline{\mathrm{TxCB}}$	In	Transmit clock, channel B
8	TxDB	Out	Transmit data, channel B
9	RxDB	In	Receive data, channel B
10	$\overline{\mathrm{RTSB}}/\overline{\mathrm{SYNCB}}$	In/Out	Request to send, channel B/Synchronization, channel B
11	$\overline{\mathrm{WAITB}}$/DRQTxA	Out	Wait, channel B/DMA request, transmit, channel A
12-19	D_7-D_0	In/Out	Data bus
20	GND		Ground potential
21	$\overline{\mathrm{WR}}$	In	Write
22	$\overline{\mathrm{RD}}$	In	Read
23	$\overline{\mathrm{CS}}$	In	Chip select
24	C/$\overline{\mathrm{D}}$	In	Control/data
25	B/$\overline{\mathrm{A}}$	In	Channel B/Channel A
26	$\overline{\mathrm{DTRB}}/\overline{\mathrm{HAI}}$	In/Out	Data terminal ready, channel B/Hold acknowledge input
27	$\overline{\mathrm{INTAK}}$	In	Interrupt acknowledge
28	$\overline{\mathrm{INT}}$	Out	Interrupt request
29	$\overline{\mathrm{PRI}}$/DRQRxB	In/Out	Priority input/DMA request, receive, channel B
30	$\overline{\mathrm{PRO}}$/DRQTxB	Out	Priority output/DMA request, transmit, channel B
31	$\overline{\mathrm{DTRA}}/\overline{\mathrm{HAO}}$	Out	Data terminal ready, channel A/Hold acknowledge output
32	$\overline{\mathrm{WAITA}}$/DRQRxA	Out	Wait, channel A/DMA request, receive, channel A
33	$\overline{\mathrm{SYNCA}}$	In/Out	Synchronization, channel A
34	RxDA	In	Receive data, channel A
35	$\overline{\mathrm{RxCA}}$	In	Receive clock, channel A
36	$\overline{\mathrm{TxCA}}$	In	Transmit clock, channel A
37	TxDA	Out	Transmit data, channel A
38	$\overline{\mathrm{RTSA}}$	Out	Request to send, channel A
39	$\overline{\mathrm{CTSA}}$	In	Clear to send, channel A
40	V_{CC}		Power supply

The pins of the MPSCC function in three interface categories: CPU, channel, and modem. This section describes the pin functions of each interface category.

Figure 2-1 shows pinouts of the MPSCC by each interface category.

Figure 2-1. Functional Pinout



49-000385A

CPU INTERFACE SIGNALS

The following signals function as the CPU interface:

$\overline{RESET}$ (Reset)

A low-level input to this pin for at least one clock cycle causes the MPSCC to perform a system reset. Since a system reset may cause data loss to control registers, write or rewrite data to the control registers before data transmission or reception.

The state of the MPSCC when the system is reset is:

Transmitter/receiver	Disabled
Interrupt/DMA	Disabled
TxDA/TxDB output	Marking 1s
Modem control output	1

See Section 4, D_5-D_3 (Command Bits) for a comparison of the states of each register and pin when the system is reset and when the channel reset command is issued.

CLK (System Clock)

Inputs a single-phase, system clock that is TTL-compatible. The system clock rate must be 4.5 times faster than the data rate.

D_7 - D_0 (Data Bus)

D_7-D_0 is an 8-bit bidirectional data bus. These tri-state pins connect the MPSCC to a standard CPU such as the μPD8080AF, μPD8085A, or μPD8086. The data bus transmits data, commands, or status signals between the MPSCC and CPU.

$\overline{CS}$ (Chip Select)

Provides access to the Tx or Rx buffer or the status or control register specified by the register pointer, B/A, or C/$\overline{D}$ pins.

$\overline{WR}$ (Write)

Host sends a low-level signal when data or commands are transmitted from the CPU or memory to the MPSCC.

$\overline{RD}$ (Read)

Host sends a low-level signal when data or commands are transmitted from the CPU or memory to the MPSCC.

C/$\overline{D}$ (Control/Data)

Indicates the type of data on the data bus when a write or read operation is performed. A high level indicates that commands or status data is on the data bus. A low level indicates that transmit or receive data is on the data bus.

B/$\overline{A}$ (Channel B/Channel A)

Indicates the channel to or from which data on the data bus is written or read, when a write or read operation is performed. A high level specifies channel B. A low level specifies channel A.

Table 2-1 shows the different states and functions of the C/$\overline{D}$, $\overline{WR}$, $\overline{RD}$, $\overline{CS}$, and B/$\overline{A}$ signals.

Table 2-1. Signal Functions

C $\overline{D}$	$\overline{WR}$	$\overline{RD}$	$\overline{CS}$	B/$\overline{A}$	Channel	Function
0	0	1	0	0	A	Writes transmit data to the Tx buffer
				1	B	
0	1	0	0	0	A	Reads receive data from the Rx buffer
				1	B	
1	0	1	0	0	A	Writes data to the command parameter (CR7-CR0) registers.
				1	B	
1	1	0	0	0	A	Reads data from the status vector (SR4-SR0) registers.
				1	B	

$\overline{\text{INT}}$ (Interrupt Request)

Outputs an interrupt request signal. A low-level signal indicates an interrupt within the MPSCC. $\overline{\text{INT}}$ is an open-drain output.

$\overline{\text{INTAK}}$ (Interrupt Acknowledge)

Inputs an interrupt acknowledge signal in response to an interrupt request signal. When several MPSCCs simultaneously request an interrupt, the one with the top priority accepts this signal. The MPSCC may also output an interrupt vector to the data bus, depending on the specified mode. If this pin is not used, it must be held high with a pull-up resistor.

$\overline{\text{PRI}}$/DRQRxB (Priority Input/DMA Request RxB)

Transmits or receives the $\overline{\text{PRI}}$ and DRQRxB signals. The state of the INT/DMA mode bits (D_1 and D_0) of the CR2A register specify the function of this pin.

$\overline{\text{PRI}}$ is used when two or more interrupt devices are used to fom a daisy chain. A low level indicates that an interrupt device assigned a higher priority is not receiving service by the CPU's interrupt routine. When only one MPSCC is used, this pin must be held low (except when using a µPD7201A in 85-3 Vectored Mode).

DRQRxB is used as a DMA request signal to the DMA controller. The signal is active when data is input to the receive buffer of channel B (Rx Character Available).

$\overline{\text{PRO}}$/DRQTxB (Priority Output/DMA Request TxB)

The state of the INT/DMA mode bits (D_1 and D_0) of the CR2A register specify the function of this pin.

$\overline{\text{PRO}}$ is paired with the $\overline{\text{PRI}}$ signal to control interrupt priority by using a daisy chain. When $\overline{\text{PRI}}$ is high, $\overline{\text{PRO}}$ outputs a high-level signal. When $\overline{\text{PRI}}$ is low, the $\overline{\text{PRO}}$ signal goes high when the MPSCC requests an interrupt from the CPU. If the interrupt signal is not ouput to the CPU, $\overline{\text{PRO}}$ remains low even if $\overline{\text{PRI}}$ is high.

DRQTxB outputs a DMA request signal to the DMA controller and becomes active when the Tx buffer of channel B becomes empty.

$\overline{\text{WAITA}}$/DRQRxA (Wait A/DMA Request RxA)

The state of the INT/DMA bits (D_1 and D_0) of the CR2A register specify the function of this pin.

$\overline{\text{WAITA}}$ requests the CPU to enter the WAIT state via the CPU's READY pin. Used as an open-drain output.

DRQRxA outputs a DMA request signal to the DMA controller. This signal becomes active when a character is input to the Rx buffer of channel A.

$\overline{\text{WAITB}}$/DRQTxA (Wait B/DMA Request TxA)

The state of the INT/DMA mode bits (D_1 and D_0) of the CR2A register specify the function of this pin.

$\overline{\text{WAITB}}$ requests the CPU to enter the WAIT state via the CPU's READY pin. Used as an open-drain output when the WAIT function is used.

DRQTxA outputs a DMA request signal to the DMA controller and becomes active when the Tx buffer of channel A becomes empty.

$\overline{\text{DTRB}}$/$\overline{\text{HAI}}$ (Data Terminal Ready B/Hold Acknowledge Input)

The state of INT/DMA mode bits (D_1 and D_0) of the CR2A register specify the function of this pin.

The level of $\overline{\text{DTRB}}$ changes according to the state of the DTR bit (D_7 of the CR5 register) of channel B. When the DTR bit is 1, $\overline{\text{DTRB}}$ becomes 0. When the DTR bit is 0, the $\overline{\text{DTRB}}$ signal becomes 1.

$\overline{\text{HAI}}$ is used to control the DMA priority with a daisy chain. Normally, HLDA is input to the $\overline{\text{HAI}}$ pin of the MPSCC with the highest DMA priority. A low-level input to this pin indicates that this particular MPSCC is selected for the DMA operation.

$\overline{\text{DTRA}}$/$\overline{\text{HAO}}$ (Data Terminal Ready A/Hold Acknowledge Output)

The state of the INT/DMA mode bits (D_1 and D_0) of the CR2A register specify the function of this pin.

The level of $\overline{\text{DTRA}}$ changes according to the state of the DTR bit of channel A (D_7 of the CR5 register). When the DTR bit is 1, $\overline{\text{DTRA}}$ is 0. When the DTR bit is 0, $\overline{\text{DTRA}}$ is 1.

$\overline{\text{HAO}}$ is paired with $\overline{\text{HAI}}$ to control the DMA daisy chain. When $\overline{\text{HAI}}$ is at a high level, $\overline{\text{HAO}}$ outputs a high. When $\overline{\text{HAI}}$ is at a low level, $\overline{\text{HAO}}$ outputs a high when the MPSCC requests a DMA operation. This inhibits another MPSCC daisy chained in the lower order from performing the DMA operation. If the DMA request signal is not output from the MPSCC, a low-level signal is output and DMA operations by the lower-order MPSCCs in the daisy chain are enabled.

CHANNEL INTERFACE SIGNALS

The following signals function as the channel interface.

RxDA, RxDB (Receive Data A/B)

Inputs receive data.

TxDA, TxDB (Transmit Data A/B)

Outputs transmit data.

$\overline{\text{RxCA}}$, $\overline{\text{RxCB}}$ (Receive Clock A/B)

Inputs clocks for sampling receive data. Sampling is performed on the leading edge of the $\overline{\text{RxC}}$ signal.

When the asynchronous mode is specified, the clock rate of the $\overline{\text{TxC}}$ signal must be a 1, 16, 32, or 64 multiple of the data rate.

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$ (Transmit Clock A/B)

Inputs clocks for transmit data. Data is output at the leading edge of the $\overline{\text{TxC}}$ signal. When the asynchronous mode is specified, the clock rate of the $\overline{\text{TxC}}$ signal must be a 1, 16, 32, or 64 multiple of the data rate.

$\overline{\text{SYNCA}}$ (Synchronization A)

The function of this pin depends on the operating protocol of the MPSCC.

Asynchronous Mode (Input). If external/synchronous (E/S) interrupts are enabled, an interrupt occurs at the leading or trailing edge of the SYNC signal (or according to the state of the E/S bit). Inverted data from the $\overline{\text{SYNCA}}$ pin is latched at the SYNC/Hunt bit (D_4 of the SR0 register).

External Asynchronous Mode (External/Sync) (Input). The E/S interrupt operation and function of the E/S bit in this mode is similar to the asynchronous mode.

If the MPSCC is in the hunt phase (Section 4, D_4 (Enter Hunt Phase)) when the program specifies the external/sync mode, the MPSCC goes out of the hunt phase at the leading edge of an input SYNC signal. The MPSCC then starts assembling characters, beginning from the character sampled at the leading edge of the RxC signal input before the SYNC signal.

When the SYNC character has been detected, the external synchronization circuit resets the SYNC signal to a low level after two cycles of the $\overline{\text{RxC}}$ signal. The SYNC signal must be held low until the character synchronization is lost or a new message starts.

Internal Synchronous Mode (Monosync) (Output). Outputs a low-level signal each time the SYNC character is detected in received data.

Internal Synchronous Mode (Bisync) (Output). Outputs a low-level signal only when the first SYNC character is received after the MPSCC has entered the hunt phase.

$\overline{\text{RTSB}}$/$\overline{\text{SYNCB}}$ (Request to Send B/Synchronization B)

Transmits or receives the $\overline{\text{RTSB}}$ and $\overline{\text{SYNCB}}$ signals, respectively. The state of the RTSB/SYNCB select bit (D_7 of the CR2A register) specifies which function the pin performs.

When the RTSB/SYNCB select bit is 0, the $\overline{\text{RTSB}}$ signal is selected. When the RTSB/SYNCB select bit is 1, the $\overline{\text{SYNCB}}$ signal is selected.

The functions of the $\overline{\text{RTSB}}$ and $\overline{\text{SYNCB}}$ signals are the same as those of the $\overline{\text{RTSA}}$ and $\overline{\text{SYNCA}}$ signals, except that the $\overline{\text{RTSB}}$ and $\overline{\text{SYNCB}}$ signals are applied to channel B.

MODEM INTERFACE SIGNALS

The following signals function as the modem interface.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$ (Clear to Send A/B)
$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$ (Data Carrier Detect A/B)

The $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ pins have similar functions. The E/S interrupt is generated (when interrupts are enabled) at the leading or trailing edge of these signals and latched at the E/S bits (D_5 and D_3) of the SR0 register. The CTS and DCD pins can also be used as a trigger input in the auto enable mode. In this case, when bit D_3 of the CR3 is 1, either the transmitter ($\overline{\text{CTS}}$) or receiver ($\overline{\text{DCD}}$) is enabled by the presence of a low-level signal. When the MPSCC is not in auto enable mode, these two pins are used as general purpose input pins.

$\overline{\text{RTSA}}$ (Request to Send A)

The state of the RTS bit (D_1 of the CR5 register) controls this pin. If the RTS bit is reset in the asynchronous mode, a high level will not be output on the $\overline{\text{RTS}}$ pin until all transmit characters are written and the all sent bit (D_0 of the SR1 register) is set.

In the synchronous mode, the state of the RTS bit is used as is. That is, when the RTS bit is 0, the $\overline{\text{RTS}}$ pin is 1. When the RTS bit is 1, the $\overline{\text{RTS}}$ pin is 0.

The MPSCC consists of a transmitter/receiver and CPU interface section. The transmitter/receiver section has two full-duplex channels (A and B) and control circuits for communication with other serial data processing devices via a modem. Both channels A and B have a receiver and transmitter and identical structure. Figure 3-1 shows the flow of transmit/receive data in each channel.

The CPU interface section has status and control registers, interrupt and DMA control circuitry, and a data bus buffer.

RECEIVER SECTION

Receive (Rx) data input from the RxD terminal is converted into 8-bit data by the Rx shift register and transferred to the Rx data buffer. The operation mode and character length of the data determines the path along which data flows.

Serial Data Path

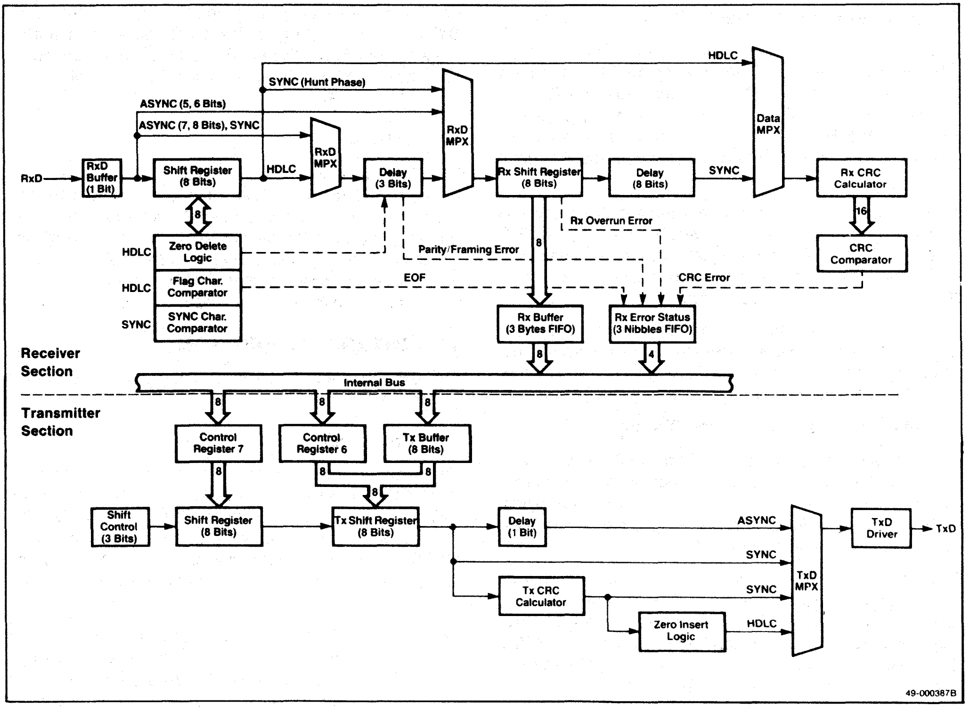
In the asynchronous mode, serial data received by the MPSCC is transmitted to the Rx shift register by two methods. If the character of the received data consists of 7 or 8 bits, it is transferred via a 3-bit buffer. If it consists of 5 or 6 bits, the data is directly input to the Rx shift register.

In the synchronous mode, the serial data is compared with the SYNC character which was input in the 8-bit shift register while the MPSCC is in the hunt phase. If the MPSCC goes out of the hunt phase, or if it is set in the external synchronous mode, serial data is transferred to the 3-bit buffer, bypassing the 8-bit shift register. When the MPSCC is in the hunt phase, input data is compared at the 8-bit shift register with the contents of the CR6 or CR7 register by the Rx SYNC register.

In the monosync mode (8-bit SYNC character), data is compared with the contents of the CR7 register.

In the bisync mode (16-bit SYNC character), the first 8 bits of the input data are compared with the contents of the CR6 register and the last 8 bits with the contents of the CR7 register. If the input data is matched with the contents of the above registers, the MPSCC goes out of the hunt phase. Input data then bypasses the 8-bit shift register as long as this synchronization is maintained. Note that the syn-

Figure 3-1. Flow of Transmit/Receive Data



chronization may not be properly established when xxH SYNC characters (for example 55H or 33H) are used in the monosync or bisync mode.

In the HDLC mode, input data follows the same data path regardless of whether the MPSCC is in the hunt phase or synchronization is established. Input data is first transferred to the 8-bit shift register and the zeros of the input data are deleted if necessary. The MPSCC then also determines whether the input data contains a flag or abort sequence. The input data is subsequently transferred to the 3-bit buffer, then to the Rx shift register. If the register finds that the first five consecutive bits of the input data are 1s, the MPSCC then performs a function according to the state of the next (6th or 7th) bit(s) as follows:

Rx Shift Register Data Bits 1 2 3 4 5 6 7	Function
1 1 1 1 1 0	Deletes 0
1 1 1 1 1 1 0	Receives flag sequence
1 1 1 1 1 1 1	Receives abort sequence

In the 7201A, an abort sequence is only reported after a flag sequence has been detected.

Cyclic Redundancy Check (CRC)

The CRC calculation operation is performed by the Rx CRC calculator and the CRC comparator. Receive data is transferred to the Rx CRC calculator from the Rx shift register via the CRC delay register in the synchronous mode. In the HDLC mode, the receive data is directly transferred to the Rx CRC calculator from the 8-bit shift register.

In the bisync mode, the program must make a decision whether to include each receive character in the CRC calculation. The CRC delay register holds the receive data for 8 bits to compensate for the process. It then transfers the data to the Rx CRC calculator.

In the HDLC mode, flag patterns are not included in the CRC calculation and the calculation is performed on data other than flag patterns. The MPSCC determines whether the CRC calculation is performed on a given character.

Parallel Data Path and Error Display

Serial data is transferred to the Rx buffer (three 8-bit buffers) after it is converted into 8-bit parallel data by the Rx shift register. The Rx buffer has a capacity of three bytes, sufficient capacity for data processing to transfer parallel-converted Rx data into the CPU.

Error data concerning receive data is stored in three 4-bit error registers. Each register indicates parity, overrun, CRC/framing, and end of frame errors. Each 4-bit error register has error data that corresponds to each byte of the 3-byte Rx buffer. The contents of the error registers can be read from bits D_7 to D_4 of the SR1 register.

TRANSMITTER SECTION

The 8-bit transmit data sent to the MPSCC is transferred to the Tx buffer via the internal data bus. At the same time, the SYNC character in the synchronous mode or a secondary address and a flag in the HDLC mode are loaded into control register 6 (CR6) and control register 7 (CR7). The contents of the Tx buffer and the CR6 and CR7 registers are then sent to the Tx shift register (3-bit shift control, two 8-bit shift registers and 1-bit delay) and converted to serial data.

Asynchronous Mode

In this mode, asynchronous data in the Tx buffer is transferred to the Tx shift register and converted to serial data. Then parity, start, and stop bits are appended to the serial-converted data. The data is then subsequently transferred to the Tx multiplexer at a specified clock rate for transmittal from the TxD pin.

Synchronous Mode

As in the asynchronous mode, transmit data is sent to the Tx shift register from the Tx buffer. The contents of the CR6 or CR7 registers are sent to the Tx shift register as the SYNC characters, or as a time-fill character for a starting message or when a Tx underrun error occurs. Data output from the Tx shift register is also sent to the Tx multiplexer and to the Tx CRC calculator.

HDLC Mode

The Tx data is sent to the Tx shift register from the Tx buffer, and a flag pattern is sent to the Tx shift register from the Tx buffer when a message starts and ends. The output data from the Tx shift register is sent to the zero insert logic circuit and to the Tx CRC calculator. In the zero insert logic circuit, a zero is inserted after any five consecutive 1s in all fields other than the flags (address, control, frame check).

CPU INTERFACE SECTION

The CPU interface consists of a register section and a control section.

Register Section

The register section consists of eight control registers (CR7-CR0) and five status registers (SR4-SR0) for the 7201A, or three status registers (SR0-SR2) for the 7201. The control registers control the operation mode of the MPSCC, the internal data flow, and retain SYNC characters or flag patterns. The status registers hold status information.

Note that the CR2 register performs different functions depending on the channel to which the register belongs. The CR2 register of channel A (CR2A) is used to determine the system configuration. The CR2 register of channel B

(CR2B) receives an interrupt vector. The interrupt vector written to the CR2B register can be read from the SR2B register. In addition to the CR0-CR7 registers, each channel of the 7201A has a 2-byte transmit length register. Section 4 gives detailed descriptions of the functions of each register.

Control Section

The control section provides an interface between the MPSCC and the main system, and controls data transmission/reception. Data communication between the MPSCC and CPU is performed via an 8-bit data bus consisting of the D_7-D_0 pins. The type of data transmitted and data transmission directions are controlled by input control signals. Table 3-1 shows how to:

- Select channels using control signals
- Specify a read or write operation
- Select control or status registers

Three data transmission methods exist for communication between the MPSCC and CPU: polling, interrupt, and DMA modes. Each channel (A or B) is set in one of these modes by the contents of the CR2A register. However, each channel's CR1 register determines whether to enable or disable either the interrupt or DMA mode.

The CR2A register also determines the priority of an MPSCC interrupt or DMA request. Table 3-2 shows how the CR2A register specifies the mode (interrupt or DMA), pin function, or priority. Table 3-3 shows the interrupt mode's selection by the CR2A register and the contents of the data bus (interrupt vector) in the interrupt acknowledge cycle.

Table 3-1. MPSCC Control Signals

C/$\overline{D}$	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	$\overline{INTAK}$	$\overline{PRI}$	$\overline{HAI}$	B/$\overline{A}$	Channel	Function
0	0	0	1	1	x	1	0	A	Reads receive data from Rx buffer and reads data received by DMA operation in DMA-2 mode (7201A only)
							1	B	
0	0	1	0	1	x	1	0	A	Writes transmit data to Tx buffer and writes data transmitted by DMA operation in DMA-2 mode (7201A only)
							1	B	
1	0	0	1	1	x	1	0	A	Reads data from status vector registers (SR0-SR4).
							1	B	
1	0	1	0	1	x	1	0	A	Writes data to command parameter registers (CR0-CR7).
							1	B	
x	0	Combination other than the above when $\overline{CS}$ = 0.							Prohibited
x	1	x	x	0	1	x	x		Reads CALL command and interrupt vector in interrupt acknowledge cycle.
x	1	0	1	1	x	0	x		Reads receive data by DMA operation in DMA-1 mode.
x	1	1	0	1	x	0	x		Writes transmit data by DMA operation in DMA-1 mode.
x	1	Combination other than the above when $\overline{CS}$ = 1.							No read or write operation is performed.

Notes: 1. x = Don't care

2. If the $\overline{HAI}$ pin is not used for a hold acknowledge input, it will be treated as a don't care pin.

Table 3-2. CR2A Register Control

CR2A			INT/DMA		Pin Function						Priority					
D_2	D_1	D_0	Channel A	Channel B	32	11	29	30	26	31	High					Low
0	0	0	INT	INT	$\overline{\text{WAITA}}$	$\overline{\text{WAITB}}$	$\overline{\text{PRI}}$	$\overline{\text{PRO}}$	$\overline{\text{DTRB}}$	$\overline{\text{DTRA}}$	RxA	TxA	RxB	TxB	E/SA	E/SB
1	0	0	INT	INT							RxA	RxB	TxA	TxB	E/SA	E/SB
x	0	1	DMA		DRQRxA	DRQTxA	$\overline{\text{PRI}}$	$\overline{\text{PRO}}$	$\overline{\text{HAI}}$	$\overline{\text{HAO}}$	RxA	TxA				
				INT							RxA		RxB	TxB	E/SA	E/SB
0	1	0	DMA	DMA	DRQRxA	DRQTxA	DRQRxB	DRQTxB	$\overline{\text{HAI}}$	$\overline{\text{HAO}}$	RxA	TxA	RxB	TxB		
											RxA		RxB		E/SA	E/SB*
1	1	0	DMA	DMA							RxA	RxB	TxA	TxB		
											RxA	RxB			E/SA	E/SB*
x	1	1 (7201A only)	DMA	DMA					$\overline{\text{DTRB}}$	$\overline{\text{DTRA}}$	No priority					
											RxA	RxB			E/SA	E/SB*

Notes: x = Don't care

* These priorities are for the special receive and external status interrupts which can occur when using dual channel DMA operation.

Table 3-3. Vectored Interrupt Control

CR2A				INTA CYCLE			
D_5	D_4	D_3	Interrupt Mode Select	PRI	1	2	3
1	0	0	85-1	0	CD HEX (CALL OP)	VECTOR	0
				1	CD HEX (CALL OP)	HI-Z	HI-Z
1	0	1	85-2	0	HI-Z	VECTOR	0
				1	HI-Z	HI-Z	HI-Z
1	1	1	85-3 Note 1	0	HI-Z	VECTOR	0
				1	HI-Z	HI-Z	HI-Z
1	1	0	86 Note 2	0	HI-Z	VECTOR	
				1	HI-Z	HI-Z	

Notes: 1. 85-3 mode activates the $\overline{\text{INT}}$ line if an internal interrupt exists whether or not the $\overline{\text{PRI}}$ line is high. Therefore, a daisy chain using the $\overline{\text{PRI}}$ and $\overline{\text{PRO}}$ pins cannot be performed. (This mode is available on the μPD7201A only.)

2. 8086 mode issues two $\overline{\text{INTAK}}$ pulses instead of three.

When operating in the 8080/5 modes, the MPSCC issues an 8080-type CALL CD vv (hex) instruction where vv is the contents of control register 2B. Register 2B is modified by the cause of the interrupt if the status affects vector feature is enabled. An MPSCC programmed for 8085 master mode always places the CALL opcode on the data bus whether or not that MPSCC has a pending interrupt request. To avoid problems caused by momentary bus contention, never program more than one device to operate in this mode.

Data Transmission Between MPSCC and CPU

Data transmission between the CPU and MPSCC (or memory and the MPSCC) is performed in either polling, interrupt, or DMA mode.

Polling Mode. When data transmission is performed in the polling mode, the CPU must monitor the status of the MPSCC to learn the timing of the data transmission. The CPU reads status information from the MPSCC's SR0 and SR1 registers and determines whether:

- a Tx data transmit (MPSCC to CPU) request exists,
- an Rx data receive (CPU to MPSCC) request exists, or
- an error occured in the MPSCC.

If an Rx character available bit (D_0 of the SR0 register) is set, for example, valid Rx data exists in the Rx buffer of the MPSCC, then the CPU must immediately read the data.

Also, the interrupt vector may be used in the SR2B register (vector bits V_4-V_2 or V_2-V_0) for checking the status of the MPSCC. In this case, channels A and B must be set in the interrupt mode by bits D_0 and D_1 of the CR2A register, or in a nonvectored mode. The 85 mode is set when bits D_5-D_3 of the CR2A register are 000, 001, or 011. The 86 mode is set when bits D_5-D_3 of the CR2A register are 010. The status affects vector bit (D_2 of the CR1B register) must also be set. vector bits V_4-V_2 of the interrupt vector are modified in the nonvectored 85 mode and V_2-V_0 in the nonvectored 86 mode.

Interrupt Mode. The MPSCC has one interrupt request terminal ($\overline{INT}$). When an interrupt source occurs inside the MPSCC, the $\overline{INT}$ signal becomes active and informs the CPU or interrupt controller of an interrupt request.

In a system using more than one MPSCC, the interrupt process can be performed through a daisy chain using the $\overline{PRI}$ and $\overline{PRO}$ pins. If an interrupt source occurs inside the MPSCC, the interrupt vector corresponding to the interrupt source can be generated when the vector mode is specified and the status affects vector bit is set.

Eight types of interrupt vectors, each corresponding to a given interrupt source inside the MPSCC, can be generated by setting the status affects vector bit (D_2 of the CR1B register). These interrupt vectors allow direct branching to the process routine for each interrupt source. The MPSCC also has nonvectored modes for CPU's that cannot read vectors in an interrupt cycle. In this case, the CPU determines what interrupt source is issuing the request by reading the contents of the status or vector registers of the MPSCC. The CPU will then execute the appropriate process routine.

Three interrupt sources can occur in the MPSCC.

- Receive (Rx)
- Transmit (Tx)
- External/Status (E/S)

The Rx interrupt occurs when the following conditions exist: the MPSCC receives data; the valid (character) data for the read operation to be performed by the CPU is loaded to the Rx buffer; and Rx interrupts are enabled. There are three modes of Rx interrupts.

First Rx Character. In this mode an interrupt occurs only when the first character is received by the MPSCC. In other words, an interrupt is caused by the first character loaded to the Rx buffer after this mode is set. Once an interrupt occurs, subsequent interrupts in this mode can be enabled by issuing the enable interrupt on next Rx character command.

This mode is generally used for data transmission controlled by the software. It is also used for DMA data transmission. This interrupt can be masked by setting bit D_6 or the CR2A register to 1 (7201A only).

All Rx Characters. This mode is used for data transmission by using an interrupt each time a character is loaded to the Rx buffer.

Special Rx Conditions. This interrupt is a special case of Rx interrupt and occurs when either the first Rx character interrupt or all Rx character interrupt is specified. Special Rx conditions refer to parity, overrun, framing, or end of frame errors. When the first Rx character interrupt mode is specified, parity errors are not treated as a special Rx condition. When all Rx character interrupt mode is specified, specify whether or not parity errors are included in the special Rx conditions.

The Tx interrupt occurs when the Tx buffer is empty and Tx interrupts are enabled. This interrupt indicates that the MPSCC is requesting transmit data. The interrupt is satisfied when data is written to the Tx buffer or when the reset Tx INT/DMA pending command is issued. The Tx interrupt also occurs when transmission of CRC characters has been completed in the synchronous or HDLC mode. However, immediately after the MPSCC has been reset ($\overline{RESET}$ signal or channel reset command), the Tx buffer is empty and the Tx interrupt does not occur even if enabled. Note: If bit D_6 of the CR1 register is set to 1 (7201A only) in the HLDC mode, the Tx interrupt will occur if the first data is written to the Tx buffer.

The E/S interrupt checks whether the state of the $\overline{CTS}$, $\overline{DCD}$, or $\overline{SYNC}$ pin has changed. This interrupt is also used to detect the Tx underrun error, break state (asynchronous mode), abort sequence, or completion of data transmission (HDLC mode).

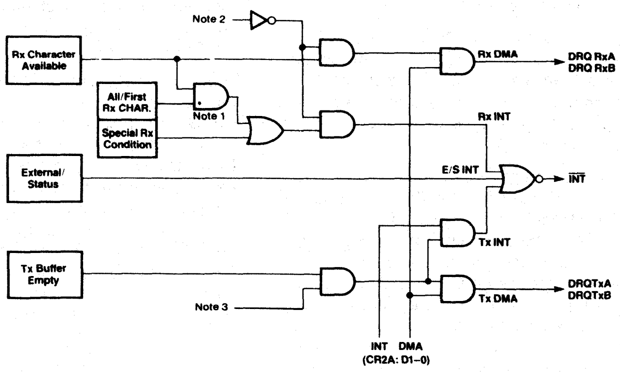
DMA. The MPSCC has four DMA request pins. When DMA is enabled by D_1, D_3, and D_4 of the CR1 register, a DMA request occurs as follows:

DRQTxA, DRQTxB: When the Tx buffer becomes empty. However, the DMA request does not occur if the Tx buffer is empty immediately after it has been reset. This does not apply when bit D_6 of the CR1 register is 1 (HDLC mode, 7201A only).

DRQRxA, DRQRxB: When a character exists in the Rx buffer.

Interrupt and DMA. Bits D_1 and D_0 of the CR2A register specify whether data transmission is performed by means of an interrupt or DMA. An interrupt or DMA is enabled or disabled by bit D_1 (Tx interrupt/DMA) and D_4, D_3 (Rx interrupt/DMA) of the CR1 register. When DMA is specified to a channel, the Tx interrupt of that channel is inhibited. However, the Rx, special Rx condition, and E/S interrupts can still occur when interrupts are enabled. Normally, the first Rx character interrupt is specified as the Rx interrupt mode of the channel that has been specified for DMA operation. This first Rx character interrupt can be masked by setting bit D_6 of the CR2A register to 1 (7201A only). See table 3-2 for details on specifying interrupts or DMA. Figure 3-2 shows interrupt/DMA request generation logic.

Figure 3-2. Interrupt/DMA Request Generation Logic



Notes: 1. In All Rx Character mode: Normally active
In First Rx Character Interrupt Mode: Active until first Rx Character interrupt is generated. However, this is masked when D_6 of the CR2A register = 1 (7201A only).

2. Rx Disable (CR1: D_1)

3. Tx INT/DMA Enable (CR1: D_1)

49-000386A

Both channels A and B of the MPSCC have eight control registers (CR0-CR7), two transmit length registers (TxLR: 7201A only), and two or four status registers (SR0 and SR1: 7201; and SR0, SR1, SR3, and SR4: 7201A). Channel B also has an additional status register (SR2B).

The control registers select the required functions for the MPSCC's particular application system. These functions include selecting the following:

- CPU Interface mode
- Serial transmit/receive protocol (asynchronous, bisync, or HDLC)
- MPSCC operation mode used with the selected protocol
- Various parameters
- SYNC character codes

The status registers indicate the internal state of the MPSCC, interrupt status or vectors, and the contents of the transmit length counter (7201A only).

This section describes the function of each register; table 4-1 summarizes the function of each register for both channels.

Table 4-1. Register Functions

Register	Channel	Function
CR0	A, B	Selects a status or control register and resets CRC logic and sets commands (reset of the MPSCC interrupt)
CR1	A, B	Selects the bus interface mode (data transfer mode between CPU and MPSCC)
CR2A	A	Specifies system configuration
CR2B	B	Sets the interrupt vectors
CR3	A, B	Controls data reception operation
CR4	A, B	Controls common operation of data transmission and reception
CR5	A, B	Controls data transmission operation
CR6	A, B	Specifies SYNC character, flags, and secondary address (HDLC mode)
CR7	A, B	
TxLR-L	A, B	Specifies the length of transmit data up to 16 bits (7201A only)
TxLR-H	A, B	
SR0	A, B	Specifies the states of E/S bits, interrupts, or buffers
SR1	A, B	Sets the special Rx condition or residue codes
SR2	B	Indicates interrupt vectors
SR3	A, B	Indicates contents of a 16-bit Tx length counter (7201A only)
SR4	A, B	

Note: The function CR2 performs is determined by its channel. The CR2 register of channel A (CR2A) defines the system configuration, and the CR2 register of channel B (CR2B) specifies the interrupt vector to be transmitted in the interrupt acknowledge cycle. The contents of the CR2B register can also be read from the SR2B register. Channel A does not have the SR2 register.

CONTROL REGISTER 0 (CR0)

The functions of the CR0 register bits are summarized in table 4-2.

Table 4-2. CR0 Bit Functions

Bit	Name	Definition	Function
D_7	CRC1	CRC initialization code	Initializes CRC logic
D_6	CRC0		
D_5	CMD2	Command bits	Sets commands used to reset the MPSCC, enables, interrupts, etc.
D_4	CMD1		
D_3	CMD0		
D_2	PTR2	Pointer bits	Selects a status or control register
D_1	PTR1		
D_0	PTR0		

D_7 and D_6 (CRC Initialization Code)

Table 4-3 shows the bit pattern and related function for the CRC bits.

Table 4-3. CRC Bit Functions

CRC1	CRC0	Function	
0	0	No operation	
0	1	Initializes Rx CRC calculator	In Synchronous mode: all 0s (CCITT-0 CRC-16) In HDLC mode: all 1s (CCITT-1)
1	0	Initializes Tx CRC calculator	
1	1	Resets Tx underrun/EOM bit (D_6 of the SR0 register) (See Note)	

Note: Effective when one character is written and Tx is enabled.

D_5-D_3 (Command Bits)

D_5	D_4	D_3	
0	0	0	(No operation)

No operation is performed. Specified when the CR0 register is used for purposes other than issuing commands.

D_5	D_4	D_3	
0	0	1	(Send Abort)

HDLC abort bits (8 bits, all 1s) are transmitted followed by a flag that causes the CRC circuit of the receiver to reset. 8 to 13 bits are transmitted depending on the number of bits (1s) preceding the abort bits. The contents of the Tx (transmit) buffer are lost when this command is issued.

D_5	D_4	D_3	
0	1	0	(Reset External/Status Interrupt)

External/status (E/S) interrupt is re-enabled, and status latch operations to E/S bits (D_7-D_3 of the SR0 register and D_0 of SR1 register) can be performed. Once an E/S interrupt occurs, issue this command to allow other E/S interrupts. Also, issue this command to allow status latch operations to the E/S bits.

D_5 D_4 D_3
0 1 1 (Channel Reset)

Performs almost the same operation on channel A or B as the external reset. When executed to reset a channel, part of the register contents of the channel is lost. Therefore, it is necessary to rewrite data to the register whose contents have been lost. Issue a new command after the lapse of four system clock cycles when this command has been executed.

Tables 4-4 and 4-5 show the states of the registers and pins after a system reset and after execution of the channel reset command.

Table 4-4. Register State at Reset

Register	Bits	At System Reset	At Channel Reset
CR2A	0-5, 7	0	See table 4-5
CR0	0-2	0	0
CR1	0, 1, 3, 4, 7	0	0
CR3	0-7	0	0
CR5	1-4, 7	0	0
SR0	0-1	0	0
	2, 6	1	1
	3-5, 7	Undefined	
SR1	4-7	0	0
	0-3	Undefined	
SR3, SR4 (7201A only)	0-7	0	0

Table 4-5. Pin State at Reset

		CR2A, D_2-D_0			
		000 or 100	001 or 101	010 or 110	011 or 111
Pin	At System Reset	At Channel Reset			
$\overline{\text{WAITA}}$/ DRQRxA	$\overline{\text{WAITA}}$ = Hi–Z		DRQRxA = 0		
$\overline{\text{WAITB}}$/ DRQRxA	$\overline{\text{WAITB}}$ = Hi–Z		DRQTxA = 0		
$\overline{\text{PRI}}$/DRQRxB	$\overline{\text{PRI}}$ = input			DRQRxB = 0	
$\overline{\text{PRO}}$/DRQTxB	$\overline{\text{PRO}}$ = by PRI			DRQTxB = 0	
$\overline{\text{DTRB}}$/$\overline{\text{HAI}}$	$\overline{\text{DTRB}}$ = 1		$\overline{\text{HAI}}$ = input		$\overline{\text{DTRB}}$ = 1
$\overline{\text{DTRA}}$/$\overline{\text{HAO}}$	$\overline{\text{DTRA}}$ = 1		$\overline{\text{HAO}}$ = $\overline{\text{HAI}}$		$\overline{\text{DTRA}}$ = 1
INT	Hi–Z				
$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$	1				
$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$	Input state				

D_5 D_4 D_3
1 0 0 (Enable Interrupt on Next Receive Character)

If an interrupt occurs in the first Rx character mode (first character has been received), this command causes another interrupt with the first character after the command is issued. Prepares for new messages after receiving a series of messages.

D_5 D_4 D_3
1 0 1 (Reset Transmitter Interrupt/DMA Pending)

When the Tx buffer becomes empty, a Tx interrupt/DMA request is generated to request that new data be written to the Tx buffer. In some instances (when a message ends) this request may not be acknowledged. Issue this command to clear the interrupt/DMA request inside the MPSCC.

D_5 D_4 D_3
1 1 0 (Error Reset)

Resets an error bit and the EOF bits (D_4-D_7 of the SR1 register). If the special Rx condition interrupt occurs in the first Rx character interrupt mode, the characters loaded to the Rx buffer following the character that caused the interrupt will not be assembled until this command is issued.

D_5 D_4 D_3
1 1 1 (End of Interrupt)

Resets the in service latch that has the higher priority among the in service latches which are set. Set on channel A only. If there is an interrupt request in service whose process currently is terminated, interrupt request processing will be resumed unless $\overline{\text{INTAK}}$ is generated to request an interrupt assigned a higher priority than the interrupt in service. In this manner, the end of interrupt command enables an interrupt with lower priority in the daisy chain. Priority status will be reanalyzed to determine which interrupt takes precedence, if an INTAK sequence occurs from the interrupt with a priority higher than the one in service.

D_2 - D_0 (Pointer Bits)

These bits specify the register to be used for the next read or write operation. Generally, each register is accessed by the MPSCC according to this procedure. The pointer indicates 000 after an external reset or a channel reset command.

Pointer Specification. The pointer bits specify a binary register number used to perform a read or write operation.

Read/Write Operation. The register specified as a pointer will be accessed if the read or write operation is performed when C/$\overline{\text{D}}$ is 1.

Pointer Restoration. The pointer bits return to 000 after the read or write operation has been performed. Therefore, do not respecify the pointer bits when the CR0 or SR0 register is accessed.

Example 1 (Setting CR2A). The pointer bits of the CR0A register specify 2 (D_2 - D_0 = 010). The CR2A register inputs data written during the next write to the MPSCC, for instance, $\overline{CS}$ = 0, C/$\overline{D}$ = 1.

Example 2 (Reading SR3B). The pointer bits of the CR0B register specify 3 (D_2 - D_0 = 011). The SR3B register outputs the contents of the Tx length register.

The Tx length registers (TxLR-L, TxLR-H; 7501A only) use a special data setting method. The values of these registers are set (without specifying the pointer) by a series of two write cycles that follow writing a 1 to bit D_6 of the CR1 register.

Example 3 (Setting TxLR of Channel A). The pointer bits of the CR0A register specify 1 (D_2 - D_0 = 001). Bit D_6 of the CR1A register is set to 1 and other necessary specifications in this register are then performed. The number of transmit data characters is set in TxLR-L and TxLR-H by the successive write cycles of CRxA.

CONTROL REGISTER 1 (CR1)

The functions of the CR1 register bits are summarized in table 4-6.

Table 4-6. CR1 Bit Functions

Bit	Name	Function
D_7	Wait Enable	Controls operation of $\overline{WAIT}$ pin
D_6	Tx Length Register Set	Instructs data setting of Tx length registers (7201A only)
D_5	Wait on Rx/Tx	Controls operation of $\overline{WAIT}$ pin
D_4	Rx INT Mode 1	Selects the mode for Rx interrupts
D_3	Rx INT Mode 0	
D_2	Status Affects Vector	Modifies the interrupt vector caused by an interrupt
D_1	Tx INT/DMA Enable	Enables the Tx interrupt/DMA
D_0	E/S Interrupt Enable	Enables the E/S interrupt

D_7 (Wait Enable) and D_5 (Wait on Rx/Tx)

These bits are meaningful only when both channels A and B are set in the interrupt mode (D_0 and D_2 of the CR2A register = 0). When D_5 is 1, $\overline{WAIT}$ becomes active (goes low) when an attempt is made to read receive (Rx) data when the MPSCC is not in the Rx character available state; that is, when the MPSCC has not received data. $\overline{WAIT}$ becomes inactive (high impedance) when the MPSCC enters the Rx character available state. When D_5 is 0, $\overline{WAIT}$ becomes active when data is written to the MPSCC when the Tx buffer is full. When the Tx buffer becomes empty, $\overline{WAIT}$ becomes inactive. Table 4-7 shows the state of bits D_7 and D_5 during receive and transmit operations.

Table 4-7. Bits D_7 and D_5 of CR1

		Function		
D_7	D_5	Rx/Tx Operation	Condition	WAIT Output
0	x	—	—	Hi-Z
1	0	Tx buffer	When data is written to the Tx buffer while it is full	Low
			When the Tx buffer is or becomes empty	Hi-Z
1	1	Rx buffer	When data is read from the Rx buffer when it is empty	Low
			When the Rx buffer is or becomes full	Hi-Z

Note: x = Don't care

D_6 (Tx Length Register Set)

This bit is used in the μPD7201A only. It is used to program the number of transmit data characters in the Tx length registers (TxLR-L and TXLR-H). The 16-bit value sent to the μPD7201A after this bit is set to 1 will be loaded to the Tx length registers. The first write to the chip ($\overline{WR}$ = 0) with the C/$\overline{D}$ line equal to 1 will load the lower eight bits of the Tx length from the data bus into the TxLR-L register. The next command write will load the high-order byte into TxLR-H. When transmission is begun after this mode has been set, the transmission length counter will be incremented each time the $\overline{INT}$ or DRQTx lines become active. When this counter value equals the value loaded into the TxLR, the Tx interrupt or DRQ request will be masked. When a DMA request or interrupt is masked, the value of the counter will be set to 0.

The Tx length registers can be rewritten by again setting bit D_6 of CR1. If the TxLR is rewritten, the interrupt or DMA request masking is reset. Transmission should not be enabled (D_3 of CR5 should not be set) until the TxLR is programmed.

During HDLC transmission in this mode, a transmission underrun condition will cause the transmission of the CRC bytes if the value in the Tx length counter equals the value in the TxLR. If these values are not equal at the Tx underrun, then an abort sequence is automatically transmitted.

Once this bit has been set, a channel reset command or an external reset must be given to reset this mode. Writing a 0 to D_6 in CR1 will not reset it.

D_4 and D_3 (Rx Interrupt Modes 0-3)

These bits determine the mode of an interrupt caused by character reception. In modes 1-3, the special receive condition interrupt occurs. When the DMA mode is selected by

INT DMA mode bits (D_0, D_1 of the CR2A register), and when the Rx interrupt mode bits are set in any mode other than mode 0, DMA requests are generated on the Rx character available condition. Table 4-8 shows the function of bits D_4 and D_3 in Rx interrupt mode.

Table 4-8. Bits D_4 and D_3 of CR1

Mode	D_4	D_3	Interrupt Mode	Parity Error
0	0	0	Rx INT/DMA disabled	— —
1	0	1	First character interrupt	Excluded from Special Rx Condition
2	1	0	All character interrupt	Included in Special Rx Condition
3	1	1	All character interrupt	Excluded from Special Rx Condition

D_2 (Status Affects Vector)

This bit is set only at channel B. If a vectored mode (D_5 of the CR2A register = 1) is specified, an interrupt vector (vector bits V_7 - V_0) is output to the data bus in the interrupt acknowledge cycle. The status affects vector bit is used to determine whether the contents of an interrupt vector should be modified according to the interrupt source.

When this bit is 0, the interrupt vector set at the CR2B register is output as it is. When the bit is 1, Vector bits V_4-V_2 of the interrupt vector set at the CR2B register are modified if 85 mode is specified. In 86 mode, vector bits V_2-V_0 are modified as shown in table 4-9.

Table 4-9. Modification of Vector Bits V_2-V_0 or V_4-V_2

		V_2	V_1	V_0	86 mode
Interrupt Cause	**Channel**	V_4	V_3	V_2	**85 mode**
Tx Buffer Empty	A	1	0	0	
	B	0	0	0	
External/Status Change	A	1	0	1	
	B	0	0	1	
Rx Character Available	A	1	1	0	
	B	0	1	0	
Special Rx Condition	A	1	1	1	
	B	0	1	1	

D_1 (Tx Interrupt/DMA Enable)

This bit enables the Tx interrupt or DMA. When D_1 is set and when the first Tx data byte is written to the MPSCC, an interrupt or DMA takes place each time the Tx buffer becomes empty. However, when bit D_6 of the CR1 register is 1, an interrupt/DMA request will be generated each time the Tx buffer becomes empty even if the first Tx data byte is not yet written to the MPSCC (7201A only).

D_0 (External/Status Interrupt Enable)

This bit enables the E/S interrupt. The E/S interrupt is enabled when the following occurs:

- The level of an input signal to the $\overline{CTS}$, $\overline{DCD}$, or $\overline{SYNC}$ pin changes.
- The start or end of a break condition or abort sequence is detected.
- The transmission of CRC, SYNC, or flag characters starts in the Tx Underrun condition.
- The transmission of a frame has been completed in HDLC mode (7201A only).

CONTROL REGISTER 2, CHANNEL A (CR2A)

The functions of the CR2A register bits are summarized in table 4-10.

Table 4-10. CR2A Bit Functions

Bit	Name	Function
D_7	RTSB/SYNC Select	Selects the function of the RTSB/SYNCB pin
D_6	Rx INT Mask (7201A only)	Masks an interrupt caused by a first Rx character
D_5	Vector Mode	Selects nonvectored or vectored mode
D_4	INT Mode 1	Selects interrupt mode 85-1, 85-2, 85-3, or 86
D_3	INT Mode 0	
D_2	Priority Select	Selects the priority of an interrupt or DMA request
D_1	INT/DMA Mode 1	Specifies the interrupt or DMA mode of each channel
D_0	INT/DMA Mode 0	

D_7 (RTSB/SYNCB Select)

This bit specifies the function of the $\overline{RTSB}/\overline{SYNCB}$ pin. When 0, selects the RTSB function; when 1, selects the $\overline{SYNCB}$ pin.

D_6 (Rx INT Mask)

If bits D_4 and D_3 of the CR1 register are 0 and 1 (INT on first character mode), an interrupt caused by the first received character is masked when D_6 is set (7201A only). While D_6 is set, the Rx $\overline{INT}$ signal will not become active even if the enable $\overline{INT}$ on next Rx character command is issued. When D_6 = 0, no interrupt is masked. This bit must be zero on the μPD7201.

D_5 (Vector Mode)

This bit specifies the nonvectored or vectored mode. When 0, the nonvectored mode is selected. Nonvectored mode is used for the CPU that cannot read interrupt vectors in the interrupt acknowledge cycle. When 1, vectored mode is specified. In vectored mode, when the MPSCC responds to the state of the interrupt mode bits, $\overline{\text{PRI}}$ and an internal interrupt source, the MPSCC outputs either a CALL code or an interrupt vector to the data bus when the $\overline{\text{INTAK}}$ pin receives an active signal. If the CALL or an interrupt vector is not output, the data bus becomes high impedance.

D_4 and D_3 (Interrupt Mode)

These bits selects the mode in which the MPSCC responds when it receives the $\overline{\text{INTAK}}$ signal. Also, they determine the vector bits (V_7 - V_0 of the SR2B register) of an interrupt vector to be modified in status affects vector mode. Table 4-11 shows the states of bits D_4 and D_3 in the interrupt modes.

Table 4-11. Bits D_4 and D_3 of CR2A

D_4	D_3	Mode	Interrupt Vector Bits for Modification (Status Affects Vector Mode)
0	0	85-1	$V_4\ V_3\ V_2$
0	1	85-2	
1	1	85-3 (7201A only)	
1	0	86	$V_2\ V_1\ V_0$

D_2 (Priority Select)

This bit Determines the priority order of TxA (channel A) and RxB (channel B) when an interrupt or DMA occurs inside the MPSCC. The priority of the other interupt sources, such as an E/S interrupt, is fixed (see table 4-12).

D_1 and D0 (INT/DMA Modes 1 and 0)

These bits specify an interrupt or DMA mode for each channel. The pin functions necessary for each system configuration (DMA request generation and recognition performed by the DRQRx, DRQTx, or $\overline{\text{HAI}}$ pin, and MPSCC daisy chain performed by the $\overline{\text{HAI}}$, $\overline{\text{HAO}}$, $\overline{\text{PRI}}$, or $\overline{\text{PRO}}$ pin) is selected and defined by D_1 and D_0 (see Section 6). When both D_1 and D_0 = 1, there is no priority for DMA operations (7201A only). The priority of the special Rx condition is the same for each channel.

In table 4-12, when bit D_1 or D_0 is 1, channel A is set in the DMA mode, and a DMA request is generated when the MPSCC is in the Rx character available state (DRQRxA specified) or when the Tx buffer is empty (DRQTxA specified). Channel B is set in the interrupt mode. It is also possible that the E/S interrupt or Rx interrupt occurs at channel A. Usually in this case, the INT or first character bit is selected when channel A is set in the Rx interrupt mode. Consequently, the first received character will cause an interrupt. In the Rx interrupt mode, bit D_2 can be set to either 0 or 1 without causing any change in interrupt or DMA priority.

Table 4-12. INT/DMA Priorities

CR2A			INT/DMA Mode		Pin Function						Priority					
D_2	D_1	D_0	Channel A	Channel B	$\overline{\text{WAITA}}$/DRQRxA $\overline{\text{WAITB}}$/DRQTxA		$\overline{\text{PRI}}$/DRQRxB $\overline{\text{PRO}}$/DRQTxB		$\overline{\text{DTRB}}$/$\overline{\text{HAKI}}$ $\overline{\text{DTRA}}$/HAKO		High					Low
0	0	0	INT	INT	$\overline{\text{WAITA}}$	$\overline{\text{WAITB}}$	$\overline{\text{PRI}}$	$\overline{\text{PRO}}$	$\overline{\text{DTRB}}$	$\overline{\text{DTRA}}$	RxA	TxA	RxB	TxB	E/SA	E/SB
1	0	0	INT	INT							RxA	RxB	TxA	TxB	E/SA	E/SB
x	0	1	DMA		DRQRxA	DRQTxA	PRI	$\overline{\text{PRO}}$	$\overline{\text{HAI}}$	$\overline{\text{HAO}}$	RxA	TxA				
			INT								RxA*		RxB	TxB	E/SA*	E/SB
0	1	0	DMA	DMA	DRQRxA	DRQTxA	DRQRxB	DRQTxB	$\overline{\text{HAI}}$	$\overline{\text{HAO}}$	RxA	TxA	RxB	TxB		
											RxA*		RxB*		E/SA*	E/SB*
1	1	0	DMA	DMA							RxA	RxB	TxA	TxB		
											RxA*	RxB*			E/SA*	E/SB*
x	1	1	DMA	DMA					$\overline{\text{DTRB}}$	$\overline{\text{DTRA}}$	No Priority					
											RxA*	RxB*			E/SA*	E/SB*

CONTROL REGISTER 2, CHANNEL B (CR2B)

The functions of the CR2B register bits are summarized in table 4-13.

Table 4-13. ***CR2B Bit Functions***

Bit	Name	Function
D_7 V_7 D_6 V_6 D_5 V_5 D_4 V_4 D_3 V_3 D_2 V_2 D_1 V_1 D_0 V_0	Interrupt Vector	Sets interrupt vectors sent in the interrupt acknowledge cycle. When Status Affects Vector bit = 0, vector bits V_7 - V_0 are output. When Status Affects Vector bit = 1, Vector Bits V_7 - V_5, and V_1, V_0 are modified. Vector Bits V_4 - V_2 are output in 85 mode. Vector bits V_2 - V_0 are output in 86 mode.

CONTROL REGISTER 3 (CR3)

The functions of the CR3 register bits are summarized in table 4-14.

Table 4-14. ***CR3 Bit Functions***

Bit	Name	Function
D_7	Rx Bit/Character 1	Define number of bits per serial data character.
D_6	Rx Bit/Character 0	
D_5	Auto Enable	Controls transmission/reception operation by input of a $\overline{CTS}$ or $\overline{DCD}$ signal.
D_4	Enter Hunt Phase	Sets the Hunt phase.
D_3	Rx CRC Enable	Starts (or restarts) the CRC calculation operation.
D_2	Address Search Mode	Receives a message if the receive address matches either that of CR6 or a global address.
D_1	SYNC CHAR Load Inhibit	Inhibits SYNC characters from being loaded to the Rx buffer.
D_0	Rx Enable	Starts reception operation.

D_7 and D_6 (Rx Bits/Characters 1 and 0)

These bits define the number of bits per character of serial data to be received. Although the number of bits in a character can be modified during serial data reception, any modification must be done before the character reaches the MPSCC. Table 4-15 shows the states of bits D_7 and D_6 and how they define the received character.

Table 4-15. ***Bits D_7 and D_6 of CR3***

			8-bit Character After Assembly								
D_7	D_6	Bits/ Character	MSB							LSB	Mode
0	0	5	1	1	1/P[1]	D_4	D_3	D_2	D_1	D_0	Asynchronous
			x[2]	x	x/P[3]	D_4	D_3	D_2	D_1	D_0	Synchronous HDLC
1	0	6	1	1/P	D_5	D_4	D_3	D_2	D_1	D_0	Asynchronous
			x	x	D_5	D_4	D_3	D_2	D_1	D_0	Synchronous HDLC
0	1	7	1/P	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Asynchronous
			x/P	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Synchronous HDLC
1	1	8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Asynchronous
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Synchronous HDLC

Notes: [1] 1/P is 1 when the parity enable bit (D_0 of the CR4 register) is 0. Used for parity when the parity enable bit is 1.

[2] x = the unused bit(s) of the received character.

[3] x = the unused bit in the parity disable state (D_0 of the CR4 register = 0). Used as a parity bit in the parity enable state (D_0 of the CR4 register = 1).

D_5 (Auto Enable)

This bit enables or disables data transmission (Tx) or reception (Rx) in response to the modem signal input to either the $\overline{CTS}$ or $\overline{DCD}$ pin. When the auto enable bit is 1 and the Tx or Rx enable bit is 1, the transmitter or receiver will be enabled while the $\overline{CTS}$ or $\overline{DCD}$ signal is active. The Tx or Rx operation is disabled when the $\overline{CTS}$ or $\overline{DCD}$ signal becomes inactive or when the Tx/Rx enable bit is 0. When the auto enable bit is 0, the $\overline{CTS}$ and $\overline{DCD}$ inputs are treated only as general-purpose inputs. In other words, input of the two signals has no effect on the Tx or Rx operation; it only causes the status bits (D_5 and D_3 of the SR0 register) to be set.

D_4 (Enter Hunt Phase)

This bit directs the MPSCC to enter the hunt phase when the synchronous or HDLC mode is specified. The hunt operation occurs when the MPSCC enters the Rx enable state after the enter hunt phase bit is set. The MPSCC goes out of the hunt phase when it is synchronized and detects the next state.

When the enter hunt phase bit is set in the internal synchronous or HDLC mode, the SYNC/hunt bit (D_4 of the SR0 register) also is set. When the MPSCC is reset, it enters the hunt phase after it has been released from the Rx enable state. For this reason, the enter hunt phase bit need not be set. This bit also enables the MPSCC to reenter the hunt phase if for some reason established character synchronization is lost in the synchronous mode, or if an input frame is unnecessary in the HDLC mode.

D_3 (Receive CRC Enable)

This bit directs the start or restart of CRC calculation (1 = start). This bit must be changed before the character following the one involved in the CRC calculation is loaded into the Rx buffer. If there is no character stored in the Rx buffer, check the Rx character available bit to find when to set the bit.

D_2 (Address Search Mode)

This bit is valid only in the HDLC mode. When the address search mode bit is set in the HDLC receive mode, a received secondary address (the first character following the flag) is compared with either the secondary reference address in the CR6 register or the global address 11111111. If no match is made, the Rx interrupt is prevented, the message following the received secondary address will not be accepted, and the character will not be assembled. This mode is effective in systems using more than one secondary MPSCC and when a secondary MPSCC selectively receives transmitted messages.

D_1 (SYNC Character Load Inhibit)

This bit prevents SYNC characters from being loaded into the Rx buffer. This mode must normally be reset after the MPSCC detects a SYNC character. This prevents the mode from being applied to SYNC characters inserted in a message. In this mode, CRC calculation is also performed on SYNC characters that have not been loaded into the Rx buffer. This bit remains valid even after the MPSCC has exited the hunt phase.

D_0 (Rx Enable)

This bit enables start of the receive operation. The Rx enable bit is set after receive parameters are set and the receiver is initialized.

CONTROL REGISTER 4 (CR4)

The functions of the CR4 register bits are summarized in table 4-16.

Table 4-16. ***CR4 Bit Functions***

Bit	Name	Function
D_7	Clock Rate 1	Specify the clock rate. Must be the data rate times 1, 16, 32, or 64
D_6	Clock Rate 0	
D_5	SYNC Mode 1	Select the character synchronization method
D_6	SYNC Mode 0	
D_3	Stop Bit 1	Select the transmit stop bit length
D_2	Stop Bit 0	
D_1	Parity Even Odd	Selects even or odd parity
D_0	Parity Enable	Enables transmission or checks parity bit

D_7 and D_6 (Clock Rates 1 and 0)

These bits specify the multiple by which the clock rate (RxC and TxC) should be divided to obtain the desired data transmission rate (asynchronous mode). In the synchronous mode, a clock rate of x1 must be used. Table 4-17 shows how the D_7 and D_6 bits select the clock rate.

Table 4-17. ***Bits D_7 and D_6 of CR4***

D_7	D_6	Clock Rate (RxC, TxC)	System Clock Rate (CLK)
0	0	(Data rate) × 1	(Data Rate) × 4.5 minimum
0	1	(Data rate) × 16	
1	0	(Data rate) × 32	
1	1	(Data rate) × 64	

Note: When clock rate × 1 is specified, bit synchronization must be established externally.

D_5 and D_4 (SYNC Modes 1 and 0)

These bits select the character synchronization method and the SYNC character length. Set these bits to 00 in the asynchronous mode and 10 in the HDLC mode. Table 4-18 shows how D_5 and D_4 specify the synchronous mode and character.

Table 4-18. ***Bits D_5 and D_4 of CR4***

D_5	D_4	Synchronous Mode	Synchronization Character
0	0	Monosync	8-bit SYNC
0	1	Bisync	16-bit SYNC
1	0	HDLC	Flag pattern (01111110)
1	1	External Sync	None

D_3 and D_2 (Stop Bits 1 and 0)

These bits specify the stop bit length for transmission in the asynchronous transmit mode. In the receive mode, a stop bit check is performed assuming that the length of a stop bit is 1. Set these bits to 00 in the synchronous mode (including HDLC mode). Table 4-19 shows how bits D_3 and D_2 specify the stop bit length.

Table 4-19. ***Bits D_3 and D_2 of CR4***

D_3	D_2	Stop Bit Length (Bit/Char)
0	0	Synchronous mode (including HDLC)
0	1	1
1	0	1½
1	1	2

D_1 (Parity Even/Odd)

This bit selects even or odd parity and is valid while the parity enable bit is set. When the parity even/odd bit is 1, even parity is selected. When it is 0, odd parity is selected.

D_0 (Parity Enable)

Setting this bit to 1 enables transmission of a parity bit and allows checks of parity error. In the Tx mode, a parity bit is added to the data bits of each character and the character is transmitted. The number of bits making up a character is specified by bits D_6 and D_5 of the CR5 register. In the Rx mode, the MPSCC performs a parity check, assuming that the data bits for each character are followed by a parity bit. Also, the number of data bits making up a character is determined by bits D_7 and D_6 of the CR3 register. If a parity error occurs, the error bit is set.

Note that the parity bit is treated as part of a character and transmitted that way to the CPU. The parity bit, however, is not included in a character if the character is already 8 bits long.

CONTROL REGISTER 5 (CR5)

The functions of the CR5 register bits are summarized in table 4-20.

Table 4-20. ***CR5 Bit Functions***

Bit	Name	Function
D_7	Data Terminal Ready	Controls the DTR pin
D_6	Tx Bits/Character 1	Define the number of bits of the serial transmit data character
D_5	Tx Bits/Character 0	
D_4	Send Break	Transmits a break sequence
D_3	Tx Enable	Starts the Tx operation
D_2	CRC-16/CCITT	Controls the CRC polynomial
D_1	Request to Send	Controls the RTS pin
D_0	Tx CRC Enable	Enables the CRC calculation or transmission

D_7 (Data Terminal Ready)

This bit controls the $\overline{DTR}$ pin. When the $\overline{DTRA}/\overline{HAO}$ or $\overline{DTRB}/\overline{HAI}$ pin is used as the $\overline{DTR}$ pin (D_1 and D_0 of the CR2A register = 00), the state of this bit is inverted as follows:

DTR bit	$\overline{DTR}$ pin
1	0
0	1

D_6 and D_5 (Transmit Bits/Character 1 and 0)

These bits define the number of bits per character of each serial transmit data byte. Bits D_7 - D_0 are valid transmit data bits. Table 4-21 shows how D_6 and D_7 specify the number of bits/character and the data byte written to the MPSCC.

Table 4-21. Bits D_6 and D_5 of CR5

D_6	D_5	Number of bits/character	Data byte written to the MPSCC							
			MSB							LSB
0	0	1	1	1	1	1	0	0	0	D_0
		2	1	1	1	0	0	0	D_1	D_0
		3	1	1	0	0	0	D_2	D_1	D_0
		4	1	0	0	0	D_3	D_2	D_1	D_0
		5	0	0	0	D_4	D_3	D_2	D_1	D_0
1	0	6	x	x	D_5	D_4	D_3	D_2	D_1	D_0
0	1	7	x	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

D_4 (Send Break)

When D_4 is a 1, the output signal from the TxD pin will be set immediately in the space state (0), and the contents of the Tx buffer will be lost. When 0, the output signal from the TxD pin will return to the mark state (1). However, if this bit remains 1 for a shorter period of time than that required for transmission of one character, the output from the TxD pin will not be in the mark state even after this bit has been reset to 0. In this case, part of the internal transmit data will be transmitted. Therefore, the time during which this bit is set to 1 must be long enough for the specified one character transmission rate.

D_3 (Transmit Enable)

When D_3 is a 1, it enables the start of data transmission. The auto enable mode (see D_5, auto enable in this section) makes it possible to enable the transmitter after the $\overline{CTS}$ pin becomes active. The transmitter is disabled when the transmit enable bit is 0. This sets the signal output from the TxD pin in the mark state. If the transmit enable bit becomes 0 while data is being transmitted, the TxD pin will return to the mark state as soon as all transmit characters (such as data, SYNC character, and flag) have been transmitted. If a CRC character is transmitted when the transmit enable bit is 0, it will be replaced by either a SYNC character or a flag.

D_2 (CRC-16/CCITT)

This bit selects the CRC polynomial to be used by the Tx and Rx CRC calculator as shown in table 4-22.

Table 4-22. CRC Polynomial

D_2	Polynomial	Operation Mode	Preset Data (CR0: D_7, D_6)
1	CRC-16 (x16 + x15 + x2 + 1)	Synchronous	All 0s
0	CCITT-0 (x16 + x12 + x5 + 1)		
	CCITT-1 (x16 + x12 + x5 + 1)	HDLC	All 1s

D_1 (Request to Send)

D_1 controls the $\overline{RTS}$ pin as shown in table 4-23.

Table 4-23. $\overline{RTS}$ Pin Control

Mode	D_1	$\overline{RTS}$ Pin
Asynchronous	1	0
	0	Becomes 1 when all characters have been transmitted and the Tx buffer is empty.
Synchronous	1	0
HDLC	0	1

D_0 (Transmit CRC Enable)

D_0 enables both CRC calculation and CRC character transmission. Normally, it must be set before the first data (bisync mode) or address field (HDLC mode) is transmitted to the MPSCC. The transmit CRC enable bit becomes valid when a character is transferred to the Tx shift register from the Tx buffer. Therefore, when the CRC calculation is enabled or disabled during message transmission, this bit must be set or reset before the data character involved in the CRC calculation is transmitted.

If the Tx underrun condition occurs when the transmit CRC enable bit is set and when the Tx underrun/EOM bit is reset, a CRC character will be transmitted. If the Tx underrun condition occurs when the transmit CRC enable bit is reset, no CRC character will be transmitted, and a SYNC character or flag will be transmitted instead.

CONTROL REGISTERS 6 AND 7 (CR6 and CR7)

The functions of the CR6 and CR7 register bits are summarized in table 4-24.

Table 4-24. ***CR6 and CR7 Bit Functions***

Bit	**Name**		**Function**
	CR6	**CR7**	
D_7	SYNC Bit 7	Sync Bit 15	SYNC characters (Synchronous mode)
D_6	SYNC Bit 5	SYNC Bit 14	
D_5	SYNC Bit 5	SYNC Bit 13	Flag, secondary address (HDLC mode)
D_4	SYNC Bit 4	SYNC Bit 12	
D_3	SYNC Bit 3	SYNC Bit 11	
D_2	SYNC Bit 2	SYNC Bit 10	
D_1	SYNC Bit 1	SYNC Bit 9	
D_0	SYNC Bit 0	SYNC Bit 8	

These characters are written to the CR6 and CR7 registers according to the specified mode shown in table 4-25.

Table 4-25. ***CR6 and CR7 Characters***

Mode	**CR6**	**CR7**
Monosync	Tx SYNC Character	Rx SYNC Character
Bisync	SYNC character 1 (First 8 of 16 bits)	SYNC character 2 (Last 8 of 16 bits)
EXT Sync	Tx SYNC character	Not used
HDLC	Secondary address (For comparison with an address field)	Flag character (011111110)

STATUS REGISTER 0 (SR0)

The functions of the SR0 register bits are summarized in table 4-26.

Table 4-26. ***SR0 Bit Functions***

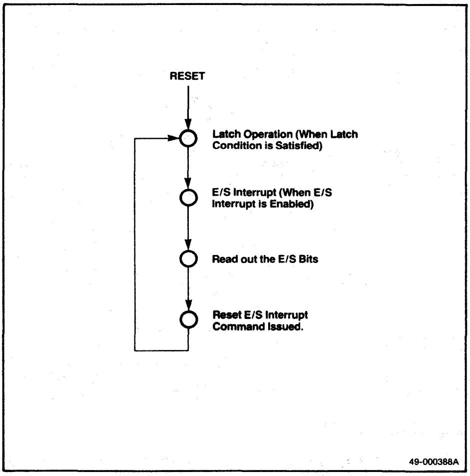
Bit		**Name**	**Function**
D_7	E/S bits	Break/Abort	Detects the break or abort sequence.
D_6		Tx Underrun/EOM	Detects the Tx Underrun condition.
D_5		CTS	Monitors input to the $\overline{CTS}$ pin.
D_4		SYNC/Hunt	Monitors input to the $\overline{SYNC}$ pin or the Hunt phase.
D_3		DCD	Monitors input to the $\overline{DCD}$ pin.
D_2		Tx Buffer Empty	Checks for data in the Tx buffer.
D_1		Interrupt Pending	Checks for an interrupt source whose process has not been completed.
D_0		Rx Character Available	Checks for data in the Rx buffer.

D_7 - D_3 (E/S Bits)

These bits indicate (or latch) the states of the modem control pins and data transmission/reception.

The E/S bits latch states in a special manner. When one satisfies a predetermined condition, such as inversion of a bit, each bit latches the respective state at that point. The E/S interrupt then occurs. Once a latch occurs, another latch operation will not be performed until the reset E/S INT command is issued, even if the condition of each bit required to perform another latch operation is satisfied. Figure 4-1 shows E/S bit operation.

Figure 4-1. ***E/S Bit Operation***



D_7 (Break/Abort)

This bit indicates whether the MPSCC has received a break sequence in the asynchronous mode or an abort sequence in the HDLC mode. If the MPSCC receives a break or abort sequence during latch operation of the E/S bits, the break/abort bit becomes 1 and latches that state as long as the sequence is received. If the MPSCC does not receive a break or abort sequence, the break/abort bit remains 0. In other words, this bit becomes 1 when a break or abort sequence is detected among receive data and remains 0 when either the MPSCC detects the end of the break or abort sequence or when an E/S interrupt occurs. The reset E/S INT command is normally issued ($D_7 = 0$ or 1) immediately after the E/S interrupt occurs, enabling detection of either the end of the break or abort sequence or of the next break or abort sequence.

After the MPSCC detects the end of a break sequence, a null character (all 0s) received during the break sequence and consisting of one or more bytes will remain in the Rx buffer. Therefore, ignore the first data received immediately after the MPSCC has detected an end of the break sequence.

The break/abort bit is not used in synchronous mode.

D_6 (Transmit Underrun/End of Message)

This bit becomes 1 if the MPSCC is in the Tx underrun condition during latch operation of the E/S bits. If the MPSCC is not in the Tx underrun condition, this bit is 0. However, the transmit underrun/end of message bit also becomes 1 when the MPSCC is reset by either the channel reset command or the RESET signal and can be reset to 0 only when the Tx underrun/EOM command is issued by setting bits D_7 and D_6 of the CR0 register to 11. Bit D_6 is set to 0 when the first data before the flag pattern is written into the MPSCC in HDLC mode. However, the Tx underrun/EOM bit will become 1 and the E/S interrupt will occur if the MPSCC enters the Tx underrun condition when CRC character transmission is enabled. Therefore, the E/S interrupt will not occur when the reset Tx underrun/EOM command is issued (Tx underrun/EOM bit changes from 1 to 0). Additionally, latch operation of the other E/S bits will not occur. The Tx underrun/EOM command affects the operation of the MPSCC when it detects the end of a message, just as the Tx buffer empty bit does. See Section 5 for a description of operations in the synchronous and HDLC modes.

D_5 (Clear to Send)

This bit inverts the state of the $\overline{CTS}$ pin during a latch operation. When the clear to send bit is 1, the $\overline{CTS}$ pin is 0; when clear to send is 0, the $\overline{CTS}$ pin is 1. At the leading (0 to 1) or trailing (1 to 0) edge of the input signal to the CTS pin, the clear to send bit becomes 1 or 0 and the E/S interrupt occurs.

D_4 (SYNC/Hunt)

The function of this bit varies depending on the specified operating mode. In the asynchronous or external synchronous mode, the SYNC/hunt bit indicates inversion of the $\overline{SYNC}$ pin's signal level during latch operation of the E/S bits. That is, if this bit is 1, it indicates the $\overline{SYNC}$ pin is 0; if the bit is 0, the SYNC pin is 1. At the trailing edge (1 to 0) and leading edge (0 to 1) of the $\overline{SYNC}$ signal, this bit becomes 1 and 0, respectively. The E/S interrupt occurs whether the bit is 0 or 1.

In the internal synchronous mode (monosync or bisync) or HDLC mode, a 1 indicates that the hunt phase has been entered, and a 0 indicates that a SYNC character has been received. In either case, an E/S interrupt is generated.

D_3 (Data Carrier Detect)

This bit inverts the state of the $\overline{DCD}$ pin during latch operations. When the D_3 bit is 1, the $\overline{DCD}$ pin becomes 0; when it is 0, the $\overline{DCD}$ pin becomes 1.

At the leading edge of the input signal to the $\overline{DCD}$ pin, the data carrier detect bit becomes 0. At the trailing edge of the signal, the bit becomes 1 and the E/S interrupt occurs. Table 4-27 shows E/S bit operations.

Table 4-27. E/S Bit Operations

Bit	Name	Mode	Latch Condition of E/S Bits	State at Read of E/S Bits
D_7	Break/Abort	Async	Start of completion of break sequence	1: Break sequence 0: Other than break sequence
		HDLC	Start of completion of abort sequence	1: Abort sequence 0: Other than abort sequence
D_6	Tx Underrun/ EOM		Tx Underrun state when Tx CRC is enabled	1: Tx Underrun state 0: Other than Tx Underrun state (Reset Tx Underrun/EOM command. After issuance: 0. After reset: 1.
D_5	CTS		Leading or trailing edge of $\overline{CTS}$ input	1: $\overline{CTS}$ = 0 0: $\overline{CTS}$ = 1
D_4	SYNC/Hunt	Async EXT sync	Leading or trailing edge of $\overline{SYNC}$ input	1: $\overline{SYNC}$ = 0 0: $\overline{SYNC}$ = 1
		Monosync Bisync HDLC	Hunt phase <--> Character synchronization	1: During hunt 0: During character synchronization
D_3	DCD		Leading or trailing edge of $\overline{DCD}$ input	1: $\overline{DCD}$ = 0 0: $\overline{DCD}$ = 1

D_2 (Transmit Buffer Empty)

When this bit is a 1, it indicates that no data exists in the Tx buffer. This is not true while a CRC character is being transmitted in the synchronous or HDLC mode. When 0, it indicates that data exists in the Tx buffer or that a CRC character is being transmitted.

D_1 (Interrupt Pending)

This bit is applicable to only the SR0 register of channel A. SR0B bit D_1 is always 0. When the interrupt pending bit is 1, it indicates that an interrupt source exists in either channel A or B, or both. This bit is set if an interrupt source exists in the MPSCC whose service is in progress or being terminated (in service latch set). When this bit is 0, it indicates that the processes of all interrrupt sources have been completed (all in service latches are reset). If neither channel is in the DMA mode (D_1 and D_0 of the CR2A register = 1), the interrupt pending bit remains 0 as long as the input signal to the $\overline{PRI}$ pin is 1.

D_0 (Receive Character Available)

When 1, this bit indicates that one or more bytes of receive data are in the Rx buffer. When 0, it indicates that no receive data is in the Rx buffer.

STATUS REGISTER 1 (SR1)

The functions of register SR1 bits are summarized in table 4-28.

Table 4-28. SR1 Bit Functions

Bit	Name	Function
D_7	End of Frame	Indicates reception of flag pattern (usually a closing flag pattern).
D_6	CRC/Framing Error	Indicates CRC or Framing error.
D_5	Rx Overrun Error	Indicates an overrun error.
D_4	Parity Error	Indicates parity error.
D_3	Residue Code 2	Indicates the valid bit range of an information (I) field.
D_2	Residue Code 1	
D_1	Residue Code 0	
D_0	All Sent	Indicates completed transmission of all characters in the transmitter.

D_7 (End of Frame)

This bit is valid only in the HDLC mode. It is used to indicate reception of a valid closing flag and the validity of the CRC/framing bit and residue codes. On reception of a valid closing flag, the end of frame bit is set (special Rx condition interrupt occurs) and the error reset commmand is issued to clear this condition. When the character of the following frame is received and stored to the Rx buffer, the end of frame bit corresponding to the received character is reset. Also, the result of a CRC check (CRC/framing error bit) and residue codes are valid as long as this bit remains set.

D_6 (CRC/Framing Error)

This bit signals the occurrence of a CRC or framing error. If a framing error is detected in the asynchronous mode (0 is detected at the position of the stop bit), the CRC/framing error bit is set to 1, and the special Rx condition interrupt occurs. The framing error is reset when an error-free character is received or when the error reset command is executed.

Because detection of the stop bit is performed in response to the specified character length, the stop bit of the character will never be mistaken for the start bit of the following character even if the parity bit and stop bit of a character are respectively 1 and 0. This is because detection of the start bit of the following character is triggered at the leading edge of the reception line.

In the synchronous or HDLC mode, the CRC/framing error bit indicates the result of a comparison between the contents of the Rx CRC calculator and a received CRC character. As a result, if the received CRC character agrees with the contents of the Rx CRC calculator, the CRC/framing error bit becomes 0, indicating the normal state. If it does not agree, this bit becomes 1, indicating an error. However, the CRC/framing error bit also becomes 0 when the MPSCC is reset or when the error reset command is issued. The special Rx condition interrupt does not occur in the synchronous or HDLC mode.

CRC comparison is done on a character-by-character basis. Because the results of each comparison are indicated by the CRC/framing error bit, the CRC/framing error bit normally indicates the error state (1) during message reception. In other words, the bit remains 1 because the result of the comparison (performed up to the point when the received character is stored to the Rx buffer) is set in this bit, which corresponds to the Rx buffer. For this reason, this bit should be read after a 20-bit interval following completion of the last CRC character transmission (synchronous mode) or when the end of frame bit becomes 1 (HDLC mode).

D_5 (Receive Overrun Error)

The MPSCC sometimes receives more than three characters because the CPU delayed in fetching receive data. If data is written to the Rx buffer that exceeds the buffer's capacity, the receive overrun error bit will be set each time an attempt is made to write a character. When this bit is set, the special Rx condition interrrupt occurs when a character has caused the overrun error. This state is latched until the error reset command is issued.

D_4 (Parity Error)

This bit sets when a parity error occurs (parity enable bit set) because the specified odd or even parity does not match. Once the parity error bit is set, the state is retained until the error reset command is issued. When Rx interrupt mode 2 is specified (D_4 and D_3 of the CR1 register = 1 and 0 respectively), the parity error bit causes the special Rx condition interrupt.

D_3 - D_1 (Residue Codes 2 through 0)

Since the data portion of the an HDLC message can consist of a number of bits and not necessarily an integral number of characters, the MPSCC determines when the end of frame flag has been received and the boundary between valid data and the CRC character in the last few data characters read. When the end of frame condition is indicated (D_7 of SR1 is 1) and there is a special receive condition interrupt (if enabled), the last bits of the CRC character are in the receive buffer. The residue code for the frame is valid for the byte of SR1 that is associated with that data character.

Table 4-29. Residue Codes

8 Bits/Character

D_3 D_2 D_1	Previous Character	2nd Previous Character
1 0 0	C C C C C C C C	C C C C C D D D
0 1 0	C C C C C C C C	C C C C D D D D
1 1 0	C C C C C C C C	C C C D D D D D
0 0 1	C C C C C C C C	C C D D D D D D
1 0 1	C C C C C C C C	C D D D D D D D
0 1 1	C C C C C C C C*	D D D D D D D D*
1 1 1	C C C C C C C D	D D D D D D D D
0 0 0	C C C C C C C D	D D D D D D D D

7 Bits/Character

D_3 D_2 D_1	Previous Character	2nd Previous Character
1 0 0	C C C C C C C	C C C C C D D
0 1 0	C C C C C C C	C C C C D D D
1 1 0	C C C C C C C	C C C D D D D
0 0 1	C C C C C C C	C C D D D D D
1 0 1	C C C C C C C	C D D D D D D
0 1 1	C C C C C C C*	D D D D D D D*
0 0 0	C C C C C C D	D D D D D D D

6 Bits/Character

D_3 D_2 D_1	Previous Character	2nd Previous Character
1 0 0	C C C C C C	C C C C C D
0 1 0	C C C C C C	C C C C D D
1 1 0	C C C C C C	C C C D D D
0 0 1	C C C C C C	C C D D D D
1 0 1	C C C C C C	C D D D D D
0 0 0	C C C C C C*	D D D D D D*

5 Bits/Character

D_3 D_2 D_1	Previous Character	2nd Previous Character
1 0 0	C C C C C*	D D D D D*
0 1 0	C C C C D	D D D D D
1 1 0	C C C D D	D D D D D
0 0 1	C C D D D	D D D D D
0 0 0	C D D D D	D D D D D

Notes: 1. C = CRC bit
2. D = Valid data
3. * = No residue

The meaning of the residue code depends on the number of bits/character specified for the receiver. Table 4-29 shows the residue codes for 8, 7, 6, and 5 bits/character specified for the receiver. The previous character refers to the last character read before the end of frame, and so on.

D_0 (All Sent)

In the asynchronous transmit mode, this bit is set when all characters have been transmitted and the Tx buffer and Tx shift register become completely empty. No interrupt is caused by a change in the state of the all sent bit. In the synchronous and HDLC modes (D_6 of the CR1 register = 0), this bit is always set.

In HDLC mode (D_6 of the CR1 register = 1; 7201A only) the level of this bit changes from 0 to 1 after a series of flags have been transmitted causing the E/S interrupt to occur. The host system is notified that the transmission of one frame is complete. Although the level of this bit changes from 1 to 0 on transmittal of the CRC character of the next frame, this change does not cause the E/S interrupt.

STATUS REGISTER CHANNEL B (SR2B)

The bits in this register indicate an interrupt vector that can be read only through channel B. When the status affects vector is 0, SR2B will indicate the vector which was last programmed in CR2B. If the status affects vector bit is 1 and an interrupt request exists in the MPSCC, the bits of SR2B will be modified according to the interrupt source. The bits modified are V_4-V_2 (8085 mode) or V_2-V_0 (8086 mode) of CR2B.

This interrupt vector corresponds to the interrupt source assigned the top priority among the interrupt requests. If no interrupt request exists, vector bits V_4-V_2 of the CR2B register are set to 1s in 85 mode or V_2-V_0 are set to 1s in 86 mode. See table 4-30.

Table 4-30. Vector Bits V_4-V_0

8085 Mode 8086 Mode	V_4 V_2	V_3 V_1	V_2 V_0	Condition
	1	1	1	No Interrupt Pending
	0	0	0	Channel B Transmitter Buffer Empty
	0	0	1	Channel B External/Status Change
	0	1	0	Channel B Received Character Available
	0	1	1	Channel B Special Receive Condition
	1	0	0	Channel A Transmitter Buffer Empty
	1	0	1	Channel A External/Status Change
	1	1	0	Channel A Received Character Available
	1	1	1	Channel A Special Receive Condition

STATUS REGISTERS 3 AND 4 (SR3 and SR4)

SR3

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Tx Length Counter Bits 7 - 0							

SR4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Tx Length Counter Bits 15 - 8							

These counters are only available on the µPD7201A. They can be used when data transmission is performed in the TxLR set mode (D_6 of the CR1 register = 1) to indicate the number of transmit data characters. The values of the counters are cleared when the system is reset, when the contents of the Tx length register high byte (TxLR-H) and low byte (TxLR-L) coincide with those of the SR4 and SR3 registers, or when the Tx length register is set by setting D_6 of the CR1 register to 1. However, in HDLC mode, when the TxLR set mode is specified as well (D_6 of the CR1 register = 1), the contents of the TxLR-H and TxLR-L registers are compared with those of the SR4 and SR3 registers when a Tx underrun condition exists. If the contents of the counters are not equal to those of the Tx length registers, an abort sequence will be transmitted. When this occurs, the values of the SR4 and SR3 registers will be retained until the system is reset or the TxLR is set again by setting D_6 of the CR1 register to 1.

Tx LENGTH REGISTER, HIGH BYTE AND LOW BYTE (TxLR-H and TxLR-L)

TxLR-H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Tx Length Counter Bits 15 - 8							

TxLR-L

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Tx Length Counter Bits 7 - 0							

These registers are only available on the µPD7201A. They set the number of data characters to be transmitted. They can be set by two successive command write cycles which follow the setting of D_6 of the CR1 register to 1. During the first cycle, the command is written to the low-byte register. During the second cycle, the command is written to the high-byte register.

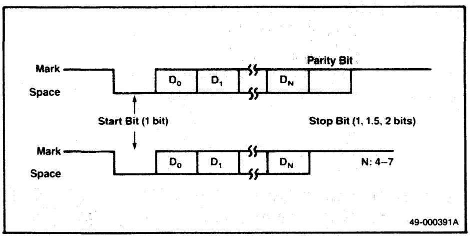
When the system has been externally reset, or when the channel has been reset, transmission must be enabled (D_3 of the CR5 register = 1) after the desired number of transmit data characters is sent to these registers. If the number of transmit data coincides with the values of these registers, the Tx $\overline{\text{INT}}$/DRQ signal will not become active (even if the Tx buffer is empty) until the desired number of transmit data characters is sent to the registers again (D_6 of the CR1 register = 1).

The MPSCC has three communication operating modes: asynchronous, synchronous, and HDLC. The contents of control registers (mainly CR3, CR4, and CR5) specify these modes. This section describes the operations performed in each mode.

ASYNCHRONOUS MODE

Figure 5-1 shows the format of serial data in the asynchronous mode. When data is transmitted, the MPSCC adds a start and a stop bit, and if necessary a parity bit. The transmit data consists of data bits D_0 through D_N. Each set of transmit data provided with a start, stop, and parity bit is a "character". When no transmit data exists, the serial circuit and transmit data pin TxD are set in the mark state.

Figure 5-1. Asynchronous Message Format



The start bit of the receive data indicates the start of the data. The stop bit indicates the end of the data block. When a data block is received, the MPSCC detects the start and stop bits, checks the parity bit, and deletes these bits from the data block. Therefore, the MPSCC transfers only data D_0 through D_N to the host system.

In asynchronous mode, data transmission or reception is controlled by registers CR0 to CR5. The CR6 and CR7 registers are not used. Registers CR1 and CR2 specify the mode of the host system that controls the MPSCC. In other words, these registers specify whether the host system should use interrupt, DMA, or polling mode to perform data transmission to and from the MPSCC. Register CR3 specifies the receive mode and CR5 specifies the transmit mode. Register CR4 specifies the mode commonly used for both transmission and reception.

Table 5-1 shows the set contents of registers CR3, CR4, and CR5.

Data Transmission in Asynchronous Mode

The CR4 register specifies clock mode, number of bits per character, and state of parity bits. The clock mode sets the data transmission rate by dividing the frequency input at TxC by 1, 16, 32, or 64. For example, if 64 is selected as the clock mode and the TxC pin has a clock of 64 kHz (15.625 µs clock cycle), the transmission data rate will be 1 kb/s.

Table 5-1. Contents of Registers CR3, CR4, and CR5 in Asynchronous Mode

CR3	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	Rx Bits/ CHAR 1	**Rx Bits/ CHAR 0**	**Auto Enable**	**Enter Hunt Phase**	**Rx CRC Enable**	**Address Search Mode**	**SYNC CHAR Load Inhibit**	**Rx Enable**
	00, 5 Bits/CHAR 01, 7 Bits/CHAR 10, 6 Bits/CHAR 11, 8 Bits/CHAR		0 Disable 1 Enable	0	0	0	0	0 Disable 1 Enable
CR4	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	Clock Rate 1	**Clock Rate 0**	**SYNC Mode 1**	**SYNC Mode 0**	**Stop Bits 1**	**Stop Bits 0**	**Parity Even/Odd**	**Parity Enable**
	00, x1 Clock Mode 01, x16 Clock Mode 10, x32 Clock Mode 11, x64 Clock Mode		0	0	01, 1 Stop Bit/CHAR 10, 1½ Stop Bits/CHAR 11, 2 Stop Bits/CHAR		0 Odd 1 Even	0 Disable 1 Enable
CR5	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	DTR	**Tx Bits/ CHAR 1**	**Tx Bits/ CHAR 0**	**Send Break**	**Tx Enable**	**CRC-16/ CCITT**	**RTS**	**Tx CRC Enable**
	0 $\overline{\text{DTR}}$ = 1 1 $\overline{\text{DTR}}$ = 0	00, 5 or Less Bits/CHAR		0 Marking 1 Spacing	0 Disable 1 Enable	0	0 $\overline{\text{RTS}}$ = 1 When all Sent 1 $\overline{\text{RTS}}$ = 0	0

Note: Set the parameters of register CR4 before those of the CR1, CR3, and CR5 registers.

The CR1 and CR2 registers specify the mode in which data is transmitted from the host system to the MPSCC. After setting the registers and the Tx enable bit (D_3 of the CR5 register), the TxD pin outputs the data's start bit. The start bit is output at the trailing edge of $\overline{TxC}$ at the transmission rate preset when the data is written to the MPSCC.

The data written to the MPSCC from the host system is then output starting from the LSB (least significant bit) of the data specified by bits D_5 and D_6 of the CR5 register. If data with a specified character length of 7 or 6 bits is transmitted to the MPSCC from the host system, the MPSCC deletes the unused bit(s) from the 8-bit data. Section 4 (Control Register 5) describes data transmitted with fewer than six bits. Parity and stop bits are automatically transmitted to the MPSCC following the data transmitted from the TxD pin.

When the transmit (Tx) buffer becomes empty while the first data byte is being transmitted, the Tx buffer empty bit (D_2 of the SR0 register) is set. If the interrupt mode is specified, an interrupt occurs and the MPSCC asks the host system to send the next data. If no transmit data characters remain for transmission, the TxD pin is set in the mark state (1).

Even if the Tx enable bit is set and data is written to the MPSCC from the host system, no data is output from the TxD pin in the auto enable mode (D_6 of the CR3 register is 1) unless $\overline{CTS}$ becomes 0. Data transmission is complete when no further data is transmitted to the MPSCC or when the Tx enable bit (D_3 of the CR5 register) is set to 0. The TxD pin is then set in the mark state, and the transmission operation is complete. Note the the TxD pin can also be set in the mark state by turning the $\overline{CTS}$ pin to 1 when the auto enable mode is specified.

To set the TxD pin in the space state (0), set the send break bit (D_4 of the CR5 register). This state will continue until the send break bit (independent of the Tx enable bit) is reset.

If the external status (E/S) interrupt mode is enabled (D_0 of the CR1 register is 1), the states of the $\overline{DCD}$, $\overline{CTS}$, and SYNC pins can be monitored. That is, each time the level of signals input to these pins changes, an interrupt occurs. In this interrupt routine, use the reset E/S INT command (D_5, D_4, and D_3 of the CR0 register are 0, 1, and 0, respectively) to again enable the E/S interrupt that occurs in response to these changes. The $\overline{RTS}$ pin outputs 1 when the transmission has been completed if the request to send bit (D_1 of the CR5 register) is reset. Under this condition, the all sent bit (D_0 of the SR1 register) is set.

Reception in Asynchronous Mode

Reception. Before characters are received in the asynchronous mode, set register CR4 to specify the length of transmit characters, data transmission rate, and whether a parity bit(s) is present. Set the character length with bits D_7 and D_6 of the CR3 register. The received data assembled by the MPSCC is transferred to the host system in the mode specified by the contents of the CR1 and CR2 registers.

After these registers are set and the Rx enable bit of the CR3 register is set, if RxD changes from 1 to 0 and remains 0 after a half-bit period of time, the MPSCC takes the bit as the start bit and assembles the data following it according to the character bit length specified.

If the parity enable bit is set (D_4 of the SR1 register), the MPSCC also detects it and checks for a parity error. When the MPSCC detects the stop bit following the parity bit, it finishes assembling the character. If the character length is eight bits and the parity enable bit has been set, the parity bit will not be transferred to the host system. If the length of a character is 7 bits (parity enable bit set) or less, the MPSCC assembles each character into character bit length + parity bit + 1s for the unused bits. For example, if a character is five bits long and the parity enable bit is set, the format of the assembled data is:

MSB LSB

1	1	P	D_4	D_3	D_2	D_1	D_0

Error Information During Reception. The MPSCC detects three types of errors: parity, framing, and overrun. If the parity bit of each received character does not coincide with the parity bit calculated by the MPSCC, the parity error bit (D_4 of the SR1 register) is set. The error status continues until the error reset command is issued.

Normally, the stop bit of a receive character is 1. If 0 is detected at that bit position, the framing error bit (D_6 of the SR1 register) will be set. If no error exists in the next character received, the framing error bit is reset. If x1 clock mode is not specified, the $\overline{RxC}$ input signals are counted after the trailing edge of the start bit has been detected and the MPSCC has determined the sampling period of the received data. If x1 clock mode is specified, however, the synchronization of the RxD and $\overline{RxC}$ pins must be externally established because the sampling of the received data is performed at the trailing edge of the $\overline{RxC}$ input signal. Even if a framing error occurs, the bit at the position of the stop bit of the preceeding data will not be taken as the bit of the next. However, if a null character is received with a framing error in it, the framing error bit will not be set. Instead, the break bit (D_7 of the SR0 register) is set.

An overrun error occurs when data exceeding the capacity of the Rx buffer is written to the MPSCC. This error occurs when the host CPU cannot read the receive data from the Rx buffer on time. Since the capacity of the MPSCC's buffer is three bytes, the Rx overrun error bit (D_5 of the SR1 register) is set if the MPSCC receives a character exceeding that number. The host system is then notified of the overrun. Should an overrun state occur, the fourth received character (or the excess character) will replace the third character in the Rx buffer, causing bit D_5 of the SR1 register to be set. An overrun error can be detected by reading the contents of the status register (SR1). This register indicates the state of the third receive character in the Rx buffer. The overrun error status continues until the error reset command is issued.

Using Receive Data and Received Status. Each channel of the MPSCC has a 3-byte receive buffer and error status register (SR1) that corresponds to the 3-byte buffer. In addition, the MPSCC has an Rx character available bit (D_0 of the SR0 register) and a break/abort bit (D_7 of the SR0 register) that are used as receive status bits.

When receive data is processed in the polling mode controlled by the software, the Rx character available bit is set if at least one character is loaded to the receive buffer. By polling this bit, the host system knows if the MPSCC has received a character.

The contents of the SR1 register must be read before characters are read from the receive buffer. A character is then read from the received buffer to relate the character read to the status of that character. In an interrupt mode, an interrupt occurs if the Rx character available bit is set. The contents of the SR1 register and receive data can then be read after this interrupt signal. The receive character must be read out after these contents have been read in the same manner as in the polling mode. If a parity, framing, or overrun error occurs, each will be treated as the special Rx condition interrupt. After an interrupt in first Rx character mode, receive interrupts occur only when the special Rx condition is satisfied, causing an error status to continue until the error reset command is issued. In other words, the SR1 register retains an error status corresponding to the character that caused it until the error reset command is issued, even if the receive character has been read from the buffer. This interrupt mode is effective when performing block transfer by software or DMA processing.

In all Rx character interrupt mode, receive characters must be read after the contents of the SR1 register have been read to detect any errors of the receive character. If a special Rx condition interrupt occurs in the all Rx interrupt mode, the SR1 register must be read to fetch receive characters. The error reset command is used to clear an error status. If the special Rx condition interrupt occurs, and the status affects vector bit (D_2 of the CR1 register) is set to 1, it can be distinguished from the normal receive interrupts by using interrupt vectors in both the first and the all Rx character modes.

In addition to the interrupts that occur in either the first or the all Rx character mode, an E/S interrupt occurs (if enabled) when a break sequence is detected and the state of the $\overline{DCD}$ pin changes. A break sequence is the continuation of a space state that exceeds the number of bits of a null character, resulting in a framing error. When a break sequence is detected, an E/S interrupt occurs at the start (trailing edge) and end (leading edge) of the break sequence.

Asynchronous Mode Sample

Figure 5-2 shows an example of full-duplex transmit or receive operations in the asynchronous mode, using both channels A and B. Figure 5-3 shows the interrupt process.

Figure 5-2. Asynchronous Mode Flowchart

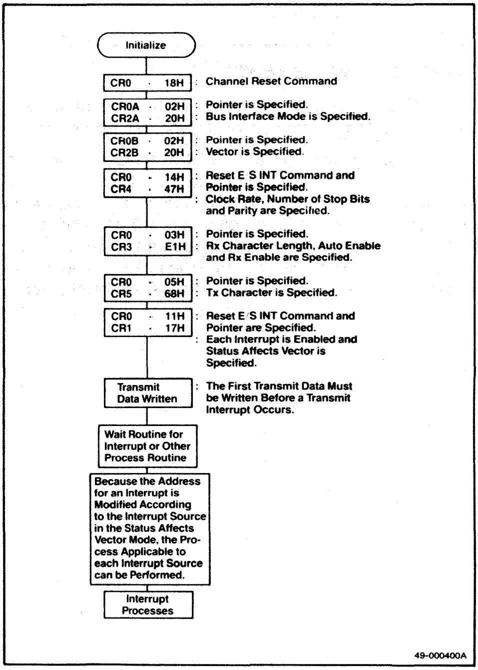
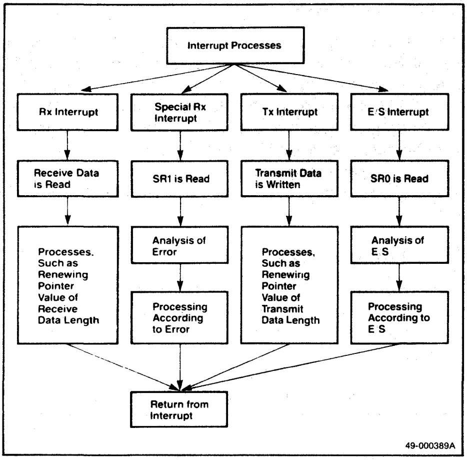


Figure 5-3. Interrupt Process



SYNCHRONOUS MODE

The MPSCC uses three types of byte-oriented synchronous protocols: monosync, bisync, and external sync.

In monosync protocol, an 8-bit SYNC character in the CR7 register coincides with serially received 8-bit data, and the MPSCC assumes character synchronization has been established. The characters following the 8-bit SYNC character(s) are then treated as received data. A SYNC character consisting of one or more bytes must be added to data at the start of data transmission to establish character synchronization. The SYNC character transmitted is set in the CR6 register.

In bisync protocol, a 16-bit SYNC character in the CR6 and CR7 registers coincides with serially received 16-bit data, and the MPSCC assumes that character synchronization has been established. The characters following the 16-bit SYNC character(s) are treated as receive data. A SYNC character consisting of one word (16 bits) or more must therefore be added to data at the start of data transmission. When the SYNC character is transmitted, the same SYNC character from the CR6 and CR7 registers is used.

In external sync protocol, character synchronization for reception must be externally established. The SYNC pin specifies the character synchronization timing. Detection of a SYNC character is therefore external. The input signal to the $\overline{\text{SYNC}}$ pin becomes 0 after a SYNC character has been detected. The SYNC character set in the CR6 register is sent at the start of a transmission.

Figure 5-4 shows the data format in the three synchronous protocols.

Figure 5-4. Synchronous SYNC Message Format

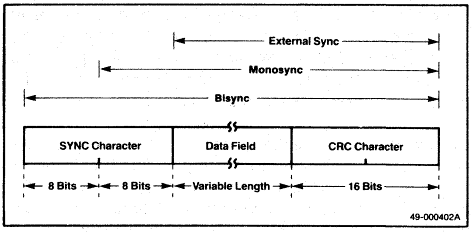


Table 5-2 shows the contents of registers CR3 through CR7 used in the synchronous mode.

Table 5-2. Contents of Registers CR3-CR7 in Synchronous Mode

CR3	**D_7**	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
	Rx Bits/ CHAR 1	**Rx Bits/ CHAR 0**	**Auto Enable**	**Enter Hunt Phase**	**Rx CRC Enable**	**Address Search Mode**	**SYNC CHAR Load Inhibit**	**Rx Enable**
	00, 5 Bits/CHAR 01, 7 Bits/CHAR 10, 6 Bits/CHAR 11, 8 Bits/CHAR		0 Disable 1 Enable	0 Nop 1 Re-enabled	0 Disable 1 Enable		0 Nop 1 Inhibit	0 Disable 1 Enable

CR4	**D_7**	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
	Clock Rate 1	**Clock Rate 0**	**SYNC Mode 1**	**SYNC Mode 0**	**Stop Bits 1**	**Stop Bits 0**	**Parity Even/Odd**	**Parity Enable**
	00		00, 8-Bit SYNC 01, 16-Bit SYNC 11, EXT SYNC		00 SYNC Mode		0 Odd 1 Even	0 Disable 1 Enable

CR5	**D_7**	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
	DTR	**Tx Bits/ CHAR 1**	**Tx Bits/ CHAR 0**	**Send Break**	**Tx Enable**	**CRC-16/ CCITT**	**RTS**	**Tx CRC Enable**
	0 $\overline{DTR}$ = 1 1 $\overline{DTR}$ = 0	00, 5 or Less Bits/CHAR 01, 7 Bits/CHAR 10, 6 Bits/CHAR 11, 8 Bits/CHAR		0 Marking 1 Spacing	0 Disable 1 Enable	0 CCITT-0 1 CRC-16	0 $\overline{RTS}$ = 1 1 $\overline{RTS}$ = 0	0 Disable 1 Enable

CR6	**D_7**	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
	SYNC Bit 7 – 0							
Monosync	Tx SYNC Character							
Bisync	SYNC Character Bit 7 – 0							
Ext Sync	Tx SYNC Character							

CR7	**D_7**	**D_6**	**D_5**	**D_4**	**D_3**	**D_2**	**D_1**	**D_0**
	SYNC Bit 15 – 8							
Monosync	Rx SYNC Character							
Bisync	SYNC Character Bit 15 – 8							
Ext Sync	not used							

Data Transmission in Synchronous Mode

Initialization for Transmission. Before transmitting data in the synchronous mode, initialize the control registers for the transmit operation. The CR4 register specifies whether monosync, bisync, or external sync mode is used. Set the parameters of the CR4 register before the parameters or commands of the CR1, CR3, CR5, and CR7 registers.

Specify the bit length of a transmit character and the CRC polynomial using the CR5 register. Load a SYNC character to the CR6 or CR7 register, or to both.

In the monosync or external sync mode, the contents of the CR6 register is used as the SYNC character. In the bisync mode, the contents of the CR6 and CR7 registers are used as the SYNC character. The CR1 and CR2 registers specify the mode in which data is transferred to the MPSCC from the host system.

After initialization of the transmit operation, the contents (SYNC character) of register CR6 (monosync or external sync modes) or CR6 and CR7 registers (bisync mode) are transmitted when the transmitter is enabled. If there are a number of SYNC characters to be transmitted before data transmission, delay writing the first transmit data character to the MPSCC until after transmission of several SYNC characters.

Write the first transmit data to the MPSCC without using an interrupt or DRQ signal. Afterwards, the interrupt or DRQ signal becomes active each time the transmit buffer becomes empty. If auto enable mode is selected, the data transmit operation will start when the transmitter has been enabled and the $\overline{\text{CTS}}$ pin has become active (low level).

In the synchronous mode, specify either CRC-16 or CCITT as the CRC polynomial and reset it to the initial value (0) by the initialize Tx CRC calculator command.

The TxD output pin remains in the mark state when the MPSCC is reset or when data transmission is disabled. When the send break bit (D_4 of the CR5 register) is set, the TxD (output) pin is immediately set in the space state (0) whether the transmitter is enabled or disbled. If the space state lasts too long, the contents of the transmit buffer and transmit shift register will be lost.

Data Transfer Mode. When transmit data is transferred to the MPSCC by an interrupt, set the parameters of the CR1 and CR2 registers first. The channel from which data is transferred must be set in the interrupt mode by D_1 and D_0 of the CR2A register. The Tx INT/DMA enable bit (D_1 of the CR1 register) must be set. When the first transmit data is written to the MPSCC after the transmitter has been enabled, the Tx interrupt will subsequently occur each time the transmit (Tx) buffer becomes empty. When an interrupt signal is used for data transfer, the data transfer is performed on a byte-by-byte basis until the message is completely transmitted.

When data transfer is by means of DMA, the DRQTx signal is used in place of the interrupt signal. When DMA mode is used, the transmit channel uses the DRQ signal as selected by the CR2A register, and the Tx INT/DMA enable bit in the CR1 register is set. When the transmitter has been enabled and the first transmit byte is written to the MPSCC, the DRQTx signal becomes active each time the Tx buffer becomes empty. The subsequent transmit data characters are written to the MPSCC by means of DMA on a one-by-one basis using the DRQTx signal.

If the MPSCC uses the $\overline{\text{WAIT}}$ pin before a data transfer, the wait on Rx/Tx bit (D_5 of the CR1 register) must to set to 0, and the wait enable bit (D_7 of the CR1 register) must be set to 1. The MPSCC prepares for the host system to fetch a transmit byte when the host system writes the first transmit byte to the MPSCC after the transmitter has been enabled. The host system then writes the next transmit byte to the MPSCC. The $\overline{\text{WAIT}}$ pin becomes active if the Tx buffer is not empty and inactive when it is empty. This mode is effective when the host system writes transmit data to the MPSCC in response to the software block transfer command.

When data transfer by software polling is used, transmit data is written to the MPSCC under the control of the Tx buffer empty bit (D_2 of the SR0 register). The Tx empty bit is polled, and the data transfer operation waits until the bit is set. When the bit is set, the host system writes the transmit data into the MPSCC.

Tx Underrun Condition and CRC Character Transmission. When the Tx buffer is empty, the Tx buffer empty bit is set, causing the interrupt or DRQTx signal to become active. The MPSCC requests that the host system write the next transmit data. Should the host system not respond, in other words, should the Tx buffer remain empty too long, all transmit data held in the shift register in the MPSCC will be transmitted. As a result, no data to be transmitted will remain in the MPSCC. This state is called the Tx underrun condition, and once it occurs, the Tx underrun/EOM bit (D_6 of the SR0 register) is set.

Unless the reset Tx underrun/EOM bit command (D_7 and D_6 of the CR0 register = 1) has been issued immediately before the Tx underrun condition occurs, a SYNC character will be transmitted. If SYNC character(s) are inserted, CRC calculation will not be performed on them, even if the Tx CRC is enabled. The Tx buffer is empty in this case, and writing transmit data to the MPSCC causes it to be followed by the inserted SYNC character(s).

If transmission of CRC characters is enabled, and the transmit enable command of the CRC and the reset Tx underrun/EOM command have been issued, a 2-byte CRC character will be transmitted, followed by SYNC character(s). When the Tx underrun condition occurs, the Tx underrun/EOM bit is set by the $\overline{\text{RESET}}$ signal or by the channel reset command. The bit is reset only by the reset Tx underrun/EOM bit command. When a CRC character is inserted, there is no time requirement for issuing the reset Tx underrun/EOM command. Issue it only immediately before the Tx underrun condition occurs. Normally the command is issued when the host system detects the end of a message. This command can also be issued when the first transmit data is written to the MPSCC. In that event, monitoring the end of a message is not required.

If no transmit data is sent from the host system to the MPSCC and the Tx underrun condition occurs, a CRC character is transmitted (7201A only). Note, however, that the CRC character will be inserted if the host system delays writing transmit data to the µPD7201A and the Tx underrun condition occurs prematurely.

While the CRC character is transmitted, the Tx buffer empty bit remains in the reset state. This bit is set, however, when a CRC character has been transmitted and writing the first data byte of the next message to the Tx buffer is enabled. Should the interrupt or DMA mode be specified, an interrupt will occur, or the DRQTx signal will become active.

Using the Tx CRC Enable Bit. When a protocol that is dependent on characters (bisync) is used, character(s) may sometimes need to be excluded from the CRC calculation. The MPSCC can exclude unnecessary characters by using the Tx CRC enable bit (D_0 of the CR5 register). If transmit data is written to the MPSCC when the Tx buffer is empty and the Tx CRC enable bit is set, the CRC calculation of the written transmit data is performed. If the Tx CRC enable bit is reset, and transmit data is written to the MPSCC, CRC calculation will not be performed.

PAD Character and Transparent Mode. To confirm that the first and last characters have been correctly transmitted to the modem, the bisync protocol appends PAD character(s) to the first and last character each time a character transmission operation is performed. The character placed in front of the first character of the transmit data is called a "leading PAD character" (55H), and the character placed after the last character is called a "trailing PAD character" (FFH). The MPSCC can append these two PAD characters to transmit characters.

First, 55H (leading PAD) is loaded to the CR6 register and a SYNC character to the CR7 before the transmitter is enabled. When the transmitter is enabled, register CR6's contents are changed into a SYNC character after the transmission of 55H followed by SYNC characters (55H, SYN, SYN, SYN).

If the trailing PAD character (FFH) is written to the Tx buffer while the MPSCC is transmitting a CRC character, it, instead of a SYNC character, will be transmitted after the CRC character has been transmitted. Monitoring bit D_6 of the SR0 register (Tx underrun/EOM bit), determines whether or not the MPSCC is in the CRC transmission operation or when the E/S interrupt occurs in response to the Tx underrun/EOM bit setting.

The bisync protocol employs a transmission procedure dependent on characters. Processing binary data that cannot be regarded as characters is usually difficult. The transparent test mode should be employed with the bisync protocol for transmission of binary data. In this mode, the DLE.SYN pattern is used as the synchronization pattern instead of SYN.SYN. By loading the DLE character to the CR6 register and SYN to CR7, the MPSCC can process the synchronization pattern of the transparent text mode. The CRC calculation of the SYNC character, transmitted from the CR6 and CR7 registers, is not performed. However, if a character whose CRC calculation is not perormed is among data to be transmitted, CRC calculation will be enabled by manipulating the Tx CRC enable bit. Because the MPSCC cannot process control characters other than DLE.SYN (such as DLE.DEL) they must be processed by the host system.

Transmit Operation in Bisync Mode

Figure 5-5 shows an example of transmit operation in the bisync mode. Data is transferred between the MPSCC and the host system by interrupt. Figure 5-6 shows the interrupt wait routine.

Figure 5-5. Transmit Operation, Bisync Mode

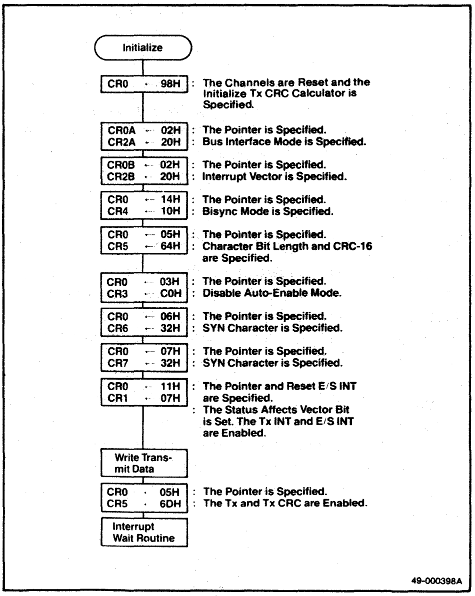
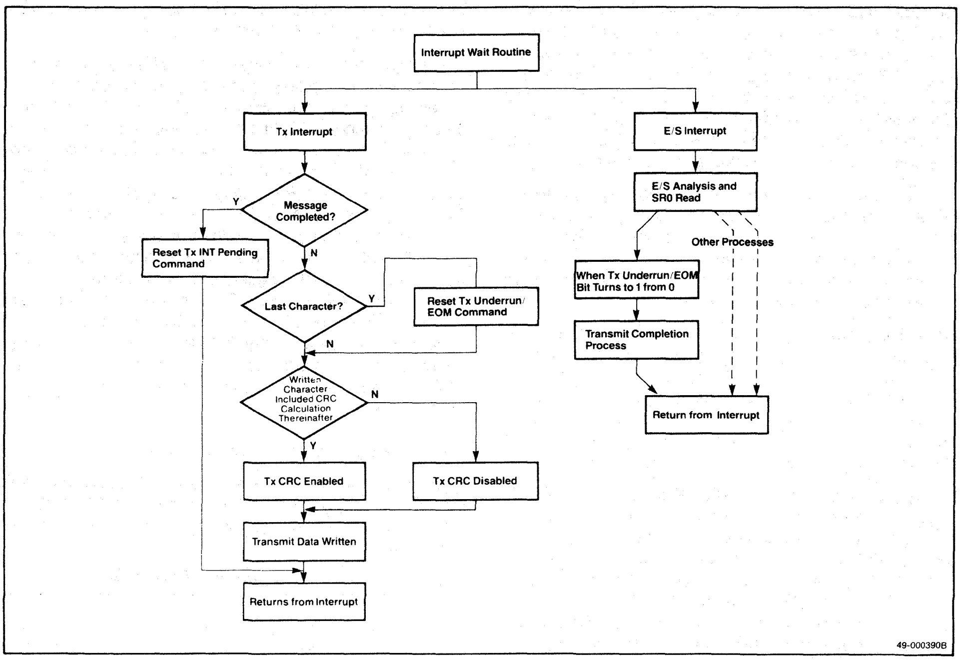


Figure 5-6. Interrupt Wait Routine



When the transmitter is enabled after transmit data has been written, two bytes of SYN.SYN are transmitted, followed by transmit data. If a number of SYN characters need to be transmitted, write transmit data after the predetermined period of time since the transmission was enabled.

Reception in Synchronous Mode

Initialization for Reception. Before performing data reception in the synchronous mode, initialize the MPSCC for reception. The CR4 register specifies the monosync, bisync, or external sync mode. The CR3 register specifies the bit length of the receive characters. The CR5 register determines the CRC polynomial. A SYNC character is loaded to the CR6 register in monosync mode and to the CR6 and CR7 registers in bisync mode. There is no need to load a SYNC character in the external sync mode. The CR1 and CR2 registers specify the mode in which receive data transfer takes place between the host system and the MPSCC.

After the Rx enable bit (D_0 of the CR2 register) is set and initialization is complete, the MPSCC starts to assemble received serial data and compares it with the SYNC character loaded in the CR6 and CR7 registers. The synchronization starts when the assembled character coincides with the SYNC characters. Once synchronization is establised, the MPSCC exits the hunt phase and starts assembling the characters it receives. If an assembled character equals the SYNC character in the CR6 register when the reception operation starts, it can be inhibited from being loaded into the Rx buffer. This is made possible by setting the SYNC CHAR load inhibit bit (D_1 of the CR3 register). Even after the MPSCC has gone out of the hunt phase, this bit remains valid. It therefore must be reset when loading a SYNC character in received data.

Transfer of Receive Data. When character synchronization has been established, an assembled receive character is transferred to the Rx buffer. The contents of the Rx buffer can be transferred to the host system by an interrupt or software polling.

A first Rx character interrupt occurs only when receive data is loaded to the Rx buffer after reception has been enabled (D_6 of CR2 register is 0). This interrupt can be used to detect when data transfer using DMA should be performed or when data transfer by software should start. When using a protocol that is dependent on characters (bisync), the process of checking receive characters must be performed in parallel with the process of first Rx character interrupt. For this reason, data transfer by means of DMA is not normally performed. The first Rx character interrupt is also caused only once by the data loaded into the Rx buffer immediately after the enable interrupt on next Rx character command is issued.

When bit D_6 of the CR2A register is 1 (7201A only) in this mode, no interrupt is caused by the character received by either of the channels. Also, the Rx interrupt cannot be enabled by the enable interrupt on next Rx character command. The other operations are performed in the same manner as when bit D_6 of the CR2A register is 0.

In all Rx character interrupt mode, an interrupt occurs each time a character is loaded into the Rx buffer.

In special Rx condition interrupt mode, if receive data contains an error when the first Rx character or all Rx character interrupt mode is specified, these modes are treated as the special Rx condition interrupt mode. Although two types of errors - parity and Rx overrun - are possible, the special Rx condition interrupt will not occur if a parity error occurs in the first Rx character interrupt mode. When the all Rx character interrupt mode is specified, a parity error may or may not be included in the special Rx condition interrupt.

Data transfer can be performed by polling the Rx character available bit (D_0 of the SR0 register) by the software. When bit D_0 of the SR0 register is set, it indicates that receive data is loaded into the Rx buffer for the host system to read from the MPSCC. In this case, to maintain the correspondence between the receive data and its status, this data must be read out to the host system after the contents of the SR1 register are read.

The Rx CRC enable bit (D_3 of the CR3 register), controls whether the MPSCC performs CRC calculation of a receive character. That is, when the character has been loaded into the Rx buffer, manipulation of the Rx CRC enable bit before the next character is loaded into the buffer will determine whether or not to perform the CRC calculation. If the Rx CRC enable bit is set after a character has been loaded into the Rx buffer (Rx interrupt or Rx character available bit is set) and before the next character is loaded, CRC calculation is performed on the first character.

If the Rx CRC enable bit is reset under that condition, the CRC calculation will not be performed. The 3-byte Rx buffer cannot be used to determine whether to perform the CRC calculation on a case-by-case basis.

The CRC/framing error bit (D_6 of the SR1 register) indicates the result of a CRC calculation after a 16-bit interval since a character for the CRC calculation was transferred to the Rx buffer from the Rx shift register. This bit indicates the result of the calculation performed before the character was transferred. If the CRC/framing error bit is 0, no error occurred during the calculation. Only the last result of the CRC check is valid, which means results from early in the calculation are meaningless. Normally, if this bit is 0 during data reception, an error is indicated.

The CRC error check is done in this manner when reception is completed. After a 16-bit interval since the second byte of a CRC character was loaded to the Rx buffer, the CRC/framing error bit is monitored. If the bit is 0, no error has occurred. If an error occurs, the bit turns to 1.

The CRC/framing error bit indicates the state of a CRC error after a 20-bit interval since the last CRC character bit was input to the RxD pin. Accordingly, the RxC clock must be input continously for a 16-bit interval after the second byte of a CRC character has been loaded to the buffer, or for a 20-bit interval after the last CRC character bit has been input to the RxD pin.

Receive Operation in Bisync Mode

Figure 5-7 shows the receive operation in the bisync mode. Interrupts transfer data between the MPSCC and host system. Figure 5-8 shows the interrupt routine.

Figure 5-7. Receive Operation, Bisync Mode

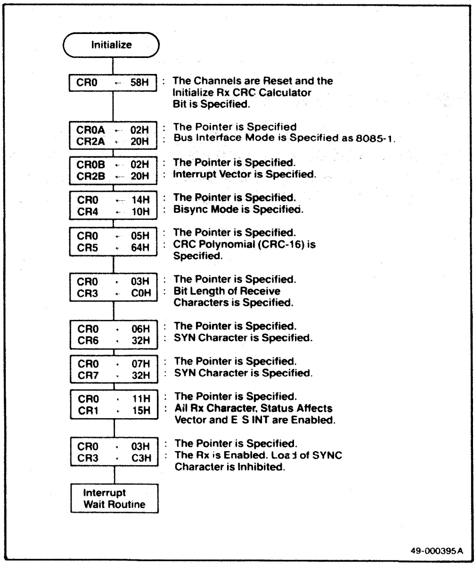
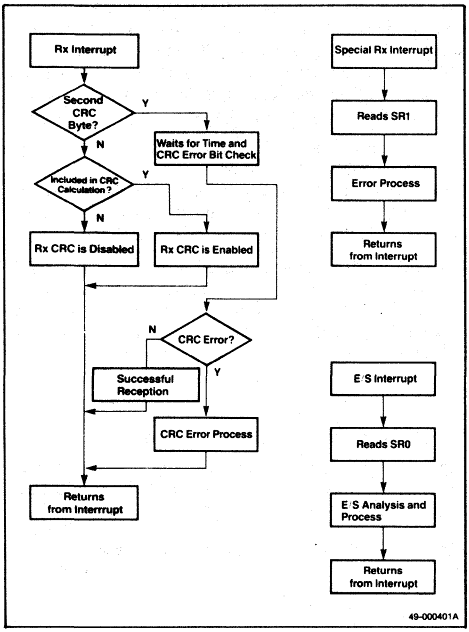


Figure 5-8. Rx Interrupt



HDLC (SDLC) MODE

The MPSCC can perform HDLC (high-level data link control) and IBM SDLC (synchronous data link control). The HDLC is a bit-oriented protocol different from byte-oriented protocol, such as BSC (binary synchronous communication). Being bit-oriented, HDLC is a protocol flexible in message length and bit pattern. The MPSCC has several features to process messages of different lengths. For details on the HDLC, refer to the CCITT standards.

Figure 5-9 shows the HDLC message format. An HDLC message, or frame, starts with a flag and ends with a flag. The MPSCC transmits and detects the flag to indicate the frame opening and closing. Flags received by MPSCC are not transferred to the CPU as data.

The 8-bit address field in an HDLC frame indicates the secondary address. If the address search mode has been enabled, detection of the secondary address determines whether or not the MPSCC receives a frame. The control field of an HDLC frame is transparent to the MPSCC; the MPSCC receives it merely as data and transfers it to the CPU.

The MPSCC has a Tx CRC calculator and an Rx CRC calculator to check frames. The Tx CRC calculator is initialized by commands (all bits are preset to 1), and the Rx CRC calculator is automatically reset when the MPSCC detects the opening flag of a frame. The MPSCC also has automatic zero insert and delete functions.

Figure 5-9. HDLC Message Format

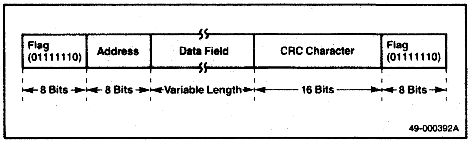


Table 5-3 shows the contents of registers CR3-CR7 when the MPSCC is in HDLC mode.

Table 5-3. Contents of Registers CR3-CR7 in HDLC Mode

CR3	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	Rx Bits/ CHAR 1	**Rx Bits/ CHAR 0**	**Auto Enable**	**Enter Hunt Phase**	**Rx CRC Enable**	**Address Search Mode**	**SYNC CHAR Load Inhibit**	**Rx Enable**
	00, 5 Bits/CHAR 01, 7 Bits/CHAR 10, 6 Bits/CHAR 11, 8 Bits/CHAR		0 Disable 1 Enable	0 NOP 1 Re-enable	0 Disable 1 Enable	0 NOP 1 Available	0	0 Disable 1 Enable
CR4	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	Clock Rate 1	**Clock Rate 0**	**SYNC Mode 1**	**SYNC Mode 0**	**Stop Bits 1**	**Stop Bits 0**	**Parity Even/Odd**	**Parity Enable**
	10		10 HDLC		00		0	0
CR5	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	DTR	**Tx Bits/ CHAR 1**	**Tx Bits/ CHAR 0**	**Send Break**	**Tx Enable**	**CRC-16/ CCITT**	**RTS**	**Tx CRC Enable**
	0 $\overline{DTR}$ = 1 1 $\overline{DTR}$ = 0	00, 5 or Less Bits/CHAR 01, 7 Bits/CHAR 10, 6 Bits/CHAR 11, 8 Bits/CHAR		0	0 Disable 1 Enable	0 CCITT-1	0 $\overline{RTS}$ = 1 1 $\overline{RTS}$ = 0	0 Disable 1 Enable
CR6	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	SYNC Bit 7 – 0							
	Secondary address *							
CR7	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	SYNC Bit 15 – 8							
	Flag character (01111110)							

Data Transmission in HDLC Mode

Before performing data transmission in the HDLC mode, initialize the MPSCC for transmission. The CR4 register specifies the HDLC mode and the CR5 register specifies the bit length of a transmit character and an HDLC polynomial (CCITT-1: $X^{16} + X^{12} + X^5 + 1$). The CR1 and CR2 registers set the mode in which data is transferred from the host system to the MPSCC. Set bit D_3 of the CR5 register to 1 to enable the transmitter.

After the registers have been properly initialized, flags are transmitted from the TxD pin. Flag transmission will continue until the first transmit data byte is written to the MPSCC. Normally after the transmitter has been enabled, writing the first data to the MPSCC is delayed and two or more flags are transmitted so that the synchronization of the reception side is securely established.

Data transfer to the MPSCC is done by means of an interrupt, DMA, or polling operation. When the transmitter is enabled, the interrupt or DRQ signal is not activated by the empty state of the Tx buffer. Therefore, do not use interrupts, DMA, or polling operations to transfer the first transmit data to the MPSCC after the transmitter has been enabled. Note, however, that an interrupt or DMA request is automatically generated when a flag is loaded to the shift register after the transmitter has been enabled, provided that bit D_6 of the CR1 register is 1 (7201A only).

If auto enable mode is specified, the transmit operation is enabled when the $\overline{CTS}$ pin becomes active after the transmitter has been enabled. Of the frame configurations regulated by the HDLC protocol, address, control, and data fields are transparent to the MPSCC. The MPSCC merely transmits these written data without any change. After the transmitter has been enabled, the first data written to the MPSCC is normally treated as an address field. An 8-bit character transferred then from the MPSCC is transmitted sequentially with the MPSCC starting from its least significant bit (LSB) in the number of bits per character specified by D_6 and D_5 of the CR5 register. Therefore, if a character is made up of seven bits or less, the data byte to be transferred to the MPSCC must accordingly be shifted to the right by the number of unused bit(s) in compliance with the data format described in Section 4, Control Register 5 (CR5).

Change or specify the transmit character length as desired. The bit length can be changed by modifying the specification of the CR5 register immediately after a transmit character has been written to the MPSCC. This is true because characters are transmitted from the Tx buffer to the transmitter by the specified bit length.

In HDLC mode, CCITT-1 ($X^{16} + X^{12} + X^5 + 1$) is selected as a CRC polynomial. In this case, the CPU will either issue the initialize Tx CRC calculator command or (in the 7201A only) it will be cleared (all bits will become 1) when the flag is loaded to the Tx shift register. This allows faster transmission of sequential frames in the 7201A.

The MPSCC transmits CRC character by using the Tx underrun state. When the Tx underrun state occurs when the Tx underrun/EOM bit is 0, CRC characters are transmitted. In the 7201, the Tx underrun/EOM bit must be reset by command. In the 7201A, it is reset by the first bit of the transmission. If the TxLR set bit (D6 of the CR1 register) is 1, CRC or abort sequences will be transmitted according to the value of the Tx length counter. This applies to the 7201A only. The 7201 has no Tx length counter.

The MPSCC has a zero-insertion function. If five consecutive 1s are detected in a frame when data is transmitted using HDLC protocol, a 0 is inserted after the 1s. However, no 0 is inserted after flags or abort sequences.

Data Transfer Modes. When the MPSCC transfers data to the host, it uses three modes: polling, interrupt, and DMA.

In polling mode, data transfer is performed without using interrupts or DMA. The CPU writes the transmit data to the MPSCC by checking the state of the Tx buffer empty bit (D_2 of the SR0 register). If the bit is 1, the CPU can write transmit data to the MPSCC. If the bit is 0 (Tx buffer is full), the MPSCC is not ready to accept any transmit data and the CPU cannot write the transmit data.

If the Tx buffer is full when $\overline{\text{WAIT}}$ is enabled (D_7 of the CR1 register is 1), an attempt to write data to the MPSCC will cause the $\overline{\text{WAIT}}$ signal to become active. The write cycle of the CPU is prolonged by this $\overline{\text{WAIT}}$ signal until the Tx buffer becomes empty. The WAIT function is used to perform data transmission controlled by the software, such as block transmission.

In interrupt mode, data transfer is performed using an interrupt. Bits D_1 and D_0 of the CR2A register set both channels A and B in the interrupt mode when they are 00. Only channel B is set in the interrupt mode when these bits are 01. If the Tx INT/DMA enable bit is set, the Tx interrupt occurs when the Tx buffer becomes empty. The CPU writes data to the MPSCC using this Tx interrupt. The Tx interrupt is satisfied when data is written to the Tx buffer or when the reset Tx INT/DMA pending command is issued.

When the reset Tx INT/DMA command is issued without the CPU writing data to the MPSCC, the Tx interrupt is temporarily reset, but the Tx buffer remains empty. The MPSCC then enters the Tx underrun state when data in the Tx shift register has been completely transmitted. The Tx interrupt will occur again on completion of CRC character transmission and when the MPSCC requests the CPU to transfer the next message. This interrupt occurs when a flag is internally loaded. When the transmitter is enabled first, the Tx buffer is already empty. Therefore, the Tx interrupt does not occur until the first character has been written to the MPSCC when bit D_6 of the CR1 register is 0. When this bit is set to 1, the interrupt occurs each time the Tx buffer becomes empty.

In DMA mode, data transfer is performed using DMA. Bits D_1 and D_0 of the CR2A register must be set either to 01 (channel A to the DMA mode), to 10 (channel A and B to the DMA mode), or to 11 (channels A and B to the DMA-2 mode, 7201A only).

The DRQTx signal becomes active when the Tx buffer empty bit is 1 and the DMA controller can write transmit data from the CPU to MPSCC. The DRQTx signal does not become active just by enabling the transmission. The DRQTx signal becomes active when the Tx buffer empty bit is set to 1 following the first data written to the MPSCC when D_6 of the CR1 register is 0. When bit D_6 of the CR1 register is 1 (7201A only), the signal becomes active when a flag is loaded to the Tx shift register after the transmitter has been enabled. Even if the last data of a frame is transferred to the MPSCC, the DRQTx signal becomes active when the Tx buffer becomes empty. Should the DMA controller not respond under this condition, the Tx underrun condition occurs after data currently existing in the shift register has been transmitted.

If necessary, the DRQTx signal can be reset temporarily by using the reset Tx INT/DMA pending command. This command is valid, however, only when the Tx buffer empty bit is set to 1. The DRQTx signal becomes active again when the MPSCC completes transmission of a CRC character and requests the CPU to transfer the next message.

Tx Underrun/EOM in HDLC Mode. When all data has been transmitted from the Tx buffer and Tx shift register (Tx underrun condition occurs), the MPSCC closes the frame currently in process and transmits a 2-byte CRC character, followed by a flag.

The Tx underrun condition is indicated by the Tx underrun/EOM bit. This bit is set by an external reset signal input from the $\overline{\text{RESET}}$ pin, by the channel reset command, or when the Tx underrun State occurs. The Tx underrun/EOM bit is automatically reset when the first character is loaded to the Tx shift register. A CRC character is, therefore, transmitted when the Tx underrun state occurs because the Tx underrun/EOM bit is already 0.

After the MPSCC has been reset, the Tx underrun/EOM bit is 1. This prevents transmission of CRC characters when the MPSCC has no data to transmit. Consequently, a flag is transmitted when the MPSCC is set in the Tx enable state. When data is written to the Tx buffer, the MPSCC starts to transmit the data.

When the Tx underrun condition occurs after the last data of a message has been transmitted, the Tx underrun/EOM bit becomes 1, causing the E/S interrupt to occur and transmission of a CRC character to start. During the CRC character transmission, the Tx underrun/EOM bit remains 1 and the Tx buffer empty bit 0. These states indicate the Tx buffer is filled with CRC. In the 7201, the reset Tx underrun/EOM command must be given before the underrun occurs to assure that CRC bytes are sent. If this bit is not reset, SYNC characters will be transmitted.

When the CRC characters have been completely transmitted, the Tx buffer empty bit becomes 1 again, and transmission of the next message is enabled. If no message is to be transmitted, the transmitter will be disabled, and the transmit operation will terminate.

Abort Sequence Generation. An abort sequence is a series of 8 to 13 “1” bits. A maximum of 13 bits is possible because it is possible for a maximum of 5 consecutive bits to precede an 8-bit sequence.

When an abort sequence is transmitted, characters currently being transmitted and the contents of the Tx buffer are lost. Then, flags follow completion of the abort sequence transmission.

When the MPSCC is reset, the Tx Underrun/EOM bit is 1 and flags are transmitted until a character is written for the first time since the transmitter was enabled. When the next character (secondary address) is written, the Tx underrun/EOM bit resets to 0 (7201A only) and starts the transmission of frames.

If data transfer from the CPU is delayed despite all data transfers having not yet been completed (Tx underrun condition occurs), the frame must be aborted with an abort sequence. This abort sequence can be generated in the MPSCC automatically (7201A only) or by software.

In automatic generation, if the TxLR set bit (D_6 of the CR1 register) is set to 1 and the number of transmit bytes is set to the Tx length register as the initial condition, the MPSCC compares the contents of the Tx length register with those of the Tx length counter (SR3 and SR4 registers) to determine whether to transmit an abort sequence. If the number of data characters to be transmitted (set value of the Tx length register) does not agree with the number of data characters which have been transmitted (set value of the Tx length counter) in the Tx underrun condition, the MPSCC will automatically transmit an abort sequence. Since the 7201 does not have Tx length registers and counters, this is only possible for the 7201A.

The abort sequence can also be generated by issuing a send abort command (CR0). The host system, rather than the MPSCC, decides to generate an abort sequence. The Tx underrun/EOM function of the MPSCC makes this decision possible. The 7201 abort sequence must be generated by the host.

If data transfer to the MPSCC is delayed, the Tx underrun condition will occur. If this condition occurs while the CRC characters are being transmitted, then the host system must decide if the occurence of the Tx underrun state is caused normally. If the condition is abnormal, the host system secures the time required to issue the send abort command.

If a flag, transmitted when the Tx underrun error occurs, erroneously results in a transmit data pattern that matches the results of the Rx CRC calculator + flag, the frame is probably normal. When that happens, the CRC character transmission function prevents the abnormal frame from being seen as normal. The E/S interrupt generally occurs when the Tx underrun/EOM bit is set to 1 from 0. The host system makes the decision based on this interrupt.

The MPSCC performs the following operations in response to a send abort command and the CPU responds to them.

(1) The Tx buffer of the MPSCC becomes empty, and the Tx interrupt and DRQTx signals become active.

(2) If either the CPU or DMA controller does not respond to operation (1) within a fixed interval, the MPSCC enters the underrun state.

(3) The MPSCC then sets the Tx underrun/EOM bit, and the E/S interrupt occurs.

(4) The MPSCC starts transmitting CRC characters.

(5) The CPU accepts the E/S interrupt described in operation (3), checks the Tx underrun/EOM bit, and decides if the interrupt was caused by completion of a message, as it should have been.

(6) If the CPU decides that the interrupt has not been caused by a normal completion of a message, it immediately issues the send abort command (D_5, D_4, and D_3 of the CR0 register are 001).

(7) In response to the send abort command, the MPSCC transmits an abort sequence.

The CPU response time for the above sequence is from 22 to 30 Tx clock cycles.

Abort sequences become valid as soon as the send abort command is written to the CR0 register causing loss of the transmit data. The send abort command is automatically reset after transmission of abort.

CRC Calculation in HDLC Mode. In HDLC protocol, the bits between the opening flag of each frame and a CRC character are CRC-calculated using the polynomial $X^{16} + X^{12} + X^5 + 1$. The CRC calculation is performed in the

HDLC mode as follows:

(1) The HDLC polynomial is selected by bit D$_2$ of the CR5 register. This specification must be performed prior to resetting the Tx CRC calculator.

(2) The Tx CRC enable bit is set by bit D$_0$ of the CR5 register to enable the CRC calculation. This must be performed before transmitting the first data to the MPSCC (address field).

(3) The Tx CRC calculator is automatically reset (all bits become 1) at the beginning of each frame when a flag is loaded to the Tx shift register inside the MPSCC (7201A only).

(4) The CRC calculation starts when the first data (address field) is transferred to the MPSCC.

(5) Upon completion of data transmission to the MPSCC and when the Tx underrun condition occurs, a CRC character is transmitted onto the data line with each bit of the value generated by the Tx CRC calculator inverted.

Completion of Transmit Operation (7201A only). The MPSCC can notify the host system that transmission of a frame has been completed (all sent bit is 0) by using the E/S interrupt in HDLC mode. This E/S interrupt occurs when a flag pattern of one byte or more is transmitted from the TxD pin after a CRC character has been transmitted. Inside the MPSCC, however, the interrupt occurs when flag characters of three bytes or more are about to be transmitted.

Because the E/S interrupt caused by the Tx underrun/EOM bit does not indicate completion of a frame transmission, this interrupt is used to detect it.

The E/S interrupt occurs when bit D$_0$ of the SR1 register is set to 1. When it occurs, the contents of the SR0 and SR1 registers must be read. Should the transmitter be disabled, the characters held in the Tx buffer will remain there unchanged and will not be transmitted.

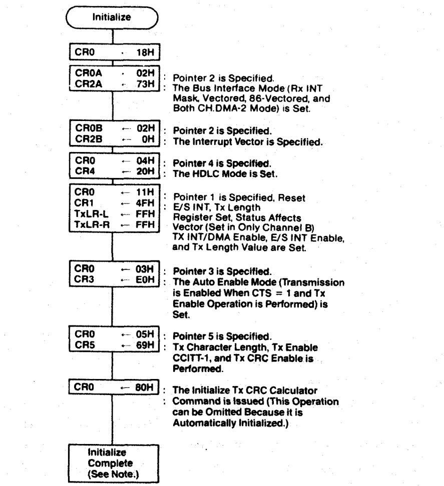
Transmit Operation in Tx Length Set Mode (D$_6$ of the CR1 Register is 1) (7201A only). Data transmission using the Tx length register is performed differently from transmission without using it. This mode, in which the desired number of bytes is set to the Tx length register, is especially effective when using both channels A and B in the DMA mode (especially in the both CH.DMA-1 mode when bits D$_1$ and D$_0$ of the CR2A register are set to 1 and 0). In this mode, the Tx DMA request DRQTx signals automatically become inactive when the number of bytes sent to the Tx length register has been completely transmitted. When bit D$_6$ of the CR1 register is set to 1 again, the DRQTx signal becomes active again. If the Tx underrun condition occurs before the number of data transmitted has reached the number set in the Tx length register, the MPSCC will automatically abort the frame. The contents of the SR3 and SR4 registers can confirm whether the frame has been aborted or the transmission has been normally completed. If the frame has been aborted, the contents of the registers indicates the number of data character transmitted up to the point where the abort has occurred + 1. If the contents are 00s, the transmission has completed normally.

To resume transmission after the frame has been automatically aborted, activate the DRQTx signal by setting bit D$_6$ of the CR1 register to 1 as if the transmission had been completed normally. the DRQTx signal becomes active after the contents of the Tx length register have been modified.

Transmit Example in HDLC Mode

Figure 5-10 shows an example of a transmit operation in HDLC mode. The example assumes data is transmitted using DMA. Figure 5-11 shows the E/S interrupt process.

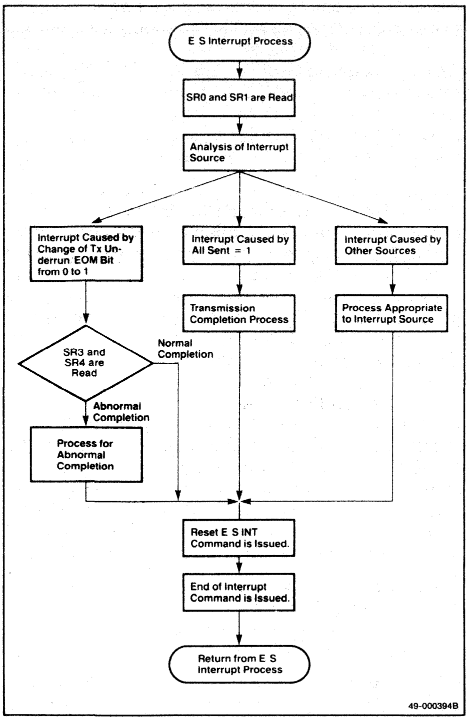
Figure 5-10. Transmit Operation, HDLC Mode



Note: In this example, the DRQTx signal becomes active and requests the first data transmission when $\overline{\text{CTS}}$ becomes low after initialization. Afterward, the DRQTx signal becomes active each time the Tx buffer becomes empty. When the number of data set in the TxLR has been written, subsequent MPSCC DRQTx signals do not become active.

49-000396A

Figure 5-11. E/S Interrupt Process



Reception In HDLC Mode

Receive Operation in HDLC Mode. To receive data in the HDLC mode, perform the following steps:

(1) Specify the HDLC mode by setting the necessary parameters in the CR4 register. Do this before setting parameters in other registers.

(2) Use the CR5 register to specify the HDLC polynomial (CCITT-1: $X^{16} + X^{12} + X^5 + 1$).

(3) Specify the secondary address and flag character with the CR6 and CR7 registers, respectively.

The MPSCC enters the hunt phase when the enter hunt phase bit (D_4 of the CR3 register)is set. This state continues until the first flag (opening flag) is detected after the receiver is enabled. In the HDLC mode, once the MPSCC goes out of the hunt phase and synchronization is established, the synchronization will be maintained until the receiver is disabled. When the MPSCC detects the valid opening flag, it sets the SYNC/hunt bit to 0 and the E/S interrupt occurs. Subsequently, the MPSCC assembles and transfers the serial data input from the RxD pin to the Rx buffer. Data that exists between the opening and closing flags is transparent to the MPSCC. The MPSCC performs no special process on the received data.

The MPSCC has an address search mode function that determines whether the MPSCC should receive the transmitted data. In this mode (D_2 of the CR3 register is 1), the MPSCC will only start assembling data when the first data byte (secondary address) following a valid flag matches either the address set in the CR6 register or the global address 11111111. The address search function is used only to assemble 1-byte address data. The host system therefore must decide, when a frame configured of 2 bytes or more of extended address is used, whether the frame should be received. In the event that it is received, the MPSCC transfers the extended address following the first address byte to the CPU if the first address byte coincides with the contents of the CR6 register or with the global address. The CPU, in turn, must check the extended address. Should the CPU decide the frame is unnecessary, it will set the enter hunt phase bit again, temporarily stop reception, and wait for the next frame.

Because the length of the I field is not regulated by HDLC protocol, the MPSCC can modify the transmit character length as desired. The modification must take place while the character whose character length has been changed is being assembled. In other words, the modification must be made at an appropriate time so that it becomes valid before the number of bits specified as the character length is completely assembled. For example, if the address and control field are eight bits long and the I field folllowing them is seven bits long, the modification must be carried out after the control field has been assembled and while the 7-bit character is being assembled.

The abort detection function of the 7201A is valid when the opening flag is detected after reception is enabled. The abort detect function for the 7201 is valid during the hunt phase as well. When eight or more 1s are received (abort sequence), the break/abort bit (D_7 of the SR0 register) is set and the E/S interrupt occurs if the interrupt has been enabled. After detecting the abort sequence, the MPSCC automatically enters the hunt phase. If the E/S interrupt occurs because of detection of the abort sequence, the MPSCC will issue the reset E/S INT command. To distinguish whether the detected bit string is an abort sequence or an idle sequence, time is measured from when the reset E/S INT command was issued. If the abort state is reset (zero following the abort sequence is detected), the E/S interrupt occurs. The special Rx condition interrupt occurs, which is caused by the end of frame, when the MPSCC detects the closing flag. Normally, by using this interrupt, the MPSCC detects the completion of frame transmission. If the data length between the opening and closing flags is seven bits or less, the frame will not be received. If the data length is more than seven bits, the frame will be received. The MPSCC automatically deletes the zero inserted in the data, using its zero deletion function. Flags are not transmitted to the CPU. If the E/S interrupt is enabled, the E/S interrupt will also occur when the state of the $\overline{DCD}$ pin changes.

Transfer Modes of Receive Data. Receive data can be transferred from the MPSCC to the CPU in HDLC mode by polling, interrupt, or DMA.

In polling mode, the CPU reads receive data from the MPSCC as it checks the state of the Rx character available bit (D_0 of the SR0 register). The MPSCC turns the Rx character available bit to 1 each time it receives a character and requests the CPU to fetch the receive data out of the MPSCC. If the Rx character available bit is 1, the CPU will read receive data from the MPSCC.

In data transfer by interrupts, data transfer using an interrupt can be in first Rx character interrupt mode or all Rx character interrupt mode.

In the first Rx character interrupt mode (D_4 and D_3 of the CR1 register are 01), an interrupt is caused by the first received character only. Subsequently, an interrupt will occur only each time the special Rx condition is detected. However, when the enable interrupt on next Rx character command (D_5, D_4, and D_3 of the CR0 register are 1, 0, and 0, respectively) is issued, an interrupt can reoccur when a character is received after the command. In the Rx INT mask mode (D_6 of the CR2A register is 1, 7201 A only), an interrupt is not caused by the first received character. The interrupt which is caused by a received character cannot be enabled by issuing the enable interrupt on next Rx character command. This mode can be used to start a polling operation by the software or to block transmission by using the $\overline{WAIT}$ signal or to start a DMA operation. In the all Rx character interrupt mode (D_4 and D_3 of the CR1 register are either 10 or 11), an interrupt occurs whenever a character exists in the Rx buffer. Also in this mode, an interrupt provided with an interrupt vector indicating the interrupt, occurs when the special Rx condition is detected (status affects vector is selected). The character immediately before a closing flag (normally the second byte of a CRC character) is read from the MPSCC by the process routine for the special Rx condition interrupt, because of an end of frame.

In data transfer in DMA mode, the first Rx character interrupt mode is selected. The DMA controller usually is enabled by this interrupt process routine, and the DMA operation starts. When an opening flag is detected, the DRQRx signal becomes active each time the MPSCC receives a character and the MPSCC requests the DMA controller to read the receive data. On the other hand, in the first Rx character mode, an Rx interrupt occurs upon reception of the first character. This interrupt can be masked by setting D_6 of the CR2A register to 1 (Rx INT mask mode, 7201A only). If the DMA controller is enabled before the first Rx character interrupt occurs, it processes the interrupt before the CPU accepts it. When the receive data is read from the MPSCC to the CPU, the interrupt request signal is reset. When a closing flag is received, the special RX condition interrupt occurs because of an end of frame, and the MPSCC informs the CPU of the completion of frame reception.

Special Rx Condition in HDLC Mode. In the HDLC mode, the special Rx condition interrupt occurs when the Rx Overrun state or end of frame state is detected. When the MPSCC enters either state, an interrupt with a vector indicating the special Rx condition can be generated (status affects vector is selected). The special Rx condition interrupt, however, is not an independent interrupt mode, and either the first Rx character or the all Rx character interrupt mode must have been selected before the conditions necessary to generate this interrupt are established. The Rx overrun status is latched, and the error information read from the MPSCC indicates an error occurred either in the data stored in the Rx buffer or after the error reset command was issued. The special Rx condition interrupt due to an Rx overrun error is generated when the MPSCC reads the data that caused the Rx overrun state.

The special Rx condition interrupt due to the end of frame state is generated when a valid closing flag is received. If the end of frame bit is set, the CRC error bit and residue codes will become valid.

CRC Calculation of Receive Data. The MPSCC performs control of the Rx CRC calculator during a receive operation. The Rx CRC calculator is reset by an opening flag (all bits checked are set to 1) and completes the CRC calculation in response to a closing flag. It can also be reset by the reset Rx CRC checker command when bits D_7 and D_6 of the CR0 register are 0 and 1, respectively.

In the HDLC mode, CRC calculation of all receive data is performed, but no 8-bit delay occurs as in the synchronous mode. Because results of the CRC calculation are inverted when they are transmitted, a special sequence is used for the CRC check of the MPSCC during the receive operation. A 2-byte CRC character is used for the CRC check inside the MPSCC and can be received as normal data. The CRC calculation is complete when the second CRC character has been transferred to the RX buffer. In the 7201A, both bytes of the CRC character can be received as data. In the 7201, only the first CRC byte is received as data correctly.

Completion of Receive Operation. When the 7201A receives a closing flag, the end of frame bit (D_7 of the SR1 register) is set, and the special Rx condition interrupt occurs. Under this condition, the CRC framing error bit (D_7

of the SR1 register) and residue codes (D_3, D_2 and D_1 of the SR1 register) are valid. The residue codes indicate the number of bits on the boundary between a 16-bit CRC character bit and an I field.

The CPU must immediately issue the error reset command (D_5, D_4, and D_3 of the CR0 register are 110) to clear this state after performing necessary processes in the special Rx condition interrupt process routine caused by the end of frame bit.

The MPSCC continues receiving input frames even when it has entered the end of frame state after detecting a closing flag. These receive data characters, however, are not transferred to the last byte of the Rx buffer until the error reset command is issued. This is in case of an INT on first Rx character mode. If no subsequent message is to be received by the MPSCC, the receiver will be disabled to terminate the receive operation.

Figure 5-12 shows a receive operation in HDLC mode. This example assumes that data is transmitted using the DMA mode. Figure 5-13 shows the special Rx condition interrupt process. Figure 5-14 shows the reset E/S interrupt.

Figure 5-12. Receive Operation, HDLC Mode

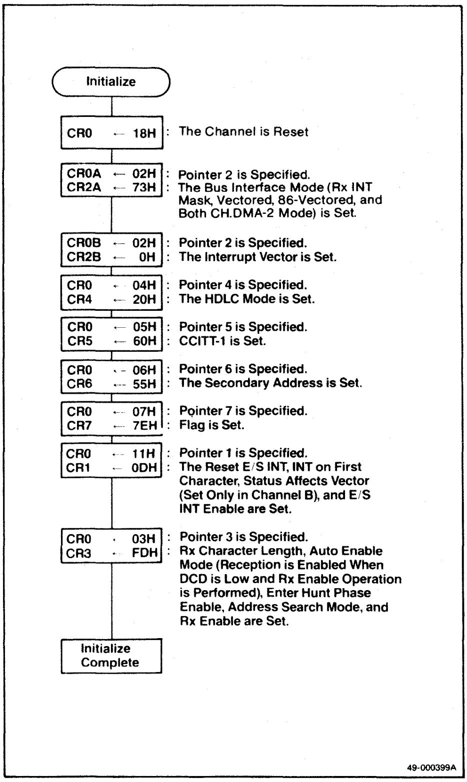
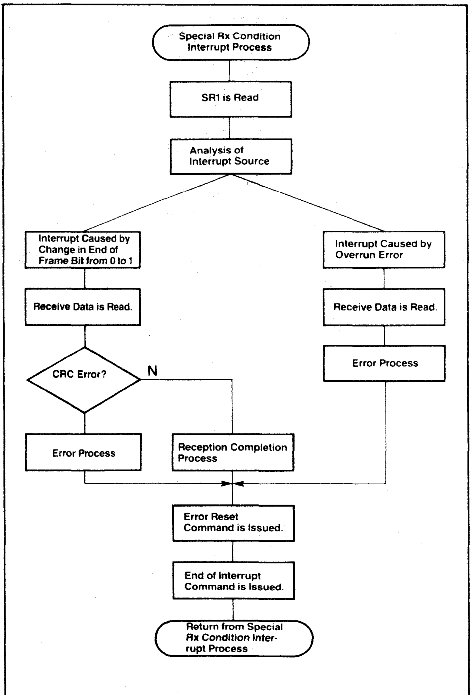


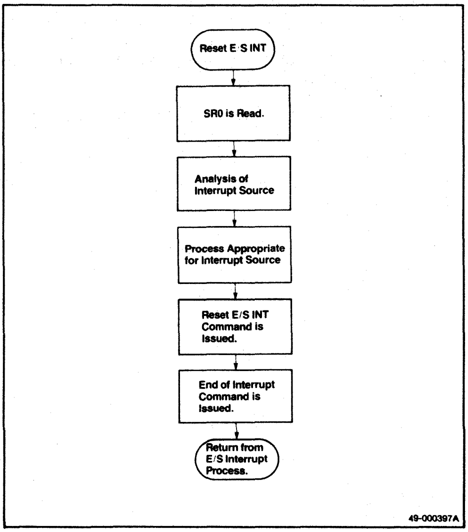
Figure 5-13. Special Rx Condition Interrupt Process



Note: In this example, a receive operation starts when the level of $\overline{DCD}$ becomes low after initialization. When a valid flag is detected, the first data (other than flags following the valid flag) is compared to the contents of the CR6 register. If this data is the same. (address agreement). DRQRx becomes active each time data including this address data is received and the transfer of receive data is requested.

49-000393B

Figure 5-14. Reset E/S Interrupt Process



CPU INTERFACE TIMING

CPU interface timing consists of read/write, WAIT, interrupt, and DMA timing.

Read/Write Timing

Figure 6-1 is a diagram of the interface between the CPU and the MPSCC. It also shows timing for reading out data or status information in the MPSCC (read cycle) and for writing commands, parameters, or data to the MPSCC (write cycle). The CPU is an 8085A.

WAIT Timing

Figure 6-2 shows the basic timing chart of a WAIT operation. Both the read cycle in which the CPU fetches data from the MPSCC and the write cycle in which it writes data to the MPSCC can be prolonged by inputting $\overline{\text{WAITA}}$ and $\overline{\text{WAITB}}$ signals to the CPU'S READY pin. These two signals are active low.

The $\overline{\text{WAITA}}$ pin also functions as the DRQRxA pin and the $\overline{\text{WAITB}}$ as the DRQTxA pin. When D_1 and D_0 of CR2A register are set to zero, these two pins function as the WAIT pins.

The WAIT pins are open drain pins. Their impedance is high until they become active. Both WAITA and WAITB pins can be wired OR.

Both pins become active low level under the following special conditions:

- D_1 and D_0 of the CR2A register are both 0s in the $\overline{\text{WAIT}}$ mode.
- D_7 of the CR1 register is 1 in the wait enable mode.
- The Tx buffer is full or the Rx buffer is empty and the level of the $\overline{\text{CS}}$ and the C/D pin is 0.

The $\overline{\text{WAIT}}$ pins go to a high level when the Tx buffer becomes empty or the MPSCC is in the Rx character available state, thus enabling the read/write operation.

Interrupt Timing

Vectored Mode. Figure 6-3 shows the basic timing chart of an interrupt in the vectored mode. Figure 6-4 shows a timing chart when more than one interrupt source exists in this mode.

The MPSCC accepts its own internally generated interrupt request when the internal $\overline{\text{INTAK}}$ signal (set at the trailing edge of the first externally input $\overline{\text{INTAK}}$ signal and reset at the leading edge of the second externally input $\overline{\text{INTAK}}$ signal) is inactive. An internal interrupt is disabled when the internal $\overline{\text{INTAK}}$ signal is active.

When an interrupt assigned a priority is enabled, interrupts with lower priority will remain disabled. When an internal interrupt is enabled, the $\overline{\text{INT}}$ pin of the MPSCC goes low and notifies the interrupt controller (for example, the 8259A). At the same time, the $\overline{\text{PRO}}$ pin turns high, and interrupts by an MPSCC connected in the lower order of the daisy chain are disabled.

When the $\overline{\text{PRI}}$ input is high, an output signal from the $\overline{\text{INT}}$ pin will be kept at a high level until the input to the PRI pin becomes low. Because the $\overline{\text{INT}}$ output pin is an open drain pin, the output must be pulled up to turn it to a high level. When the MPSCC enters the interrupt acknowledge cycle, the $\overline{\text{INTAK}}$ signal is input to it from the CPU, and an interrupt request is latched at an in service latch. The MPSCC then analyzes the priority of the interrupt in the time interval between the first and second falling edges of the $\overline{\text{INTAK}}$ signal. As a result, the MPSCC sets the in service latch corresponding to the highest-priority interrupt at the second falling edge of the $\overline{\text{INTAK}}$ signal. As the in service latch is set, the $\overline{\text{INT}}$ signal is reset at a high level.

When an in service latch is set, an interrupt assigned a lower priority will not be enabled, and the in service latches of lower levels also will be reset.

In service latches are reset when the EOI command is issued (D_3, D_4, and D_5 of the CR0 register are all 1s). At this point, the in service latch assigned top priority will be reset. An input signal to the $\overline{\text{CS}}$ pin must be inactive ($\overline{\text{CS}} = 1$) while the INTAK pin remains at a low level (0).

Even when an in service latch with priority is set, if an interrupt request signal with a higher priority is generated, the interrupt can be enabled during the interrupt enabled period.

Figure 6-4 shows interrupt timings of three interrupts, labeled A, B, and C. Interrupt A has highest priority, interrupt B the second highest, and interrupt C the lowest. After the $\overline{\text{INT}}$ signal is output because of interrupt C, interrupt B will be enabled if it occurs before the $\overline{\text{INTAK}}$ signal is received. When the MPSCC enters the interrupt acknowledge cycle, the in service latch corresponding to interrupt B is set, and the interrupt vector corresponding to interrupt B will be sent to the CPU if the status affects vector mode is specified.

If interrupt A occurs, however, while interrupt B is being serviced (in service latch of interrupt B is 1), it will be enabled instead of B because of its higher priority, and the $\overline{\text{INT}}$ signal will be generated. When the CPU issues the EOI command after servicing interrupt A, the in service latches of interrupt A and interrupt B are both reset. After all three interrupts have been serviced and all interrupt sources in the MPSCC and in service latches have been cleared, the output signal from the $\overline{\text{PRO}}$ pin is reset, and the interrupt of the MPSCC connected in the lower order of the daisy chain will be enabled.

Figure 6-1. ***Read/Write Timing***

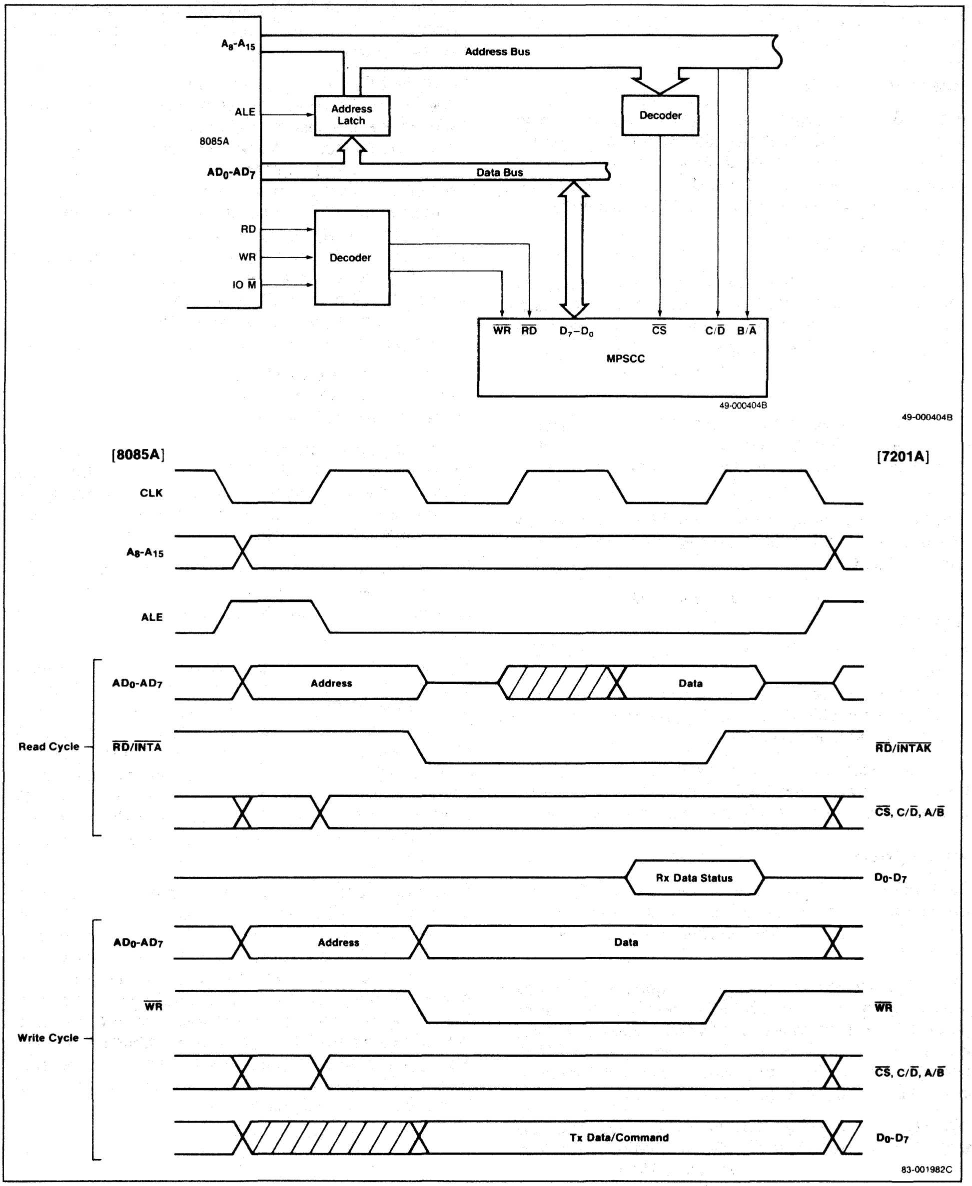


Figure 6-2. WAIT Timing

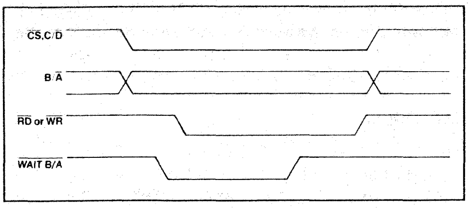


Figure 6-3. Interrupt Timing in Vectored Mode (1)

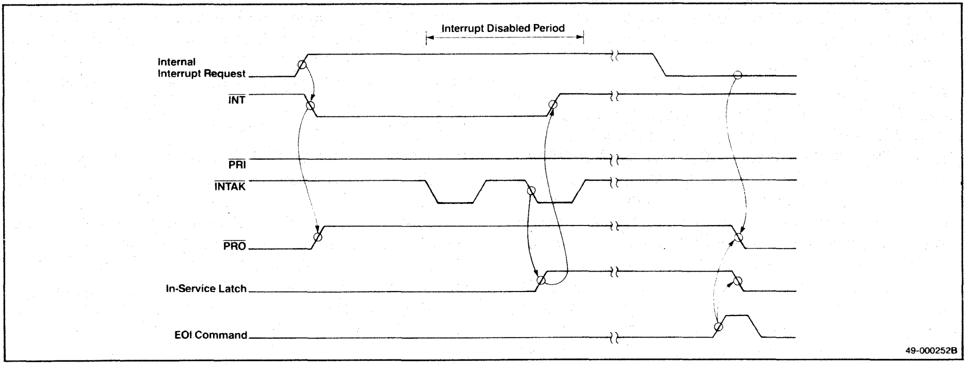
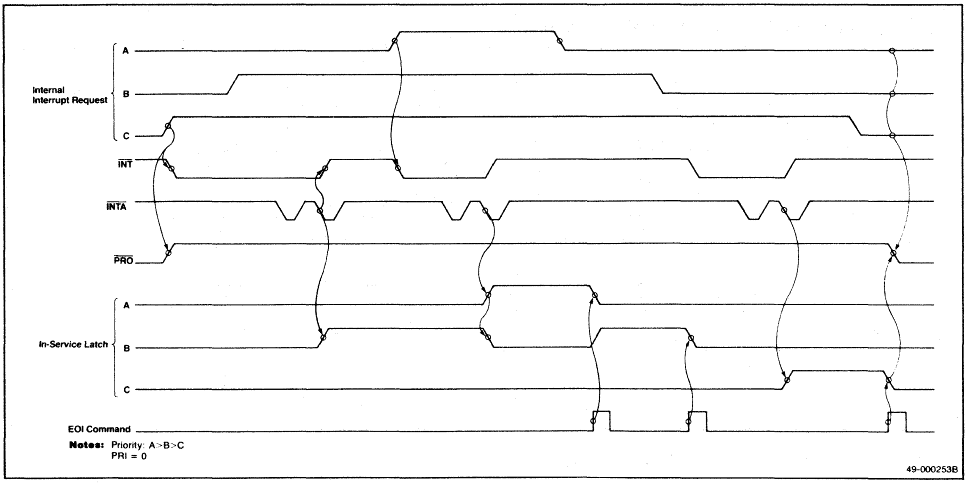


Figure 6-4. Interrupt Timing in Vectored Mode (2)



Notes: Priority: A>B>C
PRI = 0

The interrupt operation is applicable in the 85-1 and 85-2 interrupt modes (D_3 and D_4 of the CR2A register are either 00 or 01). However, in the 85-3 interrupt mode (D_3 and D_4 of the CR2A register are 11, 7201A only) the $\overline{\text{INT}}$ output will become active (low level) if there is an interrupt in the MPSCC even when the $\overline{\text{PRI}}$ input is high.

Nonvectored Mode. Figure 6-5 shows an interrupt timing chart in the nonvectored mode. If an interrupt assigned a priority is enabled, no interrupt with a lower priority can be enabled.

When an interrupt is enabled, the $\overline{\text{INT}}$ signal of the MPSCC becomes active, and the MPSCC notifies the CPU. The CPU specifies the SR2B register (pointer two) for the MPSCC to analyze the interrupt source and fetch the interrupt vector.

The MPSCC then sets an in service latch at the falling edge of the $\overline{\text{RD}}$ signal while the SR2B register is being specified and latches the information that an interrupt has been enabled. Once the SR2B register is specified, the MPSCC disables all additional interrupts. It analyzes the priority of the enabled interrupt until the $\overline{\text{RD}}$ signal becomes low, and the in service latch with the highest priority then becomes active. In this manner, specifications of the SR2B register and the $\overline{\text{RD}}$ signal are used in the nonvectored mode to confirm that an interrupt has been enabled by the CPU. This is different from interrupt operation in the vectored mode. Other operations performed in the nonvectored mode, however, are the same as those in the vectored mode.

Also, no subsequent interrupt will be enabled unless the CPU performs a read operation (RD) in the nonvectored mode after the SR2B register has been specified. In the nonvectored mode, the $\overline{\text{INTAK}}$ pin must be kept in the inactive state ($\overline{\text{INTAK}} = 1$).

DMA Timing

DMA Timing in Both CH.DMA-2 Mode. (7201A only) The both CH.DMA-2 mode is set by setting D_1 and D_0 of the CR2A register to 1s. In this mode, DMA priority in the MPSCC is not fixed by a command and DMA process can be performed without using the $\overline{\text{HAI}}$ or $\overline{\text{HAO}}$ pins.

An internally generated DMA request is accepted during the low-level period of the CLK signal. However, in this low-level period, data-read and data-write periods are not included. This acceptance of DMA requests is independent of the priority, and an accepted DMA request causes the corresponding pin to become active.

The DMA request is then sent to the DMA controller (μPD8237A, for example). The $\overline{\text{CS}}$, C/$\overline{\text{D}}$, and B/$\overline{\text{A}}$ pins of the MPSCC must be controlled using the DACK signal of the DMA controller during each acknowledge period. The DACK signal determines whether an accepted DMA request has been satisfied. The DACK, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ signals were input during the DMA acknowledge period. Therefore, the DMA controller determines and controls the priority of a DMA operation.

Figures 6-6 and 6-7 show the DMA timing and an example for interfacing the μPD8237A with the MPSCC in this mode.

Figure 6-5. Interrupt Timing in Nonvectored Mode

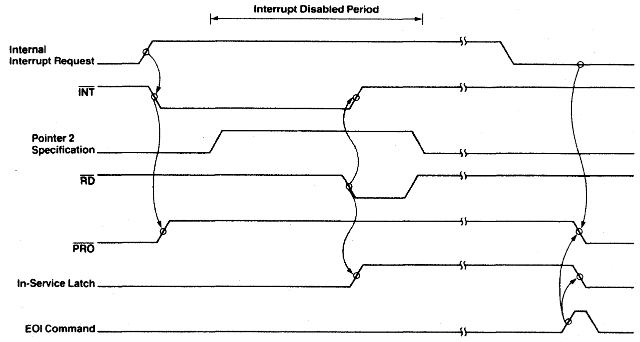


Figure 6-6. DMA Timing, Both CH. DMA-2 Mode

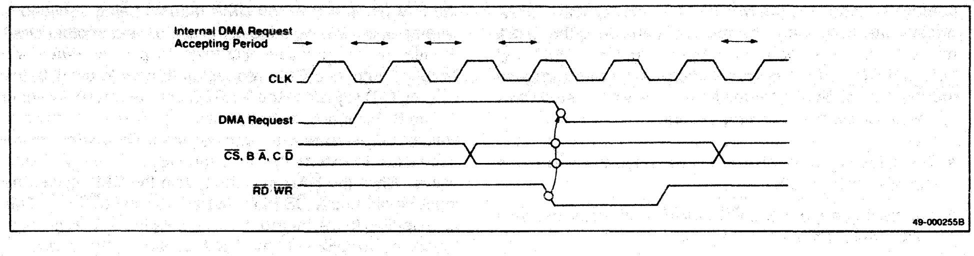
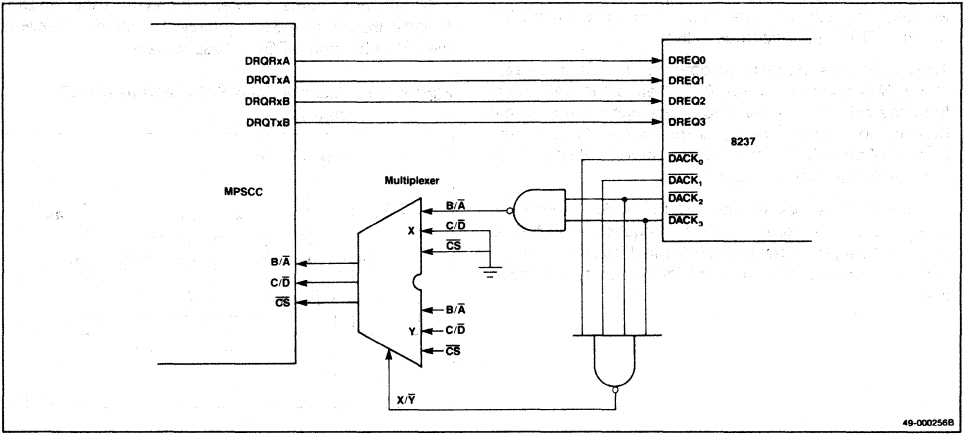


Figure 6-7. Example of MPSCC and µPD8237A Interface



CH.A DMA and Both CH.DMA-1 Modes. When both channels A and B of the MPSCC (D$_1$ and D$_0$ of the CR2A register are 10) or only channel A (D$_1$ and D$_0$ of the CR2A register are 01) is set in the DMA mode, CH.A DMA and both CH.DMA-1 modes are respectively specified. In these modes, the MPSCC enables an internally requested DMA operation while the following occurs:

- The CLK signal is at a low level when the $\overline{\text{HAI}}$ pin is inactive (high level).
- The trailing edge of the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal coincides with an active $\overline{\text{HAI}}$ pin.

DMA requests are accepted independent of the priority assigned to each. Whenever one is accepted, each DMA request (DRQ) pin becomes active.

The output DMA requests accepted in this manner are sent to the DMA controller (µPD8257). If either the $\overline{\text{HAO}}$ signal from the MPSCC, connected in the higher order of the daisy chain, or the HLDA signal from the CPU is input to the $\overline{\text{HAI}}$ pin of an MPSCC, that MPSCC enters the DMA cycle and starts the DMA process.

When a DMA request is received, the MPSCC sets the in service latch corresponding to the request at the trailing edge of the input signal to the $\overline{\text{HAI}}$ pin. If more than one in service latch is set, the one with highest priority is serviced first.

Subsequently, if either the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal is input while the $\overline{\text{HAI}}$ pin is active, the DMA request being serviced is cleared at the trailing edge of the signal, and another DMA request is the serviced according to priority. When the MPSCC outputs a DMA request and the HAI pin is 0, the $\overline{\text{CS}}$ and C/$\overline{\text{D}}$ signals in the MPSCC are automatically set to 0. The B/$\overline{\text{A}}$ signal becomes 0/1 according to the channel at which a DMA request is being serviced. Therefore, it is not necessary to externally make the $\overline{\text{CS}}$, C/$\overline{\text{D}}$ and B/$\overline{\text{A}}$ pins active. While the $\overline{\text{HAI}}$ pin is set to 0 in the DMA cycle, the input signal to the $\overline{\text{CS}}$ pin must be inactive ($\overline{\text{CS}}=1$). This also applies to the interrupt cycle while the $\overline{\text{INTAK}}$ pin is 0. When the µPD8257 is the DMA controller, the priority of DMA requests used in the MPCC and those in the µPD8257 must agree. Figures 6-8 and 6-9 show the DMA timing in this mode and an example of an interface between the µPD8257 and MPSCC, respectively.

Figure 6-9. Example of MPSCC and µPD8257 Interface

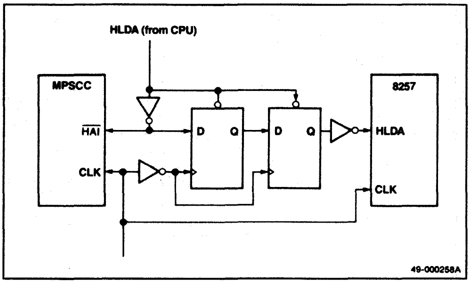
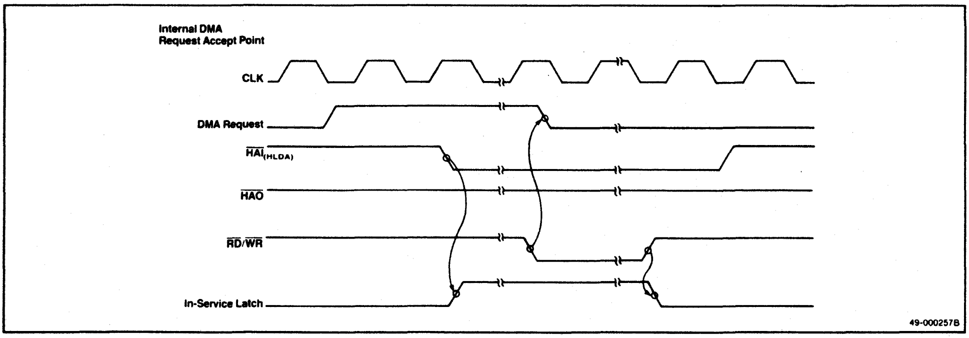


Figure 6-8. DMA Timing, CH.A DMA and Both CH.DMA-1 Modes



TRANSMISSION/RECEPTION TIMING

Figures 6-10 through 6-17 show the timing of MPSCC transmission/reception in the asynchronous, synchronous, and HDLC modes. These figures show timings related both to transmit/receive data and interrupt signals.

Figure 6-10. Transmission in Asynchronous Mode

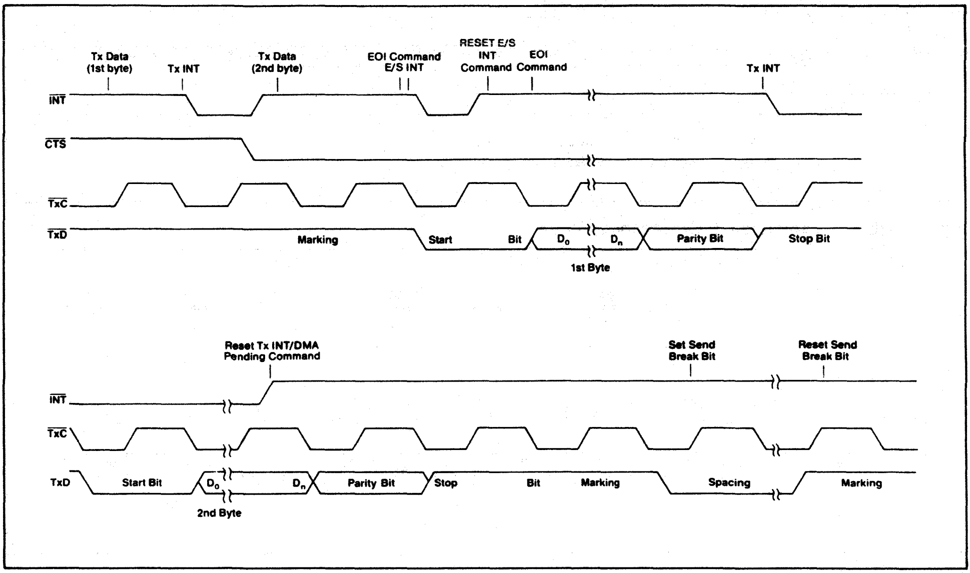
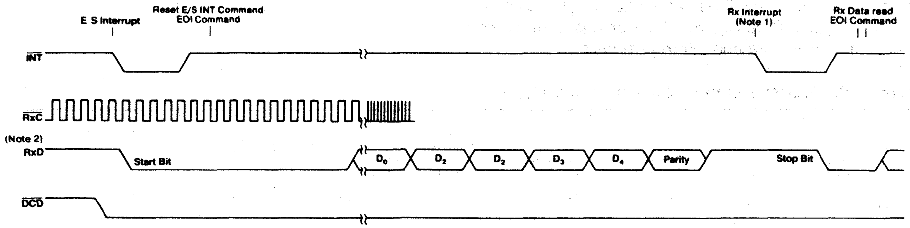


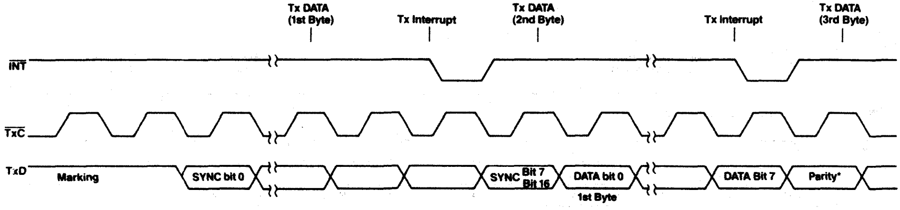
Figure 6-11. Reception in Asynchronous Mode



Notes: 1. The Special Rx Condition Interrupt occurs if a Parity, Framing, or Overrun error occurs.
2. The x16 clock mode is specified and each character is made up of 5 bits.

49-000260B

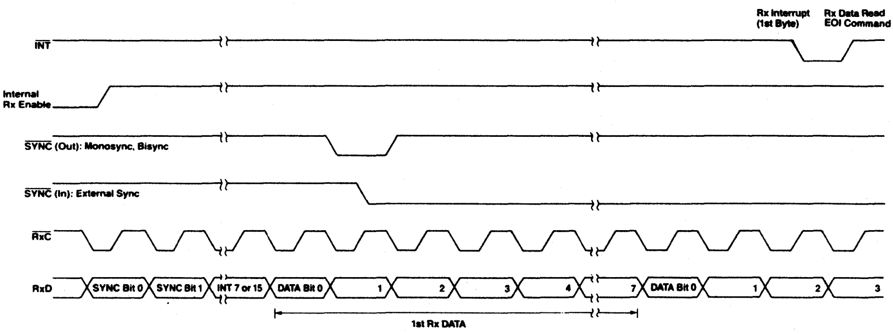
Figure 6-12. Transmission in Synchronous Mode



Note: *Parity enabled

49-000261B

Figure 6-13. Reception in Synchronous Mode



49-000262B

Figure 6-14 Reception in Synchronous Mode (CRC Calculation)

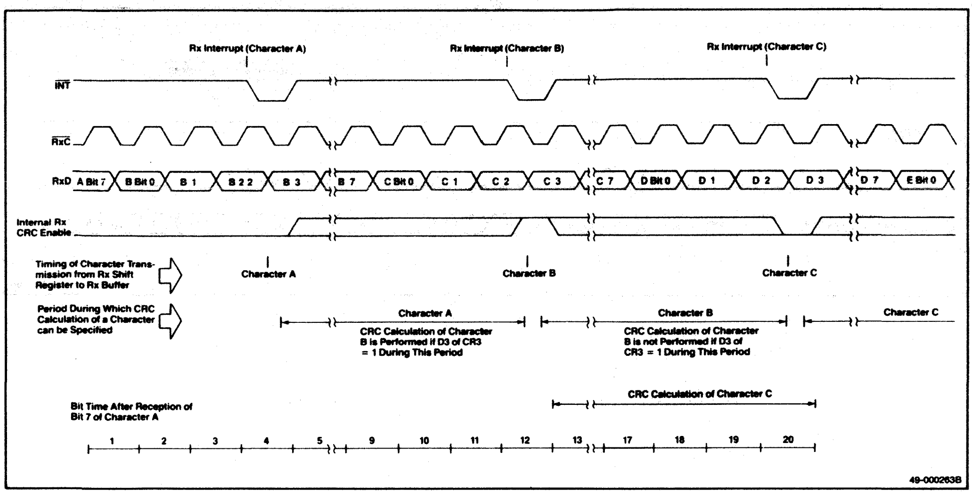
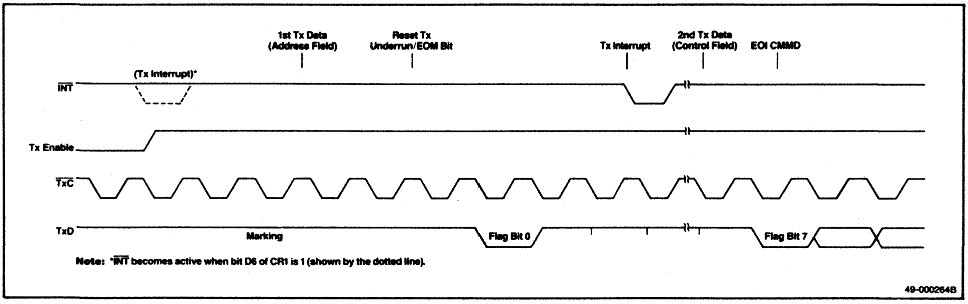


Figure 6-15. Transmission in HDLC Mode (1)



Note: *INT becomes active when bit D6 of CR1 is 1 (shown by the dotted line).

Figure 6-16. Transmission in HDLC Mode (2)

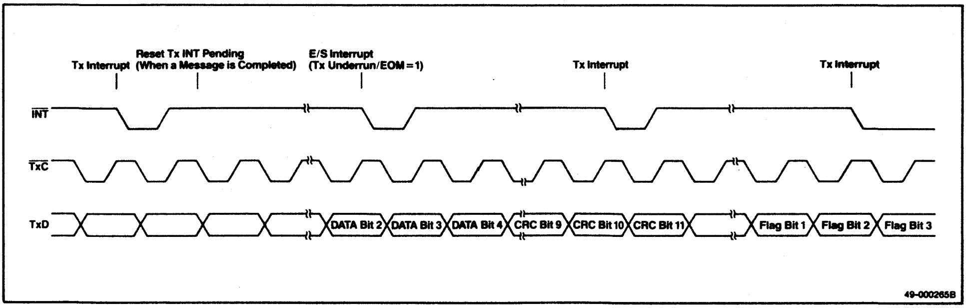
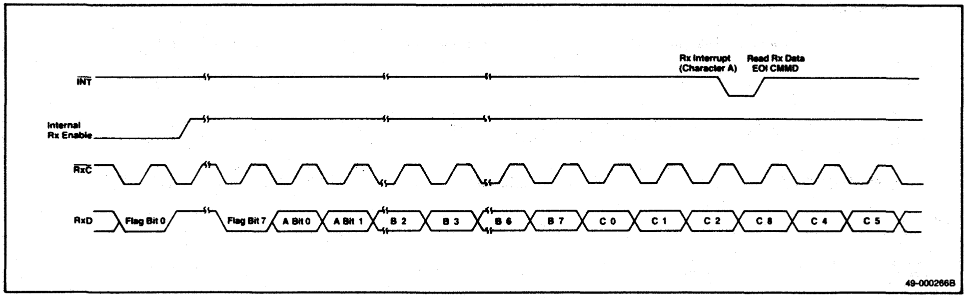


Figure 6-17. Reception in HDLC Mode



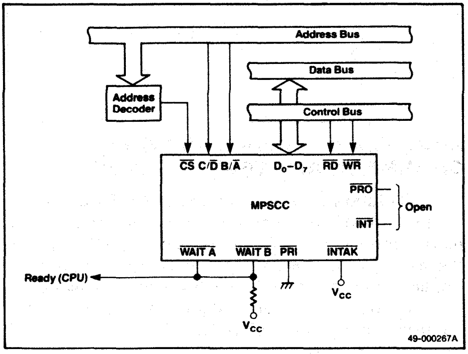
This section describes the various µPD7201A MPSCC system configurations.

STATUS POLLING MODE

The status polling mode is the simplest system configuration since it does not use the interrupt or DMA request pins. Data transmission between the MPSCC and the main system is by polling the SR0, SR1, and SR2 status registers of the MPSCC. The $\overline{\text{WAIT}}$ pins can be used if needed.

Figure 7-1 shows an example of the status polling mode system configuration during data transmission. If the MPSCC is set in the interrupt mode (both channels A and B) or the nonvectored mode, with the status affects vector mode enabled, the interrupt vector in the SR2B register can analyze interrupt sources.

Figure 7-1. Status Polling System



CHANNELS A AND B IN INTERRUPT MODE

In this type of system configuration, interrupts of both channels A and B perform data transmission between the MPSCC and the main system.

Vectored Mode

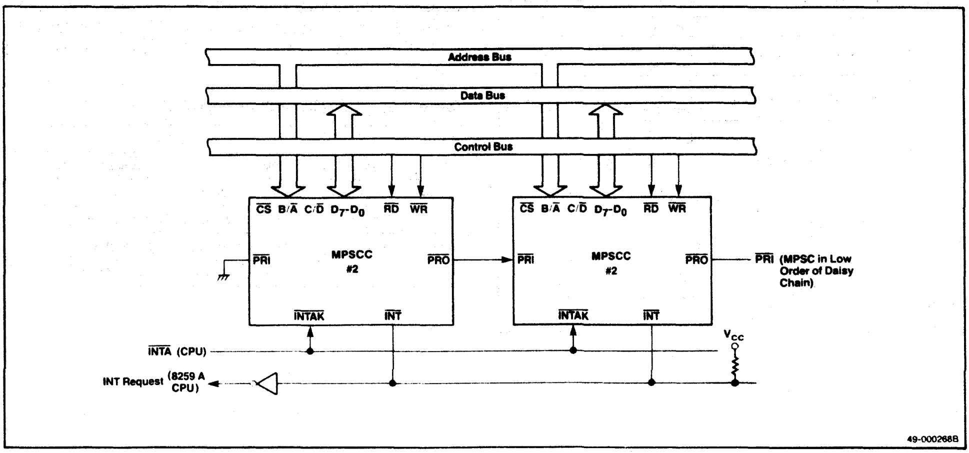
If the MPSCC is set in the vectored mode, an interrupt request signal is input from the MPSCC to the controller (µPD8259A, for example) or directly to the CPU. Set the vectored modes in the MPSCC as shown in table 7-1.

Table 7-1. MPSCC Vectored Mode

		MPSCC Mode	
CPU	8259A	MPSCC #1	Other than MPSCC #1
8085A	Not provided	85-1 Vectored	85-2 Vectored
	Provided	85-2 Vectored	85-2 Vectored
8086	Not provided	86 Vectored	86 Vectored
	Provided	86 Vectored	86 Vectored

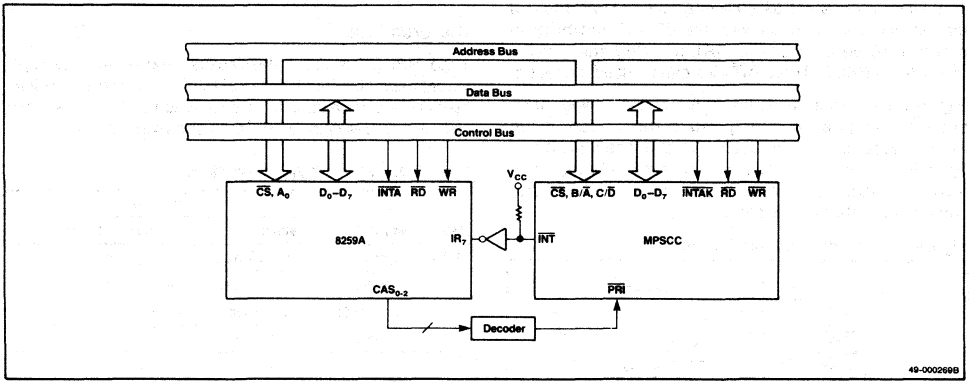
Figure 7-2 shows a system configuration using the interrupt vectored mode.

Figure 7-2. Interrupt System (Channels A and B in Interrupt, Vectored Mode)



Please note that the 85-3 vectored mode differs in its configuration. This mode does not allow a daisy chain using the $\overline{\text{PRI}}$ and $\overline{\text{PRO}}$ pins. However, the output signal from the $\overline{\text{INT}}$ pin becomes active even if the level of the $\overline{\text{PRI}}$ pin is 1. Therefore, the 85-3 vectored mode can be used as a slave of the 8259A. Figure 7-3 shows a system configuration using this mode. Note that 85-3 vectored mode is available for the 7201A only.

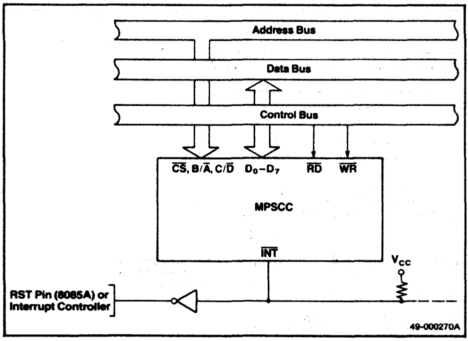
Figure 7-3. Interrupt System (85-3 Vectored Mode)



Nonvectored Mode

Figure 7-4 shows a system configuration in the nonvectored mode. An output signal from the $\overline{\text{INT}}$ pin of the MPSCC is input either to the RST pin of the µPD8085A (RST5, RST6.6, RST7.5) or to the interrupt controller (such as µPD8259A). When more than two MPSCCs are connected, software must also be used to decide which MPSCC is requesting an interrupt.

Figure 7-4. Interrupt System (Channels A and B in Interrupt, Nonvectored Mode)



CHANNEL A IN DMA MODE AND CHANNEL B IN INTERRUPT MODE

Figure 7-5 shows a system configuration with channel A set in the DMA mode and channel B in the interrupt mode for data transmission between the MPSCC and the main system. This configuration permits high-speed transmission on channel A and low-speed transmission on channel B. DMA request signals are input to a DMA controller, such as the µPD8257.

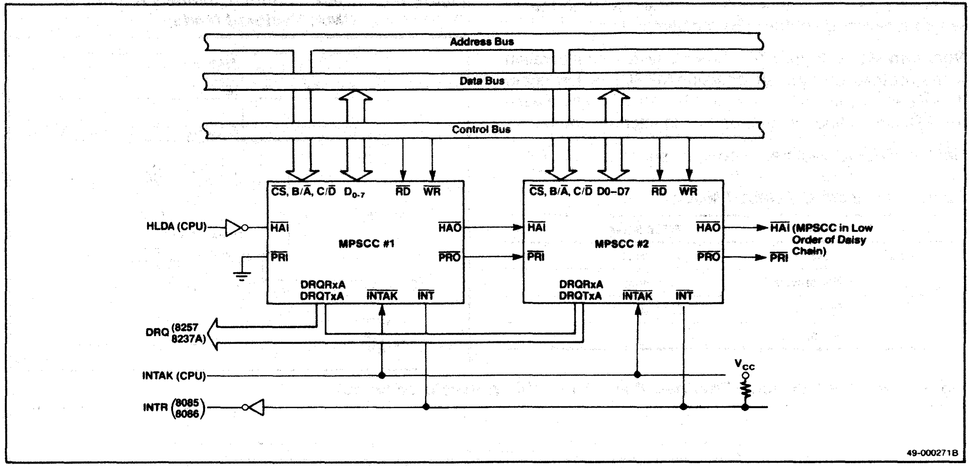
If the MPSCC is set in the nonvectored mode, interrupt request signals from the MPSCC are input either to the interrupt controller (such as the µPD8259) or directly to the CPU. In this case, set the MPSCC modes as shown in table 7-2.

Table 7-2. MPSCC Nonvectored Mode

		MPSCC Mode	
CPU	**8259A**	**MPSCC #1**	**Other than MPSCC #1**
8085A	Not provided	85-1 Nonvectored	85-2 Nonvectored
	Provided	85-2 Nonvectored	85-2 Nonvectored
8086	Not provided	86 Nonvectored	86 Nonvectored
	Provided	86 Nonvectored	86 Nonvectored

In the nonvectored mode, interrupt request signals can be input to the RST pin of the µPD8085A.

Figure 7-5. Interrupt/DMA System (Vectored Mode)



CHANNELS A AND B IN DMA MODE

It is possible to use two system configurations with both channels set in the DMA mode. The DMA mode used decides which system is configured. In either case, data communication between the MPSCC and the main system is implemented by performing a DMA operation with both channels. In this manner, high-speed data transmission is possible.

The DMA request signal is input to the DMA controller (for example, the µPD8237A). The interrupt signal is generated in response to the E/S, the first Rx character, or the special Rx condition interrupt. Figure 7-6 shows a DMA system configuration.

Both CH.DMA-2 Mode

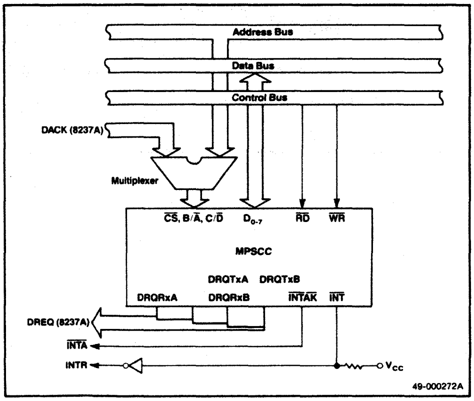
In both CH.DMA-2 mode (7201A only), the DMA process is performed using the DACK signal instead of the DMA controller's $\overline{HAI}$ and $\overline{HAO}$ pins. Figure 7-6 shows a system configuration in this mode. If the configuration uses only one MPSCC, the vectored mode can be used. If so, set the MPSCC vectored modes as shown in table 7-3.

Table 7-3. Both CH.DMA-2 Mode

CPU	8259A	Mode to be set in MPSCC
8085A	Not provided	85-1 mode
	Provided	85-2, 85-3 modes
8086	Not provided	86 mode
	Provided	

Note: If two or more MPSCCs are used, specify the nonvectored mode.

Figure 7-6. DMA System (Both CH.DMA-2 Mode)



Both CH.DMA-1 Mode

In both CH.DMA-1 mode, DMA processes are carried out using the $\overline{HAI}$ and $\overline{HAO}$ pins. Both channels can simultaneously perform transmission operations using the Tx length register (D_6 of the CR1 register = 1).

Vectored Mode. Figure 7-7 shows a system configuration in the vectored mode. Use only one MPSCC in this mode. The $\overline{INTAK}$ signal functions inside the MPSCC as if it were the $\overline{PRI}$ signal and assigns priority to interrupt sources.

Set the MPSCC vectored modes as shown in table 7-4.

Table 7-4. Both CH.DMA-1 Mode

CPU	8259A	MPSCC Modes
8085A	Provided	85-1 mode
	Not provided	85-2 mode
8086	Provided	86 mode
	Not provided	

Nonvectored Mode. Figure 7-8 shows a system configuration in the nonvectored mode. In this type of system, software must analyze interrupt sources.

Figure 7-7. DMA System (Channels A and B in DMA, Vectored Mode)

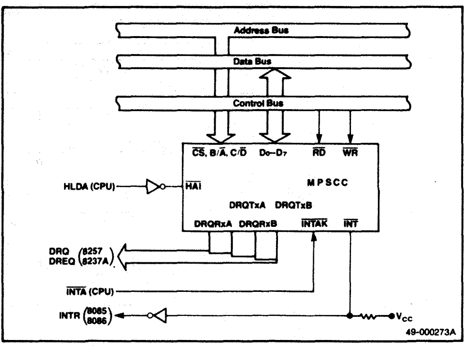
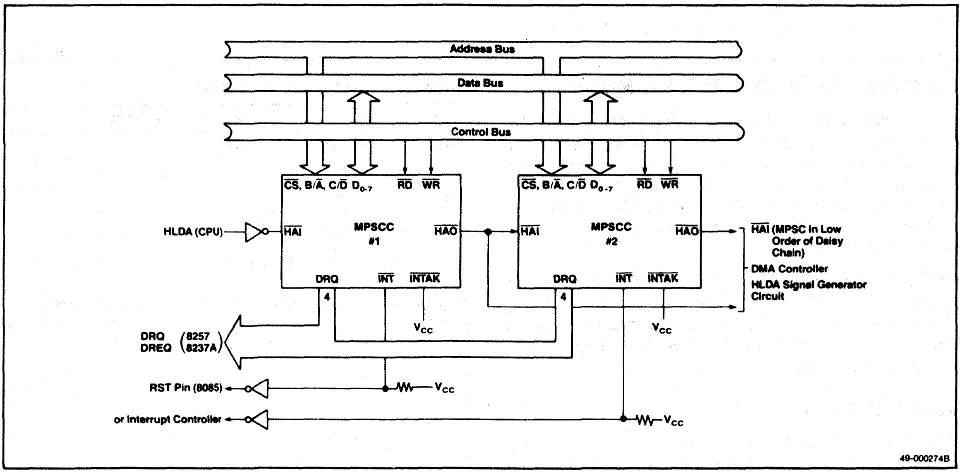


Figure 7-8. DMA System (Channels A and B in DMA, Nonvectored Mode)



μPD7201A Specifications

Absolute Maximum Ratings

T_a = 25°C	
Power Supply, V_{CC}	−0.5V to +7.0V
Input Voltages, V_I	−0.5V to +7.0V
Output Voltages, V_O	−0.5V to +7.0V
Operating Temperature, T_{OPT}	0°C to +70°C
Storage Temperature, T_{STG}	−65°C to +125°C

*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.

DC Characteristics

T_a = 0°C to +70°C; V_{CC} = +5V ± 10%

		Limits				
Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Input Low Voltage	V_{IL}	−0.5		+0.8	V	
Input High Voltage	V_{IH}	+2.0		V_{CC} + 0.5	V	
Output Low Voltage	V_{OL}			+0.45	V	I_{OL} = +2.0mA
Output High Voltage	V_{OH}	+2.4			V	I_{OH} = 200μA
Input Leakage Current	I_{IL}			±10	μA	V_{IN} = V_{CC} to 0V
Output Leakage Current	I_{OL}			±10	μA	V_{OUT} = V_{CC} to 0V
V_{CC} Supply Current	I_{CC}			230	mA	

Capacitance

T_a = 25°C; V_{CC} = GND = 0V

		Limits				
Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Input Capacitance	C_{IN}			10	pF	fc = 1MHz Unmeasured pins returned to GND.
Output Capacitance	C_{OUT}			15	pF	
I/O Capacitance	$C_{I/O}$			20	pF	

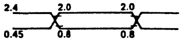
AC Characteristics

T_a = 0°C to +70°C; V_{CC} = +5V ± 10%

		Limits				
Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Clock Cycle	t_{CY}	200		4000	ns	
Clock High Width	t_{CH}	70		2000	ns	
Clock Low Width	t_{CL}	70		2000	ns	
Clock Rise Time	t_r	0		30	ns	
Clock Fall Time	t_f	0		30	ns	
Address Setup to RD	t_{AR}	0			ns	
Address Hold from RD	t_{RA}	0			ns	
RD Pulse Width	t_{RR}	200			ns	
Data Output Delay from Address	t_{AD}			200	ns	
Data Output Delay from RD	t_{RD}			200	ns	
Data Float Delay from RD	t_{DF}	10		100	ns	
Address Setup to WR	t_{AW}	0			ns	
Address Hold from WR	t_{WA}	0			ns	
WR Pulse Width	t_{WW}	200			ns	
Data Setup to WR	t_{DW}	130			ns	
Data Hold from WR	t_{WD}	0			ns	
PRO Delay from PRI	t_{PIPO}			100	ns	
PRO Delay from INTA	t_{IAPO}			200	ns	
PRI Setup to INTA	t_{PIW}	0			ns	
PRI Hold from INTA	t_{IP}	20			ns	
INTA Pulse Width	t_{II}	200			ns	
Data Output Delay from INTA	t_{ID}			200	ns	
Data Float Delay from INTA	t_{DF}	10		100	ns	
Request Hold from RD/WR	t_{CQ}			150	ns	
HAI Setup to RD/WR	t_{LR}	300			ns	
HAI Hold from RD/WR	t_{RL}	0			ns	
HAO Delay from HAI	t_{HIHO}			100	ns	
Data Clock Cycle	t_{DCY}	400			ns	RxC, TxC
Data Clock High Width	t_{DCH}	180			ns	RxC, TxC
Data Clock Low Width	t_{DCL}	180			ns	RxC, TxC
Tx Data Delay from TxC	t_{TD}			300	ns	x1 Mode
				1000		x16, 32, 64
Rx Data Setup to RxC	t_{DS}	0			ns	
Rx Data Hold from RxC	t_{DH}	140			ns	
INT Delay Time from Tx Data	t_{ITD}			4–6	t_{CY}	
INT Delay Time from RxC	t_{IRD}			7–11	t_{CY}	
CTS, DCD, SYNC High Pulse Width	t_{PH}	200			ns	
CTS, DCD, SYNC Low Pulse Width	t_{PL}	200			ns	
External INT from CTS, DCD, SYNC	t_{PO}			500	ns	
Recovery Time Between Controls	t_{RV}	300			ns	
WAIT Delay Time from Address	t_{CW}			120	ns	
SYNC Setup to RxC	t_{DRxC}			100	ns	

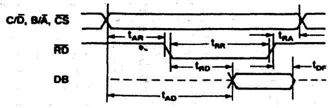
Notes: 1: $\overline{\text{RESET}}$ must be active for a minimum of one complete CLK cycle.
2: In all modes system clock rate must be 4.5 times data rate.

AC Waveform Measurement Points

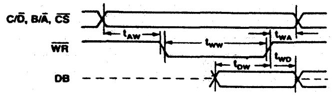


Timing Waveforms

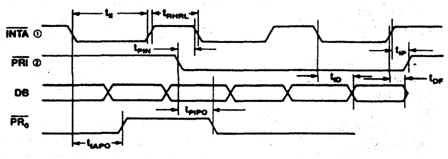
Read Cycle



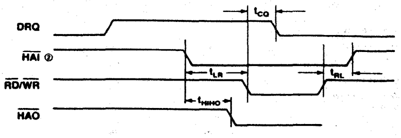
Write Cycle



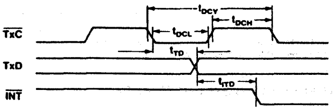
INTA Cycle



DMA Cycle

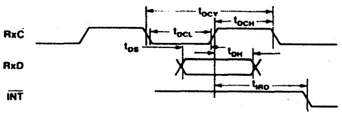


Transmit Data Cycle

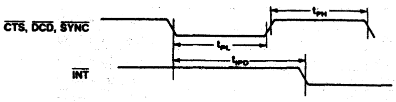


Notes: ① $\overline{INTA}$ signal acts as $\overline{RD}$ signal.
② $\overline{PRI}$ and $\overline{HAI}$ signals act as $\overline{CS}$ signal.

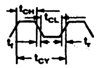
Receive Data Cycle



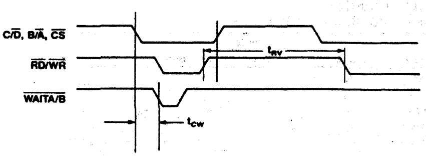
Other Timing



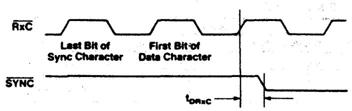
Clock



Read/Write Cycle (Software Block Transfer Mode)



Sync Pulse Generation (External Sync Mode)



Appendix
µPD7201A and 7201 Comparison

<table>
<tr><th>µPD7201A</th><th>µPD7201</th></tr>
<tr><td colspan="2">(1) TxLR set mode (bit D_6 of CR1 register = 1)</td></tr>
<tr><td>
<table>
<tr><th colspan="3">CR1</th></tr>
<tr><td>D_7</td><td>D_6</td><td>D_5 D_0</td></tr>
<tr><td></td><td>TxLR Set</td><td></td></tr>
</table>
Data transmission by using an interrupt or DMA is simplified.
0: The same as 7201
1: TxLR set mode
To realize this mode operation, these two 16-bit registers are additionally provided.
● TxLR
This is a transmit length register. Immediately after the setting of bit D_6 of the CR1 register, a desired number of transmit data characters are written to this register.
● TxLC (SR3 and SR4)
This is a transmit length counter which indicates the number of transmit data characters.
The value of the TxLC is incremented by 1 each time the TxINT or DRQTx signal becomes active and when it matches that of the TxLR, a Tx interrupt or DRQ signal, which generates in response to a Tx empty state, is masked. Once set, this mode will not be cleared until the system is reset.
</td><td>
<table>
<tr><th colspan="3">CR1</th></tr>
<tr><td>D_7</td><td>D_6</td><td>D_5 D_0</td></tr>
<tr><td></td><td>0</td><td></td></tr>
</table>
This bit is always “0”.
</td></tr>
</table>

µPD7201A	µPD7201

(2) DMA mode (bit D_1, D_0 of CR2A register = 1, 1)

µPD7201A

CR2A		
D_7 … D_2	D_1	D_0
	INT/DMA MODE 1	INT/DMA MODE 0
	0 0 Both CH.INT 0 1 CH.A: DMA,CH.B:INT 1 0 Both CH.DMA-1 1 1 Both CH.DMA-2	

- DMA-1 Mode

The same as the DMA mode of the 7201.

- DMA-2 Mode

Priority is not determined in the MPSC.

The DMA controller, in response to a DMA request, determines the priority.

The DACK signal controls the CS, C/D, B/A, RD, and WR pins of the MPSC. The DTRB/HAI and DTRA/HAO pins respectively function as DTRB and DTRA pins.

µPD7201

CR2A		
D_7 … D_2	D_1	D_0
	INT/DMA MODE 1	INT/DMA MODE 0
	0 0 Both CH.INT 0 1 CH.A: DMA,CH.B:INT 1 0 Both CH.DMA-1	

DMA-2 mode is not available in the 7201.

(3) 85-3 Vectored mode (bits D_4, D_3 of CR2A register = 1, 1)

µPD7201A

CR2A			
D_7 … D_5	D_4	D_3	D_2 … D_0
	INT MODE 1	INT MODE 0	
	0 0 85-1 Vectored 0 1 85-2 Vectored 1 0 86 Vectored 1 1 85-3 Vectored		

85-3 Vectored Mode

In this mode the INT signal becomes active when an internal interrupt occurs even if the PRI is 1. Therefore, a daisy-chaining system using the PRI and PRO signals cannot be configured. The µPD7201A can be used as a slave of µPD8259A.

µPD7201

CR2A			
D_7 … D_5	D_4	D_3	D_2 … D_0
	INT MODE 1	INT MODE 0	
	0 0 85-1 Vectored 0 1 85-2 Vectored 1 0 86 Vectored		

85-3 Vectored mode is not available in the 7201.

<table>
<tr><th>μPD7201A</th><th>μPD7201</th></tr>
<tr><td colspan="2">(4) Masking first Rx character interrupt (bit D_6 of CR2A register = 1)</td></tr>
<tr><td>
<table>
<tr><td colspan="3">CR2</td></tr>
<tr><td>D_7</td><td>D_6</td><td>D_5 D_0</td></tr>
<tr><td></td><td>Rx INT Mask</td><td></td></tr>
</table>
The first Rx character interrupt can be masked.
0: The same as the 7201
1: The first Rx character interrupt is masked when the INT on first Rx character mode is selected. This mode is not cleared by the enable INT on next Rx character command.
</td><td>
<table>
<tr><td colspan="3">CR2</td></tr>
<tr><td>D_7</td><td>D_6</td><td>D_5 D_0</td></tr>
<tr><td></td><td>0</td><td></td></tr>
</table>
This bit must be always "0".
</td></tr>
<tr><td colspan="2">(5) Reception of 2nd CRC character (HDLC mode)</td></tr>
<tr><td>
<table>
<tr><td>........</td><td>CRC #2</td><td>Closing Flag</td></tr>
</table>
Rx buffer (normal data)
Rx Buffer (normal data)
The character received immediately before a flag (normally, CRC #2) is correctly loaded to the Rx buffer.
</td><td>
<table>
<tr><td>........</td><td>CRC #2</td><td>Closing Flag</td></tr>
</table>
Rx buffer (abnormal data)
Rx Buffer (erroneous data)
The character received immediately before a flag (normally, CRC #2) is not correctly loaded to the Rx buffer.
</td></tr>
<tr><td colspan="2">(6) Tx interrupt/DRQ signal in Tx buffer empty state (HDLC mode)</td></tr>
<tr><td>When TxLR set bit (bit D_6 of the CR1 register) is 1, the Tx interrupt/DRQ signal automatically becomes active in response to issuance of the Tx enable command. Therefore, the first data byte can be transferred to the 7201A by using an interrupt or DMA.</td><td>The Tx interrupt/DRQ signal does not become active until the first data byte is written into the 7201. Therefore, the first data byte after a channel has been reset must be written from the CPU to the 7201.</td></tr>
<tr><td colspan="2">(7) Abort detection before detecting flag (HDLC mode)</td></tr>
<tr><td>Abort sequences are detected only after a flag has been received. Therefore, if a receive line is in the mark state when Rx has been enabled, this will not be detected as an abort sequence.</td><td>An abort sequence is detected even before a flag has been received (during hunt phase) when the Rx has been enabled.</td></tr>
</table>

µPD7201A	µPD7201
(8) Detection of frame transmission completion (HDLC mode)	
When transmission of a frame has been completed (i.e., when closing flags are transmitted), the all sent bit (bit D_0 of the SR1 register) becomes 1 and an E/S interrupt occurs at the same time.	There is no information indicating completion of frame transmission (i.e., transmission of a closing flag).
(9) Transmission of sequential frames (HDLC mode)	
The overhead of the CPU is lessened when a series of frames are transmitted by using DMA mode. • Initialize Tx CRC calculator function The Tx CRC calculator is automatically reset (all the bits are reset to 1's) when a flag is loaded to the Tx shift register. • Reset Tx underrun/EOM function When the first character is loaded to the MPSCC, the Tx underrun/EOM bit is automatically reset.	When a series of frames are transmitted by using DMA, the CPU is required to perform such things as issuance of the initialize Tx CRC calculator or Tx underrun/EOM command and managing the timing of the command issuance.
(10) CRC transmission in Tx underrun (HDLC mode)	
If the Tx underrun condition occurs when the TxLR set bit (bit D_6 of the CR1 register) is 1, an abort sequence or CRC is automatically transmitted.	If the Tx underrun condition occurs before all the data in a frame have been transmitted, a send abort command must be issued.

μPD7201A

APPLICATION NOTE

Chapter 1

Introduction

This application note describes a uPD7201A application for the design of the hardware of software of a system using the uPD7201A.

Detailed explanations of the uPD7201A functions and operation are omitted. For this information, refer to the other parts of the Product Description and to the Data Book.

The first chapter explains uPD7201A operation from the viewpoint of hardware and software designers for reference in examination or discussion of system configuration. Particularly, actual process flow for data transmission and reception is shown to facilitate understanding of system operation. Chapters 4 and 5 provide configuration examples of systems using the uPD7201A and explain the software required for each.

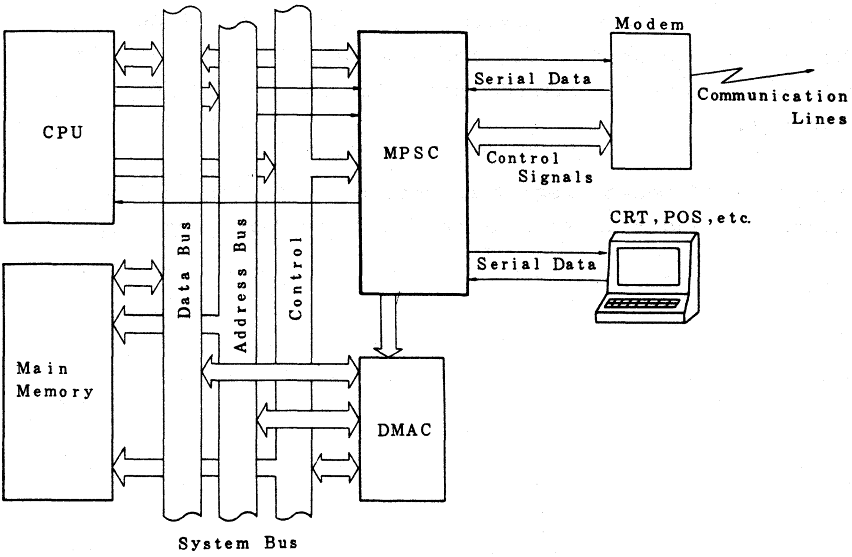


Figure 1.1 Example of configuration of system using MPSC

Chapter 2

μPD7201A Register Configuration

The μPD7201A contains six control registers (CR0 - CR5), two synchronization pattern registers (CR6 and CR7), one transmit length register (TxLR), and four status registers (SR0, SR1, SR3, and SR4) for channel A and for channel B, and a vector register (SR2B) for channel B.

Table 2.1 lists the names and main functions of the registers.

Table 2.2 lists the structure of each register.

For the bit functions and meanings of the registers, refer to the User's Manual.

Selection of the registers is made by combining the B/$\overline{A}$, C/$\overline{D}$, and $\overline{CS}$ lines. The registers CR1 to CR7 and SR1 to SR4 are selected by using the CR0) pointer bit. Some registers or register bits are contained only in either channel A or channel B. However, if they are set for the channel in which they are contained, they function commonly to both channels.

Figure 2.1 shows register setting examples to operate channels A and B in Asynchronous mode by using interrupts.

Table 2.1 μPD7201A register configuration

Classification	Register name	Channel	Function
Control register	CR0	A, B	Register selecting pointer, CRC logic initialization, various control commands.
	CR1	A, B	Interface mode with CPU. Interrupt/DMA control.
	CR2	A	Pin selection, vector mode selection, interrupt/DMA control.
		B	Initial value of interrupt vector.
	CR3	A, B	Receive operation control, Auto Enable mode.
	CR4	A, B	Operation protocol mode.
	CR5	A, B	Send operation control, general-purpose output line control.
	CR6	A, B	SYNC character or secondary station address.
	CR7	A, B	SYNC character or flag.
	TxLR	A, B	Setting of the number of send data words (16 bits).
Status register	SR0	A, B	E/S bit, send and receive buffer status, interrupt state.
	SR1	A, B	Special Rx Condition source, Residue Code, transmitter status.
	SR2	B	Interrupt vector value.
	SR3, SR4	A, B	TxLC value

Table 2.2 µPD7201A register configuration

Bit Register name	D7	D6	D5	D4	D3	D2	D1	D0
CR0	CRC1	CRC0	CMD2	CMD1	CMD0	PTR2	PTR1	PTR0
	00 Null 01 Initialize Rx CRC Calculator 10 Initialize Tx CRC Calculator 11 Reset Tx underrun/EOM Bit		000 Null 001 Send Abort (HDLC) 010 Reset E/S INT 011 Channel Reset 100 Enable INT on NEXT Rx CHAR 101 Reset Tx INT/DMA Pending 110 Error Reset 111 End of INT (CH.A only)			(000 CR0,SR0) 001 CR1,SR1 010 CR2,SR2 (CH.B only) 011 CR3,SR3 100 CR4,SR4 101 CR5 110 CR6 111 CR7		

Bit Register name	D7	D6	D5	D4	D3	D2	D1	D0
CR1	Wait Enable	TxLR Set	Wait on Rx/Tx	Rx INT Mode I	Rx INT Mode 0 Vector	Status Affects	TX INT/ DMA	E/S INT Enable
	0 Disable 1 Enable	0 No operation 1 Tx Length Register Set (HLDC only)	0 Tx 1 Rx	00 INT/DMA Disable 01 INT on First CHAR. 10 INT on ALL CHAR. (With Parity) 11 INT on ALL CHAR. (No Parity)		0 Fixed Vector 1 Modified Vector (CH.B only)	0 Disable 1 Enable	0 Disable 1 Enable

Bit Register name	D7	D6	D5	D4	D3	D2	D1	D0
CR2A	$\overline{\text{RTSB}}$/ $\overline{\text{SYNCB}}$ Select	RxINT Mask	Vector Mode	INT Mode 1	INT Mode 0	Priority Select	INT/DMA Mode 1	INT/DMA Mode 0
	0 $\overline{\text{RTSB}}$ 1 $\overline{\text{SYNCB}}$	0 Don't Mask 1 Mask	0 Non-Vectored 1 Vectored	00 85-1 Vectored 01 85-2 Vectored 10 86 Vectored 11 85-3 Vectored		0 TxA→RxB 1 RxB→TxA	00 Both CH. INT 01 CH. A: DMA CH. B: INT 10 Both CH. DMA-1 (fixed priority) 11 Both CH.DMA-2 (free from priority)	
	* 0: RxA TxA RxB TxB E/SA E/SB 1: RxA RxB TxA TxB E/SA E/SB High → Low							

Bit Register name	D7	D6	D5	D4	D3	D2	D1	D0
CR2B	V7	V6	V5	V4	V3	V2	V1	V0

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
CR3		Rx Bits/CHAR 1	Rx Bits/CHAR 0	Auto Enable	Enter Hunt Phase	Rx CRC Enable	Address Search Mode	SYNC CHAR Load Inhibit	Rx Enable
	Async	00 5 Bits/CHAR. 01 7 Bits/CHAR. 10 6 Bits/CHAR. 11 8 Bits/CHAR.		0 Disable 1 Enable	0	0	0	0	0 Disable 1 Enable
	Sync				0 Nop 1 Re-enable	0 Disable 1 Enable		0 Nop 1 Inhibit	
	HDLC						0 Nop 1 Available	0	

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
CR4		Clock Rate 1	Clock Rate 0	SYNC Mode 1	SYNC Mode 0	Stop Bits 1	Stop Bits 0	Parity Even/Odd	Parity Enable
	Async	00 x 1 Clock Mode 01 x 16 Clock Mode 10 x 32 Clock Mode 11 x 64 Clock Mode		00		01 1stop bit/CHAR 10 1 1/2stop bit/CHAR 11 2stop bits/CHAR		0 Odd 1 Even	0 Disable 1 Enable
	Sync	00		00 8 bit SYNC 01 16 bit SYNC 11 EXT SYNC		00 SYNC Mode			
	HDLC			10 HDLC				0	0

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
CR5		DTR	Tx Bits/CHAR 1	TX Bits/CHAR 0	Send Break	Tx Enable	CRC-16/CCITT	RTS	Tx CRC Enable
	Async	0 $\overline{\text{DTR}}$=1 1 $\overline{\text{DTR}}$=0	00 5 or less Bits/CHAR. 01 7 Bits/CHAR. 10 6 Bits/CHAR. 11 8 Bits/CHAR.		0 Marking 1 Spacing	0 Disable 1 Enable	0	0 $\overline{\text{RTS}}$=1 (when All Sent) 1 $\overline{\text{RTS}}$=0	0
	Sync						0 CCITT-0 1 CRC-16	0 $\overline{\text{RTS}}$=1 1 $\overline{\text{RTS}}$=0	0 Disable 1 Enable
	HDLC				0		0 CCITT-1		

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
CR6		SYNC Bit 7 - 0							
	Async	not used							
	Monosync	Tx SYNC CHAR.							
	Bisync	SYNC CHAR. (Bit 7 - 0)							
	EXT sync	Tx SYNC CHAR.							
	HDLC	Secondary Address							

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
CR7		SYNC Bit 15 - 8							
	Async	not used							
	Monosync	Rx SYNC CHAR.							
	Bisync	SYNC CHAR. (Bit 15 - 8)							
	EXT sync	not used							
	HDLC	Flag (01111110)							

Bit / Register name	D7	D6	D5	D4	D3	D2	D1	D0
TxLR-L	Tx Length Register bit 7 - 0							

Bit / Register name	D7	D6	D5	D4	D3	D2	D1	D0
TxLR-H	Tx Length Register bit 15 - 8							

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
SR0		Break/ Abort	Tx Under-run/EOM	CTS	SYNC/ Hunt	DCD	Tx Buffer Empty	INT Pending	Rx CHAR. Available
	Async	1 Break Detected 0 Normal	Unknown	1 $\overline{\text{CTS}}$=0 0 $\overline{\text{CTS}}$=0	1 $\overline{\text{SYNC}}$=0 0 $\overline{\text{SYNC}}$=1	1 $\overline{\text{DCD}}$=0 0 $\overline{\text{DCD}}$=1	1 Tx Buff Empty 0 Tx Buff Full	1 INT Pending 0 No INT Pending (CH.A only)	1 Rx CHAR Available 0 Rx Buff. Empty
	EXT sync Monosync Bisync	0	1 Tx Underrun /EOM 0 Not Tx Underrun		1 Hunt Phase 0 Exit Hunt Phase		1 Tx Buff Empty 0 Tx Buff Full (With CRC)		
	HDLC	1 Abort Detect-ed 0 Normal							

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
SR1		End of Frame	CRC/ Framing Error	Rx/ Overrun Error	Parity Error	Residue Code 2	Residue Code 1	Residue Code 0	All Sent
	Async	0	1 Framing Error 0 No Error	1 Rx Overrun Error 0 No Error	1 Parity Error 0 No Error	Unknown			1 All Sent 0 Not All Sent
	Sync		1 CRC Error 0 No Error						1
	HDLC	1 End of Frame 0 Not End of Frame			0	When no residue exists 5 bits/CHAR 100 6 bits/CHAR 000 7 bits/CHAR 011 8 bits/CHAR 011			1 All Sent 0 Not All Sent

Register name	Bit / Mode	D7	D6	D5	D4	D3	D2	D1	D0
SR2B	CR1 bit D2=0	V7	V6	V5	V4	V3	V2	V1	V0
	CR1 bit D2=1 CR2A bits D4,D3 =00,01,11 (85 mode)	V7	V6	V5	1 CH.A 0 CH.B	00 Tx Buffer Empty 01 External/Status Change 10 Receive Character Available 11 Special Rx Condition		V1	V0
	CR1 bit D2=1 CR2A bits D4,D3 =10 (86 mode)	V7	V6	V5	V4	V3	1 CH.A 0 CH.B	00 Tx Buffer Empty 01 External/Status Change 10 Receive Character Available 11 Special Rx Condition	

Bit / Register name	D7	D6	D5	D4	D3	D2	D1	D0
SR3	Tx Length Counter bit 7 - 0							

Bit / Register name	D7	D6	D5	D4	D3	D2	D1	D0
SR4	Tx Length Counter bit 15 - 8							

```
  UCOM-85 ASSEMBLE LIST    (                                     ) PAGE 0001

E STNO ADRS OBJECT M   SOURCE STATEMENTS


 0001                  ;***********************************
 0002                  ; UPD7201 REGISTER SETTING EXAMPLE *
 0003                  ;***********************************
 0004                  ;
 0005                  ; COMMON REGISTER INITIALIZE
 0006                  ;
 0007 0000 3E18                MVI     A,18H          ;CHANNEL RESET
 0008 0002 D3F1                OUT     CHACT
 0009 0004 D3F3                OUT     CHBCT
 0010 0006 3E02                MVI     A,2H           ;PTR 2
 0011 0008 D3F1                OUT     CHACT
 0012 000A D3F3                OUT     CHBCT
 0013 000C 3E00                MVI     A.CR2A         ;BUS I/F MODE
 0014 000E D3F1                OUT     CHACT
 0015 0010 3E00                MVI     A.CR2B         ;INTERRUPT VECTOR
 0016 0012 D3F3                OUT     CHBCT
 0017                  ;
 0018                  ; CH-A OPERATION MODE
 0019                  ;
 0020 0014 3E04        STCHA:  MVI     A,4H           ;PTR 4A
 0021 0016 D3F1                OUT     CHACT
 0022 0018 3E4F                MVI     A.CR4A         ;CH-A OPERATION MODE
 0023 001A D3F1                OUT     CHACT
 0024 001C 3E11                MVI     A.11H          ;PTR 1A.RESET E/S INT
 0025 001E D3F1                OUT     CHACT
 0026 0020 3E13                MVI     A.CR1A         ;INT/DMA MODE
 0027 0022 D3F1                OUT     CHACT
 0028 0024 3E03                MVI     A,3H           ;PTR 3A
 0029 0026 D3F1                OUT     CHACT
 0030 0028 3E41                MVI     A.CR3A         ;RX PARAMETERS
 0031 002A D3F1                OUT     CHACT
 0032 002C 3E05                MVI     A.5H           ;PTR 5A
 0033 002E D3F1                OUT     CHACT
 0034 0030 3E28                MVI     A.CR5A         ;TX PARAMETERS
 0035 0032 D3F1                OUT     CHACT
 0036                  ;
 0037                  ; CH-B OPERATION MODE
 0038                  ;
 0039 0034 3E04        STCHB:  MVI     A,4H           ;PTR 4B
 0040 0036 D3F3                OUT     CHBCT
 0041 0038 3E4F                MVI     A.CR4B         ;CH-B MODE
 0042 003A D3F3                OUT     CHBCT
 0043 003C 3E11                MVI     A.11H          ;PTR 1B,RESET E/S INT
 0044 003E D3F3                OUT     CHBCT
 0045 0040 3E1F                MVI     A.CR1B         ;INT/DMA MODE
 0046 0042 D3F3                OUT     CHBCT
 0047 0044 3E03                MVI     A,3H           ;PTR 3B
 0048 0046 D3F3                OUT     CHBCT
 0049 0048 3E41                MVI     A,CR3B         ;RX PARAMETERS
 0050 004A D3F3                OUT     CHBCT
 0051 004C 3E05                MVI     A,5H           ;PTR 5B
 0052 004E D3F3                OUT     CHBCT
```

Figure 2.1 µPD7201A register setting example 1

```
   UCOM-85 ASSEMBLE LIST     (                                        ) PAGE 0002

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0053 0050 3E28                 MVI     A,CR5B             ;TX PARAMETERS
  0054 0052 D3F3                 OUT     CHBCT
  0055                 :
  0056                 ;******************
  0057                 :* PARAMETER AREA *
  0058                 ;******************
  0059                 :
  0060                 : SYSTEM I/O ADDRESS
  0061                 ;
  0062 00F0            CHADT    EQU     0F0H                ;MPSC CH-A DATA
  0063 00F1            CHACT    EQU     0F1H                ;MPSC CH-A CONTROL
  0064 00F2            CHBDT    EQU     0F2H                ;MPSC CH-B DATA
  0065 00F3            CHBCT    EQU     0F3H                ;MPSC CH-B CONTROL
  0066                 :
  0067                 : MPSC CONTROL REGISTER PARAMETERS
  0068                 ;
  0069 0013            CR1A     EQU     13H
  0070 001F            CR1B     EQU     1FH
  0071 0000            CR2A     EQU     0
  0072 0000            CR2B     EQU     0
  0073 0041            CR3A     EQU     41H
  0074 0041            CR3B     EQU     41H
  0075 004F            CR4A     EQU     4FH
  0076 004F            CR4B     EQU     4FH
  0077 0028            CR5A     EQU     28H
  0078 0028            CR5B     EQU     28H
  0079                 ;
  0080 0000                     END


 TOTAL ERROR =   00


  SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS


  CHACT  00F1    CHADT  00F0    CHBCT  00F3    CHBDT  00F2    CR1A   0013
  CR1B   001F    CR2A   0000    CR2B   0000    CR3A   0041    CR3B   0041
  CR4A   004F    CR4B   004F    CR5A   0028    CR5B   0028    STCHA  0014
  STCHB  0034
```

Figure 2.1 µPD7201A register setting example 2

CHAPTER 3 µPD7201A SYSTEM OPERATION

Chapter 3 deals with the features of the send and receive data transfer method (polling, interrupt, DMA), then explains the µPD7201A interrupt and DMA operation control method in detail. In addition the interrupt and DMA functions are explained to facilitate understanding of the µPD7201A operation.

3.1 Send and Receive Data Transfer Method

The µPD7201A supports three types of send and receive data transfer methods.* This section explains the differences and method of selection.

* The transfer method mentioned here applies to a write or read of send and receive data between the CPU and the µPD7201A or between system memory and the µPD7201A.

3.1.1 Transfer under program control

When data is transferred to and from peripheral devices under program control, normally the polling method or interrupt method is used.

- Polling method: The CPU continuously checks peripheral device status, and when ready, data is transferred.
- Interrupt method: The peripheral device issues an interrupt request to the CPU (or interrupt controller). Data is transferred under control of the interrupt processing program.

(1) Data transfer by polling

The CPU checks the µPD7201A status for send or receive data transfer timing.

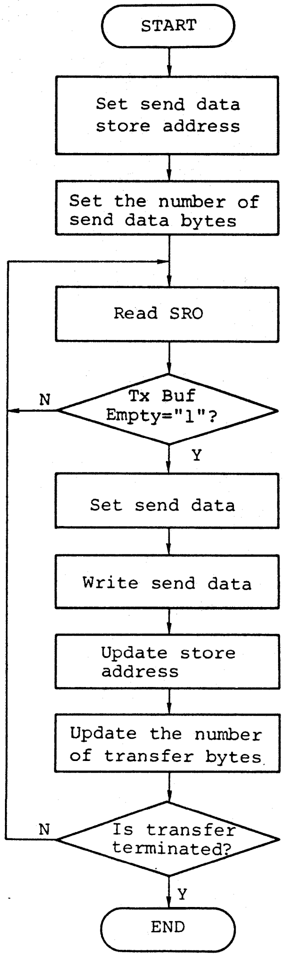
Send data write (CPU ⟶ µPD7201A) occurs when the Tx Buffer Empty bit (SR0 bit D2) is set to 1.

Receive data read (CPU ⟵ µPD7201A) occurs when the Rx Character Available bit (SR0 bit D0) is set to 1.

When data is received, the error status can be checked by the SR1 error bit.

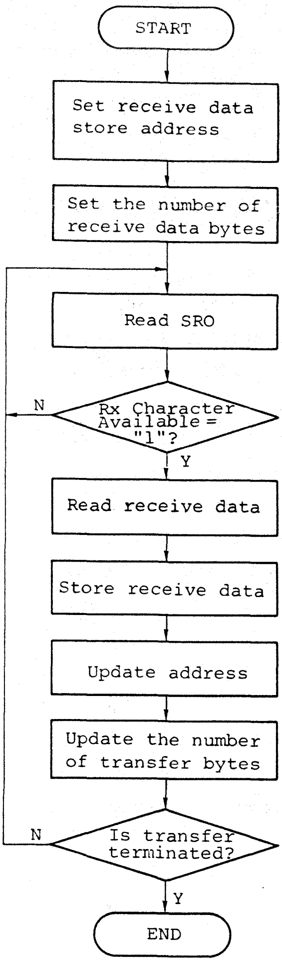
Normally, these operations are repeated until all send and receive data transfers are terminated.

Figures 3.1 and 3.2 show examples of data transfer programs using polling.



```
        LXI H, TXBUF
        MVI B, COUNT1
LOOP1:  IN  SR0
        ANI 4
        JZ  LOOP1
        MOV A, M
        OUT CHADT
        INX H
        DCR B
        JNZ LOOP1
             .
             .
             .
             .
```

Figure 3.1 Example of send data transfer program using polling



```
           LXI H, RXBUF
           MVI B, COUNT2
LOOP 2:    IN  SR0
           PRC
           JNC LOOP2
           IN  CHADT
           MOV M, A
           INX H
           DCR B
           JNZ LOOP2
                  .
                  .
                  .
                  .
```

Figure 3.2 Example of send data transfer program using polling

(2) Data transfer by using interrupt

When Tx Buffer Empty or Rx Character Available state occurs, the $\overline{\text{INT}}$ signal goes LOW (active) and an interrupt request is sent to the CPU or interrupt controller. This μPD7201A function can be used to determine the send data write or receive data read timing for the μPD7201A.

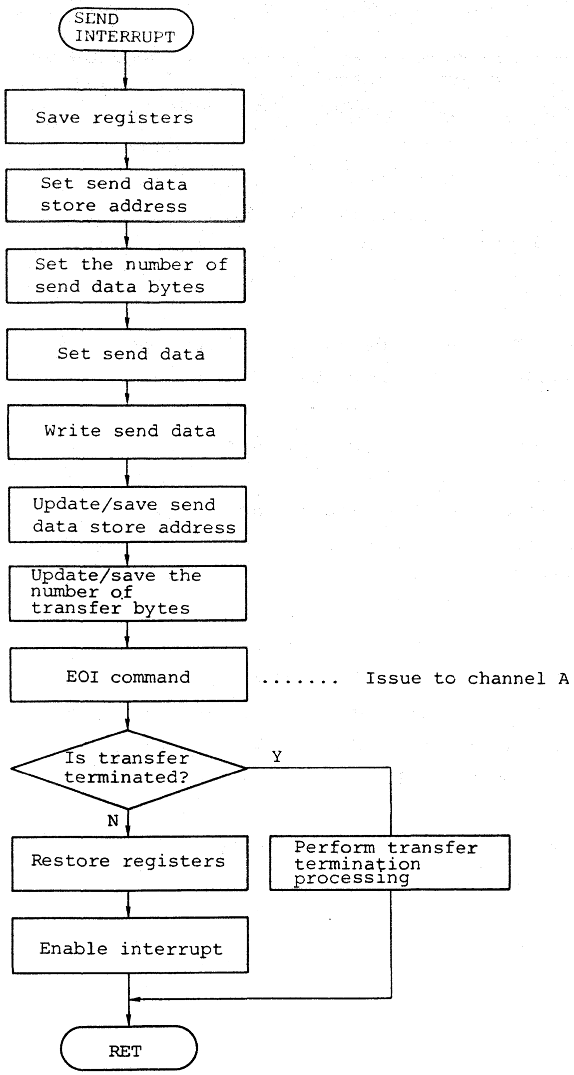
Normally, the CPU executes the main program. When an interrupt request occurs, the CPU temporarily stops main program processing and executes the interrupt processing program. The interrupt processing program first saves the current state of each register in the stack area, then performs processing on the peripheral device which requested the interrupt. When processing is terminated, the state of each register saved at the beginning of processing is restored and another interrupt is ready to be accepted, then main program processing is restarted.

Since this method eliminates the need for the CPU to always monitor the μPD7201A status and the CPU needs only to be involved when an interrupt request occurs, CPU overhead is minimized for more programming efficiency.

Figure 3.3 shows an example of a program using an interrupt to send data.

Since the μPD7201A has two interrupt modes, vector mode and nonvector mode, interrupt processing can be performed by the method most suitable for the CPU being used.

The μPD7201A interrupt mode and interrupt operation are explained in 3.2.



```
        PUSH  PSW
        PUSH  H
        PUSH  B
        LHLD  TXPTR
        LDA   TXCNT
        MOV   B, A
        MOV   A, M
        OUT   DATA
        INX   H
        SHLD  TXPTR
        DCR   B
        MOV   A, B
        STA   TXCNT
        MVI   A, 38H
        OUT   CTRLA
        JZ    TXTERM
        POP   B
        POP   H
        POP   PSW
        EI
        RET
TXTERM: Send transfer
        termination
        processing
```

Figure 3.3 Example of send data transfer program using interrupts

3.1.2 Transfer using DMA

In a transfer under program control (transfer via interrupt), the time during which the CPU reads instructions and checks pripheral device status is required in addition to the time required for actual data transfer between memory and the peripheral device. Thus, a fairly long time is generally required to transfer one data word.

Thus, if only data transfers between memory and a peripheral device occur, the transfer rate can be dramatically increased over program-controlled transfers. Such direct data transfer between memory and peripheral devices without CPU intervention is called direct memory access (DMA). It is used for high-speed data transfer. Normally, a DMA operation is executed by a DMA controller (such as the µPD8237A-5).

When the µPD7201A is placed in Tx Buffer Empty or Rx Character Available states, the DMA request signal goes HIGH (active) and a DMA transfer request is sent to the DMA controller. There are four DMA request signals: DRQRxA, DRQTxA, DRQRxB, and DRQTxB.

The µPD7201A DMA operation is explained in 3.3.

3.2 µPD7201A Interrupt Operation

3.2.1 µPD7201A interrupt mode

The µPD7201A contains two types of interrupt modes (vector mode and noninterrupt mode) according to the difference in how interrupt vectors are received. Whether the vector contents are automatically changed according to the interrupt source or are fixed can be specified by using CR1B bit D2.

In vector mode, when an interrupt source occurs, the µPD7201A asserts the $\overline{\text{INT}}$ line (LOW) and outputs an interrupt vector in synchronization with the $\overline{\text{INTAK}}$ signal output from the CPU (called an $\overline{\text{INTAK}}$ sequence). In the $\overline{\text{INTAK}}$ sequence, the $\overline{\text{CS}}$ line must be held HIGH. Figure 3.4 shows the interrupt timing in vector mode.

Since an interrupt vector is automatically generated in vector mode, the required processing program should be stored at the jump destination address determined by the µPD7201A for efficient programming.

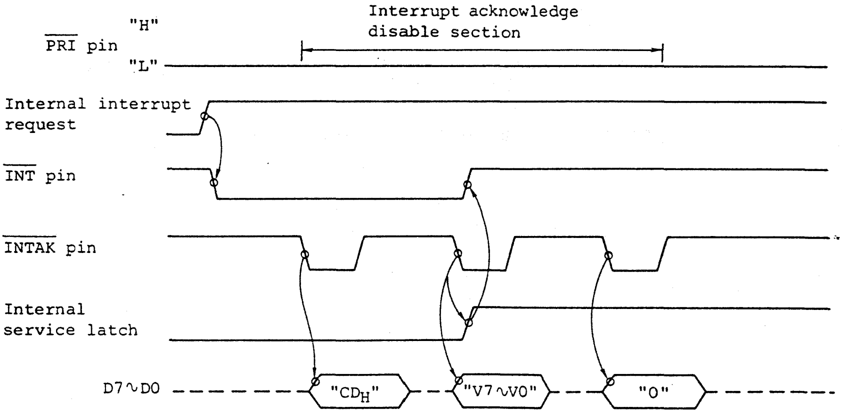


Figure 3.4 Interrupt timing in vector mode (85-1 mode)

In nonvector mode, when an interrupt source occurs, the µPD7201A asserts the $\overline{\text{INT}}$ line (LOW). When the CPU accepts the interrupt request, it reads the µPD7201A SR2B and determines the interrupt source. (If necessary, it also reads SR0 and SR1 to determine the interrupt source.) In this mode, the $\overline{\text{INTAK}}$ line is not used, but must be pulled up.

Figure 3.5 shows the interrupt timing in nonvector mode.

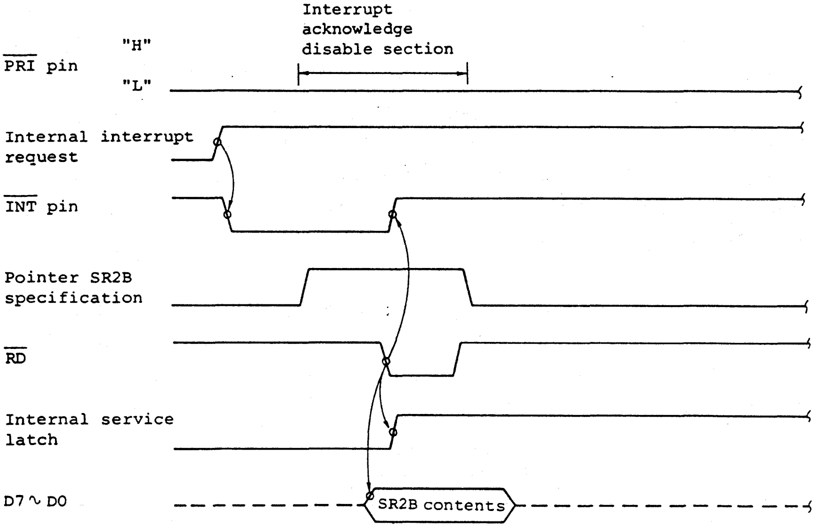


Figure 3.5 Interrupt timing in nonvector mode

Figure 3.6 shows an example of an interrupt processing sequence in nonvector mode.

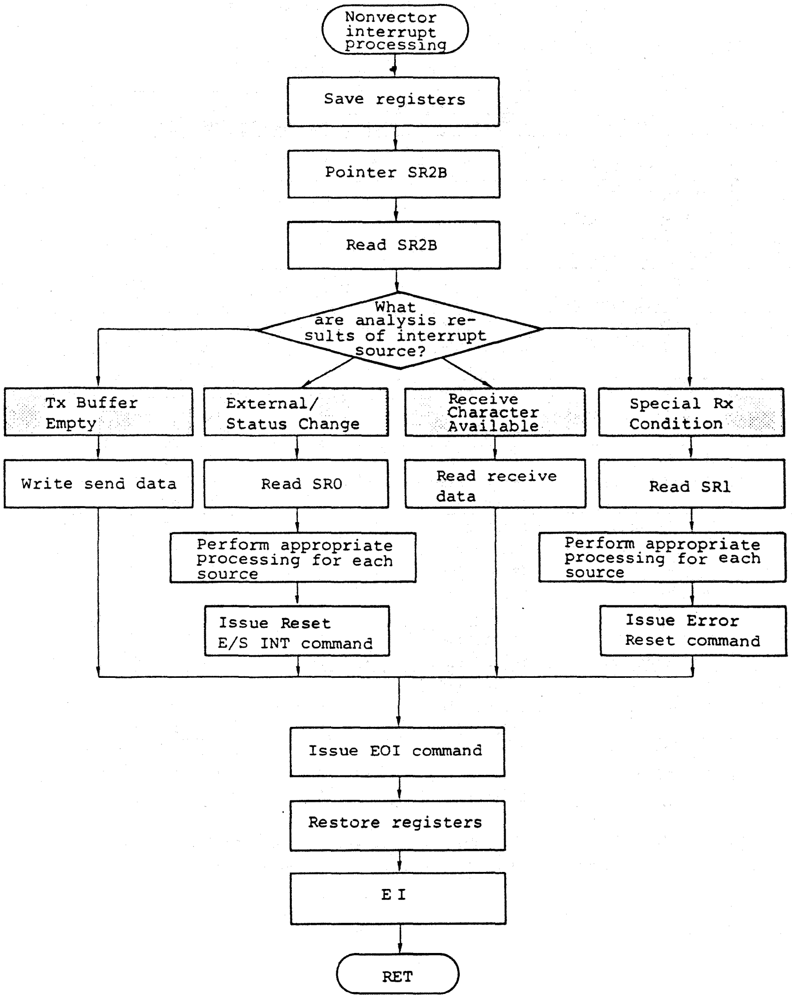


Figure 3.6 Interrupt processing in nonvector mode

Table 3.2 lists the relationship between the control registers related to each µPD7201A interrupt mode and output vectors.

Table 3.2 µPD7201A interrupt mode

CR2A			CR1B,D2	Interrupt mode	Vector read method									
D5	D4	D3			$\overline{\text{INTA}}$	SR2B	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0 1	0	Non Vectored 85 (Fixed Vector)	-	0	V7	V6	V5	V4	V3	V2	V1	V0
0	0	0 1	1	Non Vectored 85 (Modified Vector)	-	0	V7	V6	V5	$\textcircled{V4}$	$\textcircled{V3}$	$\textcircled{V2}$	V1	V0
0	1	0	0	Non Vectored 86 (Fixed Vector)	-	0	V7	V6	V5	V4	V3	V2	V1	V0
0	1	0	1	Non Vectored 86 (Modified Vector)	-	0	V7	V6	V5	V4	V3	$\textcircled{V2}$	$\textcircled{V1}$	$\textcircled{V0}$
1	0	0	0	85-1 Vectored (Fixed Vector)	First time	-	1	1	0	0	1	1	0	1
					Second time	-	V7	V6	V5	V4	V3	V2	V1	V0
					Third time	-	0	0	0	0	0	0	0	0
1	0	0	1	85-1 Vectored (Modified Vector)	First time	-	1	1	0	0	1	1	0	1
					Second time	-	V7	V6	V5	$\textcircled{V4}$	$\textcircled{V3}$	$\textcircled{V2}$	V1	V0
					Third time	-	0	0	0	0	0	0	0	0
1	0	1	0	85-2 Vectored (Fixed Vector)	First time	-	Hi - Z							
					Second time	-	V7	V6	V5	V4	V3	V2	V1	V0
					Third time	-	0	0	0	0	0	0	0	0
1	0	1	1	85-2 Vectored (Modified Vector)	First time	-	Hi - Z							
					Second time	-	V7	V6	V5	$\textcircled{V4}$	$\textcircled{V3}$	$\textcircled{V2}$	V1	V0
					Third time	-	0	0	0	0	0	0	0	0
1	1	0	0	86 Vectored (Fixed Vector)	First time	-	Hi - Z							
					Second time	-	V7	V6	V5	V4	V3	V2	V1	V0
1	1	0	1	86 Vectored (Modified Vector)	First time	-	Hi - Z							
					Second time	-	V7	V6	V5	V4	V3	$\textcircled{V2}$	$\textcircled{V1}$	$\textcircled{V0}$
1	1	1	0	85-3 Vectored (Fixed Vector)	First time	-	Hi - Z							
					Second time	-	V7	V6	V5	V4	V3	V2	V1	V0
					Third time	-	0	0	0	0	0	0	0	0
1	1	1	1	85-3 Vectored (Modified Vector)	First time	-	Hi - Z							
					Second time	-	V7	V6	V5	$\textcircled{V4}$	$\textcircled{V3}$	$\textcircled{V2}$	V1	V0
					Third time	-	0	0	0	0	0	0	0	0

Note) Such indication as $\textcircled{V4}$, $\textcircled{V3}$, or $\textcircled{V2}$ means that the value is automatically changed according to the interrupt source. In other vectors, the value set in CR2B is output as is.

-: Not used. ○: Used.

3.2.2 µPD7201A interrupt sources

The µPD7201A can detect various interrupt sources so that the CPU can perform appropriate processing. The interrupt sources are roughly classified into the following three types:

a) Transmit interrupt (Tx interrupt)
b) Receive interrupt (Rx interrupt)
c) External/Status interrupt (E/S interrupt)

This section explains each of these interrupt sources, the causes of each, and the interrupt service processing required.

(1) Transmit interrupt

A Tx interrupt occurs when Tx Buffer Empty state (SR0 bit D2 = 1) is set during INT/DMA Enable (CR1 bit D1 = 1). This interrupt indicates that the µPD7201A transmit buffer is empty and the CPU may write send data. In Synchronous or HDLC mode, the Tx interrupt also occurs when CRC character sending is terminated. Note that in the HDLC mode, the Tx interrupt occurrence condition in the TxLR Set mode (CR1 bit D6 = 1) differs from that in other cases.

The Tx interrupt source is cleared when send data is written or a Reset Tx INT/DMA Pending command is executed. However, the Reset Tx INT/DMA Pending command must be issued in the Tx Buffer Empty mode.

When DMA transfer is selected, a Tx interrupt of the channel is disabled internally, thus send data cannot be transferred via a DMA and an interrupt at the same time.

See Figure 3.3 for the send data transfer method using an interrupt.

(2) Receive interrupt

During Rx INT Enable, when received data is transferred to the receive buffer and the Rx Character Available mode is in effect, an Rx interrupt occurs. The mode to enable a receive interrupt is INT On First Character (First Rx INT) or INT on All Character. It is selected by using CR1 bits D4 and D3. First Rx INT in the INT on First Character mode can be masked by setting the Rx INT Mask bit (CR2A bit D6) to 1.

Special receive interrupt is Special Rx Condition interrupt. The Special Rx Condition interrupt occurs when receive data is erroneous (Parity, Overrun, or Framing Error). In HDLC mode, an interrupt also occurs at the end of a frame (End of Frame). The Special Rx Condition interrupt has the same priority level as an Rx interrupt.

Both Special Rx Condition and Rx interrupts do not occur at the same time for the same receive data. When one receive data word causes Special Rx Condition, only Special Rx Condition interrupt occurs, thus the receive data must be read within the processing routine. In the Special Rx Condition interrupt routine, when End-of-Frame occurs in the HDLC mode, the second byte of the CRC character is read.

The Special Rx Condition status is held in SR1, and can be checked by the CPU.

When the Special Rx Condition occurs, the Error Reset command must always be issued.

Table 3.3 lists the receive interrupt mode.

Table 3.4 lists the Special Rx Condition interrupt operation.

Table 3.3 Receive interrupt mode and operation

CR1		Receive interrupt mode	Function
D4	D3		
0	0	INT/DMA Disable	Even if data is received, receive interrupt/DMA request does not occur.
0	1	INT on First Character	A receive interrupt occurs only for the first received data after this mode is set. (It can be masked by using the Rx INT Mask bit.) After an interrupt occurs, Rx interrupt can be enabled again in the mode by using Enable Int on Next Rx Character command (except during Rx INT Mask mode). The mode is selected when receive data is transferred via DMA. In this mode, a parity error does not cause a Special Rx Condition interrupt.
1	0	INT on All Character (parity error is used as Special Rx Condition source)	A receive interrupt occurs each time data is transferred to the receive buffer. A parity error causes a Special Rx Condition interrupt.
1	1	INT on All Character (parity error is not used as Special Rx Condition source	This is the same as the mode when CR1 bits D4 and D3 = 1 and 0 except that a parity error does not cause a Special Rx Condition interrupt.

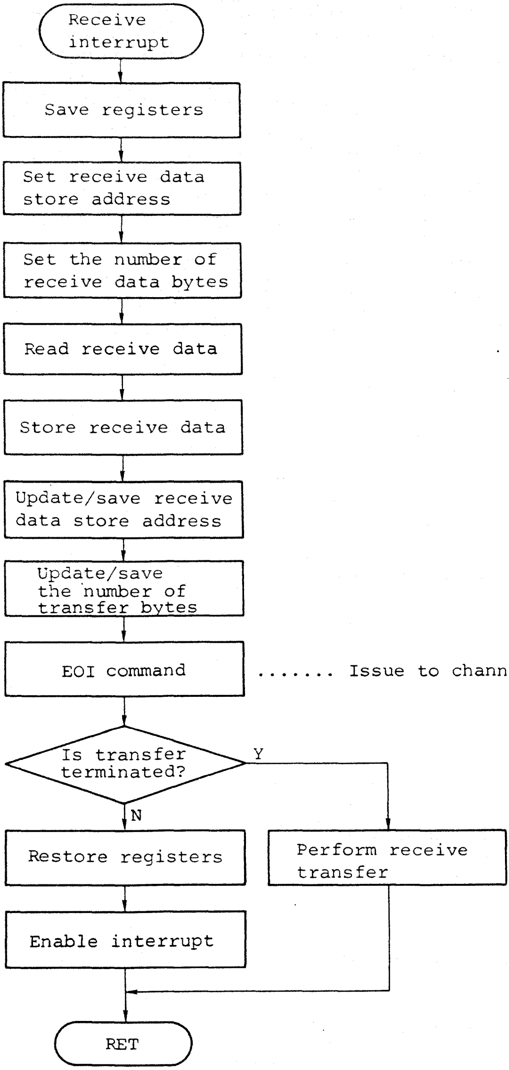
Table 3.4 Special Rx Condition interrupt operation

Source	Operation mode	Operation
Parity Error (SR1: D4)	Asynchronous Synchronous	When a Parity Error occurs. This bit is latched. It is reset when the Error Reset command is issued.
Overrun Error (SR1: D5)	Asynchronous Synchronous HDLC	When an Overrun Error occurs. This bit is latched. It is reset when Error Reset command is issued.
Framing Error (SR1: D6)	Asynchronous	When a Framing Error occurs. This bit is not latched. It is reset when next correct data is received, and also reset when an Error Reset command is issued.
End of Frame (SR1: D7)	HDLC	When valid end flag is received. This bit is latched. It is reset when an Error Reset command is issued.

Note (1) When an error status check is made, the status must be first read before receive data is read.

(2) If the First Rx INT mode (CR1 bits D4 and D3 = 0, 1) is selected, when a Special Rx Condition interrut occurs, the data subsequently input is not transferred to the last stage of the receive buffer until the Error Reset command is issued. (Up to two characters of data received after the Special Rx Condition interrupt occurs are stored in the first and second stages of the receive buffer.)

Figure 3.7 shows an example of a data transfer program using an Rx interrupt. Figure 3.8 shows an example of Special Rx Condition interrupt processing.



.... Issue to channel A

```
PUSH PSW
PUSH H
PUSH B
LHLD RXPTR
LDA  RXCNT
MOV  B, A
IN   DATA
MOV  M, A
INX  H
SHLD RXPTR
DCR  B
MOV  A, B
STA  RXCNT
MVI  A, 38H
OUT  CTRLA
JZ   RXTERM Note)
POP  B
POP  H
POP  PSW
EI
RET
```

Note)

RXTERM: Receive transfer termination processing

Figure 3.7 Example of receive data transfer program using interrupts

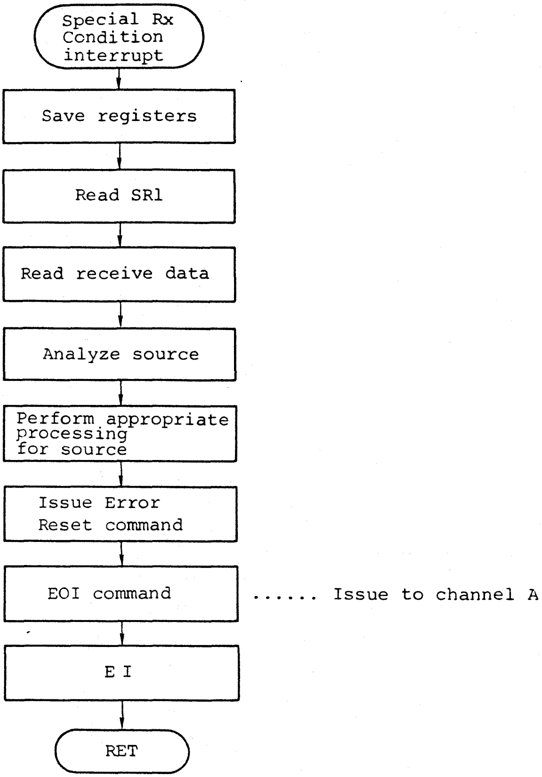


Figure 3.8 Example of Special Rx Condition interrupt processing

(3) External/Status interrupt latch operation

The External/Status (E/S) interrupt is used to detect the µPD7201A input status or change in internal state. The E/S interrupt occurs under the conditions listed in Table 3.5 if E/S INT Enable is set (CR1 bit D0 = 1).

Table 3.5 E/S interrupt occurrence conditions

Source (E/S bit)	Operation mode condition			E/S interrupt occurrence condition Note)
	Asynchronous	Synchronous	HDLC	
$\overline{\text{CTS}}$	O	O	O	0 → 1, 1 → 0
$\overline{\text{DCD}}$	O	O	O	0 → 1, 1 → 0
SYNC/HUNT	O	O	O	0 → 1, 1 → 0
Break/Abort	O	-	O	0 → 1, 1 → 0
Tx Underrun/EOM	-	O	O	0 → 1
All Sent	-	-	O	0 → 1

Note) Refer to the User's Manual for occurrence of each stage change.

The points to use the E/S interrupt are explained below:

When the state of any E/S bit changes if E/S INT Enable is set (the occurrence condition listed in Table 3.5 is satisfied and the preceding E/S interrupt processing must be terminated), the state of the other E/S bits is also latched, the values are set in SR0, and and E/S interrupt occurs. Normally, the CPU determines of the source E/S interrupt by comparing the new SR0 contents with the SR0 contents before latching the E/S interrupt handling routine, and executes interrupt service processing accordingly. Note that once an E/S bit is latched, the µPD7201A does not execute a new latch operation even if the E/S bit state changes afterwards unless the Reset E/S INT command is issued. (No E/S interrupt occurs.) Figure 3.9 shows an example operation when the $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ lines change state.

First, when the $\overline{\text{CTS}}$ line goes LOW, the change of state is latched in SR0 bit D5 and an E/S interrupt occurs. The CPU accepts the interrupt and begins interrupt processing (① in Figure 3.9). Even if $\overline{\text{DCD}}$ goes LOW while the E/S interrupt is being processed, the Reset E/S INT command is not issued for a CTS bit latch operation at the time; therefore, the $\overline{\text{DCD}}$ change of state is ignored (② in Figure 3.9). When the CTS state change processing continues and the Reset E/S INT command is issued (③ in Figure 3.9), the E/S bit latch operation is enabled again. This means that the $\overline{\text{DCD}}$ change of state is first latched in the DCD bit at the time of ④ in Figure 3.9.

The operation mentioned above also occurs when E/S INT Disable is set. Thus, it becomes difficult to recognize E/S bit change of-state timing if the E/S interrupt is not used. In such a case, the most recent E/S bit state can be known by issuing the Reset E/S INT command to the µPD7201A, then reading SR0.

Figure 3.10 shows an example of E/S interrupt processing.

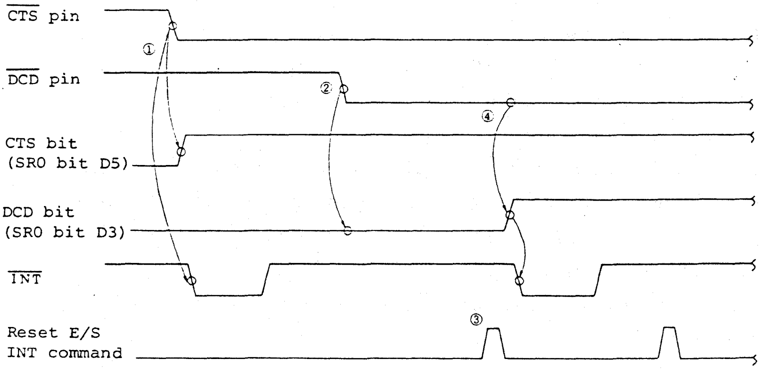


Figure 3.9 E/S bit latch operation

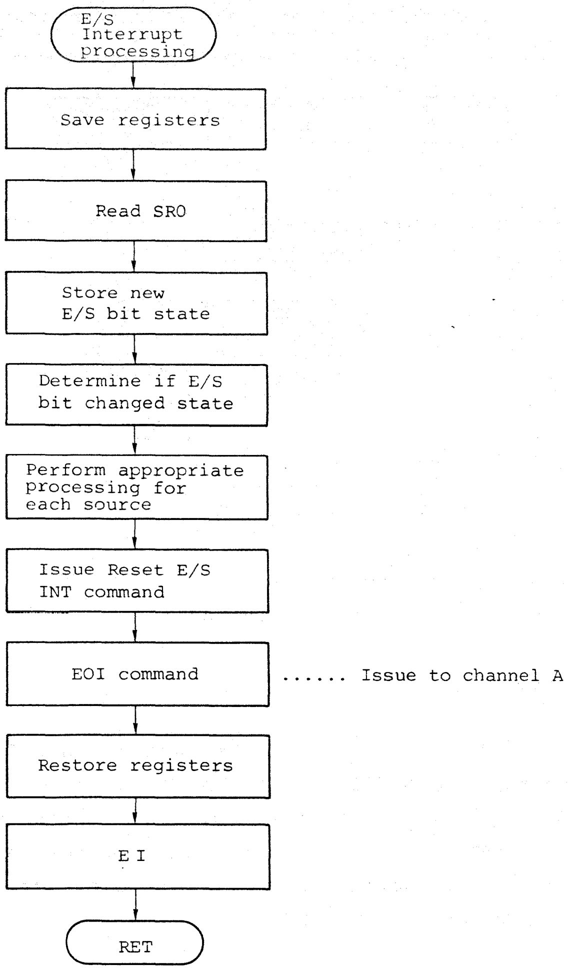


Figure 3.10 E/S interrupt processing example

Figure 3.11 summarizes the µPD7201A interrupt configuration.

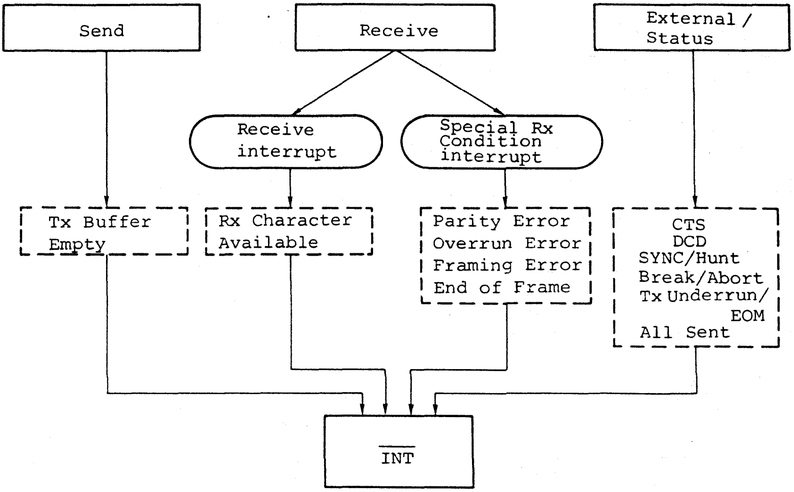


Figure 3.11 µPD7201A interrupt configuration

3.2.3 Precautions on use of interrupts

(1) External circuit must be provided to use the µPD7201A as an interrupt controller µPD8259A-2 slave in the 85-2 or 86 vector mode. If the $\overline{\text{PRI}}$ line is not LOW, the µPD7201A $\overline{\text{INT}}$ signal does not become active even if an interrupt condition occurs. Therefore, the $\overline{\text{PRI}}$ line must normally be held LOW.

However, when the µPD8259A-2 accepts any interrupt other than µPD7201A interrupts, $\overline{\text{PRI}}$ must be driven HIGH in the $\overline{\text{INTAK}}$ sequence to inhibit interrupt vector output from the µPD7201A.

Therefore, an external control circuit must be provided to perform operation control as mentioned above by using the µPD8259A-2 CAS signal.

To perform such control, although no standards are set, it is necessary to use the setup time (minimum of 280 ns) to change the $\overline{\text{PRI}}$ signal state from LOW to HIGH at the falling edge of the second $\overline{\text{INTAK}}$ signal, as shown in Figure 3.12 (a).

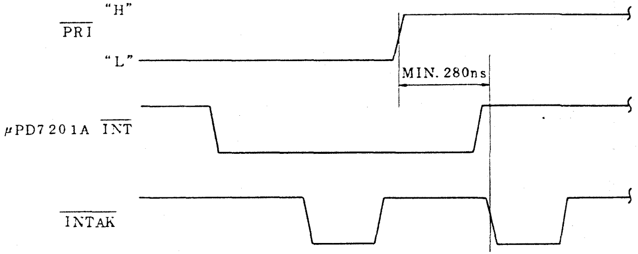


Figure 3.12 (a) Relation between $\overline{\text{PRI}}$ and $\overline{\text{INTAK}}$

Figure 3.12 (b) shows an example of an external circuit for reference when the method as mentioned above is used (in the 86 vector mode).

If the 85-3 vector mode is used, the $\overline{\text{INT}}$ signal goes active when an interrupt source occurs independently of the $\overline{\text{PRI}}$ line state, and the aforementioned external circuit is not needed. The $\overline{\text{PRI}}$ line can be controlled by directly decoding the µPD8259A-2 CAS signal. However, in this mode, a daisy chain mode system using $\overline{\text{PRI}}$ and $\overline{\text{PRO}}$ cannot be implemented.

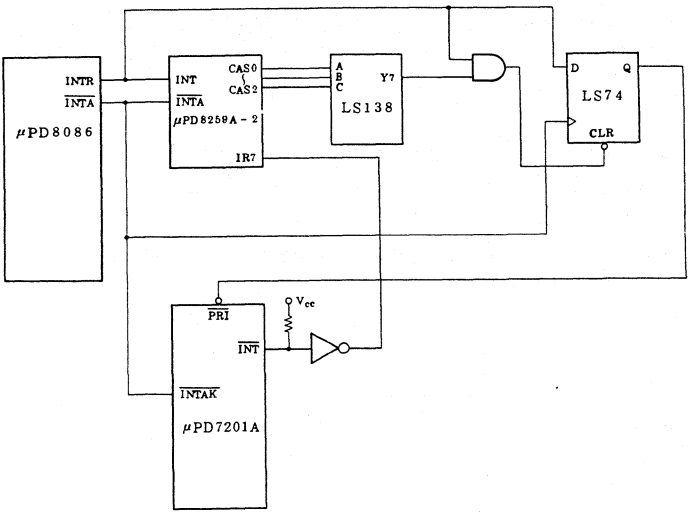


Figure 3.12 (b) µPD7201A and µPD8259A-2 connection example (reference circuit)

(2) To perform interrupt control by using µPD7201A $\overline{\text{PRI}}$ input, obey the following caution:

In the system configuration in which an interrupt request issued from another device is accepted during µPD7201A interrupt processing, issue two EOI commands in the sequence shown in Figure 3.12 (c). In this case, interrupt must be disabled in the host when the EOI command is being issued.

Such processing is not required if the system configuration enables multiinterrupts.

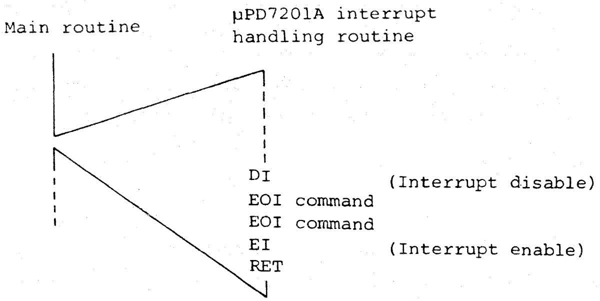


Figure 3.12 (c) Multiinterrupt processing sequence

3.3 µPD7201A DMA Operation

This section explains µPD7201A DMA operation (send DMA and receive DMA) using HDLC operation as an example.

3.3.1 Send DMA

When send data is transferred via DMA, CR2A bits D1 and D0 = 0, 1, 1, 0, or 1, 1 must be selected and CR1 bit D1 must be set to 1. A send interrupt on the channel where DMA transfer is selected will not occur.

Send DMA operation is explained by using HDLC mode as an example. The DMA operation in the HDLC mode varies depending on the value of TxLR (CR1 bit D6), as explained below:

(1) When TxLR Set = 1

This mode is a new function added to the µPD7201A. Figure 3.13 shows the timing chart of send DMA operation when the function is used (only minimum required operation is shown).

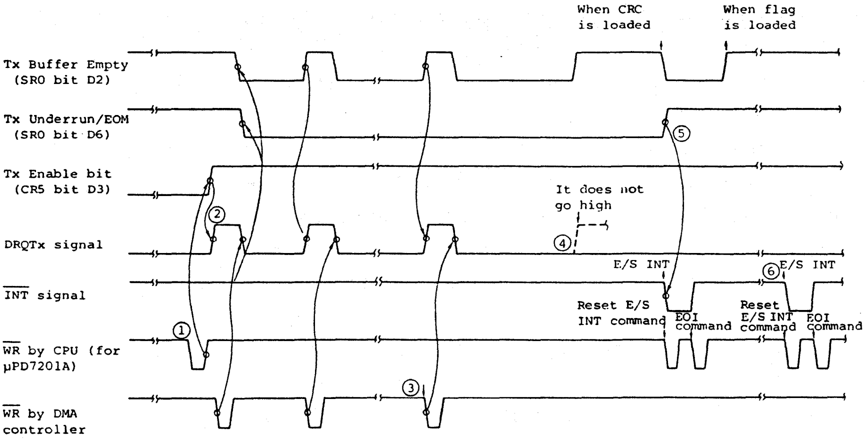
In the explanation to follow, Figure 3.13 is used.

In this made, when Tx Enable is set to 1 after initialization ends (assuming that Auto Enable mode is not entered), the DRQTx signal goes HIGH and the DMA controller is requested to transfer send data (② in Figure 3.13). As requested, the DMA controller transfers send data. After this, the same operation is repeated. When the DMA controller transfers the last data and stops operation (③ in Figure 3.13), then send data is not transferred. Thus, after the last data is sent, the µPD7201A is placed in Tx

Underrun state and an E/S interrupt occurs (⑤ in Figure 3.13). The data transferred by the DMA controller is counted each time in the µPD7201A (the result is held in TxLC). When the count reaches the value set in TxLR, then the DRQTx signal does not go active even if Tx Buffer Empty is set (④ in Figure 3.13). This state is continued until TxLR is again set. In addition, a comparison between the TxLR and TxLC values is made in the Tx Underrun state. If they match, CRC is automatically sent; if they do not match, an abort is automatically sent. Therefore, when the Tx Underrun state occurs, software must be used to read the TxLC value (SR3, SR4) and decide whether or not the normal frame end if reached.

When the µPD7201A ends sending at least one end flag, All Sent is set to 1 and an E/S interrupt occurs (⑥ in Figure 3.13). Normally, after the interrupt occurs, the next processing step (such as successive frame sending or send stopping) is executed.

Figure 3.14 shows an example of control when one frame of send data is transferred in the mode.



① Tx Enable is set to 1

② First DMA request by setting Tx Enable to 1

③ Transfer of last data

④ It does not go HIGH because TxLR = TxLC

⑤ E/S INT by setting Tx Underrun/EOM to 1

⑥ E/S INT by setting All Sent to 1

Figure 3.13 μPD7201A send DMA operation (HDLC mode, TxLR Set = 1)

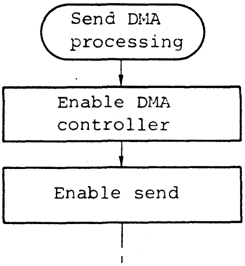
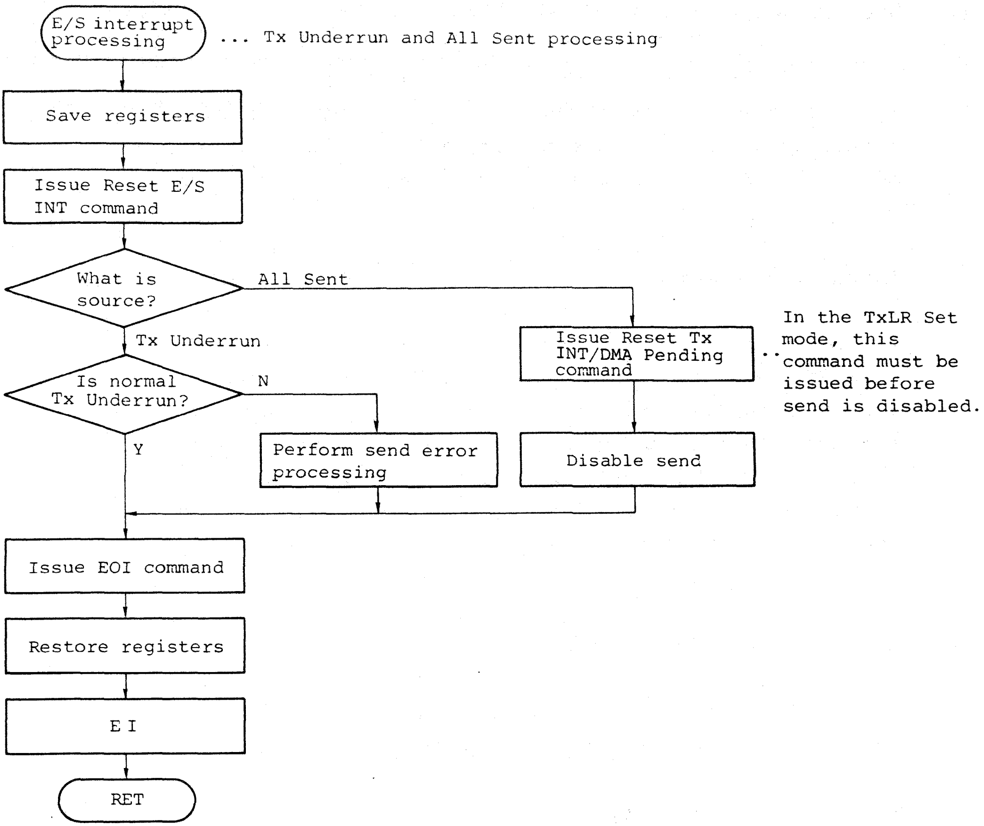


Figure 3.14 (a) Control example of send operation by DMA (main program)



Note) Interrupt processing caused by other E/S sources is omitted.

Figure 3.14 (b) Control example of send operation by DMA (part of E/S interrupt processing)

(2) When TxLR Set = 0

In this mode, similar operation to µPD7201A operation is performed except that All Sent E/S interrupt occurs after the frame end. Figure 3.15 shows timing chart in this mode (only minimum required operation is shown).

In the explanation to follow, Figure 3.15 is used.

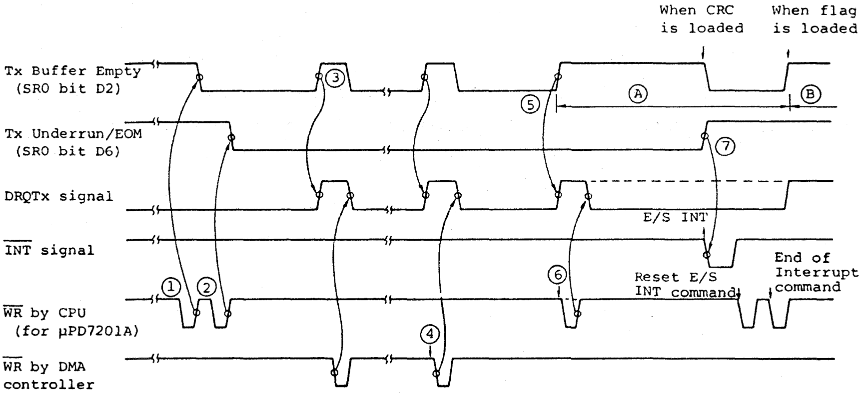
In the mode, the first send data must be written by the CPU (① in Figure 3.15). After this, when Tx Buffer Empty is set, the DRQTx signal first goes HIGH and the DMA controller is requested to transfer send data (③ in Figure 3.15). Assuming that the total number of send data words in N, the number of data words transferred by DMA equals N - 1.

When the DMA controller transfers the last data to the µPD7201A and stops operation (④ in Figure 3.15), then send data is not transferred. Thus, after the last data is sent, Tx Underrun state is set and an E/S interrupt occurs (⑦ in Figure 3.15).

When Tx Buffer Empty is set after the last data is written, the DRQTx signal goes HIGH (⑤ in Figure 3.15). The signal can be reset by issuing Reset Tx INT/DMA Pending command before CRC character is loaded internally (during Tx Buffer Empty) (⑥ in Figure 3.15). If the Reset Tx INT/DMA Pending command is not issued, the DRQTx signal remains HIGH (dotted-line portion in Figure 3.15), but does not affect the send operation.

Note that if send data is transferred to the µPD7201A in the interval of Ⓐ , frame end is not normally made.

To send continuous frames, if the next send data is transferred after the Ⓑ state is set, the next frame can be sent successively without destroying the preceding frame. However, since the µPD7201A generates an E/S interrupt when All Sent is set to 1, the extra CPU load can be eliminated if the interrupt is waited before necessary processing is performed. In this case, even if send stop (Tx Disable) is set, the DRQTx signal is not reset, but if Tx INT/DMA Disable (CR1 bit D1 = 0) or the Reset Tx INT/DMA Pending command is issued, the signal can be reset.



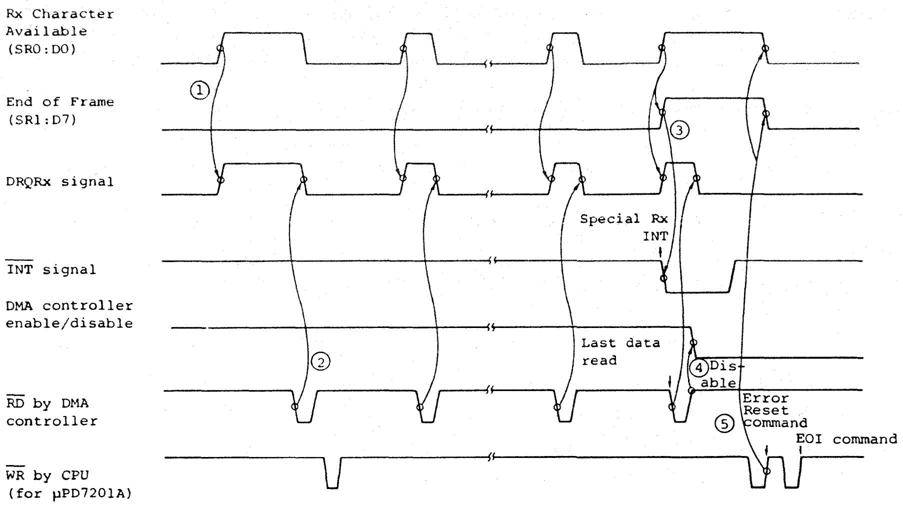
① First send data write
② Reset Tx Underrun/EOM bit
③ First DRQTx signal by setting Tx Buffer Empty to 1
④ Last data transfer
⑤ Tx Buffer Empty set after last data is written
⑥ Reset Tx INT/DMA Pending
⑦ E/S INT by setting Tx Underrun/EOM to 1

Figure 3.15 µPD7201A send DMA operation
(HDLC mode, TxLR = 0)

3.3.2 Receive DMA

To transfer receive data by using DMA, CR2A bits D1 and D2 is set to 0, 1, 1, 0, or 1, 1 and CR1 bits D4 and D3 is set to 0, 1 (First Rx INT mode). In the First Rx INT mode, an Rx interrupt caused by the first receive data can be masked by setting CR2A bits D6 to 1 (Rx INT Mask mode). This mode is used when an interrupt need not be caused by the first receive data (normal address field) or when the baud rate is fast and it is not timely if the DMA controller is enabled by using the First Rx INT handling routine. (Use of the mode is recommended to initiate a receive DMAS.)

Figure 3.16 shows the timing chart of receive DAM operation in the First Rx INT Mask mode, using the HDLC mode as an example (only minimum required operation is shown).



① DMA request by setting Rx Character Available to 1

② Receive data transfer

③ EOF is set to 1 when end flag is detected

④ DMA controller operation stops when the last data is transferred

⑤ EOF state is cleared

Figure 3.16 µPD7201A receive DMA operation (HLDC, First Rx INT Mask mode)

In the explanation to follow, Figure 3.16 is used.

The µPD7201A receives data after the flag is detected. When Rx Character Available state is set, the DRQRx signal is driven HIGH and a request is made to the DMA controller to transfer receive data (① in Figure 3.16). As requested, the DMA controller transfers receive data (② in Figure 3.16). The µPD7201A generates DMA requests for all receive data containing a CRC.

When the µPD7201A detects the end flag, End of Frame is set and Special Rx Condition interrupt occurs (③ in Figure 3.16). This interrupt causes the CPU to detect the receive End of Frame. When the DMA controller transfers the last data (the second byte of the CRC character), operation stops (④ in Figure 3.16). After the End of Frame state occurs, the Error Reset command must be issued to the µPD7201A to clear the state (⑤ in Figure 3.16). Note the following: If the Error Reset command is not issued, only two bytes of data of the next frame successively input in the Int on First Rx Character mode are stored in the FIFO, and it is not transferred to the last stage of the buffer. Thus, when continuous frames are received (particularly when the end flag is also used for the start flag), limitations are imposed on CPU processing time. (For example, when the end flag is also used for the start flag, it is desirable to issue the Error Reset command within the time of two bytes after End of Frame occurs).

If frame reception is not made after reception of one frame ends, necessary processing such as inhibition of reception is performed.

Figure 3.17 shows a control example when one frame of receive data is transferred via DMA.

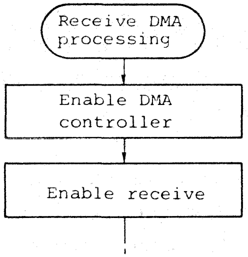


Figure 3.17 (a) Control example of receive operation via DMA (main program)

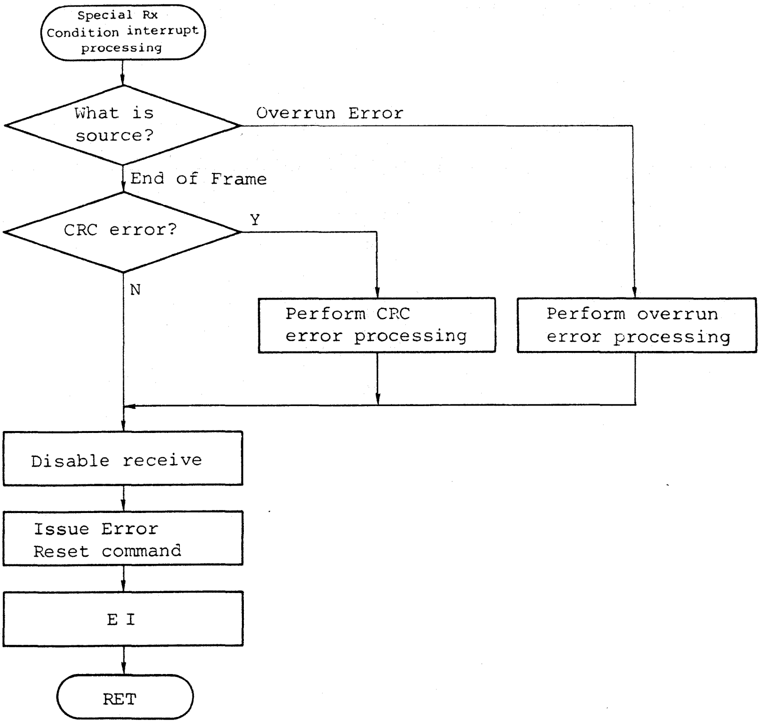


Figure 3.17 (b) Control example of receive operation via DMA (Special Rx Condition interrupt processing)

3.3.3 Caution on use of DMA

There are three µPD7201A DMA modes among which appropriate one can be selected according to the system configuration. The modes differ slightly in use of TxLR Set mode and $\overline{\text{HAKI}}$ pin, as listed in Table 3.6.

Basically, only the combination listed in Table 3.6 can be used. If the µPD7201 is replaced with the µPD7201A in a system using the µPD7201, precautions on one use of µPD7201 must be followed when a change cannot be made to the combination listed in Table 3.6. (For details, refer to the µPD7201 User's Manual or µPD7201 Application Note.)

Table 3.6 Selection of DMA Mode

Condition DMA mode		DMA mode setting	Valid DRQ signal	TxLR Set mode setting	Use of $\overline{\text{HAKI}}$ pin
Channel A DMA mode		CR2A bits D1 and D0 = 0,1	DRQTxA	As desired	Used
			DRQRxA	-	
Both channel DMA mode	Both.Ch. DMA-1	CR2A bits D1 and D0 = 1,0	DRQTxA DRQTxB	1	Used
			DRQRxA DRQRxB	-	
	Both.Ch. DMA-2	CR2A bits D1 and D0 = 1,1	DRQTxA DRQTxB	As desired	Not used
			DRQRxA DRQRxB	-	

CHAPTER 4 HARDWARE

This chapter explains a system using the µPD7201A. The hardware in this example is designed as a means to implement the operation conditions assumed, and serves as a reference for the design of a system using the µPD7201A.

4.1 System Configuration

The µPD7201A is applicable to various serial data communication protocols. Here, assume the following specifications:

Transmission format: Full or half duplex
Applicable protocol: Asynchronous, synchronous or HDLC
Baud rate: 200 Kbps, MAX.

Various methods are designed to implement the specifications. Here, the µPD8085AH, which is a typical eight-bit microprocessor, is used as the CPU. The system configuration is based on this processor.

(1) Memory capacity

16K-byte ROM is used for the system program (monitor program area) and µPD7201A control program area. 8K-byte RAM is used for the data buffer and working area. (The total memory capacity is 24K bytes.)

This memory capacity is set here for convenience. In fact, the memory capacity must be determined in accordance with the system.

(2) Transfer method of send and receive data

Polling, interrupts, and DMA can be used. Here, both interrupts and DMA can be used for both A and B channels.

Generally, the entire system configuration varies depending on what data transfer method is used. It must be determined from an overall viewpoint rather that from only the baud rate.

(3) Interrupt mode

To and similify processing and make the most efficient use of µPD7201A program control, vector processing in Modified Vector mode and nonvector mode processing can be implemented in hardware.

(4) Others

A baud rate generator is installed. I/O port, etc., can also be installed as required. Table 4.1 lists a system configuration example based on the items mentioned above.

Table 4.1 System configuration example

CPU		µPD8085AH (3.072 MHz)	Function
Memory	ROM	µPD2764x2 (16K bytes)	System program µPD7201A control program
	RAM	µPD4016x4 (8K bytes)	Data buffer Working area
DMA controller		µPD8237A-5 (3.072 MHz)	Send and receive data transfer control
Counter/timer		µPD8253-2	Baud rate generator Timer
Serial interface		µPD7201A (4.9152 MHz)	Serial data processing

4.2 Circuit Configuration

Figure 4.1 shows an actual circuit design example based on the configuration listed in Table 4.1 The circuit is configured as follows:

(1) Connection of µPD7201A and µPD8237A-5

This connection assumes that Both CH. DMA-2 mode (CR2A:D1, D0 = 1, 1) is used. Since the µPD8237A-5 DACK signal is used to select a µPD7201A DMA request channel ($\overline{\mathrm{CS}}$, B/$\overline{\mathrm{A}}$, C/$\overline{\mathrm{D}}$ control), the $\overline{\mathrm{HAKI}}$ pin is not used.

The DMA priority in this circuit is DRQRxA(DRQO)>DRQTxA (DRQ1)>DRQRxB(DRQ2)>DRQTxB(DRQ3) in accordance with µPD8237A-5 DRQ input. The priority can be set by using external circuit as desired. Select it according to the system configuration. The µPD8237A-5 DACK signal is used as active low. (For the µPD8236A-5 use method, refer to the appropriate material, etc.)

(2) Interrupt operation

To enable use of interrupts in the vector mode, the circuit configuration enables the $\overline{\mathrm{INTAK}}$ signal to be used to read interrupt vector. In this mode, the µPD7201A $\overline{\mathrm{INT}}$ signal is connected to the µPD8085AH INTR pin and the $\overline{\mathrm{INTA}}$ signal is input to the µPD7201A $\overline{\mathrm{INTAK}}$ signal.

To use interrupts in the nonvector mode, the µPD7201A $\overline{\mathrm{INT}}$ signal is connected to the µPD8085AH RST7.5 pin and the µPD7201A $\overline{\mathrm{INTAK}}$ signal is pulled up to a high level.

To use interrupts in any mode other than the Both CH. DMA-2 mode (CR2A:D1, D0 = 1, 1), the $\overline{\mathrm{PRI}}$ line must be held LOW (however, Both CH. DMA-1 mode cannot be used).

(3) Serial data send and receive

TxDA, RxDA, TxDB, and RxDB can be used. However, since interfacing with external devices (for example through an as RS-232-C port) is not considered in the circuit, an appropriate interface circuit must be added for connection to external devices.

(4) Send and receive clocks

Send and receive clocks are supplied to $\overline{\text{TxCA}}$, $\overline{\text{RxCA}}$, $\overline{\text{TxCB}}$, and $\overline{\text{RxCB}}$ from µPD8253-2 OUT0 or from external sources. For operation in Asynchronous X1 mode, Synchronous mode, or HDLC mode, receive data must be synchronized with the receive clock.

(5) µPD7201A general-purpose input and output lines

The pins $\overline{\text{CTSA}}$, $\overline{\text{DCDA}}$, $\overline{\text{CTSB}}$, and $\overline{\text{DCDB}}$, which are always LOW can be used as general-purpose input lines. The functions of the general-purpose input and output pins varies with operation mode. Select the function according to the operation mode.

(6) Address map

Figure 4.2 shows the memory address map. Figure 4.3 shows the I/O address map.

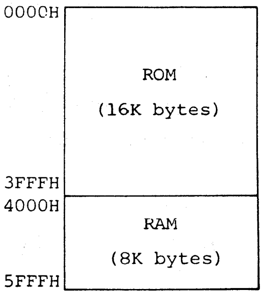


Figure 4.2 Memory map

Figure 4.3 I/O map

I/O address	Contents	I/O device
C0H	Channel A data	µPD7201A area
C1	Channel A control	
C2	Channel B data	
C3	Channel B control	
C4	Counter #0	µPD8253-2 area
C5	Counter #1	
C6	Counter #2	
C7	Control	
C8 (CF	Inhibited	
D0	Channel 0 base/current address	µPD8237A-5 area
D1	Channel 0 base/current word count	
D2	Channel 1 base/current address	
D3	Channel 1 base/current word count	
D4	Channel 2 base/current address	
D5	Channel 2 base/current word count	
D6	Channel 3 base/current address	
D7	Channel 3 base/current word count	
D8	Status/command register	
D9	Write request register	
DA	Write single mask register bit	
DB	Write mode register	
DC	Clear byte pointer flip-flop	
DD	Read temporary register/master clear	
DE	Clear mask register	
DF	Write all mask register bits	

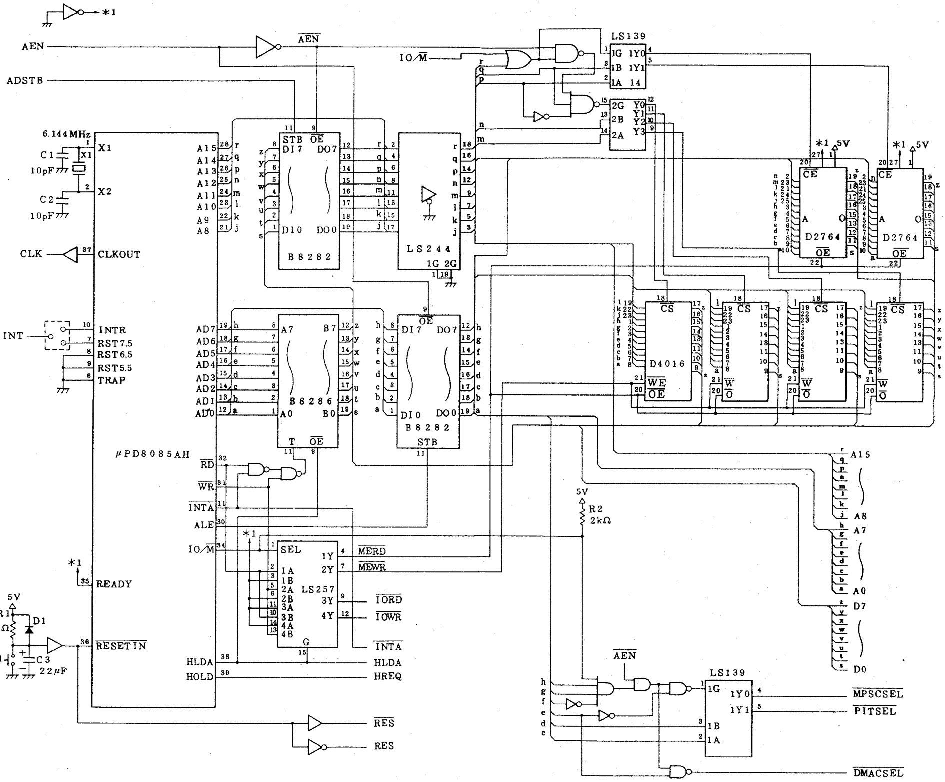


Figure 4.1 (a) μPD7201A application circuit example (reference circuit)

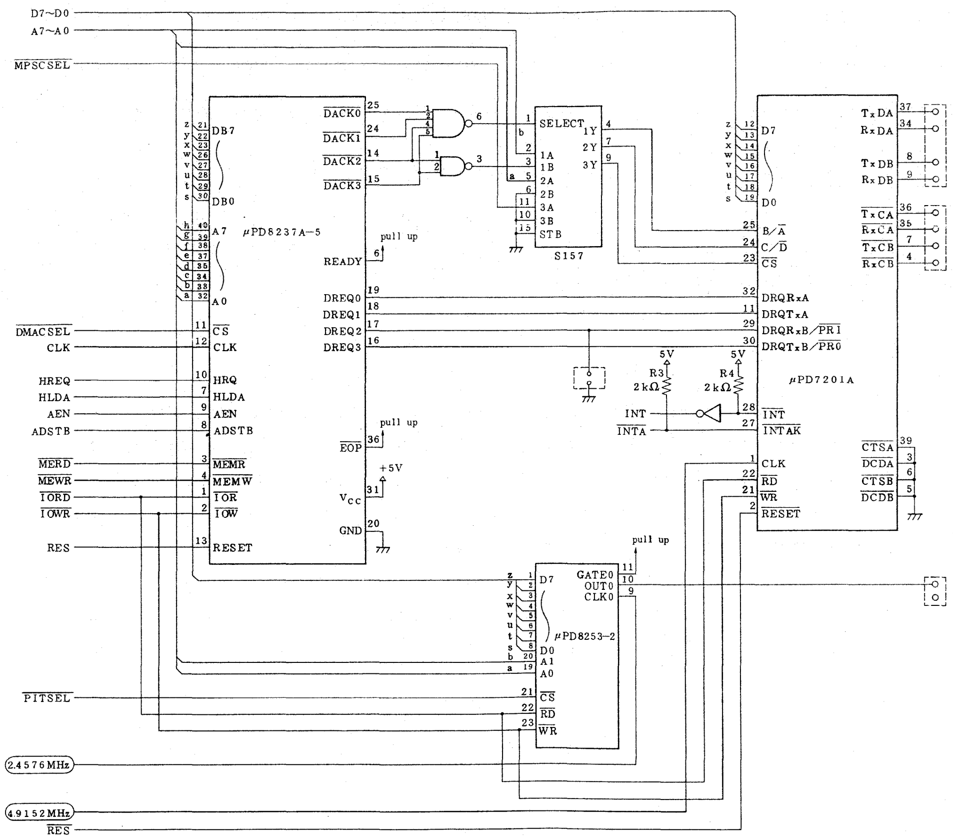


Figure 4.1 (b) μPD7201A application circuit example
(reference circuit)

CHAPTER 5 SOFTWARE

Chapter 5 explains software used to actually operate the hardware designed in Chapter 4.
First, memory assignments are made and program allocation is determined, then a specific program example in Synchronous mode and HDLC mode is discussed. Use this example to understand the basic µPD7201A control method.

5.1 Memory Assignment

Memory assignment is discussed before an actual program is prepared. Since the program explained in this chapter is intented to operate in the hardware environment described in Chapter 4, the hardware conditions are considered to when assigning memory, as shown in Figure 5.1.

The regions shaded in Figure 5.1 are mainly used for the system area. The system control program (monitor program) uses the area.

Other regions are used as the µPD7201A control program area.

The initial value of µPD7201A interrupt vector (value set in CR2B) is set to 40H. Thus, the interrupt vector address area is 40H to 50H.

The area stores the jump destination addresses corresponding to various interrupt sources when the µPD7201A is used in vector mode. (See Table 5.1.) Since the µPD8085AH RST7.5 pin is used when the µPD7201A is uses nonvector mode, and the jump destination address becomes 3CH when an interrupt occurs.

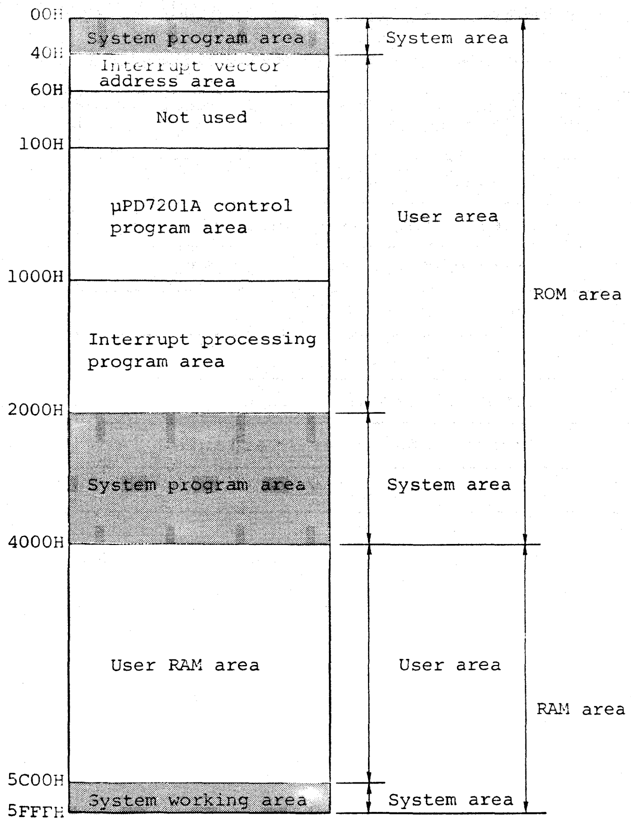


Figure 5.1 Memory assignment

Table 5.1 Interrupt vector addresses and interrupt sources

Interrupt vector address	µPD7201A interrupt source
40H	Channel B Tx Buffer Empty
44H	" External Status Change
48H	" Receive Character Available
4CH	" Special Rx Condition
50H	Channel A Tx Buffer Empty
54H	" External Status Change
58H	" Receive Character Available
5CH	" Special Rx Condition

5.2 Operation Example in Synchronous Mode

This program example is to perform send and receive operation on channel A in interrupt mode. In the example program, the receive operation is performed after completion of send operation for reasons of explanation. A program which performs send and receive operation at the same time can also be prepared.

Table 5.2 lists the µPD7201A operation mode in the present example.

The example program assumes that the send operation is to send one data block and receive operation is to receive one data block ending with ETX.

Table 5.2 μPD7201A operation mode setting (Synchronous operation)

<table>
<tr><th colspan="2">Channel
Operation contents</th><th>Channel A</th><th>Channel B</th></tr>
<tr><td colspan="2">Send and receive operation</td><td>Send (TxDA)
Receive (RxDA)</td><td>-</td></tr>
<tr><td colspan="2">Send and receive data transfer method (CR2A setting)</td><td>Interrupt</td><td>Interrupt</td></tr>
<tr><td colspan="2">Interrupt sources used</td><td>Tx interrupt
Rx interrupt
Special Rx Condition interrupt
E/S interrupt</td><td>-</td></tr>
<tr><td rowspan="4">Interrupt mode</td><td>Vector mode (CR2A: D5)</td><td colspan="2">Vector mode</td></tr>
<tr><td>Interrupt mode (CR2A: D4, D3)</td><td colspan="2">85-1 vector mode</td></tr>
<tr><td>Status Affects Vector (CR1B: D2)</td><td></td><td>Modified Vector</td></tr>
<tr><td>Initial value of interrupt vector (CR2B)</td><td></td><td>40H</td></tr>
<tr><td colspan="2">Synchronous character</td><td>"16H" x 2 bytes</td><td>-</td></tr>
<tr><td colspan="2">Synchronization system</td><td>Internal synchronization</td><td>-</td></tr>
</table>

(1) Send operation

This example assumes that the characters shown in Figure 5.2 are sent. When the characters shown in Figure 5.2 are actually sent in Synchronous mode, normally control characters, etc., are added to normal characters. (For details, refer to the Standard Book, etc.) Figure 5.3 shows the configuration of actually sent characters.

PAD1 and PAD2 in Figure 5.3 are pad characters. Normally, PAD1 sent before send is set to 55H and PAD2 sent following CRC2 is set to FFH. Synchronous characters (SYN) of at least two bytes must be sent as character Synchronization code following PAD1. In the example, 55H is set in CR6A as PAD1 and 16H is set in CR7A as SYN at initialization time, then send is enabled. This enables PAD1 and SYN to be automatically sent before characters are sent. Immediately after send is enabled, 16H is rewritten into CR6A and again SYN is set.

This sequence enables PAD1, SYN, SYN, and SYN to be sent. The number of SYNs to be sent can be controlled by reserving the time until the first character is written. All characters other than th first character (STX in the example) are written by using the interrupt handling routine. The CRC characters are automatically sent by the µPD7201A. If PAD2 is written as data when TX underrun occurs, it can be sent following the CRC characters.

CRC calculation is enabled or disabled immediately before characters are written. In the example, STX and PAD2 are excluded from CRC calculation.

CRC calculation is not applied to characters automatically inserted from CR6 and CR7, thus CRC calculation enable/disable control is not required. This applied to the four continuous characters PAD1, SYN, SYN, SYN in the example.

(2) Receive operation

The example assumes that the characters shown in Figure 5.4 are received.

In the receive operation, two continuous bytes of SYNC must first be detected to set character synchronization. Thus, the μPD7201A is first placed in Hunt mode to detect SYN. When character synchronization is set, the μPD7201A generates, receive interrupt for the subsequent characters received. The example program is designed to receive SYN to ETX characters as normal characters. In Synchronous mode, characters excluded from CRC calculation must be checked. This control can be implemented by reading a receive character and checking the character contents, then enabling or disabling CRC calculation.

In this case, CRC calculation enable or disable control must be completed before the next character is transferred to the receive buffer. This can be processed by controlling CRC calculation enable or disable within the time of five bits (during eight bits per character) after a μPD7201A receive interrupt occurs.

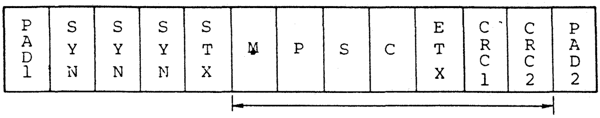
In the example, SYN, PAD2, and dummy data are excluded from CRC calculation.

The μPD7201A uses a 16-bit time interval after the last character (second byte of CRC in the example) is transferred to the receive buffer until the CRC calculation result is output. Thus, after the last character is received, on extra two bytes of dummy data are received before the CRC Error bit (SR1A: D6 bit) is checked.

Figures 5.5 to 5.9 show the process flow to perform the operation mentioned above. Figure 5.10 shows actual program listing.

M	P	S	C

Figure 5.2 Send character configuration
(send starts at the leftmost character)



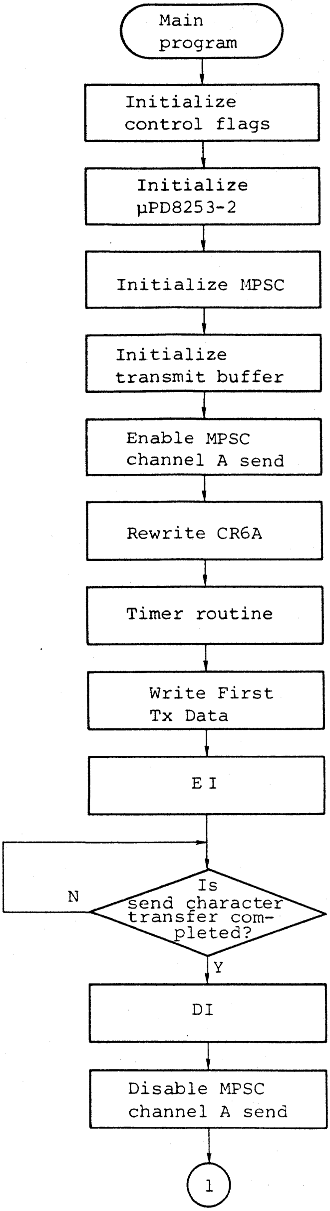
PAD1	SYN	SYN	SYN	STX	M	P	S	C	ETX	CRC1	CRC2	PAD2

These characters are contained in CRC calculation.

Figure 5.3 Actually sent characters
(send starts at the leftmost character)

SYN	SYN	Any characters	ETX	CRC1	CRC2

Figure 5.4 Receive character string format
(receive starts at the leftmost character)



... Start at 100H address

... Control flags used in the program are initialized

... Basic operation mode is set

... Send operation is started

... Tx CRC remains disabled

... CR6A is again set to SYN (16H)

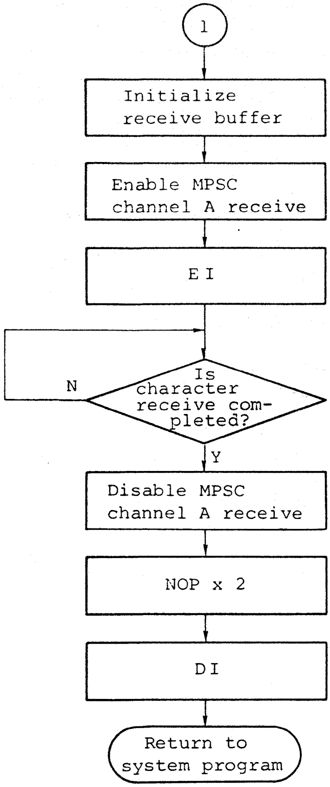
... The number of SYNs to be sent can be adjusted by changing the value.

... STX is written

... A check is made to see if all send characters have been written.

... Send operation is terminated

Figure 5.5 Synchronous operation main program 1



... Receive operation is started

... Here, Hunt mode is entered

... Wait for completion of all characters received (CRC Error check is contained)

Figure 5.5 Synchronous operation main program 2

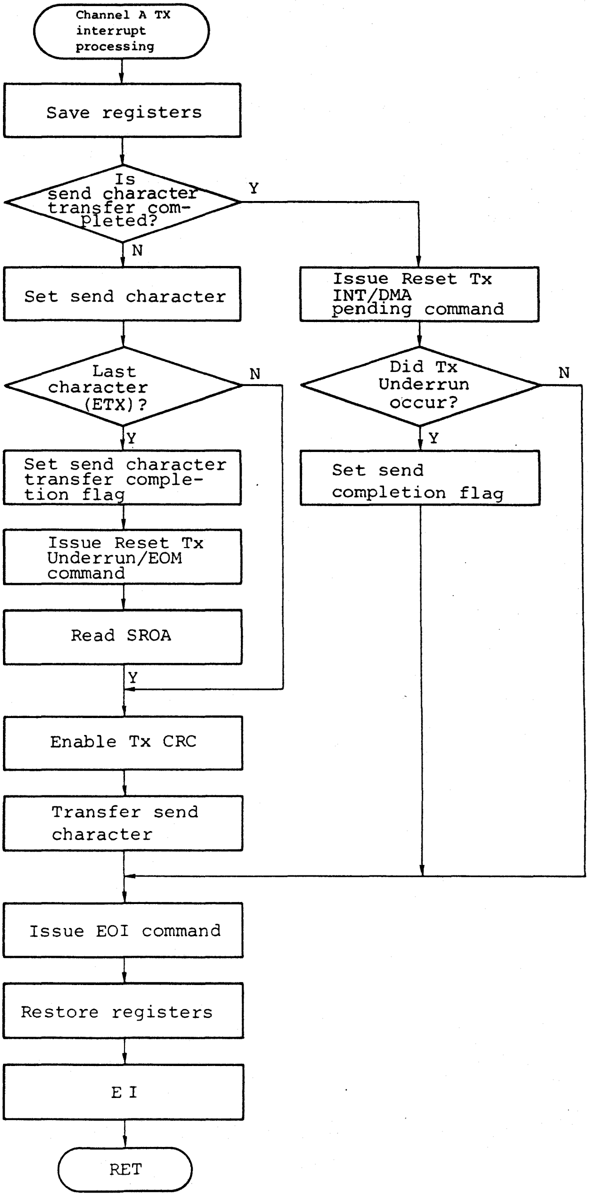


Figure 5.6 Synchronous operation Tx interrupt processing

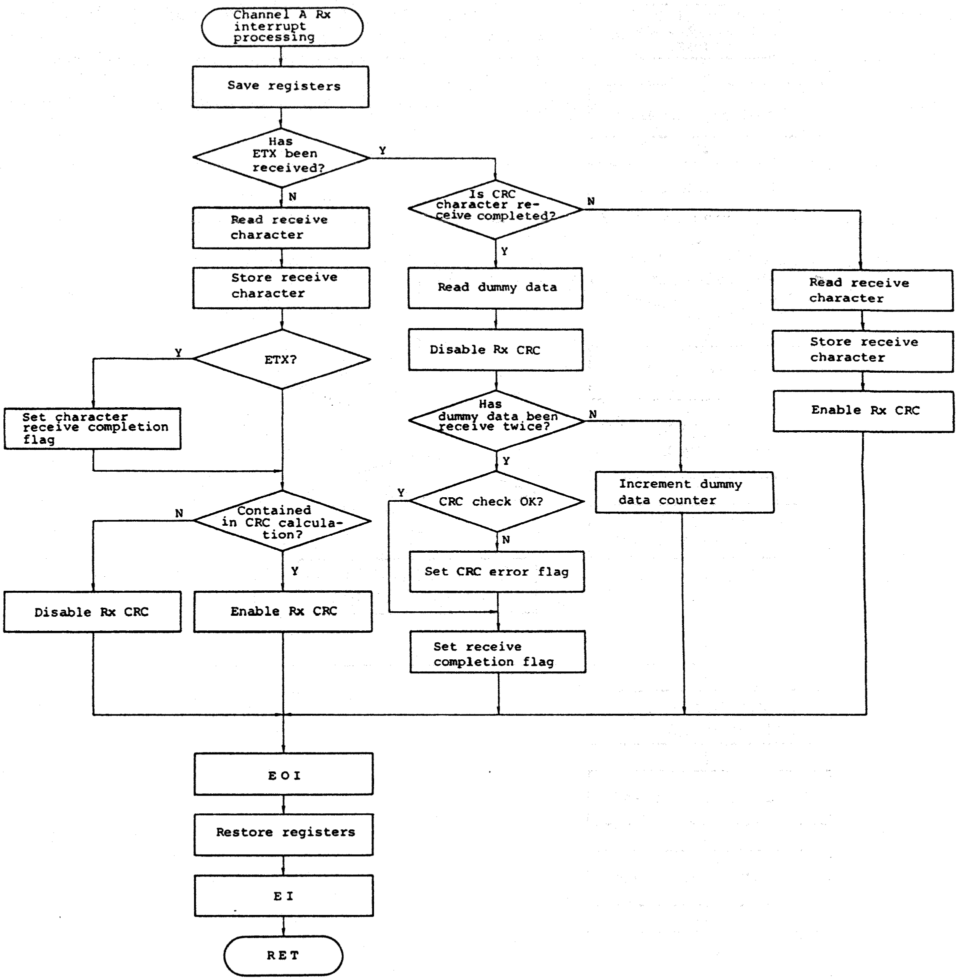


Figure 5.7 Synchronous operation Rx interrupt processing

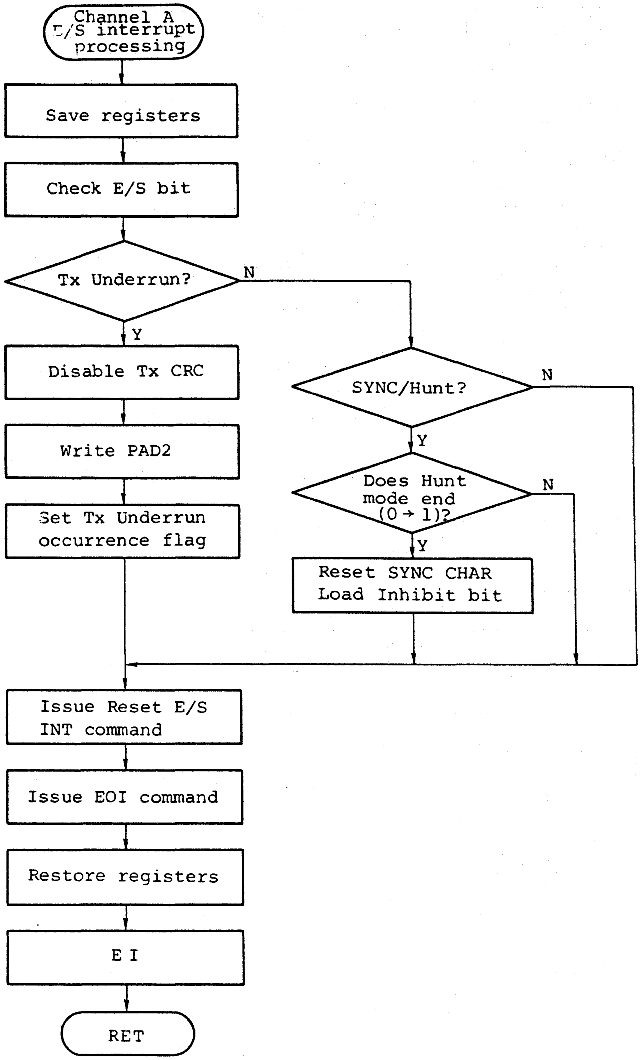


Figure 5.8 Synchronous operation E/S interrupt processing

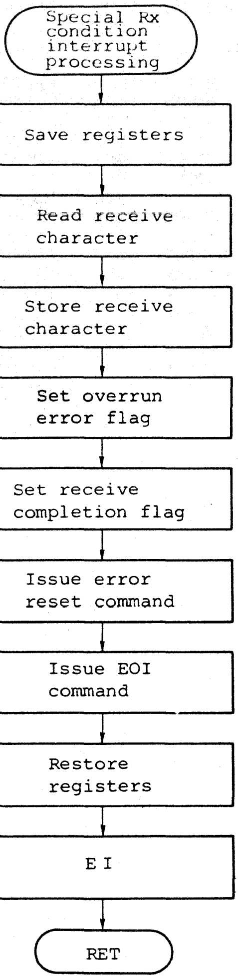


Figure 5.9 Synchronous operation Special Rx Condition interrupt processing

```
SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS


BSCCT   0100   CHACT   00C1   CHADT   00C0   CHBCT   00C3   CHBDT   00C2
CHNRS   0018   CKCNT   10D7   CKCRC   110E   CKERR   1132   CKES1   105C
CKES2   1075   CKRDE   109B   CKWRE   1003   CLRFG   0103   CR1A    0013
CR1B    0004   CR2A    0020   CR2B    0040   CR3A    00C2   CR4A    0010
CR5A    0044   CR6A    0016   CR7A    0016   CRCEF   5B09   CRCNT   5B0B
CRCSR   10B7   CRCST   1022   CUNDF   1034   DMCNT   5B0C   EOI     0038
ERDFG   5B07   ERRES   0030   ERVDF   5B08   ESINA   104D   ESRES   0010
ESRTA   1088   ETX     0003   EWDFG   5B04   EXMDF   5B06   EXTHP   1080
OVREF   5B0A   OVRER   1139   PAD1    0055   PAD2    00FF   PEND    0028
PITC0   00C4   PITC1   00C5   PITC2   00C6   PITCT   00C7   RCRDS   10CC
RCREN   10C1   RDCRC   10E1   RDDAT   10A2   RDDMY   10F6   RXBFS   01A3
RXBUF   4400   RXCRC   0040   RXEOI   1123   RXINA   1094   RXPTR   5B02
RXTRM   01B8   SPRXA   112D   SR0AS   5B0D   SRXEF   111E   STACK   6000
STCHA   0135   STPIT   010C   STX     0002   STXDT   100A   SWREF   1014
SYHNT   107A   SYN     0016   SYRCV   01AF   SYSTM   2000   SYXMT   0170
TDLY1   0180   TXBFS   016A   TXBUF   4000   TXCRC   0080   TXEOI   1044
TXINA   1000   TXPTR   5B00   TXTRM   0193   UNDEM   00C0   UNDES   1061
UNDFG   5B05   WRDAT   102A   XMTDT   0188
```

Figure 5.10 Synchronous operation program example 1

```
µCOM-85 ASSEMBLE LIST      (                                        ) PAGE 0001

E STNO ADRS OBJECT M   SOURCE STATEMENTS


 0001                  ;*****************************************
 0002                  ;* µPD7201A BSC OPERATION SAMPLE PROGRAM *
 0003                  ;*         TX : TXDA -->                 *
 0004                  ;*         RX : RXDA <--                 *
 0005                  ;*****************************************
 0006                  ;
 0007                  ;***************
 0008                  ;* I/O ADDRESS *
 0009                  ;***************
 0010                  ;
 0011 00C0             CHADT   EQU     0C0H            ;7201A CH-A DATA
 0012 00C1             CHACT   EQU     0C1H            ;7201A CH-A CNTRL
 0013 00C2             CHBDT   EQU     0C2H            ;7201A CH-B DATA
 0014 00C3             CHBCT   EQU     0C3H            ;7201A CH-B CNTRL
 0015                  ;
 0016 00C4             PITC0   EQU     0C4H            ;8253 CNT #0
 0017 00C5             PITC1   EQU     0C5H            ;8253 CNT #1
 0018 00C6             PITC2   EQU     0C6H            ;8253 CNT #2
 0019 00C7             PITCT   EQU     0C7H            ;8253 CONTROL
 0020                  ;
 0021 2000             SYSTM   EQU     2000H           ;SYSTEM PROGRAM
 0022                  ;
 0023                  ;****************
 0024                  ;* MPSC COMMAND *
 0025                  ;****************
 0026                  ;
 0027 0010             ESRES   EQU     10H             ;RESET E/S INT
 0028 0018             CHNRS   EQU     18H             ;CHANNEL RESET
 0029 0028             PEND    EQU     28H             ;RES TX PENDING
 0030 0030             ERRES   EQU     30H             ;ERROR RESET
 0031 0038             EOI     EQU     38H             ;END OF INTERRUPT
 0032                  ;
 0033 0040             RXCRC   EQU     40H             ;INIT RX CRC CAL
 0034 0080             TXCRC   EQU     80H             ;INIT TX CRC CAL
 0035 00C0             UNDEM   EQU     0C0H            ;RESET TX UND/EOM
 0036                  ;
 0037                  ;***********************************
 0038                  ;* MPSC CONTROL REGISTER PARAMETER *
 0039                  ;***********************************
 0040                  ;
 0041 0013             CR1A    EQU     13H             ;RX.TX.E/S INT EN
 0042 0020             CR2A    EQU     20H             ;INT.VECTORED MODE
 0043 00C2             CR3A    EQU     0C2H            ;RX 8 BITS
 0044 0010             CR4A    EQU     10H             ;16 BITS SYN.NO PA
 0045 0044             CR5A    EQU     44H             ;TX 8 BITS.CRC-16
 0046 0016             CR6A    EQU     16H             ;SYN #1
 0047 0016             CR7A    EQU     16H             ;SYN #2
 0048                  ;
 0049 0004             CR1B    EQU     4H              ;MODIFIED VECTOR
 0050 0040             CR2B    EQU     40H             ;VECTOR(010***00)
 0051                  ;
 0052                  ;
```

Figure 5.10 Synchronous operation program example 2

```
   UCOM-85 ASSEMBLE LIST    (                                   ) PAGE 0002

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0053                  ;***********************
  0054                  :* CONTROL CHARACTERS *
  0055                  ;***********************
  0056                  :
  0057 0016             SYN     EQU     16H
  0058 0002             STX     EQU     2H
  0059 0003             ETX     EQU     3H
  0060                  :
  0061 0055             PAD1    EQU     55H
  0062 00FF             PAD2    EQU     0FFH
  0063                  :
  0064                  ;****************************************
  0065                  :* INTERRUPT JUMP TABLE (VECTORED MODE) *
  0066                  ;****************************************
  0067                  :
  0068 0000                     ORG     50H             :CH-A TX INT
  0069 0050 C30010              JMP     TXINA
  0070                  :
  0071 0053                     ORG     54H             :CH-A E/S INT
  0072 0054 C34D10              JMP     ESINA
  0073                  :
  0074 0057                     ORG     58H             :CH-A RX INT
  0075 0058 C39410              JMP     RXINA
  0076                  :
  0077 005B                     ORG     5CH             :CH-A SPCL RX INT
  0078 005C C32D11              JMP     SPRXA
  0079                  :
  0080                  ;**********************************
  0081                  :* PERIPHERAL INITIALIZE SEQUENCE *
  0082                  ;**********************************
  0083                  :
  0084 005F                     ORG     100H
  0085                  :
  0086 0100 310060      BSCCT:  LXI     SP.STACK
  0087 0103 AF          CLRFG:  XRA     A               :CLEAR STATUS FLG
  0088 0104 21005B              LXI     H.TXPTR
  0089 0107 77                  MOV     M.A
  0090 0108 2C                  INR     L
  0091 0109 C20701              JNZ     $-2
  0092                  :
  0093                  : 8253-2 INITIALIZE
  0094                  :
  0095 010C 210001      STPIT:  LXI     H.100H          :CNT #0.9.6K BPS
  0096 010F 3E36                MVI     A.36H           :#0 MODE 2
  0097 0111 D3C7                OUT     PITCT
  0098 0113 7D                  MOV     A.L
  0099 0114 D3C4                OUT     PITC0
  0100 0116 7C                  MOV     A.H
  0101 0117 D3C4                OUT     PITC0
  0102                  :
  0103                  :
  0104                  :
  0105                  :
```

Figure 5.10 Synchronous operation program example 3

```
   UCOM-85 ASSEMBLE LIST     (                                        ) PAGE 0003

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0106                 ;****************************
  0107                 ;* MPSC INITIALIZE SEQUENCE *
  0108                 ;****************************
  0109                 ;
  0110                 ; COMMON REGISTER INITIALIZE
  0111                 ;
  0112 0119 3E18                MVI     A.CHNRS          ;CHANNEL RESET
  0113 011B D3C1                OUT     CHACT
  0114 011D D3C3                OUT     CHBCT
  0115 011F 3E02                MVI     A.2H             ;PTR 2
  0116 0121 D3C1                OUT     CHACT
  0117 0123 D3C3                OUT     CHBCT
  0118 0125 3E20                MVI     A.CR2A           ;BUS IF MODE
  0119 0127 D3C1                OUT     CHACT
  0120 0129 3E40                MVI     A.CR2B           ;INERRUPT VECTOR
  0121 012B D3C3                OUT     CHBCT
  0122 012D 3E01                MVI     A.1H             ;PTR 1B
  0123 012F D3C3                OUT     CHBCT
  0124 0131 3E04                MVI     A.4H             ;MODIFIED VECTOR
  0125 0133 D3C3                OUT     CHBCT
  0126                 ;
  0127                 ; CH-A OPERATION MODE
  0128                 ;
  0129 0135 3E04       STCHA:   MVI     A.4H             ;PTR 4A
  0130 0137 D3C1                OUT     CHACT
  0131 0139 3E10                MVI     A.CR4A           ;CH-A MODE
  0132 013B D3C1                OUT     CHACT
  0133 013D 3E03                MVI     A.3H             ;PTR 3A
  0134 013F D3C1                OUT     CHACT
  0135 0141 3EC2                MVI     A.CR3A           ;RX PARAMETER
  0136 0143 D3C1                OUT     CHACT
  0137 0145 3E05                MVI     A.5H             ;PTR 5A
  0138 0147 D3C1                OUT     CHACT
  0139 0149 3E44                MVI     A.CR5A           ;TX PARAMETER
  0140 014B D3C1                OUT     CHACT
  0141 014D 3E06                MVI     A.6H             ;PTR 6A
  0142 014F D3C1                OUT     CHACT
  0143 0151 3E55                MVI     A.PAD1           ;LEADING PAD
  0144 0153 D3C1                OUT     CHACT
  0145 0155 3E07                MVI     A.7H             ;PTR 7A
  0146 0157 D3C1                OUT     CHACT
  0147 0159 3E16                MVI     A.CR7A           ;SYN
  0148 015B D3C1                OUT     CHACT
  0149 015D 3E11                MVI     A.11H            ;PTR 1A.RS ES INT
  0150 015F D3C1                OUT     CHACT
  0151 0161 3E13                MVI     A.CR1A           ;INT MODE
  0152 0163 D3C1                OUT     CHACT
  0153 0165 DBC1                IN      CHACT            ;READ FIRST SR0A
  0154 0167 320D5B              STA     SR0AS
  0155                 ;
  0156                 ;
  0157                 ;
  0158                 ;
```

Figure 5.10 Synchronous operation program example 4

```
   UCOM-85 ASSEMBLE LIST     (                                   ) PAGE 0004

E STNO ADRS OBJECT M   SOURCE STATEMENTS


  0159                 ;******************************
  0160                 ;* TRANSMIT OPERATION ROUTINE *
  0161                 ;******************************
  0162                 ;
  0163 016A 210040     TXBFS:  LXI      H.TXBUF          ;GET TX BUF ADRS
  0164 016D 22005B             SHLD     TXPTR
  0165 0170 3E85       SYXMT:  MVI      A.85H            ;PTR 5A
  0166 0172 D3C1               OUT      CHACT
  0167 0174 3EEE               MVI      A.0EEH           ;TX EN.TX CRC DIS
  0168 0176 D3C1               OUT      CHACT
  0169 0178 3E06               MVI      A.6H             ;PTR 6A
  0170 017A D3C1               OUT      CHACT
  0171 017C 3E16               MVI      A.SYN            ;SYN
  0172 017E D3C1               OUT      CHACT
  0173                 ;
  0174 0180 3E50       TDLY1:  MVI      A.50H            ;TIME DELAY
  0175 0182 29                 DAD      H
  0176 0183 29                 DAD      H
  0177 0184 3D                 DCR      A
  0178 0185 C28201             JNZ      $-3
  0179                 XMTDT:
  0180 0188 2A005B             LHLD     TXPTR            ;GET TX DATA ADRS
  0181 018B 7E                 MOV      A.M
  0182 018C D3C0               OUT      CHADT
  0183 018E 23                 INX      H
  0184 018F 22005B             SHLD     TXPTR
  0185                 ;
  0186 0192 FB                 EI
  0187                 ;
  0188 0193 3A065B     TXTRM:  LDA      EXMDF            ;CHECK TX END FLG
  0189 0196 0F                 RRC
  0190 0197 D29301             JNC      TXTRM
  0191 019A F3                 DI
  0192 019B 3E05               MVI      A.5H             ;PTR 5A
  0193 019D D3C1               OUT      CHACT
  0194 019F 3E44               MVI      A.44H            ;TX DISABLE
  0195 01A1 D3C1               OUT      CHACT
  0196                 ;
  0197                 ;*****************************
  0198                 ;* RECEIVE OPERATION ROUTINE *
  0199                 ;*****************************
  0200                 ;
  0201 01A3 210044     RXBFS:  LXI      H.RXBUF          ;GET RX BUF ADRS
  0202 01A6 22025B             SHLD     RXPTR
  0203 01A9 AF                 XRA      A
  0204 01AA 77                 MOV      M.A              ;CLEAR RX BUFFER
  0205 01AB 2C                 INR      L
  0206 01AC C2AA01             JNZ      $-2
  0207 01AF 3E03       SYRCV:  MVI      A.3H             ;PTR 3A
  0208 01B1 D3C1               OUT      CHACT
  0209 01B3 3ED3               MVI      A.0D3H           ;RX EN.ENTER HUNT
  0210 01B5 D3C1               OUT      CHACT
  0211 01B7 FB                 EI
```

Figure 5.10 Synchronous operation program example 5

```
   UCOM-85 ASSEMBLE LIST     (                                           ) PAGE 0005

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0212                      :
  0213 01B8 3A085B      RXTRM:  LDA     ERVDF           :CHECK RCV END FLG
  0214 01BB 0F                  RRC
  0215 01BC D2B801              JNC     RXTRM
  0216 01BF 3E03                MVI     A.3H            :PTR 3A
  0217 01C1 D3C1                OUT     CHACT
  0218 01C3 3EC0                MVI     A.0C0H          :RX DISABLE
  0219 01C5 D3C1                OUT     CHACT
  0220 01C7 00                  NOP
  0221 01C8 00                  NOP
  0222 01C9 F3                  DI
  0223                      :
  0224 01CA C30020              JMP     SYSTM
  0225                      :
  0226                      :*************************************************
  0227                      :* INTERRUPT PROCESSING ROUTINE (VECTORED MODE) *
  0228                      :*************************************************
  0229                      :
  0230 01CD                     ORG     1000H
  0231                      :
  0232                      :---------------------
  0233                      : CH-A TX INT PROCESS
  0234                      :---------------------
  0235                      :
  0236                      TXINA:
  0237 1000 F5                  PUSH    PSW
  0238 1001 C5                  PUSH    B
  0239 1002 E5                  PUSH    H
  0240 1003 3A045B      CKWRE:  LDA     EWDFG           :CHECK WR END FLG
  0241 1006 0F                  RRC
  0242 1007 DA3410              JC      CUNDF
  0243 100A 2A005B      STXDT:  LHLD    TXPTR           :GET TX DATA ADRS
  0244 100D 7E                  MOV     A.M             :GET TX DATA
  0245 100E 4F                  MOV     C.A
  0246 100F FE03                CPI     ETX             :LAST TX DATA ?
  0247 1011 C22210              JNZ     CRCST
  0248 1014 3E01        SWREF:  MVI     A.1             :SET WR END FLG
  0249 1016 32045B              STA     EWDFG
  0250 1019 3EC0                MVI     A.UNDEM         :RES TX UND/EOM
  0251 101B D3C1                OUT     CHACT
  0252 101D DBC1                IN      CHACT           :READ SR0A
  0253 101F 320D5B              STA     SR0AS           :NEW SR0A STATUS
  0254 1022 3E05        CRCST:  MVI     A.5H            :PTR 5A
  0255 1024 D3C1                OUT     CHACT
  0256 1026 3EEF                MVI     A.0EFH          :TX CRC ENABLE
  0257 1028 D3C1                OUT     CHACT
  0258 102A 79          WRDAT:  MOV     A.C             :SET TX DATA
  0259 102B D3C0                OUT     CHADT
  0260 102D 23                  INX     H               :UPDATE ADRS PTR
  0261 102E 22005B              SHLD    TXPTR
  0262 1031 C34410              JMP     TXEOI
  0263 1034 3E28        CUNDF:  MVI     A.PEND          :RESET TX PENDING
  0264 1036 D3C1                OUT     CHACT
```

Figure 5.10 Synchronous operation program example 6

```
   UCOM-85 ASSEMBLE LIST    (                                      ) PAGE 0006

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0265 1038 3A055B             LDA      UNDFG            :GET UNDRN FLG
  0266 103B 0F                 RRC
  0267 103C D24410             JNC      TXEOI
  0268 103F 3E01               MVI      A.1              :SET TX END FLG
  0269 1041 32065B             STA      EXMDF
  0270 1044 3E38       TXEOI:  MVI      A.EOI            :EOI
  0271 1046 D3C1               OUT      CHACT
  0272 1048 E1                 POP      H
  0273 1049 C1                 POP      B
  0274 104A F1                 POP      PSW
  0275 104B FB                 EI
  0276 104C C9                 RET
  0277                 :
  0278                 :---------------------
  0279                 : CH-A E/S INT PROCESS
  0280                 :---------------------
  0281                 :
  0282                 ESINA:
  0283 104D F5                 PUSH     PSW
  0284 104E C5                 PUSH     B
  0285 104F 3A0D5B             LDA      SR0AS            :GET OLD E/S BITS
  0286 1052 47                 MOV      B.A
  0287 1053 DBC1               IN       CHACT            :READ NEW E/S BITS
  0288 1055 4F                 MOV      C.A
  0289 1056 320D5B             STA      SR0AS            :NEW E/S BITS
  0290 1059 A8                 XRA      B                :CHECK CHANGED E/S
  0291 105A E650               ANI      50H
  0292 105C 07         CKES1:  RLC
  0293 105D 07                 RLC
  0294 105E D27510             JNC      CKES2
  0295 1061 3E05       UNDES:  MVI      A.5              :PTR 5A
  0296 1063 D3C1               OUT      CHACT
  0297 1065 3EEE               MVI      A.0EEH           :TX CRC DISABLE
  0298 1067 D3C1               OUT      CHACT
  0299 1069 3EFF               MVI      A.PAD2           :WRITE TPAD
  0300 106B D3C0               OUT      CHADT
  0301 106D 3E01               MVI      A.1              :SET UNDRN FLG
  0302 106F 32055B             STA      UNDFG
  0303 1072 C38810             JMP      ESRTA
  0304 1075 07         CKES2:  RLC
  0305 1076 07                 RLC
  0306 1077 D28810             JNC      ESRTA
  0307 107A 79         SYHNT:  MOV      A.C              :SET NEW E/S BITS
  0308 107B E610               ANI      10H              :SYNC/HUNT BIT=1?
  0309 107D C28810             JNZ      ESRTA
  0310 1080 3E03       EXTHP:  MVI      A.3H             :PTR 3A
  0311 1082 D3C1               OUT      CHACT
  0312 1084 3EC9               MVI      A.0C9H           :RES SYN LOAD INH
  0313 1086 D3C1               OUT      CHACT
  0314 1088 3E10       ESRTA:  MVI      A.ESRES          :RESET E/S INT
  0315 108A D3C1               OUT      CHACT
  0316 108C 3E38               MVI      A.EOI            :EOI
  0317 108E D3C1               OUT      CHACT
```

Figure 5.10 Synchronous operation program example 7

```
  UCOM-85 ASSEMBLE LIST    (                                      ) PAGE 0007

E STNO ADRS OBJECT M    SOURCE STATEMENTS


 0318 1090 C1                  POP      B
 0319 1091 F1                  POP      PSW
 0320 1092 FB                  EI
 0321 1093 C9                  RET
 0322                   ;
 0323                   ;---------------------
 0324                   ; CH-A RX INT PROCESS
 0325                   ;---------------------
 0326                   ;
 0327                   RXINA:
 0328 1094 F5                  PUSH     PSW
 0329 1095 C5                  PUSH     B
 0330 1096 D5                  PUSH     D
 0331 1097 E5                  PUSH     H
 0332 1098 2A025B              LHLD     RXPTR
 0333 109B 3A075B       CKRDE: LDA      ERDFG           ;GET RX RDEND FLG
 0334 109E 0F                  RRC
 0335 109F DAD710              JC       CKCNT
 0336 10A2 DBC0         RDDAT: IN       CHADT           ;READ RX DATA
 0337 10A4 4F                  MOV      C.A
 0338 10A5 77                  MOV      M.A             ;STORE RX DATA
 0339 10A6 23                  INX      H
 0340 10A7 22025B              SHLD     RXPTR
 0341 10AA FE03                CPI      ETX             ;LAST DATA ?
 0342 10AC C2B710              JNZ      CRCSR
 0343 10AF 3E01                MVI      A.1H            ;SET RXD RDEND FLG
 0344 10B1 32075B              STA      ERDFG
 0345 10B4 C3C110              JMP      RCREN
 0346 10B7 FE16         CRCSR: CPI      SYN             ;RX DATA = 'SYN' ?
 0347 10B9 CACC10              JZ       RCRDS
 0348 10BC FE02                CPI      STX             ;RX DATA = 'STX' ?
 0349 10BE CACC10              JZ       RCRDS
 0350 10C1 3E03         RCREN: MVI      A.3H            ;PTR 3A
 0351 10C3 D3C1                OUT      CHACT
 0352 10C5 3EC9                MVI      A.0C9H          ;RX CRC ENABLE
 0353 10C7 D3C1                OUT      CHACT
 0354 10C9 C32311              JMP      RXEOI
 0355 10CC 3E03         RCRDS: MVI      A.3H            ;PTR 3A
 0356 10CE D3C1                OUT      CHACT
 0357 10D0 3EC1                MVI      A.0C1H          ;RX CRC DISABLE
 0358 10D2 D3C1                OUT      CHACT
 0359 10D4 C32311              JMP      RXEOI
 0360 10D7 110B5B       CKCNT: LXI      D.CRCNT         ;GET RX CRC CNT
 0361 10DA 1A                  LDAX     D
 0362 10DB 47                  MOV      B.A
 0363 10DC FE02                CPI      2H              ;CRC RX END ?
 0364 10DE CAF610              JZ       RDDMY
 0365 10E1 DBC0         RDCRC: IN       CHADT           ;READ RX DATA
 0366 10E3 77                  MOV      M.A
 0367 10E4 23                  INX      H
 0368 10E5 22025B              SHLD     RXPTR
 0369 10E8 3E03                MVI      A.3H            ;PTR 3A
 0370 10EA D3C1                OUT      CHACT
```

Figure 5.10 Synchronous operation program example 8

```
   UCOM-85 ASSEMBLE LIST    (                                     ) PAGE 0008

E STNO ADRS OBJECT M   SOURCE STATEMENTS


   0371 10EC 3EC9                MVI      A.0C9H            :RX CRC ENABLE
   0372 10EE D3C1                OUT      CHACT
   0373 10F0 78                  MOV      A.B
   0374 10F1 3C                  INR      A
   0375 10F2 12                  STAX     D
   0376 10F3 C32311              JMP      RXEOI
   0377 10F6 DBC0       RDDMY:   IN       CHADT             :READ DAMMY DATA
   0378 10F8 3E03                MVI      A.3H              :PTR 3A
   0379 10FA D3C1                OUT      CHACT
   0380 10FC 3EC1                MVI      A.0C1H            :RX CRC DISABLE
   0381 10FE D3C1                OUT      CHACT
   0382 1100 110C5B              LXI      D.DMCNT           :GET DAMMY RD CNT
   0383 1103 1A                  LDAX     D
   0384 1104 FE01                CPI      1H                :DAMMY READ END ?
   0385 1106 CA0E11              JZ       CKCRC
   0386 1109 3C                  INR      A
   0387 110A 12                  STAX     D
   0388 110B C32311              JMP      RXEOI
   0389 110E 3E01       CKCRC:   MVI      A.1H              :PTR 1A
   0390 1110 D3C1                OUT      CHACT
   0391 1112 DBC1                IN       CHACT
   0392 1114 07                  RLC
   0393 1115 07                  RLC
   0394 1116 D21E11              JNC      SRXEF
   0395 1119 3E01                MVI      A.1H              ;SET CRC ERROR FLG
   0396 111B 32095B              STA      CRCEF
   0397 111E 3E01       SRXEF:   MVI      A.1H              :SET RCV END FLG
   0398 1120 32085B              STA      ERVDF
   0399 1123 3E38       RXEOI:   MVI      A.EOI             :EOI
   0400 1125 D3C1                OUT      CHACT
   0401 1127 E1                  POP      H
   0402 1128 D1                  POP      D
   0403 1129 C1                  POP      B
   0404 112A F1                  POP      PSW
   0405 112B FB                  EI
   0406 112C C9                  RET
   0407                 ;
   0408                 ;
   0409                 ;---------------------------
   0410                 ; CH-A SPECIAL RX INT PROCESS
   0411                 ;---------------------------
   0412                 ;
   0413                 SPRXA:
   0414 112D F5                  PUSH     PSW
   0415 112E E5                  PUSH     H
   0416 112F 2A025B              LHLD     RXPTR             :GET RX DATA BUF
   0417 1132 DBC0       CKERR:   IN       CHADT
   0418 1134 77                  MOV      M.A
   0419 1135 23                  INX      H
   0420 1136 22025B              SHLD     RXPTR
   0421 1139 3E01       OVRER:   MVI      A.1H
   0422 113B 320A5B              STA      OVREF             :SET OVRN ERR FLG
   0423 113E 32085B              STA      ERVDF             :SET RCV END FLG
```

Figure 5.10 Synchronous operation program example 9

```
  UCOM-85 ASSEMBLE LIST    (                                          ) PAGE 0009

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0424 1141 3E30                MVI     A.ERRES           :ERROR RESET
  0425 1143 D3C1                OUT     CHACT
  0426 1145 3E03                MVI     A.3H              ;PTR 3A
  0427 1147 D3C1                OUT     CHACT
  0428 1149 3EC0                MVI     A.0C0H            ;RX DISABLE
  0429 114B D3C1                OUT     CHACT
  0430 114D 3E38                MVI     A.EOI
  0431 114F D3C1                OUT     CHACT
  0432 1151 E1                  POP     H
  0433 1152 F1                  POP     PSW
  0434 1153 FB                  EI
  0435 1154 C9                  RET
  0436                 :
  0437                 :***********************
  0438                 :* TX DATA BUFFER AREA *
  0439                 :***********************
  0440                 :
  0441 1155                     ORG     4000H
  0442                 :
  0443                 TXBUF:
  0444 4000 02                  DB      STX
  0445 4001 4D                  DB      'M'
  0446 4002 50                  DB      'P'
  0447 4003 53                  DB      'S'
  0448 4004 43                  DB      'C'
  0449 4005 03                  DB      ETX
  0450                 ;
  0451                 ;***********************
  0452                 :* RX DATA BUFFER AREA *
  0453                 ;***********************
  0454                 :
  0455 4006                     ORG     4400H
  0456                 ;
  0457 4400            RXBUF:   DS      1024
  0458                 ;
  0459                 ;********************
  0460                 :* STATUS FLAG AREA *
  0461                 :********************
  0462                 :
  0463 4800                     ORG     5B00H
  0464                 ;
  0465 5B00            TXPTR:   DS      2                 :TX DATA BUF ADRS
  0466 5B02            RXPTR:   DS      2                 ;RX DATA BUF ADRS
  0467                 ;
  0468 5B04            EWDFG:   DS      1                 :TXDATA WR END FLG
  0469 5B05            UNDFG:   DS      1                 ;TX UNDRN FLG
  0470 5B06            EXMDF:   DS      1                 ;XMIT END FLG
  0471 5B07            ERDFG:   DS      1                 ;RXD RDEND FLG
  0472 5B08            ERVDF:   DS      1                 ;RECEIVE END FLG
  0473 5B09            CRCEF:   DS      1                 ;CRC ERROR FLG
  0474 5B0A            OVREF:   DS      1                 ;OVERRUN ERROR FLG
  0475 5B0B            CRCNT:   DS      1                 ;RX CRC COUNTER
  0476 5B0C            DMCNT:   DS      1                 :DAMMY DATA RD CNT
```

Figure 5.10 Synchronous operation program example 10

```
  UCOM-85 ASSEMBLE LIST    (                                          ) PAGE 0010

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0477                 ;
  0478 5B0D            SR0AS:  DS      1                  ;SR0A STATUS
  0479                 ;
  0480                 ;**************
  0481                 ;* STACK AREA *
  0482                 ;**************
  0483                 ;
  0484 5B0E                    ORG     6000H
  0485                 ;
  0486                 STACK:
  0487 0000                    END
```

Figure 5.10 Synchronous operation program example 11

5.3 Operation Example in HDLC Mode

Send and receive operation via DMA on channel A is described as an operation example in HDLC mode. In the example, the send and receive control programs are independent of each other, but a program which performs send and receive operation at the same time can also be prepared because the µPD7201A enables full-duplex operation.

Each of the example send and receive control programs transfers data of 256 bytes per frame.

Table 5.3 Lists the µPD7201A operation mode in the program example

Table 5.3 µPD7201A operation mode setting
(HDLC operation)

Channel Operation contents		Channel A	Channel B
Send and receive operation		Send (TxDA) Receive (RxDA)	-
Send and receive data transfer method (CR2A setting)		DMA	DMA
Interrupt sources used		Special Rx Condition interrupt E/S interrupt	-
Interrupt setting	Vector mode (CR2A: D5)	Nonvector mode	
	Interrupt mode (CR2A: D4, D3)	85-1 mode	
	Status Affects Vector (CR1B: D2)		Modified Vector
	Initial value of interrupt vector (CR2B)		40H

5.3.1 Send operation

DMA transfer of data of 256 bytes per frame by using the TxLR set mode (00-FFH) is described as a µPD7201A send operation example. Since nonvector mode is used in interrupt processing, the jump destination address when an interrupt occurs is fixed to address 3CH and a single interrupt handling routine is used.

The interrupt handling routine first reads the SR2B vector register and determines the interrupt source, then executes appropriate processing.

The basic program configuration is explained below:

(1) Main program

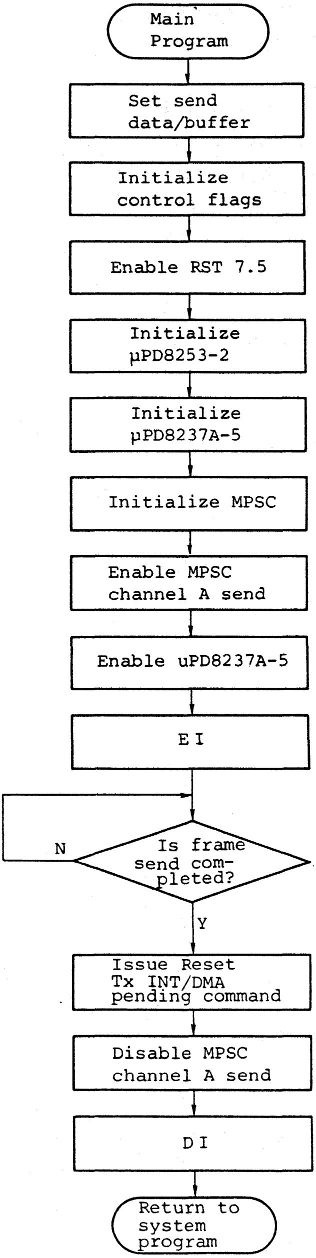
The main program sets the transmit buffer, initializes I/O devices (µPD8253-2, µPD8237A-5) and the µPD7201A, manages the send state, etc. After send starts, the program waits for the All Sent flag (which is set by the interrupt handling program) to be set and checks that the flag is set before disabling send.

(2) Interrupt handling program

Since the RST7.5 pin is used as an interrupt input pin, the µPD8085AH jumps to address 3CH when a µPD7201A interrupt request occurs. Using this address, control is passed to the actual interrupt handling routine (starting at address 1000H).

Only E/S interrupts are interrupt sources in the program example (interrupts caused by other sources and disabled). The E/S interrupt sources are Tx Underrun/EOM and All Sent only. Thus, the interrupt handling program checks only these two types of interrupt sources. The program checks to see if any of the states occurs. When the state occurs, it sets the corresponding control flag. The flag is used to manage the µPD7201A send operation state in the main program.

Figure 5.10 shows the send operation processing flow. Figure 5.11 shows actual program examples.



... Start at address 100H

... Data of 00-FFH is set

... Control flags used in the program are initialized.

... Counter 0 is set for use at a rate of 9600 bps.

... Channel 1 is set (number of transfer data words, send data, and transmit buffer address are set).

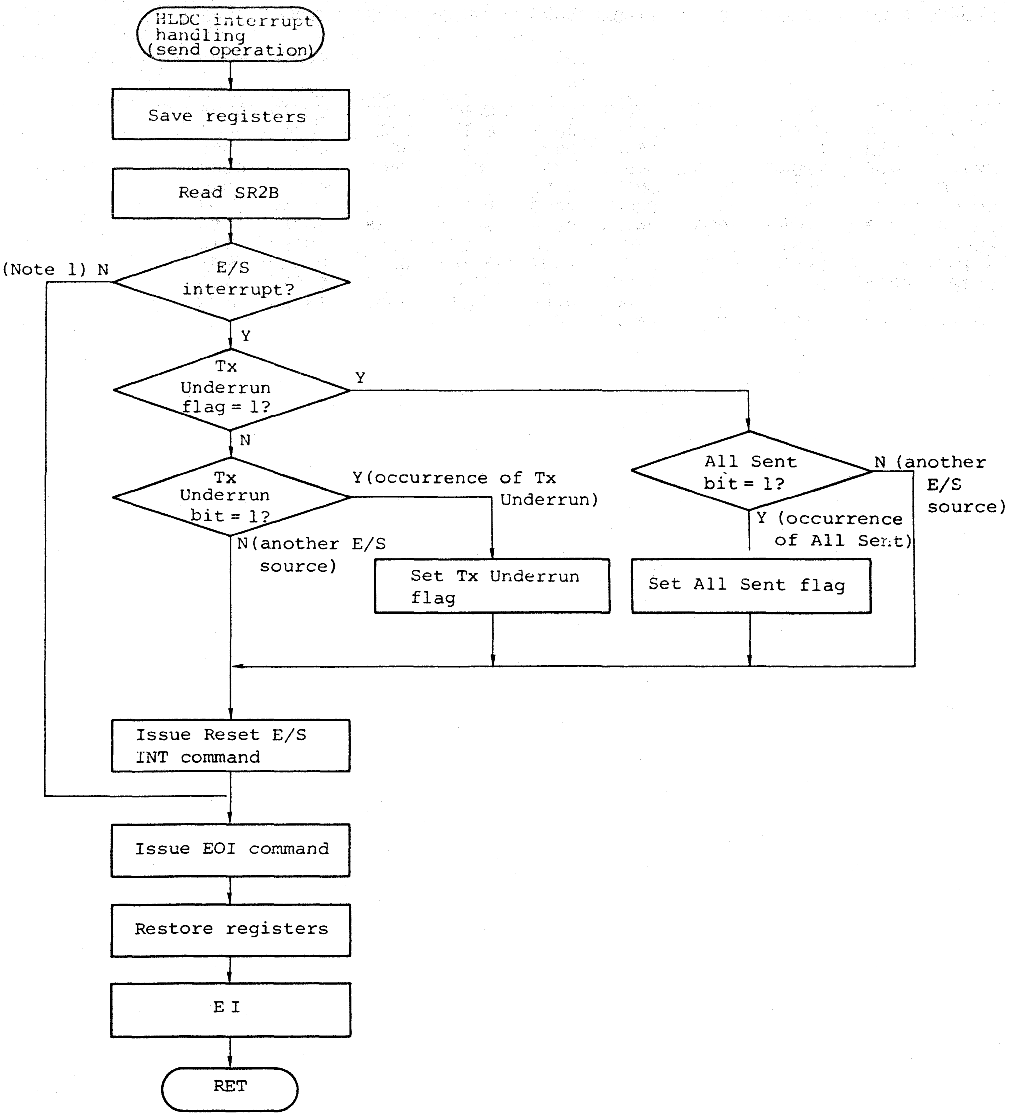
... Basic operation mode is set.

... DMA operation is enabled

... Wait for all Sent interrupt to occur.

... When TxLR Set mode is used, this command must be issued to set Tx Disable.

Figure 5.10 (a) HDLC send operation main program



Note 1: This path is not actually passed through.

Figure 5.10 (b) HDLC send operation Interrupt handling program

```
SYMBOL  ADRS    SYMBOL  ADRS    SYMBOL  ADRS    SYMBOL  ADRS    SYMBOL  ADRS


ALSFG   5B01    ALSNT   1027    BADR1   5B04    BCWC1   5B06    CH1ST   012D
CHACT   00C1    CHADT   00C0    CHBCT   00C3    CHBDT   00C2    CHNRS   0018
CKEND   01AA    CKES    100F    CR1A    004B    CR1B    0004    CR2A    0043
CR2B    0040    CR3A    00C0    CR4A    0020    CR5A    0061    CR6A    0001
CR7A    007E    DM0BC   00D0    DM0CW   00D1    DM1BC   00D2    DM1CW   00D3
DM2BC   00D4    DM2CW   00D5    DM3BC   00D6    DM3CW   00D7    DMCBP   00DC
DMCMR   00DE    DMCTR   00D8    DMMCL   00DD    DMSMR   00DA    DMWAM   00DF
DMWMD   00DB    DMWRR   00D9    ENIRX   0020    EOI     0038    EOIA    103A
ERRES   0030    ERRTA   1036    ESRES   0010    HDLCT   0100    INTEN   0112
NVINT   1000    PEND    0028    PITC0   00C4    PITCT   00C7    RXCRC   0040
SABRT   0008    STACK   6000    STCHA   0165    STDMA   0122    STPIT   0115
STTXD   0104    SYSTM   2000    TXBUF   4000    TXCRC   0080    TXLRA   5B02
UNDEM   00C0    UNDFG   5B00    UNDRN   1019
```

Figure 5.11 HDLC send control program example 1

```
   UCOM-85 ASSEMBLE LIST    (                                       ) PAGE 0001

E STNO ADRS OBJECT M   SOURCE STATEMENTS


  0001                 ;********************************************
  0002                 ;* UPD7201A HDLC TX OPERATION SAMPLE PROGRAM *
  0003                 ;*         TX : TXDA -->                     *
  0004                 ;********************************************
  0005                 ;
  0006                 ;***************
  0007                 ;* I/O ADDRESS *
  0008                 ;***************
  0009                 ;
  0010 00C0            CHADT    EQU     0C0H            ;7201A CH-A DATA
  0011 00C1            CHACT    EQU     0C1H            ;7201A CH-A CONTRO
  0012 00C2            CHBDT    EQU     0C2H            ;7201A CH-B DATA
  0013 00C3            CHBCT    EQU     0C3H            ;7201A CH-B CONTRO
  0014                 ;
  0015 00C4            PITC0    EQU     0C4H            ;8253 CNT #0
  0016 00C7            PITCT    EQU     0C7H            ;8253 CONTROL
  0017                 ;
  0018 00D0            DM0BC    EQU     0D0H            ;8237 BASE/CURAD
  0019 00D1            DM0CW    EQU     0D1H            ;8237 BASE/CWC
  0020 00D2            DM1BC    EQU     0D2H            ;8237 BASE/CURAD
  0021 00D3            DM1CW    EQU     0D3H            ;8237 BASE/CWC
  0022 00D4            DM2BC    EQU     0D4H            ;8237 BASE/CURAD
  0023 00D5            DM2CW    EQU     0D5H            ;8237 BASE/CWC
  0024 00D6            DM3BC    EQU     0D6H            ;8237 BASE/CURAD
  0025 00D7            DM3CW    EQU     0D7H            ;8237 BASE/CWC
  0026 00D8            DMCTR    EQU     0D8H            ;8237 STS/CMD R
  0027 00D9            DMWRR    EQU     0D9H            ;8237 WR REQ R
  0028 00DA            DMSMR    EQU     0DAH            ;8237 WR SIG MRR
  0029 00DB            DMWMD    EQU     0DBH            ;8237 WR MODE R
  0030 00DC            DMCBP    EQU     0DCH            ;8237 CL BYTP FF
  0031 00DD            DMMCL    EQU     0DDH            ;8237 RDMCL.MSCL
  0032 00DE            DMCMR    EQU     0DEH            ;8237 CL MSK R
  0033 00DF            DMWAM    EQU     0DFH            ;8237 WR ALMS RB
  0034                 ;
  0035 2000            SYSTM    EQU     2000H           ;SYSTEM PROGRAM
  0036                 ;
  0037                 ;****************
  0038                 ;* MPSC COMMAND *
  0039                 ;****************
  0040                 ;
  0041 0008            SABRT    EQU     8H              ;SEND ABORT
  0042 0010            ESRES    EQU     10H             ;RESET E/S INT
  0043 0018            CHNRS    EQU     18H             ;CHANNEL RESET
  0044 0020            ENIRX    EQU     20H             ;EN INT NEXT CHAR
  0045 0028            PEND     EQU     28H             ;RES TX PENDING
  0046 0030            ERRES    EQU     30H             ;ERROR RESET
  0047 0038            EOI      EQU     38H             ;END OF INTERRUPT
  0048                 ;
  0049 0040            RXCRC    EQU     40H             ;INIT RX CRC CAL
  0050 0080            TXCRC    EQU     80H             ;INIT TX CRC CAL
  0051 00C0            UNDEM    EQU     0C0H            ;RESET TX UND/EOM
  0052                 ;
```

Figure 5.11 HDLC send control program example 2

```
   UCOM-85 ASSEMBLE LIST    (                                    ) PAGE 0002

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0053                 ;***********************************
  0054                 ;* MPSC CONTROL REGISTER PARAMETER *
  0055                 ;***********************************
  0056                 ;
  0057 004B            CR1A    EQU     4BH             ;TXLR SET
  0058                                                 ;FIRST RX INT
  0059                                                 ;TX DMA ENABLE
  0060                                                 ;E/S INT ENABLE
  0061 0043            CR2A    EQU     43H             ;RX INT MASK
  0062                                                 ;NON-VECTORED
  0063                                                 ;85-1 VECTORED
  0064                                                 ;BOTH CH. DMA-2
  0065 00C0            CR3A    EQU     0C0H            ;8 BIT/CHAR
  0066 0020            CR4A    EQU     20H             ;HDLC MODE
  0067 0061            CR5A    EQU     61H             ;8 BIT/CHAR
  0068                                                 ;TX ENABLE
  0069                                                 ;CCITT-0
  0070                                                 ;TX CRC ENABLE
  0071 0001            CR6A    EQU     01H             ;SECONDARY ADDRESS
  0072 007E            CR7A    EQU     7EH             ;FLAG
  0073                 ;
  0074 0004            CR1B    EQU     4H              ;MODIFIED VECTOR
  0075 0040            CR2B    EQU     40H             ;VECTOR(010***00)
  0076                 ;
  0077                 ;********************************************
  0078                 ;* INTERRUPT JUMP TABLE (NON-VECTORED MODE) *
  0079                 ;********************************************
  0080                 ;
  0081 0000                    ORG     3CH             ;RST 7.5 ADRS
  0082                 ;
  0083 003C C30010             JMP     NVINT           ;JUMP NON-VCT INT
  0084                 ;
  0085                 ;**********************************
  0086                 ;* PERIPHERAL INITIALIZE SEQUENCE *
  0087                 ;**********************************
  0088                 ;
  0089 003F                    ORG     100H
  0090                 ;
  0091 0100 310060     HDLCT:  LXI     SP.STACK
  0092 0103 AF                 XRA     A
  0093 0104 210040     STTXD:  LXI     H.TXBUF         ;SET TX DATA
  0094 0107 75                 MOV     M.L
  0095 0108 2C                 INR     L
  0096 0109 C20701             JNZ     $-2
  0097 010C 32005B             STA     UNDFG           ;CLEAR FLAGS
  0098 010F 32015B             STA     ALSFG
  0099 0112 3E1B       INTEN:  MVI     A.1BH           ;RST 7.5 ENABLE
  0100 0114 30                 SIM
  0101                 ;
  0102                 ;
  0103                 ;
  0104                 ;
  0105                 ;
```

Figure 5.11 HDLC send control program example 3

```
   UCOM-85 ASSEMBLE LIST    (                                    ) PAGE 0003

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0106                ; 8253-2 INITIALIZE
  0107                ;
  0108 0115 210001    STPIT:  LXI     H,100H          ;CNT #0,9600 BPS
  0109 0118 3E36              MVI     A,36H           ;#0 MODE 2
  0110 011A D3C7              OUT     PITCT
  0111 011C 7D                MOV     A,L
  0112 011D D3C4              OUT     PITC0
  0113 011F 7C                MOV     A,H
  0114 0120 D3C4              OUT     PITC0
  0115                ;
  0116                ; 8237A-5 INITIALIZE
  0117                ;
  0118 0122 AF        STDMA:  XRA     A
  0119 0123 D3DD              OUT     DMMCL           ;MASTER CLEAR
  0120 0125 3E04              MVI     A,4H            ;DMAC DISABLE
  0121 0127 D3D8              OUT     DMCTR
  0122 0129 3EFD              MVI     A,0FDH          ;CH-1 ENABLE
  0123 012B D3DF              OUT     DMWAM
  0124                ;
  0125 012D AF        CH1ST:  XRA     A               ;CLEAR BYTE PTR F/
  0126 012E D3DC              OUT     DMCBP
  0127 0130 2A045B            LHLD    BADR1           ;CH-1 BADRS
  0128 0133 7D                MOV     A,L             ;CH-1 BADRS-L
  0129 0134 D3D2              OUT     DM1BC
  0130 0136 7C                MOV     A,H             ;CH-1 BADRS-H
  0131 0137 D3D2              OUT     DM1BC
  0132 0139 AF                XRA     A               ;CLEAR BYTE PTR F/
  0133 013A D3DC              OUT     DMCBP
  0134 013C 2A065B            LHLD    BCWC1           ;CH-1 TRNSFR CNT
  0135 013F 7D                MOV     A,L
  0136 0140 D3D3              OUT     DM1CW
  0137 0142 7C                MOV     A,H
  0138 0143 D3D3              OUT     DM1CW
  0139 0145 3E49              MVI     A,49H           ;DRQ1 MODE
  0140 0147 D3DB              OUT     DMWMD
  0141                ;
  0142                ;
  0143                ;***************************
  0144                ;* MPSC INITIALIZE SEQUENCE *
  0145                ;***************************
  0146                ;
  0147                ; COMMON REGISTER INITIALIZE
  0148                ;
  0149 0149 3E18              MVI     A,CHNRS         ;CHANNEL RESET
  0150 014B D3C1              OUT     CHACT
  0151 014D D3C3              OUT     CHBCT
  0152 014F 3E02              MVI     A,2H            ;PTR 2
  0153 0151 D3C1              OUT     CHACT
  0154 0153 D3C3              OUT     CHBCT
  0155 0155 3E43              MVI     A,CR2A          ;BUS IF MODE
  0156 0157 D3C1              OUT     CHACT
  0157 0159 3E40              MVI     A,CR2B          ;INERRUPT VECTOR
  0158 015B D3C3              OUT     CHBCT
```

Figure 5.11 HDLC send control program example 4

```
   UCOM-85 ASSEMBLE LIST    (                                        ) PAGE 0004

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0159 015D 3E01                MVI      A.1H                 ;PTR 1B
  0160 015F D3C3                OUT      CHBCT
  0161 0161 3E04                MVI      A.CR1B               ;MODIFIED VECTOR
  0162 0163 D3C3                OUT      CHBCT
  0163                 ;
  0164                 ; CH-A OPERATION MODE
  0165                 ;
  0166 0165 3E04       STCHA:   MVI      A.4H                 ;PTR 4A
  0167 0167 D3C1                OUT      CHACT
  0168 0169 3E20                MVI      A.CR4A               ;CH-A MODE
  0169 016B D3C1                OUT      CHACT
  0170 016D 3E03                MVI      A.3H                 ;PTR 3A
  0171 016F D3C1                OUT      CHACT
  0172 0171 3EC0                MVI      A.CR3A               ;RX PARAMETER
  0173 0173 D3C1                OUT      CHACT
  0174 0175 3E05                MVI      A.5H                 ;PTR 5A
  0175 0177 D3C1                OUT      CHACT
  0176 0179 3E61                MVI      A.CR5A               ;TX PARAMETER
  0177 017B D3C1                OUT      CHACT
  0178 017D 3E06                MVI      A.6H                 ;PTR 6A
  0179 017F D3C1                OUT      CHACT
  0180 0181 3E01                MVI      A.CR6A               ;SECONDARY ADDRESS
  0181 0183 D3C1                OUT      CHACT
  0182 0185 3E07                MVI      A.7H                 ;PTR 7A
  0183 0187 D3C1                OUT      CHACT
  0184 0189 3E7E                MVI      A.CR7A               ;FLAG
  0185 018B D3C1                OUT      CHACT
  0186 018D 3E11                MVI      A.11H                ;PTR 1A.RS ES INT
  0187 018F D3C1                OUT      CHACT
  0188 0191 3E4B                MVI      A.CR1A               ;INT/DMA MODE
  0189 0193 D3C1                OUT      CHACT
  0190 0195 2A025B              LHLD     TXLRA                ;TXLRA
  0191 0198 7D                  MOV      A,L                  ;TXLRA-L
  0192 0199 D3C1                OUT      CHACT
  0193 019B 7C                  MOV      A.H                  ;TXLRA-H
  0194 019C D3C1                OUT      CHACT
  0195 019E 3E05                MVI      A,5                  ;PTR 5A
  0196 01A0 D3C1                OUT      CHACT
  0197 01A2 3E69                MVI      A.69H                ;TX ENABLE
  0198 01A4 D3C1                OUT      CHACT
  0199 01A6 AF                  XRA      A
  0200 01A7 D3D8                OUT      DMCTR                ;DMAC ENABLE
  0201 01A9 FB                  EI
  0202                 ;
  0203 01AA 3A015B     CKEND:   LDA      ALSFG                ;CHECK ALSNT FLG
  0204 01AD 0F                  RRC
  0205 01AE D2AA01              JNC      $-4
  0206 01B1 3E28                MVI      A.PEND               ;RES TX PENDING
  0207 01B3 D3C1                OUT      CHACT
  0208 01B5 3E05                MVI      A.5                  ;PTR 5A
  0209 01B7 D3C1                OUT      CHACT
  0210 01B9 3E61                MVI      A.61H                ;TX DISABLE
  0211 01BB D3C1                OUT      CHACT
```

Figure 5.11 HDLC send control program example 5

```
  UCOM-85 ASSEMBLE LIST    (                                    ) PAGE 0005

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0212 01BD F3                  DI
  0213 01BE C30020              JMP     SYSTM
  0214                 ;
  0215                 ;**********************************************
  0216                 ;* INTERRUPT PROCESSING ROUTINTE (NON-VECTORED MOD
  0217                 ;**********************************************
  0218                 ;
  0219 01C1                     ORG     1000H
  0220                 ;
  0221 1000 F5         NVINT:   PUSH    PSW
  0222 1001 3E02                MVI     A.2H            ;PTR 2B
  0223 1003 D3C3                OUT     CHBCT
  0224 1005 DBC3                IN      CHBCT           ;READ SR2B
  0225 1007 FE54                CPI     54H             ;CH-A E/S INT ?
  0226 1009 CA0F10              JZ      CKES            ;IF YES.JUMP
  0227 100C C33A10              JMP     EOIA
  0228                 ;
  0229 100F DBC1       CKES:    IN      CHACT           ;READ SR0A
  0230 1011 47                  MOV     B.A
  0231 1012 3A005B              LDA     UNDFG           ;GET UNDRN FLAG
  0232 1015 0F                  RRC                     ;CHECK IT
  0233 1016 DA2710              JC      ALSNT
  0234                 ;
  0235 1019 78         UNDRN:   MOV     A.B
  0236 101A 07                  RLC
  0237 101B 07                  RLC
  0238 101C D23610              JNC     ERRTA           ;TX UND BIT=1?
  0239 101F 3E01                MVI     A.1             ;YES
  0240 1021 32005B              STA     UNDFG           ;SET UNDRN ELG
  0241 1024 C33610              JMP     ERRTA
  0242                 ;
  0243 1027 3E01       ALSNT:   MVI     A.1             ;PTR 1A
  0244 1029 D3C1                OUT     CHACT
  0245 102B DBC1                IN      CHACT           ;READ SR1A
  0246 102D 0F                  RRC
  0247 102E D23610              JNC     ERRTA           ;ALL SENT BIT=1?
  0248 1031 3E01                MVI     A.1             ;SET ALSNT FLG
  0249 1033 32015B              STA     ALSFG
  0250                 ;
  0251 1036 3E10       ERRTA:   MVI     A.ESRES
  0252 1038 D3C1                OUT     CHACT
  0253 103A 3E38       EOIA:    MVI     A.EOI           ;EOI TO MPSC
  0254 103C D3C1                OUT     CHACT
  0255 103E F1                  POP     PSW
  0256 103F FB                  EI
  0257 1040 C9                  RET
  0258                 ;
  0259                 ;
  0260                 ;
  0261                 ;
  0262                 ;
  0263                 ;
  0264                 ;
```

Figure 5.11 HDLC send control program example 6

```
   UCOM-85 ASSEMBLE LIST    (                                    ) PAGE 0006

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0265                ;***********************
  0266                ;* TX DATA BUFFER AREA *
  0267                ;***********************
  0268                ;
  0269 1041                   ORG     4000H
  0270                ;
  0271 4000           TXBUF:  DS      1024
  0272                ;
  0273                ;********************
  0274                ;* STATUS FLAG AREA *
  0275                ;********************
  0276                ;
  0277 4400                   ORG     5B00H
  0278                ;
  0279 5B00           UNDFG:  DS      1               ;TX UND FLAG
  0280 5B01           ALSFG:  DS      1               ;ALL SENT FLAG
  0281                ;
  0282                ;**********************
  0283                ;* I/O PARAMETER AREA *
  0284                ;**********************
  0285                ;
  0286 5B02 0001      TXLRA:  DW      100H            ;TXLR
  0287                ;
  0288 5B04 0040      BADR1:  DW      4000H           ;CH-1 DMA ADRS
  0289 5B06 FF00      BCWC1:  DW      0FFH            ;CH-1 WORD COUNT
  0290                ;
  0291                ;**************
  0292                ;* STACK AREA *
  0293                ;**************
  0294                ;
  0295 5B08                   ORG     6000H
  0296                ;
  0297                STACK:
  0298 0000                   END
```

Figure 5.11 HDLC send control program example 7

5.3.2 Receive operation

DMA transfer of data of 256 bytes per frame (00-FFH) is described as a receive operation example. For the Rx INT mode, the Int on First Rx Character mode (CR1A : D4, D3 = 0, 1) is selected, and the First Rx INT Mask mode (CR2A : D6 = 1) is used for operation. Thus, all receive data (containing CRC characters) is transferred via DMA. Interrupt processing is performed in nonvector mode as in send operation.

The basic program configuration is explained below:

(1) Main program

The main program sets the receive buffer, initializes I/O devices (µPD8253-2, µPD8237A-5) and the µPD7201A, manages the receive state, etc. When receive operation starts, the Enter Hunt bit is set to 1 for flag detection. After flag detection, the program waits for the receive completion flag (which is set by the interrupt handling program when the End of Frame state occurs) to be set, and checks that the flag is set before disabling receive.

(2) Interrupt handling program

The interrupt sources handled by the interrupt handling program are Special Rx Condition interrupts only (interrupts caused by other sources are disabled). However, a check routine to see if the interrupt is an E/S interrupt is contained in the program for convenience.

In interrupt processing, End of Frame, CRC Error, and Overrun Error and checked. When any of these occurs, the control flag indicating the occurrence of the event is set. The control flag is used to manage the µPD7201A receive operation state in the main program.

Figure 5.12 shows the receive operation processing flow. Figure 5.13 shows actual program examples.

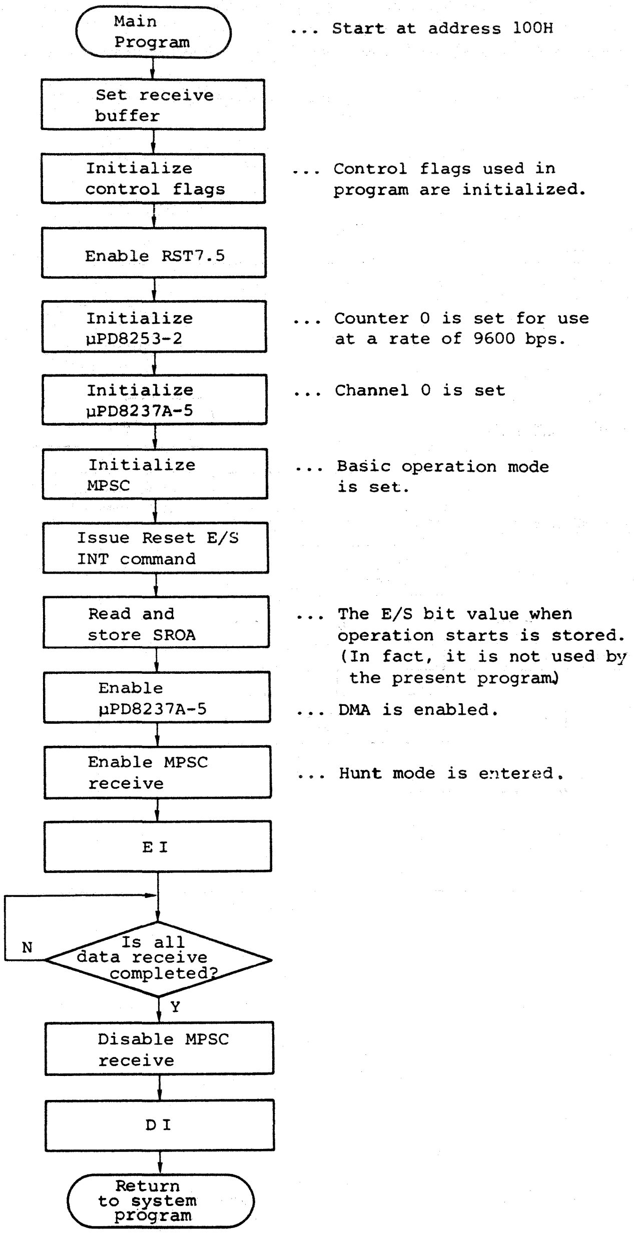
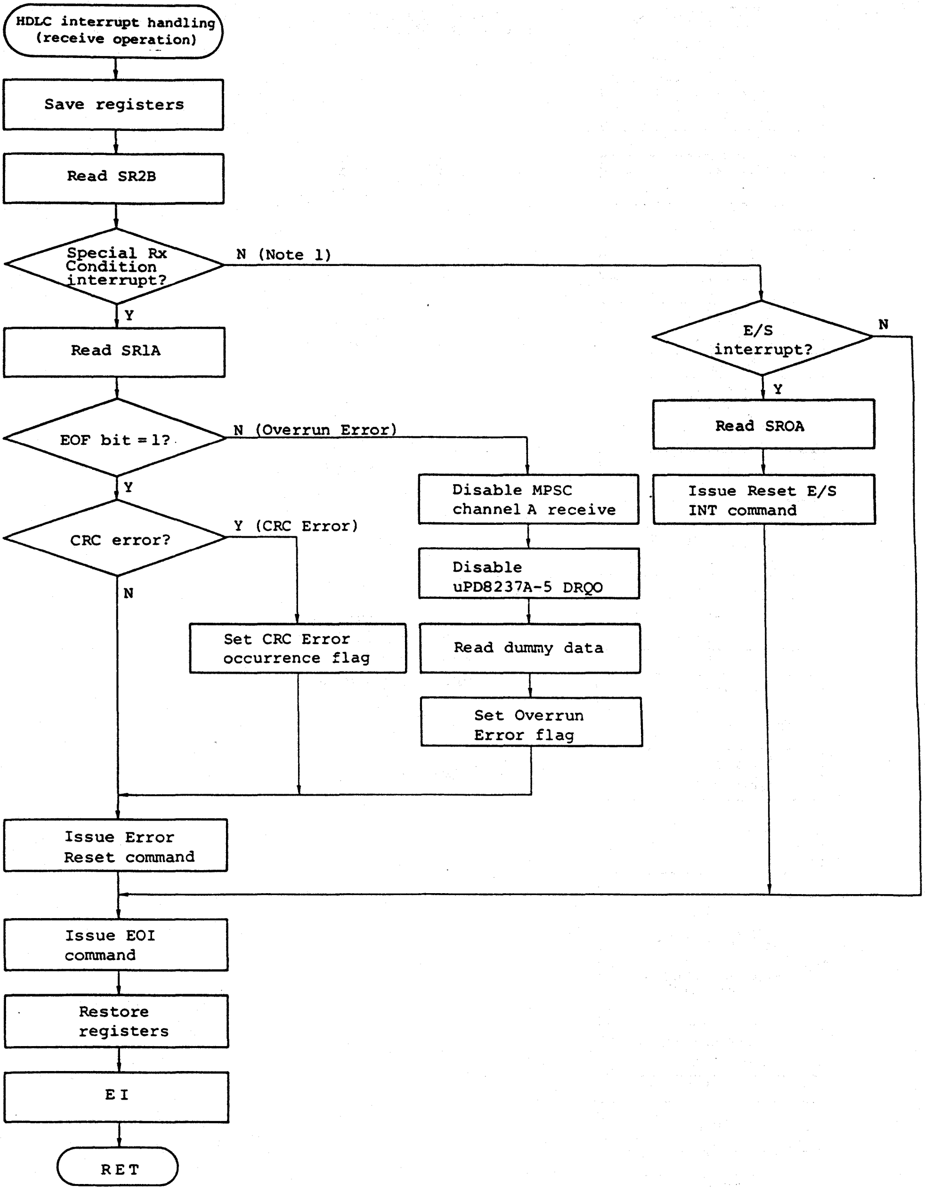


Figure 5.12 (a) HDLC receive operation main program



Note 1: This path is not actually passed through in the present program example.

Figure 5.12 (b) HDLC receive operation interrupt handling

```
SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS    SYMBOL ADRS


BADR0  5B06    BCWC0  5B08    CH0ST  0131    CHACT  00C1    CHADT  00C0
CHBCT  00C3    CHBDT  00C2    CHNRS  0018    CR1A   0048    CR1B   0004
CR2A   0043    CR2B   0040    CR3A   00C8    CR4A   0020    CR5A   0060
CR6A   0001    CR7A   007E    CRCFA  5B01    CREND  01B7    DM0BC  00D0
DM0CW  00D1    DM1BC  00D2    DM1CW  00D3    DM2BC  00D4    DM2CW  00D5
DM3BC  00D6    DM3CW  00D7    DMCBP  00DC    DMCMR  00DE    DMCTR  00D8
DMMCL  00DD    DMSMR  00DA    DMWAM  00DF    DMWMD  00DB    DMWRR  00D9
ENIRX  0020    EOI    0038    EOIA   105B    ERESA  104F    ERRES  0030
ESCHA  1016    ESRES  0010    HDLCR  0100    INTEN  0116    NVINT  1000
OVREA  1038    OVRFA  5B02    PEND   0028    PITC0  00C4    PITCT  00C7
RCVDT  1007    RENDA  5B03    RXBFC  0104    RXBUF  4400    RXCRC  0040
RXOPR  01A6    SABRT  0008    SR0AS  5B00    SRCDA  1022    STACK  6000
STCHA  0169    STDMA  0126    STPIT  0119    SYSTM  2000    TXCRC  0080
TXLRA  5B04    UNDEM  00C0
```

Figure 5.13 HDLC receive control program example 1

```
   UCOM-85 ASSEMBLE LIST    (                                  ) PAGE 0001

E STNO ADRS OBJECT M   SOURCE STATEMENTS


  0001                 ;**********************************************
  0002                 :* UPD7201A HDLC RX OPERATION SAMPLE PROGRAM *
  0003                 :*          RX : RXDA <--                    *
  0004                 ;**********************************************
  0005                 :
  0006                 ;***************
  0007                 :* I/O ADDRESS *
  0008                 ;***************
  0009                 ;
  0010 00C0            CHADT    EQU      0C0H             :7201A CH-A DATA
  0011 00C1            CHACT    EQU      0C1H             :7201A CH-A CNTRL
  0012 00C2            CHBDT    EQU      0C2H             :7201A CH-B DATA
  0013 00C3            CHBCT    EQU      0C3H             :7201A CH-B CNTRL
  0014                 :
  0015 00C4            PITC0    EQU      0C4H             :8253 CNT #0
  0016 00C7            PITCT    EQU      0C7H             :8253 CONTROL
  0017                 :
  0018 00D0            DM0BC    EQU      0D0H             :8237 BASE/CURAD
  0019 00D1            DM0CW    EQU      0D1H             :8237 BASE/CWC
  0020 00D2            DM1BC    EQU      0D2H             :8237 BASE/CURAD
  0021 00D3            DM1CW    EQU      0D3H             :8237 BASE/CWC
  0022 00D4            DM2BC    EQU      0D4H             :8237 BASE/CURAD
  0023 00D5            DM2CW    EQU      0D5H             :8237 BASE/CWC
  0024 00D6            DM3BC    EQU      0D6H             :8237 BASE/CURAD
  0025 00D7            DM3CW    EQU      0D7H             :8237 BASE/CWC
  0026 00D8            DMCTR    EQU      0D8H             :8237 STS/CMD R
  0027 00D9            DMWRR    EQU      0D9H             :8237 WR REQ R
  0028 00DA            DMSMR    EQU      0DAH             :8237 WR SIG MSR
  0029 00DB            DMWMD    EQU      0DBH             :8237 WR MODE R
  0030 00DC            DMCBP    EQU      0DCH             :8237 CL BYTP FF
  0031 00DD            DMMCL    EQU      0DDH             :8237 RDMCL.MSCL
  0032 00DE            DMCMR    EQU      0DEH             ;8237 CL MSK R
  0033 00DF            DMWAM    EQU      0DFH             :8237 WR ALMS RB
  0034                 :
  0035 2000            SYSTM    EQU      2000H            :SYSTEM PROGRAM
  0036                 :
  0037                 ;****************
  0038                 :* MPSC COMMAND *
  0039                 ;****************
  0040                 :
  0041 0008            SABRT    EQU      8H               :SEND ABORT
  0042 0010            ESRES    EQU      10H              :RESET E/S INT
  0043 0018            CHNRS    EQU      18H              :CHANNEL RESET
  0044 0020            ENIRX    EQU      20H              :EN INT NEXT CHAR
  0045 0028            PEND     EQU      28H              :RES TX PENDING
  0046 0030            ERRES    EQU      30H              :ERROR RESET
  0047 0038            EOI      EQU      38H              :END OF INTERRUPT
  0048                 :
  0049 0040            RXCRC    EQU      40H              :INIT RX CRC CAL
  0050 0080            TXCRC    EQU      80H              :INIT TX CRC CAL
  0051 00C0            UNDEM    EQU      0C0H             :RESET TX UND/EOM
  0052                 ;
```

Figure 5.13 HDLC receive control program example 2

```
   UCOM-85 ASSEMBLE LIST    (                                     ) PAGE 0002

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0053                 ;***********************************
  0054                 ;* MPSC CONTROL REGISTER PARAMETER *
  0055                 ;***********************************
  0056                 ;
  0057 0048            CR1A    EQU     48H              ;TXLR SET
  0058                                                  ;FIRST RX INT
  0059                                                  ;TX DMA DISABLE
  0060                                                  ;E/S INT DISABLE
  0061 0043            CR2A    EQU     43H              ;RX INT MASK
  0062                                                  ;NON-VECTORED
  0063                                                  ;85-1 VECTORED
  0064                                                  ;BOTH CH. DMA-2
  0065 00C8            CR3A    EQU     0C8H             ;8 BIT/CHAR
  0066                                                  ;RX CRC ENABLE
  0067 0020            CR4A    EQU     20H              ;HDLC MODE
  0068 0060            CR5A    EQU     60H              ;8 BIT/CHAR
  0069                                                  ;CCITT-0
  0070                                                  ;TX CRC DISABLE
  0071 0001            CR6A    EQU     01H              ;SECONDARY ADDRESS
  0072 007E            CR7A    EQU     7EH              ;FLAG
  0073                 ;
  0074 0004            CR1B    EQU     4H               ;MODIFIED VECTOR
  0075 0040            CR2B    EQU     40H              ;VECTOR(010***00)
  0076                 ;
  0077                 ;*********************************************
  0078                 ;* INTERRUPT JUMP TABLE (NON-VECTORED MODE) *
  0079                 ;*********************************************
  0080                 ;
  0081 0000                    ORG     3CH              ;RST 7.5 ADRS
  0082                 ;
  0083 003C C30010             JMP     NVINT            ;JUMP NON-VCT INT
  0084                 ;
  0085                 ;***********************************
  0086                 ;* PERIPHERAL INITIALIZE SEQUENCE *
  0087                 ;***********************************
  0088                 ;
  0089 003F                    ORG     100H
  0090                 ;
  0091 0100 310060     HDLCR:  LXI     SP.STACK
  0092 0103 AF                 XRA     A
  0093 0104 210044     RXBFC:  LXI     H.RXBUF          ;CLR MPSC RX BUF
  0094 0107 77                 MOV     M.A
  0095 0108 2C                 INR     L
  0096 0109 C20701             JNZ     $-2
  0097 010C AF                 XRA     A                ;CLEAR FLAGS
  0098 010D 32015B             STA     CRCFA
  0099 0110 32025B             STA     OVRFA
  0100 0113 32035B             STA     RENDA
  0101 0116 3E1B       INTEN:  MVI     A.1BH            ;RST 7.5 ENABLE
  0102 0118 30                 SIM
  0103                 ;
  0104                 ;
  0105                 ;
```

Figure 5.13 HDLC receive control program example 3

```
  UCOM-85 ASSEMBLE LIST    (                                    ) PAGE 0003

E STNO ADRS OBJECT M   SOURCE STATEMENTS


  0106                 ; 8253-2 INITIALIZE
  0107                 ;
  0108 0119 210001     STPIT:  LXI     H.100H            ;CNT #0.9600 BPS
  0109 011C 3E36               MVI     A.36H             ;#0 MODE 2
  0110 011E D3C7               OUT     PITCT
  0111 0120 7D                 MOV     A.L
  0112 0121 D3C4               OUT     PITC0
  0113 0123 7C                 MOV     A.H
  0114 0124 D3C4               OUT     PITC0
  0115                 ;
  0116                 ; 8237A-5 INITIALIZE
  0117                 ;
  0118 0126 AF         STDMA:  XRA     A
  0119 0127 D3DD               OUT     DMMCL             ;MASTER CLEAR
  0120 0129 3E04               MVI     A,4H              ;DMAC DISABLE
  0121 012B D3D8               OUT     DMCTR
  0122 012D 3EFE               MVI     A,0FEH            ;CH-0 ENABLE
  0123 012F D3DF               OUT     DMWAM
  0124                 ;
  0125 0131 AF         CHOST:  XRA     A                 ;CLR BYTE PTR F/F
  0126 0132 D3DC               OUT     DMCBP
  0127 0134 2A065B             LHLD    BADR0             ;CH-0 BADRS
  0128 0137 7D                 MOV     A,L               ;CH-0 BADRS-L
  0129 0138 D3D0               OUT     DM0BC
  0130 013A 7C                 MOV     A.H               ;CH-0 BADRS-H
  0131 013B D3D0               OUT     DM0BC
  0132 013D AF                 XRA     A                 ;CLR BYTE PTR F/F
  0133 013E D3DC               OUT     DMCBP
  0134 0140 2A085B             LHLD    BCWC0             ;CH-0 TRNSFR CNT
  0135 0143 7D                 MOV     A.L
  0136 0144 D3D1               OUT     DM0CW
  0137 0146 7C                 MOV     A.H
  0138 0147 D3D1               OUT     DM0CW
  0139 0149 3E44               MVI     A.44H             ;DRQ0 MODE
  0140 014B D3DB               OUT     DMWMD
  0141                 ;
  0142                 ;***************************
  0143                 ;* MPSC INITIALIZE SEQUENCE *
  0144                 ;***************************
  0145                 ;
  0146                 ; COMMON REGISTER INITIALIZE
  0147                 ;
  0148 014D 3E18               MVI     A.CHNRS           ;CHANNEL RESET
  0149 014F D3C1               OUT     CHACT
  0150 0151 D3C3               OUT     CHBCT
  0151 0153 3E02               MVI     A.2H              ;PTR 2
  0152 0155 D3C1               OUT     CHACT
  0153 0157 D3C3               OUT     CHBCT
  0154 0159 3E43               MVI     A.CR2A            ;BUS IF MODE
  0155 015B D3C1               OUT     CHACT
  0156 015D 3E40               MVI     A.CR2B            ;INERRUPT VECTOR
  0157 015F D3C3               OUT     CHBCT
  0158                 ;
```

Figure 5.13 HDLC receive control program example 4

```
   UCOM-85 ASSEMBLE LIST    (                                        ) PAGE 0004

E STNO ADRS OBJECT M    SOURCE STATEMENTS

  0159 0161 3E01                MVI     A,1H            ;PTR 1B
  0160 0163 D3C3                OUT     CHBCT
  0161 0165 3E04                MVI     A,CR1B          ;MODIFIED VECTOR
  0162 0167 D3C3                OUT     CHBCT
  0163                  ;
  0164                  ; CH-A OPERATION MODE
  0165                  ;
  0166 0169 3E04        STCHA:  MVI     A,4H            ;PTR 4A
  0167 016B D3C1                OUT     CHACT
  0168 016D 3E20                MVI     A,CR4A          ;CH-A MODE
  0169 016F D3C1                OUT     CHACT
  0170 0171 3E05                MVI     A,5H            ;PTR 5A
  0171 0173 D3C1                OUT     CHACT
  0172 0175 3E60                MVI     A,CR5A          ;TX PARAMETER
  0173 0177 D3C1                OUT     CHACT
  0174 0179 3E06                MVI     A,6H            ;PTR 6A
  0175 017B D3C1                OUT     CHACT
  0176 017D 3E01                MVI     A,CR6A          ;2ND ADDRESS
  0177 017F D3C1                OUT     CHACT
  0178 0181 3E07                MVI     A,7H            ;PTR 7A
  0179 0183 D3C1                OUT     CHACT
  0180 0185 3E7E                MVI     A,CR7A          ;FLAG
  0181 0187 D3C1                OUT     CHACT
  0182 0189 3E11                MVI     A,11H           ;PTR 1A,RS ES INT
  0183 018B D3C1                OUT     CHACT
  0184 018D 3E48                MVI     A,CR1A          ;INT/DMA MODE
  0185 018F D3C1                OUT     CHACT
  0186 0191 2A045B              LHLD    TXLRA           ;TXLRA
  0187 0194 7D                  MOV     A,L             ;TXLRA-L
  0188 0195 D3C1                OUT     CHACT
  0189 0197 7C                  MOV     A,H             ;TXLRA-H
  0190 0198 D3C1                OUT     CHACT
  0191 019A 3E03                MVI     A,3H            ;PTR 3A
  0192 019C D3C1                OUT     CHACT
  0193 019E 3EC8                MVI     A,CR3A          ;RX PARAMETER
  0194 01A0 D3C1                OUT     CHACT
  0195 01A2 3E10                MVI     A,ESRES         ;RESET E/S INT
  0196 01A4 D3C1                OUT     CHACT
  0197                  ;
  0198 01A6 DBC1        RXOPR:  IN      CHACT           ;READ FIRST SR0A
  0199 01A8 32005B              STA     SR0AS
  0200 01AB AF                  XRA     A
  0201 01AC D3D8                OUT     DMCTR           ;DMAC ENABLE
  0202 01AE 3E03                MVI     A,3H            ;PTR 3A
  0203 01B0 D3C1                OUT     CHACT
  0204 01B2 3ED9                MVI     A,0D9H          ;RXEN,ENTER HUNT
  0205 01B4 D3C1                OUT     CHACT
  0206 01B6 FB                  EI
  0207                  ;
  0208 01B7 3A035B      CREND:  LDA     RENDA           ;CHECK RXEND FLG
  0209 01BA FE01                CPI     1H
  0210 01BC C2B701              JNZ     CREND
  0211                  ;
```

Figure 5.13 HDLC receive control program example 5

```
   UCOM-85 ASSEMBLE LIST    (                                  ) PAGE 0005

E STNO ADRS OBJECT M   SOURCE STATEMENTS


  0212 01BF 3E03              MVI      A.3H            ;PTR 3A
  0213 01C1 D3C1              OUT      CHACT
  0214 01C3 3EC0              MVI      A.0C0H          ;RX DISABLE
  0215 01C5 D3C1              OUT      CHACT
  0216 01C7 F3                DI
  0217                 ;
  0218 01C8 C30020            JMP      SYSTM
  0219                 ;
  0220                 ;
  0221                 ;************************************************
  0222                 ;* INERRUPT PROCESSING ROUTINE (NON-VECTORED MODE)
  0223                 ;************************************************
  0224                 ;
  0225 01CB                   ORG      1000H
  0226                 ;
  0227 1000 F5         NVINT: PUSH     PSW
  0228 1001 3E02              MVI      A.2H            ;PTR 2B
  0229 1003 D3C3              OUT      CHBCT
  0230 1005 DBC3              IN       CHBCT           ;READ SR2B
  0231 1007 E61C       RCVDT: ANI      1CH             ;MASK
  0232 1009 FE1C              CPI      1CH             ;CH-A SRXCND?
  0233 100B CA2210            JZ       SRCDA           ;IF YES.JUMP
  0234 100E FE14              CPI      14H             ;CH-A E/S CHNG?
  0235 1010 CA1610            JZ       ESCHA           ;IF YES.JUMP
  0236 1013 C35B10            JMP      EOIA            ;IF OTHS.JP EOI
  0237 1016 DBC1       ESCHA: IN       CHACT           ;READ SR0A
  0238 1018 32005B            STA      SR0AS           ;STORE NEW SR0A
  0239 101B 3E10              MVI      A.ESRES         ;RESET E/S INT
  0240 101D D3C1              OUT      CHACT
  0241 101F C35B10            JMP      EOIA
  0242 1022 3E01       SRCDA: MVI      A.1H            ;PTR 1A
  0243 1024 D3C1              OUT      CHACT
  0244 1026 DBC1              IN       CHACT           ;READ SR1A
  0245 1028 07                RLC                      ;CHECK EOF BIT
  0246 1029 D23810            JNC      OVREA           ;IF NOT EOF.JMP
  0247 102C 07                RLC                      ;CHECK CRC ERR
  0248 102D D24F10            JNC      ERESA           ;IF CRC OK.JUMP
  0249 1030 3E01              MVI      A.1             ;SET CRC ERR FLG
  0250 1032 32015B            STA      CRCFA           ;STORE CRCE FLG
  0251 1035 C34F10            JMP      ERESA
  0252                 ;
  0253 1038 3E03       OVREA: MVI      A.3H            ;PTR 3A
  0254 103A D3C1              OUT      CHACT
  0255 103C 3EC8              MVI      A.0C8H          ;MPSC RX DISABLE
  0256 103E D3C1              OUT      CHACT
  0257 1040 3E04              MVI      A.4H            ;MASK DMAC CH-0
  0258 1042 D3DA              OUT      DMSMR
  0259 1044 DBC0              IN       CHADT           ;DAMMY READ
  0260 1046 DBC0              IN       CHADT
  0261 1048 DBC0              IN       CHADT
  0262 104A 3E01              MVI      A.1             ;SET OVRNER FLG
  0263 104C 32025B            STA      OVRFA           ;STORE OVRNER FG
  0264                 ;
```

Figure 5.13 HDLC receive control program example 6

```
 UCOM-85 ASSEMBLE LIST    (                                   ) PAGE 0006

E STNO ADRS OBJECT M   SOURCE STATEMENTS

  0265 104F 3E30       ERESA:  MVI      A.ERRES          :ERROR RESET
  0266 1051 D3C1               OUT      CHACT
  0267 1053 3E01               MVI      A.1              :SET RX END FLG
  0268 1055 32035B             STA      RENDA
  0269 1058 C35B10             JMP      EOIA
  0270                 ;
  0271 105B 3E38       EOIA:   MVI      A.EOI            :EOI TO MPSC
  0272 105D D3C1               OUT      CHACT
  0273 105F F1                 POP      PSW
  0274 1060 FB                 EI
  0275 1061 C9                 RET
  0276                 ;
  0277                 ;***********************
  0278                 ;* RX DATA BUFFER AREA *
  0279                 ;***********************
  0280                 ;
  0281 1062                    ORG      4400H
  0282                 ;
  0283 4400            RXBUF:  DS       1024
  0284                 ;
  0285                 ;*******************
  0286                 ;* STATUS FLAG AREA *
  0287                 ;*******************
  0288                 ;
  0289 4800                    ORG      5B00H
  0290                 ;
  0291 5B00            SR0AS:  DS       1                :SR0A STATUS
  0292 5B01            CRCFA:  DS       1                :CRC ERROR FLAG
  0293 5B02            OVRFA:  DS       1                :OVRN ERR FLAG
  0294 5B03            RENDA:  DS       1                :RX END FLAG
  0295                 ;
  0296                 ;*********************
  0297                 ;* I/O PARAMETER AREA *
  0298                 ;*********************
  0299                 ;
  0300 5B04 0001       TXLRA:  DW       100H             :TXLR
  0301                 ;
  0302 5B06 0044       BADR0:  DW       4400H            :CH-0 DMA ADRS
  0303 5B08 0101       BCWC0:  DW       101H             :CH-0 WORD COUNT
  0304                 ;
  0305                 ;**************
  0306                 ;* STACK AREA *
  0307                 ;**************
  0308                 ;
  0309 5B0A                    ORG      6000H
  0310                 ;
  0311                 STACK:
  0312 0000                    END
```

Figure 5.13 HDLC receive control program example 7

CHAPTER 6 µPD7201A SEND AND RECEIVE STATE TRANSITION DIAGRAMS

Chapter 6 shows the state transition for each µPD7201A protocol operation by using figures and tables.

Compare these with the µPD7201A operation described in Chapters 3 to 5.

° Protocols

1) Asynchronous
   Figures 6.1 and 6.2 and Tables 6.1 and 6.2

2) SYNC
   Figures 6.3 and 6.4 and Tables 6.3 and 6.4

3) HDLC
   Figures 6.5 and 6.6 and Tables 6.5 and 6.6

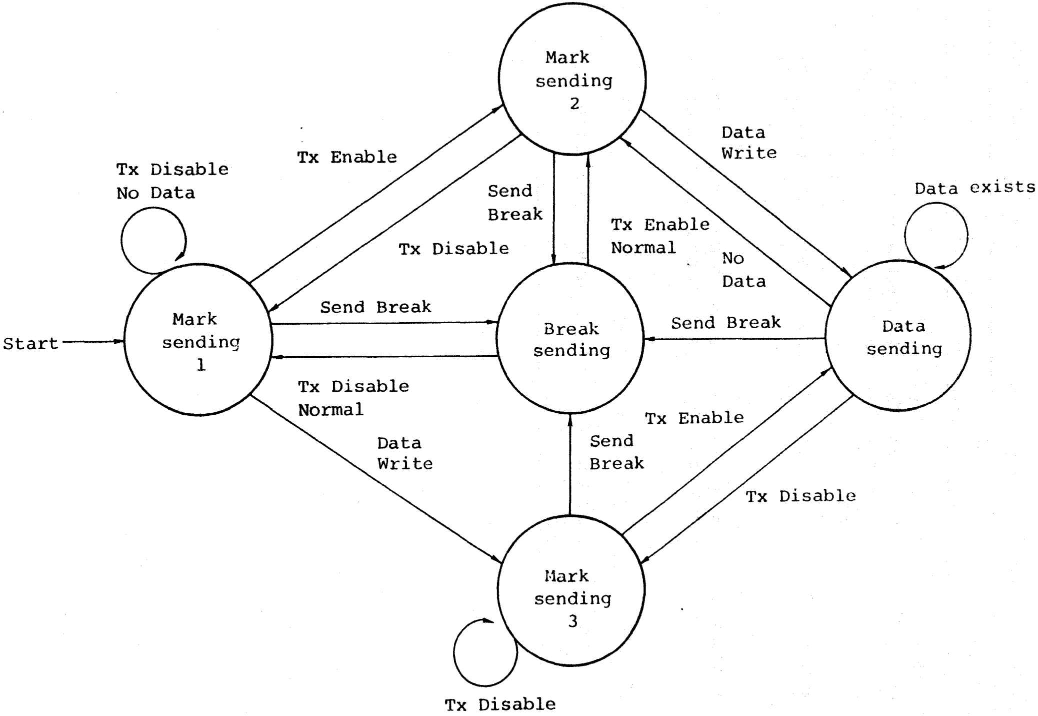


Figure 6.1 Asynchronous send state transition diagram

Table 6.1 Asynchronous send state transition diagram explanation table

No.	State name	Operation
1	Mark sending 1	Initial state. When Tx is disabled and no send data exists, TxD is placed in mark state (1).
2	Mark sending 2	When Tx is enabled and no send data exists, TxD is placed in mark state (1).
3	Mark sending 3	When Tx is disabled and send data exists, TxD is placed in mark state (1).
4	Data sending	Data in Tx shift register is sent.
5	Break sending	Space (0) is sent and send data is destroyed.

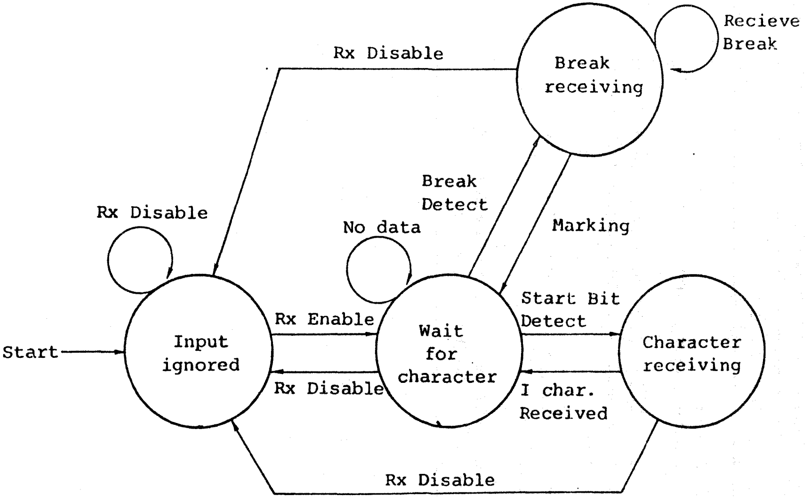


Figure 6.2 Asynchronous receive state transition diagram

Table 6.2 Asynchronous receive state transition diagram explanation table

No.	State name	Operation
1	Input ignored	Initial state. The RxD state is ignored.
2	Wait for character	A wait for start bit is continued when mark is being received after Rx is enabled.
3	Character receiving	After start bit is detected, data is assembled.
4	Break detection	If the RxD state is one-character mark, break detection is made.

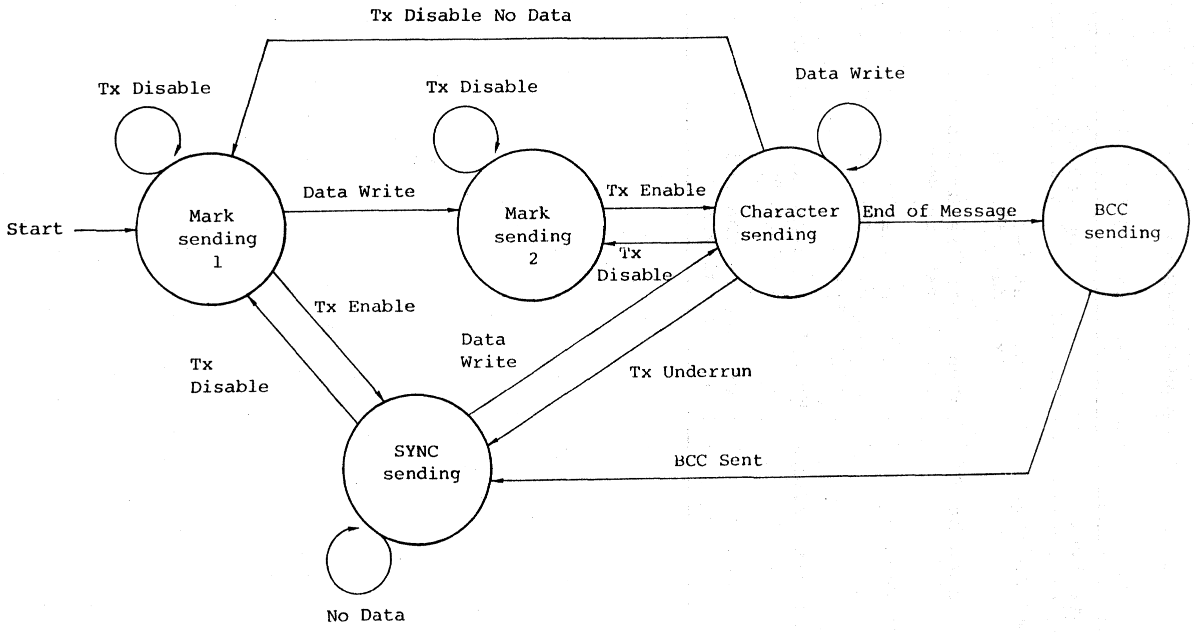


Figure 6.3 SYNC send state transition diagram

Table 6.3 SYNC send state transition diagram explanation table

No.	State name	Operation
1	Mark sending 1	Initial state. In TxD, mark is sent.
2	Mark sending 2	When Tx is disabled and Tx data exists, mark is sent.
3	SYNC sending	When Tx is enabled and no Tx data exists, SYNC character is automatically sent.
4	Character sending	When Tx is enabled and Tx data exists, data is sent.
5	BCC sending	When CRC is enabled, CRC2 byte is automatically sent.

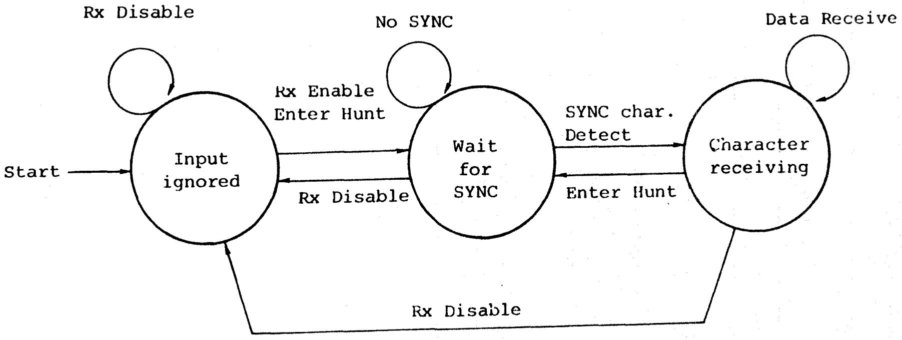


Figure 6.4 SYNC receive state transition diagram

Table 6.4 SYNC receive state transition diagram explanation table

No.	State name	Operation
1	Input ignored	Initial state. The RxD state is ignored.
2	Wait for SYNC	After Rx is enabled and Enter Hunt phase command is issued, SYNC character detection state is set. (during internal synchronization detection mode)
3	Character receiving	Character is assembled every data bit count after synchronization is set.

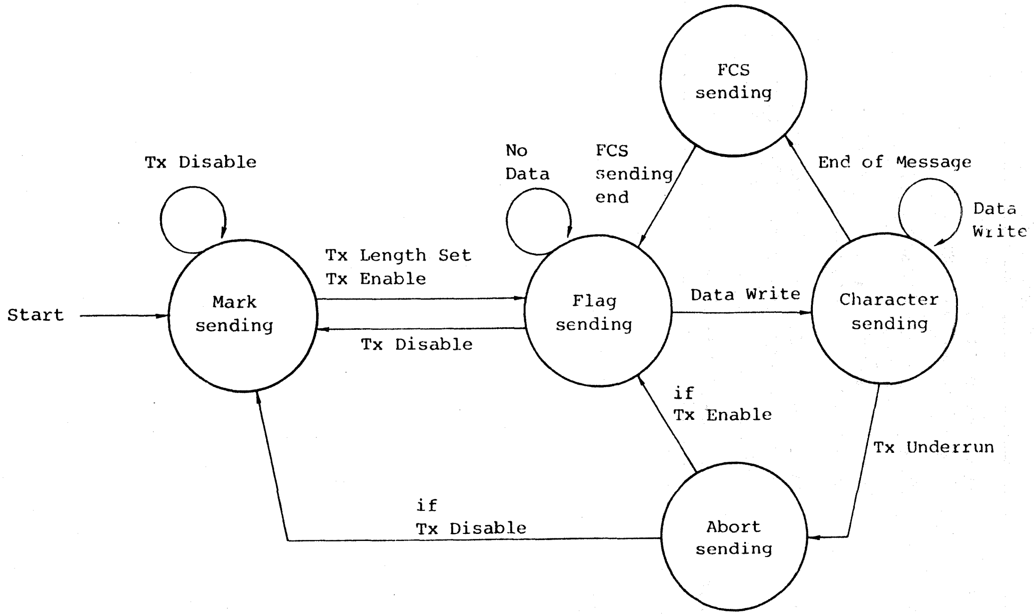


Figure 6.5 HDLC send state transition diagram (when Tx Length Counter is used and CRC exists

Table 6.5 HDLC send state transition diagram explanation table

No.	State name	Operation
1	Mark sending	Initial state. In TxD, mark is sent.
2	Flag sending	Flag is automatically sent.
3	Character sending	Data is sent. (Zero insertion function is contained.)
4	FCS sending	16 bits of CRC are automatically sent. (Zero insertion function is contained.)
5	Abort sending	1 of 8-13 bits is sent.

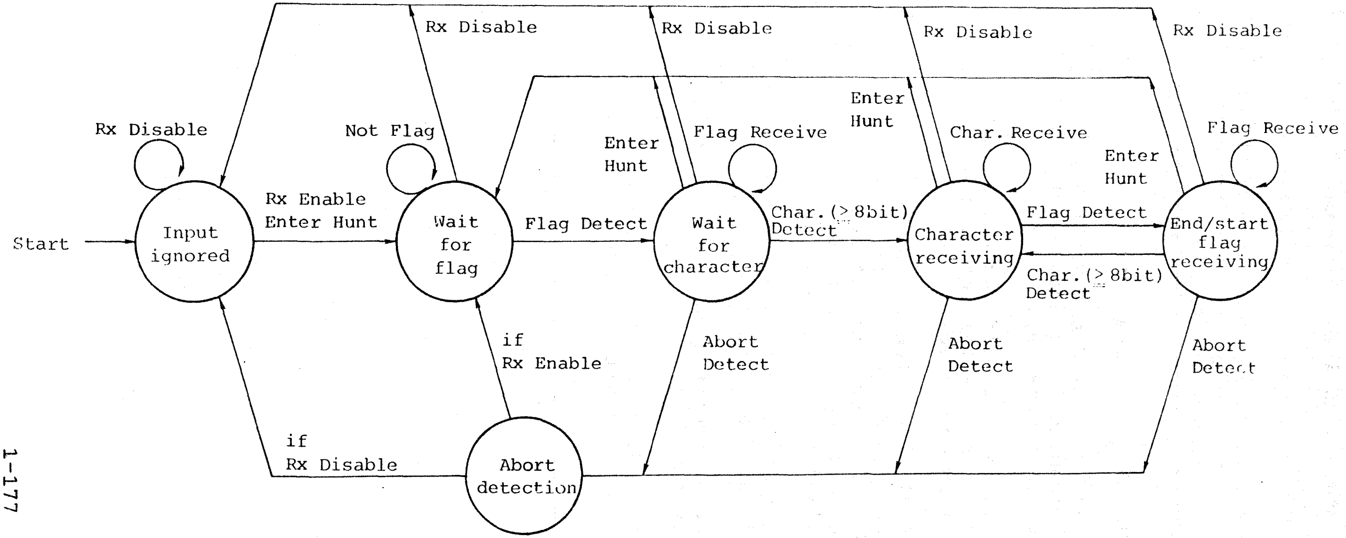


Figure 6.6 HDLC receive state transition diagram
(with no address search)

Table 6.6 HDLC receive state transition diagram explanation table

No.	State name	Operation
1	Input ignored	Initial state. RxD input is ignored.
2	Wait for flag	When Rx is enabled and Enter Hunt phase command is issued, start flag detection is started.
3	Wait for character	When start flag is detected, a wait for data (bit pattern other than flags) is continued.
4	Character receiving	After data is detected (normal address field), character is assembled every data bit count. (Zero deletion function is contained.)
4	End/start flag receiving	After A, C, I, FCS field is received, flag (end flag) is received. Character synchronization is made by subsequent flag idle.
5	Abort detection	Abort is detected by receiving 1 of 7 bit or more after synchronization is set.

PRELIMINARY

μPD72001

ADVANCED MULTIPROTOCOL SERIAL CONTROLLER

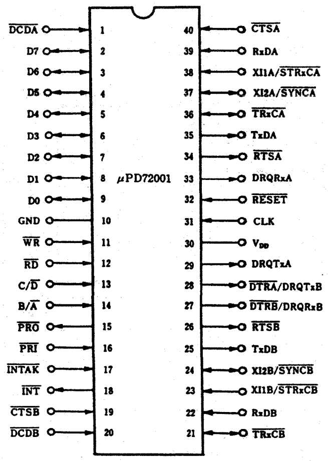
INTRODUCTION

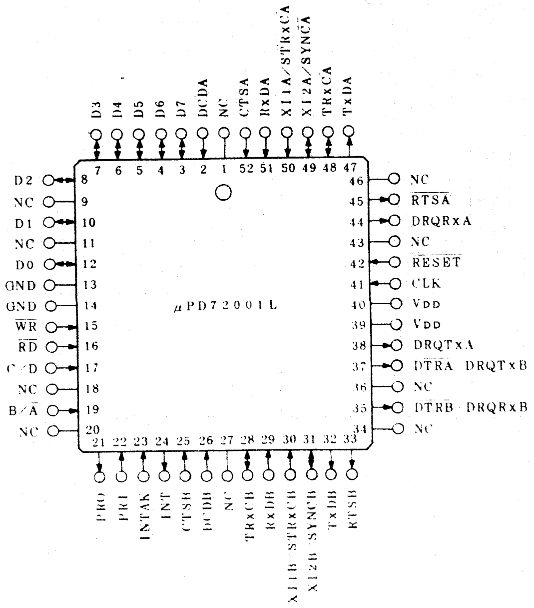
The new uPD72001 Advanced Multiprotocol Serial Controller (AMPSC) is a high performance, single chip CMOS controller applicable to a wide range of high speed serial data communication purposes. Located between a general purpose microprocessor and a data communiation equipment, the AMPSC provides data conversion according to the predetermined format on two independent serial full-duplex channels. The basic protocols are asynchronous, character oriented (COP) and bit oriented (BOP) as HDLC or SDLC. Data formats NRZ, NRZI and FM can be selected. The maximum transmission speed is 1.6 Mbits/s. Extended DMA and interrupt capabilities, on-chip crystal oscillator, baud rate generators and digital PLL circuit provide an easy hardware implementation. The baud rate generators are independent on transmit and receive and independent on both channels. The internal register structure allows 16 bit transmit data. Various types of error detection assure a highly reliable data transmission. The AMPSC is available now in 52 pin PLCC (uPD72001L) and 40 pin plastic DIP (72001C).

Features
- LSI to high-performance data communications
- Multiprotocol operations:
  - Asynchronous
  - Character-oriented protocol (COP)
  - Bit-oriented protocol (BOP)
- Two full-duplex channels
- Baud rate: DC to 1.6 Mbps
- Transmitter: Double buffer
- Receiver: Quadruple buffer
- Interrupt control capability
- DMA request signal: Transmit and Receive request/channel
- Overrun error detection
- Data formats: NRZ, NRZI, and FM
- Local self test capability
- On-chip crystal oscillator
- On-chip DPLL circuit
- On-chip baud rate generator
- General-purpose input/output pins: 4 pins per channel
- Standby function
- System clock rate: 8 MHz max.
- CMOS technology
- 40-pin DIP
- Single +5 V power supply

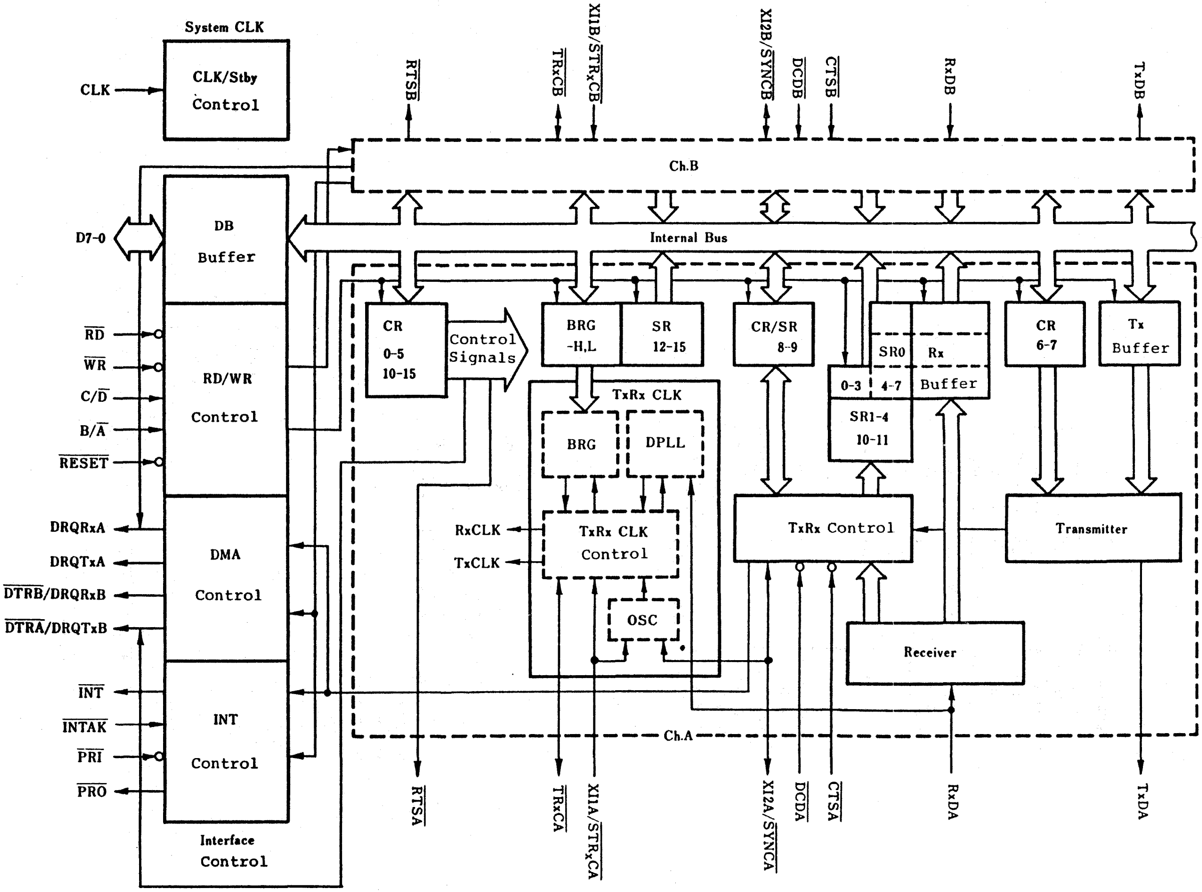
- Asynchronous operation
  - Character bit length: 5, 6, 7, or 8 bits
  - Stop bit length: 1, 1.5, or 2 bits
  - Clock rate: x1, x16, x32, or x64
  - Parity generation/checking
  - Framing error detection
  - Break generation/detection
- Character-oriented protocol (COP) operation
  - Mono-sync, Bi-sync, or External sync mode
  - Character bit length: 5, 6, 7, or 8 bits
  - Character synchronization: Internal or external
  - SYNC character bit length: 6 or 8 bits
  - BCS generation/checking (CRC-16, CRC-CCITT)
  - Parity generation/checking
  - Automatic SYNC character transmission/cheching/deletion
- Bit-oriented protocol (BOP) operation
  - HDLC, SDLC, or SDLC Loop mode
  - Flag transmission/detection
  - Zero insertion/suppression
  - Address field detection
  - FCS generation/check (CCITT-1)
  - Short frame detection
  - Automatic Abort transmission/detection
  - Idle detection
  - Go-Ahead detection
  - Transmit data count control (16-bit transmit length register/counter)

Pin Configuration (Top View)





Block Diagram



1. PIN FUNCTIONS

The AMPSC functions are basically divided in two: the system interface function which controls interfacing with a host system, and the data transmit/receive function. This section describes the AMPSC pin functions relating to system interface and send/receive control separately.
The pin functions are closely related to control register (CR) settings which specify device operations. In the following descriptions of pin functions, the related CR settings are referred to wherever needed. Refer to section 3, Register Configurations, as needed.
In the following descriptions, the input/output status of each pin is expressed, as a rule, by "H" (voltage level satisfying V_{IH} for input, and V_{OH} for output) and "L" (voltage level satisfying V_{IL} for input, and V_{OL} for output).

1.1 System Interface Pins

(1) VDD

Power supply input pin.

(2) GND

Ground pin.

(3) $\overline{\text{RESET}}$ (Reset) --- Input

This pin inputs an external Reset signal which resets the AMPSC. Applying a "L" signal continuously for 2 or more clock cycles (2 tcy) to this pin resets the device (system reset). A system reset causes the transmitter, receiver, and interrupt and DMA functions to be disabled, and the TxD and general-purpose output pins to be set to "H". It also resets (to zeros) all bits of the control register (CR). Therefore, once a system reset is executed, the CR must be set up again. Table 1-1 shows the pin status after system reset, in comparison with the pin status after channel reset (CR0: D5, D4, D3 = 0, 1, 1). For the CR and SR status after system reset, see section 3.1, Outline of Registers.
Upon system reset, the AMPSC automatically enters the standby mode, in which power consumption is reduced.

Table 1-1 Pin Status after Reset

Pin name	I/O	Pin status	
		$\overline{\text{RESET}}$ (system reset)	Channel reset
$\overline{\text{WR}}$	I	-	-
$\overline{\text{RD}}$	I	-	-
B/$\overline{\text{A}}$	I	-	-
C/$\overline{\text{D}}$	I	-	-
D7-D0	I/0	-	-
$\overline{\text{INT}}$	0	High impedance	High impedance
$\overline{\text{INTAK}}$	I	-	-
$\overline{\text{PRI}}$	I	-	-
$\overline{\text{PRO}}$	0	Depends on $\overline{\text{PRI}}$	Depends on $\overline{\text{PRI}}$
DRQTxA	0	"L"	"L"
DRQRxA	0	"L"	"L"
$\overline{\text{DTRA}}$/DRQTxB, $\overline{\text{DTRB}}$/DRQRxB	0	Enters $\overline{\text{DTR}}$ function and becomes "H"	Holds status
TxDA, TxDB	0	"H"	"H"
RxDA, RxDB	I	-	-
$\overline{\text{TRxCA}}$, $\overline{\text{TRxCB}}$	I/0	Input state	Holds current status
XI1A/$\overline{\text{STRxCA}}$ XI1B/$\overline{\text{STRxCB}}$	I	-	-
XI2A/$\overline{\text{SYNCA}}$ XI2B/$\overline{\text{SYNCB}}$	I/0	Input state	Input state
$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$	0	"H"	"H"
$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$	I	-	-
$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$	I	-	-

(4) CLK (System Clock) --- Input

This pin inputs the system clock. The system clock rate must be more than five times as high as the data transfer rate.

(5) $\overline{\text{WR}}$ (Write) --- Input (active Low)

This pin inputs a control signal used to write control words or Tx data to the device.

(6) $\overline{\text{RD}}$ (Read) --- Input (active Low)

This pin inputs a control signal used to read status or Rx data out of the device.

(7) B/$\overline{A}$ (Channel B/Channel A) --- Input
This pin inputs a channel select signal which selects the channel to be accessed for write or read operation. "L" input to this pin selects channel A; "H" input selects channel B.

(8) C/$\overline{D}$ (Control/Data) --- Input
This pin inputs a signal which determines the type of data on the data bus during read or write access to the device.

Table 1-2 shows the status of $\overline{WR}$, $\overline{RD}$, B/$\overline{A}$, and C/$\overline{D}$ and corresponding operations.

Table 1-2 AMPSC Control Signals Versus Operations

$\overline{WR}$	$\overline{RD}$	B/$\overline{A}$	C/$\overline{D}$	Operation	
L	H	L	L	Channel A	Writes transmit data to Tx buffer
		H		Channel B	
H	L	L	L	Channel A	Reads receive data from Rx buffer
		H		Channel B	
L	H	L	H	Channel A	Writes to control register
		H		Channel B	
H	L	L	H	Channel A	Reads status register
		H		Channel B	
H	H	X	X	High-impedance state	
L	L	X	X	Use inhibited	

X:Don't Care

(9) D7-D0 (Data Bus) --- Input/Output
These pins constitute a three-state, eight-bit, bidirectional data bus. This bus is connected to the host processor's data bus to transfer control words, status, and transmit/receive data.

(10) $\overline{INT}$ (Interrupt) --- Output (open drain)
This pin outputs the interrupt request signal. This pin is set to an active "L" if an interrupt condition occurred within the AMPSC. Being an open-drain output, this pin requires a pull-up resistor.

(11) $\overline{\text{INTAK}}$ (Interrupt Acknowledge) --- Input (active Low)

This pin inputs an acknowledge signal returned in response to an interrupt request. This pin is used to select the Vector mode (CR2A: D7 = 1). When selecting the Non-Vector mode (CR2A: D7 = 0), this pin must be pulled up to "H".

(12) $\overline{\text{PRI}}$ (Priority Input) --- Input (active Low)

This pin inputs a signal which controls interrupt request generation and interrupt vector output. This pin serves to interrupt generation control during general operation, and to interrupt vector output control during the INTAK sequence. The handling of this pin depends on the interrupt mode.

(a) Vectored mode (D7 of CR2A = 1)

The $\overline{\text{PRI}}$ pin usually controls the generation of interrupts. If Type A-3 or Type B-2 (D5, D4, D3 of CR2A = 0, 1, 0 or 1, 0, 0) interrupt vector output mode is selected, the $\overline{\text{PRI}}$ pin can be set to either "H" or "L". If any other interrupt vector output mode is selected, the $\overline{\text{PRI}}$ pin must be set to "L" to enable the interrupt request.

During the INTAK sequence, input of "L" level to the $\overline{\text{PRI}}$ pin causes the device to enable interrupt vector output in any interrupt vector output mode. If the $\overline{\text{PRI}}$ pin is set to "H", interrupt vector output is disabled.

(b) Non-Vectored mode (D7 of CR2A = 0)

Since this mode has no INTAK sequence, the $\overline{\text{PRI}}$ pin serves only for interrupt generation control. If a vector output mode other than Type A-3 and Type B-2 is selected, input of "L" level to the $\overline{\text{PRI}}$ pin enables interrupt generation. If the pin is set to "H", interrupt generation is disabled.

When a daisy chain is configured for interrupt, "L" input to this pin indicates that the interrupt from a device with higher priority is not serviced or the device is not in request for interrupt service. Only the AMPSC with "L" applied to this pin can request interrupts.

(13) $\overline{\text{PRO}}$ (Priority Output) --- Output (active Low)

This pin is used to configure an interrupt daisy chain. It controls interrupt requests from a device with a lower order of priority. It is usually used with the PRI pin, and is set to the following status depending on the $\overline{\text{PRI}}$ status:

If $\overline{\text{PRI}}$ = "H", $\overline{\text{PRO}}$ = "H".
If $\overline{\text{PRI}}$ = "L", $\overline{\text{PRO}}$ = "H" provided an interrupt request is present, and $\overline{\text{PRO}}$ = "L" provided no interrupt request is present.

(14) DRQTxA (DMA Request TxA) --- Output (active High)

This is a DMA request output to the DMA controller. This pin is set to "H" when the Tx buffer in transmitter channel A is emptied; The conditions under which this pin is set to "H" differ depending on the status of D2 of CR1:

CR1A: D2 = 0: DRQTxA is set to "H" when the Tx buffer is empty after the first Tx data was written into the buffer. If remains at "L" when the buffer is empty by reset operation.
CR1A: D2 = 1: DRQTxA is set to "H" when the Tx buffer is empty.

The DRQTxA is reset when transmit data is written into channel A.

(15) DRQRxA (DMA Request RxA) --- Output (active High)

This is a DMA request output to the DMA controller. This pin is set to "H" when the receiver on channel A enters the Rx Character Available state. It is reset when received data is read out of channel A.

(16) $\overline{\text{DTRA}}$/DRQTxB (Data Terminal Ready A/DMA Request TxB) --- Output

This pin serves the following two functions depending on the status of bits D1 and D0 of CR2A.

(a) When CR2A: D1, D0 = 0, 0 or 0, 1

This pin functions as $\overline{\text{DTRA}}$, which is a general-purpose output usable for modem control or other purposes. The $\overline{\text{DTRA}}$ pin status is as follows:

When CR5A: D7 = 0: $\overline{\text{DTRA}}$ = "H"
When CR5A: D7 = 1: $\overline{\text{DTRA}}$ = "L"

(b) When CR2A: D1, D0 =0, 1
This pin functions as a DRQTxB output. Its function is identical to that of the DRQTxA pin, with the exception that the former is used on channel B.

(17) $\overline{\text{DTRB}}$/DRQRxB (Data Terminal Ready B/DMA request RxB) --- Output
This pin serves the following two functions depending on the status of bits D1 and D0 of DR2A.

(a) When D1, D0 of CR2A = 0, 0 or 0, 1
This pin serves as the $\overline{\text{DTRB}}$ output. Its function is identical to that of the $\overline{\text{DTRA}}$ pin, with the exception that the former is used on channel B.

(b) When D1, D0 of CR2A = 1, 0
This pin serves as the DRQRxB output. Its function is identical to that of the DRQRxA pin, with the exception that the former is used on channel B.

(18) $\overline{\text{CTSA}}$ (Clear to Send A), $\overline{\text{CTSB}}$ (Clear to Send B) --- Inputs
These pins are general-purpose inputs usable for modem control or other purposes. A status change on these pins affects E/S bit latch operation. If E/S INT is enabled (D0 of CR1 set to 1), an E/S interrupt occurs. When the Auto Enable mode is selected (D5 of CR3 set to 1), these pins can be used with the Tx Enable bit (D3 of CR5) to control the transmitter status. This is shown in Table 1-3.

Table 1-3 Auto Enable Mode and $\overline{\text{CTS}}$ Pin

$\overline{\text{CTS}}$ pin	Tx Enable bit	Transmitter status
L	1	Enable
H	1	Disable
H or L	0	Disable

(19) $\overline{\text{DCDA}}$ (Data Carrier Detect A) --- Input
$\overline{\text{DCDB}}$ (Data Carrier Detect B) --- Input

These are general-purpose input pins usable for modem control or other purposes. A status change on these pins affects E/S bit latch operation. If E/S INT is enabled (D0 of CR1 set to 1), an E/S interrupt occurs.

If the Auto Enable mode is selected (D5 of CR3 set to 1), these pins can be used with the Rx Enable bit (D0 of CR3) to control the receiver status. This is shown in Table 1-4.

Table 1-4 Auto Enable Mode and $\overline{\text{DCD}}$ Pin

$\overline{\text{DCD}}$ pin	Rx Enable bit	Receiver status
L	1	Enable
H	1	Disable
H or L	0	Disable

(20) $\overline{\text{RTSA}}$ (Request to Send A) --- Output
$\overline{\text{RTSB}}$ (Request to Send B) --- Output

These are general-purpose output pins usable for modem control or other purposes. The status of these pins depends on the operation protocol setting and Auto Enable bit status, which is summarized in Table 1-5.

Table 1-5 Auto Enable Bit and $\overline{\text{RTS}}$ Pin

function Protocol	Auto Enable bit	RTS bit(CR5: D1)	$\overline{\text{RTS}}$ pin status
Asynchronous	0	0	H
		1	L
	1	0	Remains "L" until All Sent=1 and then becomes "H"
		1	L
COP/BOP	0 or 1	0	H
		1	L

1.2 Pins Relating to Transmit/Receive Operations

(1) TxDA (Transmit Data A), TxDB (Transmit Data B) --- Outputs
These are Tx data output pins.

(2) RxDA (Receive Data A), RxDB (Receive Data B) --- Inputs
These are Rx data input pins.

(3) XI1A/$\overline{\text{STRxCA}}$ (Crystal Input 1A/Source of Transmit Receive Clock A)
XI1B/$\overline{\text{STRxCB}}$ (Crystal Input 1B/Source of Transmit Receive Clock B) --- Inputs

The function of these pins depends on the status of D7 of CR15:

(a) When D7 of CR15 is 0
These pins function as $\overline{\text{STRxC}}$, which inputs a transmit/receive clock or a source clock for the on-chip baud rate generator (BRG) or digital phase locked loop (DPLL).

(b) When D7 of CR15 is 1
These pins function as XI1, which, along with pin XI2, connects with a crystal resonator for TxRx CLK source oscillation.

(4) XI2A/$\overline{\text{SYNCA}}$ (Crystal Input 2A/Synchronization A) --- Inputs,
XI2B/$\overline{\text{SYNCB}}$ (Crystal Input 2B/Synchronization B) Outputs

The function of these pins depends on the status of D7 of CR15:

(a) When D7 of CR15 is 0
These pins function as $\overline{\text{SYNC}}$, which provides the functions as shown in Table 1-6 depending on the setting of CR4.

(b) When D7 of CR15 is 1
These pins function as XI2, which , along with pin XI1, connects with a crystal resonator for TxRx CLK source osillation.

Table 1-6 $\overline{\text{SYNC}}$ Pin Functions versus CR4 Value (D7 of CR15 = 0)

Operation Protocol	Sync detection method	SYNC pin function	CR4						Function
			D7	D6	D5	D4	D3	D2	
Async		Input	X		X		0 1 1	1 0 1	Functions as a general-purpose input. Status changes at this pin ("H" to "L" or "L" to "H") affect Sync/Hunt bit (D4 of SR1) latch operation, and cause an E/S interrupt.
COP	Internal sync	Output	X		0 0	0 1	0	0	If a SYNC character is detected in the received data, this pin is set to "L" for 1 RxC cycle period.
	External sync	Input	0 0	0 1	1	1			Accepts character synchronization signal. If this pin is set from "H" to "L", the device exits the Hunt phase and establishes character synchronization. Character synchronization is maintained while the $\overline{\text{SYNC}}$ input is at "L". Receive character assembling is initiated at the rising edge of the receive clock which precedes the falling edge of the $\overline{\text{SYNC}}$ input.
BOP		No function	X		1	0			$\overline{\text{SYNC}}$ pin does not function.

X : Don't Care

(5) $\overline{\text{TRxCA}}$ (Transmit Receive Clock A) --- Input,Output
$\overline{\text{TRxCB}}$ (Transmit Receive Clock B) --- Input,Output

(a) If D2 of CR15 = 0
These pins function as transmit/receive clock inputs. They input an external transmit/receive clock.
(Exception):If at least one of the conditions, (D6, D5 Of CR15 = 0, 1) or (D4, D3 of CR15 = 0, 1) is satisfied, these pins function as inputs even if D2 of CR15 is set to 1.

(b) When bit D2 of CR15 = 1
These pins function as outputs. The output clock source is selectable from a crystal oscillator, BRG, DPLL, and transmit clock depending on the setting of D1 and D0 of CR15. However, when the condition given in the (Exception) of the above item (a) is fulfilled, these pins uncoditionally function as inputs, in which case the setting of D2, D1, and D0 of CR15 is invalid.

2. CONFIGURATION

The basic function of the AMPSC is to control serial data transmission/reception with other serial data communication devices. The AMPSC contains a flexible architecture to efficiently implement this function.
The internal logics of the AMPSC can be divided into system clock, system interface, and transmit/receive blocks. The system clock section supplies the system clock which is the base used to control the whole operation of the AMPSC's other internal circuits. The system interface block controls interfacing with its host system. The transmit/receive block controls data transmit/receive sequences. Fig. 2-1 shows the internal circuit configuration of the AMPSC.

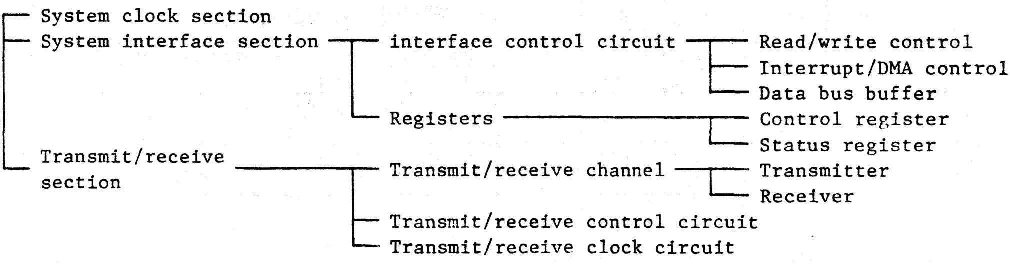


Fig. 2-1 AMPSC Internal Circuit Configuration

2.1 System Clock Section

The system clock block generates an internal reference clock based on the system clock applied to the CLK pin. This reference clock is supplied to other internal blocks to synchronize device operations.

2.2 System Interface Block

The system interface block consists of an interface control logic and registers. The interface control logic contains a read/write controller, an interrupt/DMA controller, and a data bus buffer. The registers consist of control and status registers.

2.2.1 Interface control circuit

The interface control circuit controls interfacing between the AMPSC and its host system. Data transfer between a device and host system is controlled by the read/write controller and interrupt/DMA controller via an 8-bit data bus buffer. The type of data on the data bus, channel selection, and direction of data transfer are controlled by control inputs. Table 2-1 shows combinations of control input signals and corresponding selections.

Table 2-1 Control Inputs Versus Selected Functions

$B/\overline{A}$	$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{INTAK}$	$\overline{PRI}$	Function	
L	L	L	H	H	X(1)	Channel A	Reads receive data
H						Channel B	
L	L	H	L	H	X	Channel A	Writes transmit data
H						Channel B	
L	H	L	H	H	X	Channel A	Reads status register (SR)
H						Channel B	
L	H	H	L	H	X	Channel A	Writes data to control register (CR)
H						Channel B	
X	X	H	H	L	L	Interrupt aknowledge sequence(2)	
				L	H		

Notes: 1. X denotes "Don't care".
2. Data to be output differs depending on vector type.

2.2.2 Registers

The registers within the AMPSC are used to set AMPSC operation or to indicate the device status. Each channel of the AMPSC has 20 control registers (CRs) for setting operation mode and operation control, and 12 types of status registers (SRs) for status indication.

Control words are written into these registers by the host processor. The status registers hold device status information. The status of the AMPSC can be determined by reading these registers. For more information about the CRs and SRs, see section 3, Register Configuration.

2.3 Transmitter/Receiver Block

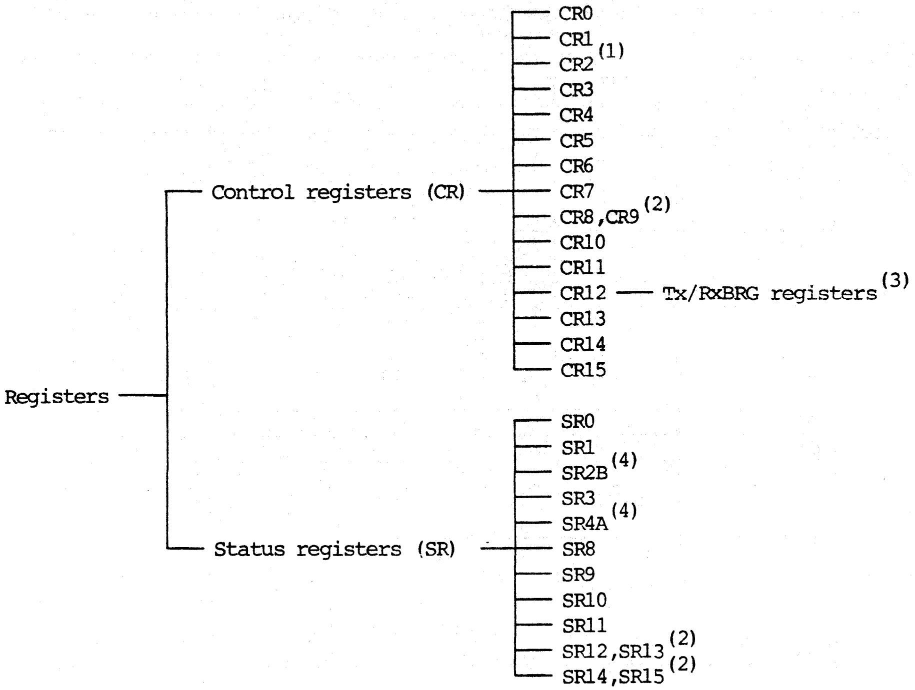
The transmitter/receiver block consists of two independent full-duplex channels, its control circuit and transmit/receive clock circuits. The operation of each transmit/receive channel is selectable from asynchronous, COP (bi-sync, etc.), and BOP (HDLC, SDLC, etc.) protocols, and is controlled by the transmit/receive control circuit. This block also contains a baud rate generator and digital PLL to supply clocks for serial data transmission and reception.

3. REGISTER CONFIGURATION

To implement operations under different communication protocols, the AMPSC contains a large group of registers. This section details the register configuration within the AMPSC and the function of each register.
The descriptions in this section assume that the reader already has some basic knowledge about each communication protocol. For details of communication protocols, refer to the respective standards for protocols. Unless otherwise stated, the following descriptions assume that relevant interrupts are enabled.

3.1 Outline of Registers

The AMPSC's internal registers are divided into control registers (CR) and status registers (SR). Control registers are used to set up the device operation mode or to control device operations. Control words are written into these registers by the host processor. Status registers hold device status information. The host processor can monitor the AMPSC's status by reading these status registers. The AMPSC's internal register configuration is shown in Fig. 3-1. The functions of the registers are outlined in Table 3-1. The initial values of the registers after resetting and the register status in the Standby mode are shown in Table 3-2.



Note: 1. Shared with both channels, though the function of this register differs on each channel.
2. These registers are used in pairs.
3. These registers are made available by setting CR12.
4. These registers do not exist on other channels.
5. The registers listed above are provided on each channel, with some exceptions.

Fig. 3-1 AMPSC Internal Register Configuration

Table 3-1 Outline of AMPSC's Internal Register Functions

Type	Reg. Name	Channel	Description
Control registers	CR0	A, B	Register selection pointer, control commands, or initialize commands
	CR1	A, B	Interrupt/DMA control, received data transfer mode
	CR2	A	System Interface mode, Interrupt/DMA mode, Vector mode
		B	Initial value of interrupt vector
	CR3	A, B	Receive control and Auto Enable control
	CR4	A, B	Operation protocol mode, parity control
	CR5	A, B	Transmission control
	CR6	A, B	SYNC character/address
	CR7	A, B	SYNC character/flag pattern
	CR8	A, B	Tramsmit data count (lower 8 bits) (TxDL-L)
	CR9	A, B	Transmit data count (higher 8 bits) (TxDL-H)
	CR10	A, B	Data format, loop mode control, SYNC character bit length
	CR11	A, B	E/S interrupt control
	CR12	A, B	$\overline{\text{TRxC}}$ pin/DPLL source selection, Tx/RxBRG interrupt control, Tx/RxBRG register control
	RxBRG-L	A, B	RxBRG count value (lower 8 bits)
	RxBRG-H	A, B	RxBRG count value (higher 8 bits)
	TxBRG-L	A, B	TxBRG count value (lower 8 bits)
	TxBRG-H	A, B	TxBRG count value (higher 8 bits)
	CR13	A, B	TxDLC control, standby mode control
	CR14	A, B	DPLL control, test mode control, Tx/Rx BRG control
	CR15	A, B	Crystal oscillator control, Tx/Rx CLK source selection, $\overline{\text{TRxC}}$ pin control

Status registers	SR0	A, B	Transmit/receive buffer status, Special Rx Condition status
	SR1	A, B	E/S bit
	SR2	B	Interrupt vector
	SR3	A, B	Residue Code, Tx/Rx BRG Zero Count
	SR4	A	INT Pending bit
	SR8	A, B	TxDLC-L (lower 8 bits)
	SR9	A, B	TxDLC-H (higher 8 bits)
	SR10	A, B	DPLL CLK Missing status, Loop status
	SR11	A, B	Interrupt Enable status (contents of CR11)
	SR12	A, B	RxBRG count value (lower 8 bits)
	SR13	A, B	RxBRG count value (higher 8 bits)
	SR14	A, B	TxBRG count value (lower 8 bits)
	SR15	A, B	TxBRG count value (higher 8 bits)

Table 3-2 (1/5) Register Status (No. 1)

(X: Undefined, –: Not affected)

Register name	Bit	D7	D6	D5	D4	D3	D2	D1	D0
CR0	Function	CRC Control		Command			Register Pointer		
	System reset	×		×		0	0		
	Channel reset	×		×		0	0		
	Standby mode	–		–		–	–		
CR1	Function	Short Frame Detect	Overrun Error INT	First Rx INT Mask	Rx INT Mode		First Tx INT/DMA Enable	Tx INT/DMA Enable	E/S INT Enable
	System reset	×	×	×	0		×	0	0
	Channel reset	–	–	–	0		–	0	0
	Standby mode	–	–	–	–		–	–	–
CR2A	Function	Vector Mode	Status Affects Vector	Output Vector Type			Priority Select	INT/DMA Mode	
	System reset	0	×	0			0	0	
	Channel reset	–	–	–			–	–	
	Standby mode	–	–	–			–	–	
CR2B	Function	Initial Vector Value							
	System rest	×							
	Channel reset	–							
	Standby mode	–							
CR3	Function	Rx Character Bit Length		Auto Enable	Enter Hunt Phase	Rx CRC Enable	Address Search Mode	SYNC Char. Load Inhibit/ Multicast	Rx Enable
	System reset	×		×	0	×	×	×	0
	Channel reset	–		–	0	–	–	–	0
	Standby mode	–		–	–	–	–	–	–
CR4	Function	Clock Rate		Protocol Mode		Tx Stop Bits		Parity Select	Parity Enable
	System reset	×		×		×		×	×
	Channel reset	–		–		–		–	–
	Standby mode	–		–		–		–	–
CR5	Function	DTR Control	Tx Character Bit Length		Send Break/ Abort	Tx Enable	CRC Poly-nomial	RTS Control	Tx CRC Enable
	System reset	0	×		0	0	0	0	×
	Channel reset	0	–		0	0	0	0	–
	Standby mode	–	–		–	–	–	–	–

Table 3-2 (2/5) Register Status (No. 2)

(X: Undefined, —: Not affected)

Register name	Bit	D7	D6	D5	D4	D3	D2	D1	D0
CR6	Function	SYNC Character/Address							
	System reset	×							
	Channel reset	—							
	Standby mode	—							
CR7	Function	SYNC Character/Flag							
	System reset	×							
	Channel reset	—							
	Standby mode	—							
CR8	Function	Tx Data Length L-Byte							
	System reset	×							
	Channel reset	—							
	Standby mode	—							
CR9	Function	Tx Data Length H-Byte							
	System reset	1	×						
	Channel reset	1	—						
	Standby mode	—	—						
CR10	Function	CRC Initialized Condition	Data Format		Auto Tx on Sync/Tx on Loop	Idle Condition	Tx Condition on Underrun	D4/Loop Enable	SYNC Character Bits
	System rest	0	0		0	0	0	0	0
	Channel reset	0	0		0	0	0	0	0
	Stanby mode	—	—		—	—	—	—	—
CR11	Function	Break/Abort/GA IE	Tx Underrun/EOM IE	CTS IE	Sync/Hunt IE	DCD IE	All Sent IE	Idle Detect IE	BRG IE
	System rest	1	1	1	1	1	1	1	0
	Channel reset	1	1	1	1	1	1	1	0
	Standby mode	—	—	—	—	—	—	—	—
CR12	Function	BRG Select for $\overline{\text{TRxC}}$	BRG Select for DPLL			Tx BRG IE	Rx BRG IE	Tx BRG Register Set	Rx BRG Register Set
	System reset	×	×			0	0	0	0
	Channel reset	—	—			0	0	0	0
	Standby mode	—	—			—	—	—	—

Table 3-2 (3/5) Register Status (No. 3)

(X: Undefined, —: Not affected)

<table>
<tr><th>Register name</th><th>Bit</th><th>D7</th><th>D6</th><th>D5</th><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th></tr>
<tr><td rowspan="4">Rx BRG Register-L</td><td>Function</td><td colspan="8">Rx BRG Constant L-Byte</td></tr>
<tr><td>System rest</td><td colspan="8">X</td></tr>
<tr><td>Channel reset</td><td colspan="8">—</td></tr>
<tr><td>Standby mode</td><td colspan="8">—</td></tr>
<tr><td rowspan="4">Rx BRG Register-H</td><td>Function</td><td colspan="8">Rx BRG Constant H-Byte</td></tr>
<tr><td>System reset</td><td colspan="8">X</td></tr>
<tr><td>Channel rest</td><td colspan="8">—</td></tr>
<tr><td>Standby mode</td><td colspan="8">—</td></tr>
<tr><td rowspan="4">Tx BRG Register-L</td><td>Function</td><td colspan="8">Tx BRG Constant L-Byte</td></tr>
<tr><td>System reset</td><td colspan="8">X</td></tr>
<tr><td>Channel reset</td><td colspan="8">—</td></tr>
<tr><td>Standby mode</td><td colspan="8">—</td></tr>
<tr><td rowspan="4">Tx BRG Register-H</td><td>Function</td><td colspan="8">Tx BRG Constant H-Byte</td></tr>
<tr><td>System reset</td><td colspan="8">X</td></tr>
<tr><td>Channel reset</td><td colspan="8">—</td></tr>
<tr><td>Standby mode</td><td colspan="8">—</td></tr>
<tr><td rowspan="4">CR13</td><td>Function</td><td colspan="6"></td><td>Tx DLC Enable</td><td>Stand-by Mode Set</td></tr>
<tr><td>System reset</td><td colspan="6" rowspan="3"></td><td>0</td><td>0</td></tr>
<tr><td>Channel reset</td><td>0</td><td>0</td></tr>
<tr><td>Standby mode</td><td>—</td><td>—</td></tr>
<tr><td rowspan="4">CR14</td><td>Function</td><td colspan="3">DPLL Command</td><td>Local Self Test</td><td>Echo Loop Test</td><td>BRG Source Select</td><td>Rx BRG Enable</td><td>Tx BRG Enable</td></tr>
<tr><td>System reset</td><td>1
1</td><td>0
1</td><td>1
1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Channel reset</td><td>1
1</td><td>0
1</td><td>1
1</td><td>—</td><td>—</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Standby mode</td><td colspan="3">—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td rowspan="4">CR15</td><td>Function</td><td>Xtal Select</td><td colspan="2">RxCLK Select</td><td colspan="2">TxCLK Select</td><td>$\overline{\text{TRxC}}$ Input/ Output</td><td colspan="2">$\overline{\text{TRxC}}$ Source Select</td></tr>
<tr><td>System reset</td><td>0</td><td colspan="2">0</td><td colspan="2">0</td><td>0</td><td colspan="2">0</td></tr>
<tr><td>Channel reset</td><td>0</td><td colspan="2">0</td><td colspan="2">0</td><td>0</td><td colspan="2">0</td></tr>
<tr><td>Standby mode</td><td>—</td><td colspan="2">—</td><td colspan="2">—</td><td>—</td><td colspan="2">—</td></tr>
</table>

Table 3-2 (4/5) Register Status (No. 4)

(× (×: Undefined, —: Not affected)

Register name	Bit	D7	D6	D5	D4	D3	D2	D1	D0
SR0	Function	End of Frame	CRC/ Framing Error	Rx Overrun Error	Parity Error	Short Frame Detect	Tx Buffer Empty	Sending Abort	Rx Data Available
	System reset	0	0	0	0	0	1	0	0
	Channel reset	0	0	0	0	0	1	0	0
	Standby mode	—	—	—	—	—	—	—	—
SR1	Function	Break/ Abort/ GA Detect	Tx Underrun/ EOM	CTS	Sync/ Hunt	DCD	All Sent	Idle Detect	BRG Zero Count
	System reset	×	1	×	×	×	0	0	0
	Channel reset	—	1	—	—	—	0	0	0
	Standby mode	—	—	—	—	—	—	—	—
SR2B	Function	Vector Value							
	System reset	×							
	Channel reset	×							
	Standby mode	—							
SR3	Function				TxBRG Zero Count	Rx BRG Zero Count	Residue Code		
	System reset				0	0	×		
	Channel reset				0	0	×		
	Standby mode				—	—	—		
SR4A	Function	Ch-A Sp. Rx Condition INT Pending	Ch-B Sp. Rx Condition INT Pending	Ch-A Rx INT Pending	Ch-A Tx INT Pending	Ch-A E/S INT Pending	Ch-B Rx INT Pending	Ch-B Tx INT Pending	Ch-B E/S INT Pending
	System reset	0	0	0	0	0	0	0	0
	Channel reset	0	0	0	0	0	0	0	0
	Standby mode	—	—	—	—	—	—	—	—
SR10	Function	One Clock Missing	Two Clocks Missing		Sending on Loop			Tx Sync/ on Loop	
	System reset	0	0		0			0	
	Channel reset	0	0		0			0	
	Standby mode	—	—		—			—	

Table 3-2 (5/5) Register Status (No. 5)

(X: Undefined, —: Not affected)

Register name	Bit	D7	D6	D5	D4	D3	D2	D1	D0
SR11	Function	Break/ Abort IE	Tx Underrun/ EOM IE	CTS IE	Sync/ Hunt IE	DCD IE	All Sent IE	Idle Detect IE	BRG IE
	System reset	1	1	1	1	1	1	1	0
	Channel reset	1	1	1	1	1	1	1	0
	Standby mode	—	—	—	—	—	—	—	—
SR12	Function	Rx BRG Counter Constant L-Byte							
	System reset	×							
	Channel reset	—							
	Standby mode	—							
SR13	Function	Rx BRG Counter Constant H-Byte							
	System reset	×							
	Channel reset	—							
	Standby mode	—							
SR14	Function	Tx BRG Counter Constant L-Byte							
	System reset	×							
	Channel reset	—							
	Stnadby mode	—							
SR15	Function	Tx BRG Counter Constant H-Byte							
	System reset	×							
	Channel reset	—							
	Standby mode	—							

The AMPSC's internal registers can be accessed through pins B/$\overline{A}$ and C/$\overline{D}$. Selection of registers is done by combining pointer bits D0-D2 of CR0 and High Pointer bit D3 of the same register. The register pointer within the AMPSC is automatically cleared when a register is accessed after being selected by the pointer bits. It indicates CR0/SR0 unless the pointer bit value is subsequently specified. The register addressing sequence is shown in Fig. 3-2.

(Example) When setting 20H into CR4A

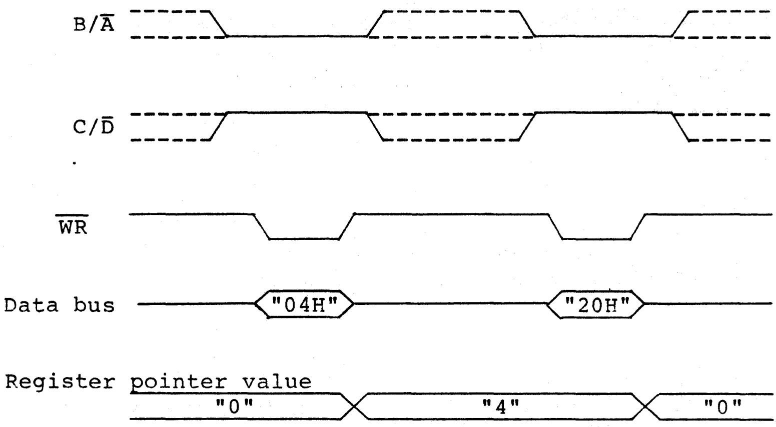


Fig. 3-2 AMPSC Register Setting Sequence

Exception:

If D1 or D0 of CR12 is set to 1, the subsequent two control word writing operations access the lower then the upper bytes of the TxBRG or RxBRG register. The register pointer is cleared only after these write accesses are completed. This sequence is shown in Fig. 3-3. If both D1 and D0 are zero, this exception is not applied.

Example: Setting the TxBRG register on channel B:

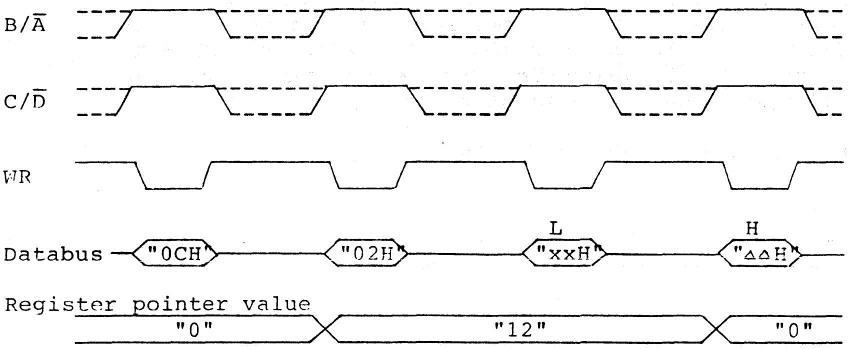


Fig. 3-3 TxBRG Register Setting Sequence

3.2 Control Registers (CR)

This section describes control registers used to specify operation modes of the AMPSC. Control register functions are listed in Fig. 3-4 below. Some control registers need not be set depending on the operation modes to be used. Carefully examine the meaning of each bit of the control registers in the desired operation mode.

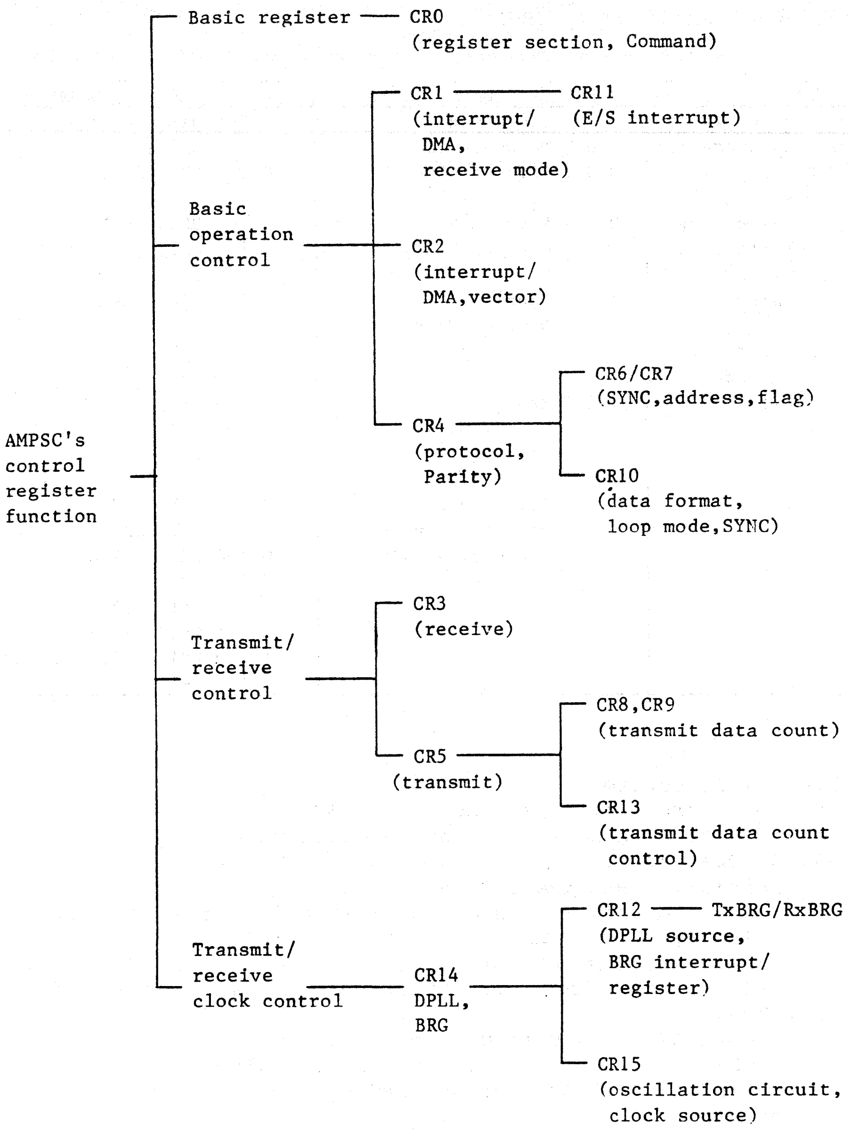


Fig. 3-4 Control Register Configuration

3.2.1 Control Register 0 (CR0)

<table>
<tr><td>Bit</td><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td></tr>
<tr><td>Function</td><td colspan="2">CRC Control</td><td colspan="3">Command</td><td colspan="3">Register Pointer</td></tr>
<tr><td>Contents</td><td colspan="2">0 0 No Operation
0 1 Initialize Rx CRC Calculator
1 0 Initialize Tx CRC Calculator
1 1 Reset Tx Underrun/EOM Bit</td><td colspan="3">0 0 0 No Operation
0 0 1 High Pointer
0 1 0 Reset E/S Bit Latch
0 1 1 Channel Reset
1 0 0 Enable Next Rx Character Interrupt
1 0 1 Reset Tx INT/DMA Pending
1 1 0 Error Reset
1 1 1 End of Interrupt (Channel A Only)</td><td colspan="3">0 0 0 CR0, SR0
0 0 1 CR1, SR1
0 1 0 CR2, SR2B
0 1 1 CR3, SR3
1 0 0 CR4, SR4A
1 0 1 CR5
1 1 0 CR6
1 1 1 CR7
0 0 0 CR8, SR8
0 0 1 CR9, SR9
0 1 0 CR10, SR10
0 1 1 CR11, SR11
1 0 0 CR12, SR12
1 0 1 CR13, SR13
1 1 0 CR14, SR14
1 1 1 CR15, SR15</td></tr>
</table>

(1) D7, D6 (CRC Control)

These bits are valid when the COP or BOP mode is selected. They have no meaning in the asynchronous mode.

0 0: No operation.

0 1: Initialize Receive CRC Calculator

This command is used to initialize the Rx CRC calculator. It must be issued before starting data reception. The initial state (value) of the Rx CRC calculator differs depending on the value of D7 of CR10:

. When D7 of CR10 is 0, the initial value is 0.

. When D7 of CR10 is 1, the initial value is 1.

Bit D7 of CR10 must be set before issuing this command.

Exception:

In the BOP mode, the CRC calculator is automatically intialized according to the value of CR10's D7 upon receiving the flag, without the need for issuing this command.

1 0: Inirialize Transmit CRC Calculator

This command is used to initialize the TxCRC calculator. It must be issued before starting data transmission. The initial value of the TxCRC calculator differs depending on the value of D7 of CR10:

. When D7 of CR10 is 0, initial value is 0.

. When D7 of CR10 is 1, initial value is 1.

D7 of CR10 must be set before issuing this command.

Exception:

If D7 of CR10 is set to 1 in the BOP mode, the TxCRC calculator is automatically initialized to 1 when a flag is transferred to the transmit shift register within the AMPSC.

1 1: Reset Transmit Underrun/End of Message Bit

This command is used to reset (1 to 0) D6 of SR1 (Tx Underrun/EOM). If Tx data is not written to the transmit buffer in the specified time period after transmit operation was started, the AMPSC enters the Tx Underrun/EOM state. At this point the AMPSC judges whether a CRC is to be sent or not depending on the value of D6 of SR1. Therefore, manipulation of bit D6 of SR1 is required before starting transmission:

. When D6 of SR1 is 0 when a Tx underrun occurs, a CRC is sent.

. When D6 of SR1 is 1 when a Tx underrun occurs, a SYNC character/flag is sent.

If CRC transmission is desired at the time of the Tx underrun, D6 of SR1 must have been set to zero by this command beforehand. This command becomes valid after at least one data byte is written into the AMPSC after transmission is enabled.

Exception:
In the BOP mode, D6 of SR1 is set to zero when the first Tx data of a frame is written into the AMPSC without issuing this command.

(2) D5-D3 (Command)
These bits of control register 0 are used to control the device's status after the device has entered a given state. The commands are valid only when they are issued.

0 0 0: No operation.

0 0 1: High Pointer
This command is used with D2-D0 (Register Pointer) of CR0 when accessing CR8-CR15 or SR8-SR15. For example, when accessing CR11, D5-D0 of CR0 are set to 001011.

0 1 0: Reset External/Status Bit Latch
This command is issued when an E/S bit (each bit of SR1) latch operation occurred. It allows latching of a new cause of E/S bit status change. If E/S interrupt is enabled, an E/S interrupt will occur when the E/S bit status has changed.

0 1 1: Channel Reset
This command is used to reset the AMPSC channels. It has a function equivalent to a system reset. Executing channel reset stops AMPSC operation. To determine the CR and SR status after channel reset, see Table 3-2. A control word should not be written within 3 system clock periods (3 tcy) after the Channel Reset command was issued.

1 0 0: Enable Next Receive Character Interrupt
This command is valid only when the First Rx INT mode (D4, D3 of CR1 = 0, 1) is selected. It is used when it is desired to guarantee that a Rx interrupt will occur when the next data character is received after this command is issued.

Exception:

This command is invalid when the First Rx INT Mask is on (D5 of CR1 = 1), even if the First Rx INT mode is selected.

1 0 1: Reset Transmit Interrupt/DMA Pending

This command is used to clear the AMPSC's Tx interrupt request or Tx DMA request which occurs when the Tx buffer is emptied (D2 of SR0 = 1), without writing Tx data into the device. It is usually used to clear a Tx interrupt or Tx DMA request caused by the Tx Buffer Empty state which occurs after the last Tx data is written into the AMPSC.

This command causes the AMPSC to operate follows:

(a) Asynchronous mode

The device issues a Tx interrupt or Tx DMA request when the Tx buffer is emptied after the next Tx data is written.

(b) COP mode

The device issues a Tx interrupt or Tx DMA request when the internal Tx buffer is emptied after the device has completed CRC transmission in the Tx Underrun state which occurred after this command was issued. If Tx data was written in the Tx Underrun state (e.g., PAD), the device issues a Tx interrupt or Tx DMA request when its Tx buffer is emptied after that data is transmitted following a CRC.

(c) BOP mode

The device issues a Tx interrupt or Tx DMA request when its internal Tx buffer is emptied after a CRC has been transmitted in the Tx Underrun state which occurred after this command was issued.

1 1 0: Error Reset

This command is used to reset the pertinent bits (D3-D7 of SR0) if a Special Rx Condition occurred. If a Special Rx Condition interrupt occurs when the First Rx INT mode is selected, data subsequently received is not transferred to the last stage of the AMPSC's internal Rx buffer but remains in the first and second stages of the buffer.

1 1 1: End of Interrupt
This command is used to let the AMPSC recognize the end of an interrupt sequence. It should be issued when an interrupt service for the AMPSC is completed. Execution of this command resets the cause of the interrupt with the highest priority at that point, and enables subsequent interrupt requests.

(3) D2-D0 (Register Pointer)
These bits specify the number of the register within the AMPSC to be accessed. They are reset to 000 when a reset operation is executed or when the AMPSC is accessed after a Register Pointer value is specified. For registers numbered 8 and above, the High Pointer command (D5-D3 = 001) is added to this Register Pointer to access those registers.

3.2.2 Control Register 1 (CR1)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function / Operation protocol	Short Frame Detect	Overrun Error INT	First Rx INT Mask	Rx INT Mode		First Tx INT/ DMA Enable	Tx INT/ DMA Enable	E/S INT Enable
Async	0	0 Normal Mode 1 Special Mode	0 Disable 1 Enable	0 0 Rx INT Disable 0 1 First Rx INT 1 0 All Rx INT-1 1 1 All Rx INT-2		0 Disable 1 Enable	0 Disable 1 Enable	0 Disable 1 Enable
COP								
BOP	0 Disable 1 Enable							

(1) D7 (Short Frame Detect; Valid in BOP Mode only)
This bit of Control Register 1 is used to detect short frames (frames less than 32 bits long).

0: Disable
Short frame detection is disabled.

1: Enable
Short frame detection is enabled. If a received frame is a short frame, D3 of SR0 (Short Frame Detect) is set to 1, causing a Special Rx Condition interrupt.

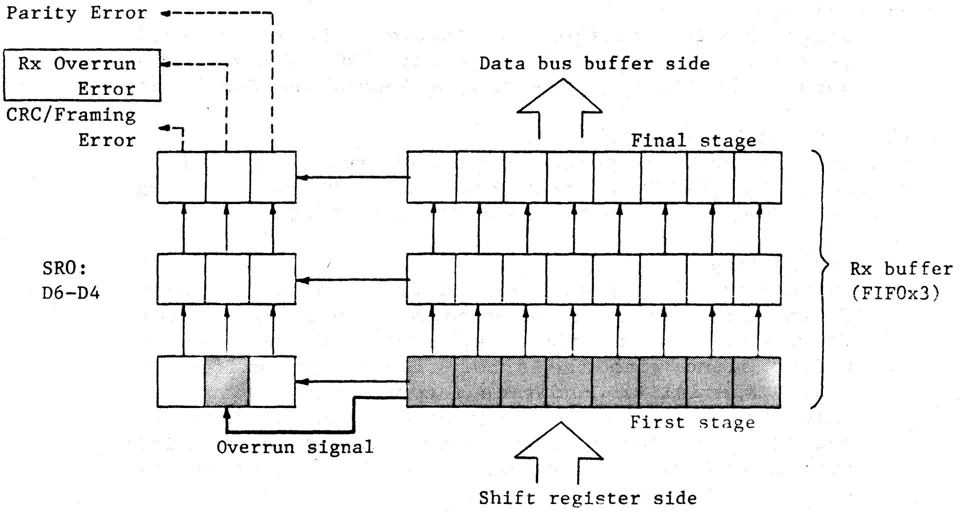
(2) D6 (Overrun Error INT)

This bit is used to specify how an overrun error will be indicated by the Rx Overrun Error bit (D5 of SR0) and when the Special Rx Condition interrupt occurs.

0: Normal Mode
In this mode, overrun error status is available only if the received data which caused the overrun error is transferred to the last stage of the receive buffer along with the error flag value shifted in SR0. A Special Rx Condition interrupt occurs at this time. In this mode the received data which caused an overrun error corresponds to the information pointed by the Rx Overrun Error bit. This is illustrated in Fig. 3-5(a).

1: Special Mode
In this mode overrun information is reflected by the Rx Overrun Error bit when an overrun error occurred within the AMPSC. A Special Rx Condition interrupt also occurs at this time. Therefore, the received data which caused the overrun error does not correspond to the information indicated by the Rx Overrun Error bit. This is illustrated in Fig. 3-5(b).

(a) If D6 of CR1 = 0 (Normal mode)



(b) If D6 of CR1 = 1 (Special mode)

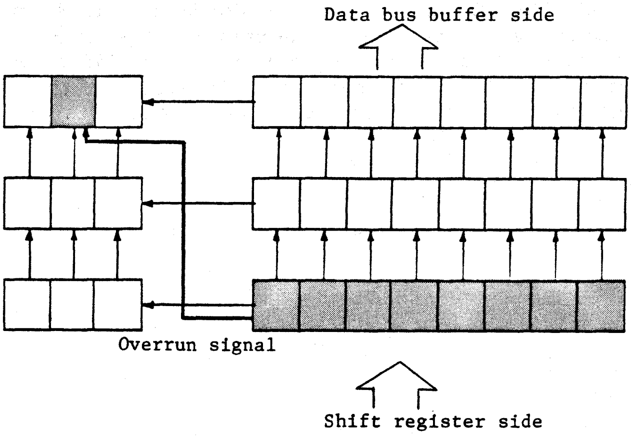


Fig. 3-5 Rx Overrun Error Bit Set Timing

(3) D5 (First Receive Interrupt Mask)
This is enabled only if the First Rx INT Mode (D4, D3 of CR1 = 0, 1) is selected. It is used to mask Rx interrupt caused by first received data.

0: Disable
The first received data causes an Rx interrupt (normal operation).

1: Enable
The first received data does not cause an Rx interrupt, and the Enable Next Rx Character INT command becomes invalid. However this bit does not affect the Special Rx Condition interrupt.

(4) D4, D3 (Receive Interrupt Mode)
These bits are used to set the Rx Interrupt mode. They specify in what way received data is to be indicated.

0 0: Receive Interrupt Disable
This mode is used to accept received data by using status polling, or to disable receive interrupt requests.

0 1: First Receive Interrupt
This mode is used to accept received data by using DMA. In this mode, an Rx interrupt occurs only by the first data is received after the Rx was enabled after initialization. In this case the first received data refers to:

. Asynchronous mode --- First character received.
. COP mode --- First character, other than SYNC character, received after synchronization was established.
. BOP mode --- First data (generally address data) first received after flag detection. In the Address Search mode it refers to valid address data that follows a flag and matches the value of CR6. These received data items can be read as normal data.

If an Enable Next Rx Character INT command is issued, data received after this command was issued can cause an Rx interrupt.

1 0: All Receive Interrupt-1
This mode is used to indicate received data by using interrupts. An interrupt occurs each time data is loaded into the receive buffer. In this mode a parity error is one of the sources of a Special Rx Condition interrupt.

1 1: All Receive Interrupt-2
Same as the All Rx Interrupt-1, with the exception that a parity error is not indicated by a Special Rx Condition interrupt.

(5) D2 (First Transmit Interrupt/DMA Enable)
This bit determines whether the Tx INT/DMA request is to be activated or not immediately after transmission is enabled. It is valid when Tx INT/DMA mode is enabled (D7 of CR1 = 1).

0: Disable
In this mode no Tx INT/DMA request occurs when transmission is enabled. The first Tx data cannot be written by using a Tx interrupt or DMA request in this mode. Therefore, the first Tx data must be directly written by the host processor immediately after transmission is enabled. Once the first Tx data is written, the Tx INT/DMA request becomes active when the Tx buffer is subsequently emptied.

1: Enable
When transmission is enabled in the AMPSC, the Tx INT/DMA request is activated. In this mode, all Tx data including the first Tx data can be transferred by using an interrupt or DMA request.

(6) D1 (Transmit Interrupt/DMA Enable)
This bit is used to transfer Tx data by using an interrupt or DMA request. It is related to the First Tx INT/DMA Enable bit (D2 of CR1), and affects occurrence of the Tx interrupt and Tx DMA request.

0: Disable
The Tx INT/DMA Request remains inactive if the Tx buffer is emptied. This mode is used to transfer Tx data by using status polling.

1: Enable
The Tx INT/DMA Request becomes activated when the Tx buffer is emptied. The activation timing differs depending on the value of the First Tx INT/DMA Enable bit. This mode is used to transfer Tx data by using an interrupt or DMA request.

(7) D0 (External/Status Interrupt Enable)
This bit determines whether or not an interrupt is to be requested when the E/S bit status has changed. It is related to the settings of the bits of CR11. For more information about the E/S interrupt, refer to subsections 3.2.13, Control Register 11 (CR11), 3.2.14, Control Register 12 (CR12), and 3.3.2, Status Register 1 (SR1).

0: Disable
In this case, no E/S interrupt occurs even if the E/S bit status has changed, regardless of the value of CR11. While no E/S bit latch operation occurs, the E/S bit of SR1 indicates the E/S cause status at the point of status change.

1: Enable
In this case, interrupts caused by each E/S change are enabled according to the bit status on CR11. Changes of the E/S bit are latched.

3.2.3 Control Register 2A (CR2A)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Vector Mode	Status Affects Vector	Output Vector Type			Priority Select	INT/DMA Mode	
Contents	0 Non-Vectored 1 Vectored	0 Fixed Vector 1 Status Affects Vector	0 0 0 Type A-1 0 0 1 Type A-2 0 1 0 Type A-3(Slave) 0 1 1 Type B-1 1 0 0 Type B-2(Slave)			0 TxA RxB 1 RxB TxA	0 0 Both Channel INT 0 1 Ch-A:DMA Ch-B:INT 1 0 Both Channel DMA 1 1 Use Prohibited	

(1) D7 (Vector Mode)
This bit determines the treatment of interrupt vectors. The value of this bit affects hardware configuration.

0: Non-Vectored
Interrupt vectors are obtained by reading the value of SR2B. In this mode the $\overline{\text{INTAK}}$ signal is not used.

1: Vectored
Interrupt vectors are obtained by using the $\overline{\text{INTAK}}$ signal.

(2) D6 (Status Affects Vector)
This bit determines whether the value of an interrupt vector is to be changed or not depending on the cause of interrupt.

0: Fixed Vector
The value of the interrupt vector is fixed. The vector value set in CR2B is directly output.

1: Status Affects Vector
Three bits of a vector value change depending on the cause of the interrupt. The bits that change differ depending on the value of the Output Vector Type bits (D5-D3 of CR2A). The values of the remaining bits set in CR2B are directly output.

(3) D5-D3 (Output Vector Type)
These bits are used to determine interrupt vector output operation and the bits of an interrupt vector which are to be changed if the Status Affects Vector is set by D6 of this register. The following shows the vector bits that are affected:

0 0 0: Type A-1 --- Bits V_4, V_3, and V_2 change.
0 0 1: Type A-2 --- Bits V_4, V_3, and V_2 change.
0 1 0: Type A-3 --- Bits V_4, V_3, and V_2 change.
0 1 1: Type B-1 --- Bits V_2, V_1, and V_0 change.
1 0 0: Type B-2 --- Bits V_2, V_1, and V_0 change.

(4) D2 (Priority Select)
This bit is used to determine the order of priority between Tx request on channel A and Rx request on channel B. The priority level for other requests are fixed. The priority is not applied to DMA transfer.

0: RxA > TxA > RxB > TxB > E/S A > E/S B
1: RxA > RxB > RxA > TxB > E/S A > E/S B
High ←——————————————→ Low

(5) D1-D0 (Interrupt/DMA Mode)
These bits are used to select the Tx/Rx data transfer mode on each channel. While E/S, Rx, and Special Rx Condition interrupts are enabled on the channel for which DMA mode is selected, the Tx interrupt is disabled on that channel.

0 0: Both Channel Interrupt
This mode allows transmit/receive data to be transferred by using interrupts on both channels A and B. An interrupt occurs when the Tx buffer is emptied or the Rx Data Available state is entered.

0 1: Channel A DMA, Channel B Interrupt
This mode allows transfer using DMA for channel A and interrupts for channel B.

1 0: Both Channels DMA
This mode allows transmit/receive data to be transferred by using DMA on both channels.

Table 3-3 shows the values of D2, D1, and D0 of CR2A and corresponding Tx/Rx data transfer modes, pin functions, and priority level.

Table 3.3 CR2A Register Values Versus Operation Modes

CR2A			Transfer mode of transmit/receive data		Pin function		Priority	Remarks
D2	D1	D0	Channel A	Channel B	$\overline{\text{DTRA}}$/ DRQTxB	$\overline{\text{DTRB}}$/ DRQRxB	High → Low	
0	0	0	INT	INT	$\overline{\text{DTRA}}$	$\overline{\text{DTRB}}$	RxA TxA RxB TxB E/SA E/SB	Interrupt priority
1	0	0	INT	INT	$\overline{\text{DTRA}}$	$\overline{\text{DTRB}}$	RxA RxB TxA TxB E/SA E/SB	Interrupt priority
X	0	1	DMA	INT	$\overline{\text{DTRA}}$	$\overline{\text{DTRB}}$	No priority	DMA priority
							RxA*>RxB>TxB >E/SA*>E/SB	Interrupt priority
X	1	0	DMA	DMA	DRQTxB	DRQRxB	No priority	DMA priority
							RxA*>RxB* >E/SA*>E/SB*	Interrupt priority

*: Rx and E/S interrupts will occur on the channel for which the DMA mode is selected, provided interrupts are enabled.

Note X: Don't care.

3.2.4 Control Register 2B (CR2B)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Initial Vector Value							
Contents	V7	V6	V5	V4	V3	V2	V1	V0

(1) D7-D0 (Initial Vector Value)

These bits are used to set the initial value of an interrupt vector.

3.2.5 Control Register 3 (CR3)

Bit / Function / Operation protocol	D7	D6	D5	D4	D3	D2	D1	D0
Function	Rx Character Bit Length		Auto Enable	Enter Hunt Phase	Rx CRC Enable	Address Search Mode	SYNC Character Load Inhibit/ Multicast	Rx Enable
Async	0 0 5 bits/char. 0 1 7 bits/char. 1 0 6 bits/char. 1 1 8 bits/char.		0 Disable 1 Enable	0	0	0	0	0 Disable 1 Enable
COP				0 No Operation 1 Enter Hunt Phase	0 Disable 1 Enable		"SYNC Char. Load Inhibit" 0 No Operation 1 Inhibit	
BOP						0 Disable 1 Enable	"Multicast" 0 No Operation 1 Enable	

(1) D7-D0 (Receive Character Bit Length)

These bits determine the number of bits per character when the received serial data is converted into parallel data. The number of bits can be changed while receiving serial data, however, the number should be changed before input of a character which has the new number of bits per character.

0 0: 5 bits/character
0 1: 7 bits/character
1 0: 6 bits/character
1 1: 8 bits/character

(2) D5 (Auto Enable)

This bit is used when performing Tx/Rx control using the $\overline{\text{CTS}}$/$\overline{\text{DCD}}$ pin and when displaying the Tx state using the $\overline{\text{RTS}}$ pin.

0: Disable

$\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ pins become input pins. If the state of these pins changes, it will be latched by the corresponding E/S bit (D5 and D3 of SR1) and an E/S interrupt will occur.

In this case, the RTS pin becomes an output pin and its output is controlled by the RTS Control bit (D1 of CR5).

1: Enable

The $\overline{\text{CTS}}/\overline{\text{DCD}}$ pin can be used for Tx/Rx control. For control methods, refer to Tables 1-3 and 1-4 of Chapter1, Pin Functions. Also, in this case, a change of the pin state is latched by the corresponding E/S bit and an E/S interrupt occurs.

The function of $\overline{\text{RTS}}$ pin changes depending on the operation protocol and state of RTS bit (refer to Table 1-5).

(3) D4 (Enter Hunt Phase; Valid in COP or BOP mode)

This bit directs the AMPSC to enter the Hunt Phase.

0: No operation

1: Enter Hunt Phase

The AMPSC enters the Hunt Phase and starts searching of the SYNC character (COP mode) or flag (BOP mode). When this bit is set, its function is retained until synchronization is established. After synchronization is established, this bit is automatically reset to 0 and the detection process stops.

(4) D3 (Receive CRC Enable: valid in COP or BOP mode)

This bit determines whether or not CRC calculation is to be done on received data.

0: Disable

CRC calculation is not done on received data.

1: Enable

CRC calculation is done on received data.

(a) COP mode

Since, in this mode, CRC calculation is started after 8-bit times have elapsed from when a character is received, the following control is required:

For example, assume that a character string, C_0, C_1, D_2,..... is received in this order (see Fig. 3.6). C_0 is first received and loaded into the Rx buffer, followed by C_1 which is shifted into the shift register in series. Whether or not CRC calculation is to be done on C_0 must be decided before the assembling for C_1 is completed (before C_1 is transferred to the Rx buffer). If the RxCRC Enable bit is set to one when C_1 is transferred to the Rx buffer, CRC calculation is done on C_0. If it is zero, CRC calculation is not done on C_0. This can be done by reading C_0 from the Rx buffer, checking the character value by the host processor, and then setting the value of this bit (RxCRC Enable bit). Since this control must be completed before all bits of C_1 reach the AMPSC, a sufficient time interval must be reserved for this control after the Rx Data Available state (D0 of SR0 = 1) is activated by C_0. Due to this control, received character buffering (for 3 bytes) using the Rx buffer is, as a rule, not available in the COP mode.

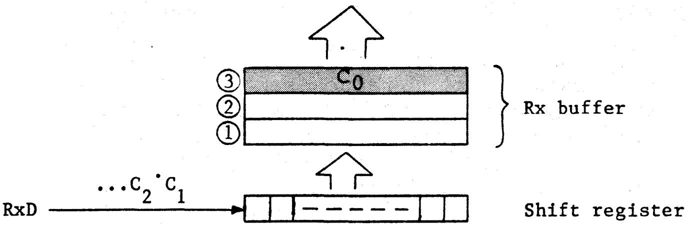


Fig.3.6 Character Reception

(b) BOP mode
Since, in this mode, all data within a frame are included in the CRC calculation, the RxCRC Enable bit should be kept at 1 during data reception.

(5) D2 (Address Search Mode; valid in BOP mode)
This bit determines whether or not the address field value of a received frame is to be compared with the value set in CR6.

0: Disable
Address field detection is not done. All frames are received unconditionally.

1: Enable

Address field detection is done. Operation in Auto Enable mode differs depending on the value of the Multicast bit (D1 of CR3).

(a) When D1 of CR3 = 0

If the address field value of a frame is equal to the value set in CR6 or equal to $\overset{\text{MSB}}{1}111111\overset{\text{LSB}}{1}$ (global address), the frame is received. In any other case, the device doensn't receive the frame and automatically returns to the Flag Detect state.

(b) When D1 of CR3 = 1

The higher 4 bits (D7-D4) of the address field are compared with bits D7-D4 of CR6. The operation in this mode is identical to that in the above case (D1 of CR3 = 0), with the exception that the bits to be compared are different.

In this mode, abort detection is not done unless the address that matches the value set in CR6 is detected. Once the address matches, abort detection is subsequently done unconditionally.

(6) D1 (Sync Character Load Inhibit/Multicast; valid in COP or BOP mode)

The meaning of this bit differs in the COP and BOP modes.

(a) COP mode

This bit serves a Sync Character Load Inhibit function. It determines the treatment of a character (within a received data block) whose data pattern is identical to that of the SYNC character (same as the value of CR6).

0: No Operation

If a character with a data pattern identical to the SYNC character is detected in a received data block, no special processing is done on that character.

1: SYNC Character Load Inhibit
Inhibits the SYNC character (for character synchronization) to be transferred to the Rx buffer as one of the received data items.
This bit is used with the Enter Hunt Phase bit (D4 of CR3) before the AMPSC starts a receive operation. The inhibit function remains valid after synchronization is established, so that the character pattern identical to the SYNC character pattern is not transferred to the Rx buffer and is not used for CRC calculation.

(b) BOP mode
This bit is used to enable/disable the Multicast function. It becomes valid when the Address Search mode is selected (D2 of CR3 = 1).
0: No Operation
The Multicast function remains ineffective.
1: Multicast
The Multicast function becomes effective. For details of Multicast operations, see the description for the Address Search Mode bit (D2 of CR3).

(7) D0 (Receive Enable)
This bit control receiver operations.
0: Disable
Disables receiver function.
1: Enable
Enables the receiver function to start a receive operation. When the Auto Enable mode is selected (D5 of CR3 = 1), the signal applied to the $\overline{\text{DCD}}$ pin also affects the start of a receive operation.

3.2.6 Control Register 4 (CR4)

Bit		D7	D6	D5	D4	D3	D2	D1	D0
Function / Operation protocol		Clock Rate		Protocol Mode		Tx stop Bits		Parity Select	Parity Enable
Async		0 0 X1 0 1 X16 1 0 X32 1 1 X64		Don't Care		0 0 Use Prohibited 0 1 1 1 0 1.5 1 1 2		0 Odd 1 Even	0 Disable 1 Enable
COP	Mono-sync	Don't Care		0 0		0 0			
	Bi-sync			0 1					
	External Sync	0 0 or 0 1		1 1					
BOP		Don't Care		1 0				0	0

(1) D7, D6 (Clock Rate), D5, D4 (Protocol Mode), D3, D2 (Stop Bits)

Combinations of these six bits are used to specify the operation protocol of the device.

(a) Asynchronous mode

In this mode, set bits D3 and D2 to 0, 1; 1, 0; or 1, 1 (which specify transmit stop bit lengths of 1, 1.5, and 2 bits, respectively). Bits D7 and D6 are used to select the clock rate as a multiple of baud rate.

0 0: x1
0 1: x16
1 0: x32
1 1: x64

When the Asynchronous mode is selected, bits D5 and D4 remain ineffective.

(b) COP mode

Set bits D3 and D2 to 0, 0. Any of the following three modes can be selected with bits D7-D4:

(i) Mono-sync mode

Bits D7 and D6 can be set to any value. Set D5 and D4 to 0, 0.

The SYNC character bit length is selectable from 6 and 8 bits (with D0 of CR10).

(ii) Bi-sync mode

Bits D7 and D6 be set to any value. Set D5 and D4 to 0, 1.

The SYNC character bit length is selectable as 12 or 16 bits (with D0 of CR10).

(iii) External Sync mode

Set bits D7 and D6 to 0, 0 or 0, 1 and D5 and D4 to 1, 1. This mode is used to establish character synchronization using an external synchronization signal. A Low level input to the SYNC pin is used to establish synchronization. For the timing, see the description of the SYNC pin.

This mode is selectable only when the Xtal Select bit (D7 of CR15) is set to zero.

(c) BOP mode

Bits D7 and D6 can be set to any value. Set D5 and D4 to 1, 0 and D3 and D2 to 0, 0. This setting selects the ordinary HDLC/SDLC operation. If the Loop Select bit (D1 of CR10) is set to 1, the Loop mode is selected.

(2) D1 (Parity Select; valid in Async and COP modes)

This bit selects parity type. It is valid only when the Parity Enable bit (D0 of CR4) is set to 1.

0: Odd parity

1: Even parity

(3) D0 (Parity Enable)

This bit enables the device to add a parity bit to Tx data and do a parity check on Rx data.

0: Disable

Disable the parity function.

1: Enable

Enable the parity function. The type of parity is selected with the Parity Select bit (D1 of CR4). If the received data length is 7 bits or less (excluding the parity bit), the parity bit can be read as part of the received data.

3.2.7 Control Registe 5 (CR5)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function / Operation protocol	DTR Control	Tx Character Bit Length		Send Break/ Abort	Tx Enable	CRC Polynomial	RTS Control	Tx CRC Enable
Async	0 DTR= "H" 1 DTR= "L"	0 0 5 or less bits/char. 0 1 7 bits/ char. 1 0 6 bits/ char. 1 1 8 bits/ char.		"Send Break" 0 No Operation 1 Send Break	0 Disable 1 Enable	0	0 RTS= "H" (at All Sent) 1 RTS= "L"	0
COP						0 CRC-CCITT 1 CRC-16	0 RTS= "H" 1 RTS= "L"	0 Disable 1 Enable
BOP				"Send Abort" 0 No Operation 1 Send Abort		0		

(1) D7 (DTR Control)
This bit controls the $\overline{\text{DTR}}$ pin status.
0: $\overline{\text{DTR}}$ pin = "H"
1: $\overline{\text{DTR}}$ pin = "L"

(2) D6-D5 (Transmit Character Bit Length)
These specify the bit count per character in serial Tx data, as follows:
0 0: 5 or less bits/character
0 1: 7 bits/character
1 0: 6 bits/character
1 1: 8 bits/character
If the bit count per character is 6 or 7 bits, down to the lower 6 or 7 bits of written Tx data become valid, respectively. Any value may be written for the higher 2 or 1 bits. If the bit count per character is 5 bits or less, the parallel data format as shown in Table 3-4 must be used when writing.

Table 3-4 Parallel Data Format for 5 Bits or Less Per Chracter

No.of bits	D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	1	1	0	0	0	D_0
2	1	1	1	0	0	0	D_1	D_0
3	1	1	0	0	0	D_2	D_1	D_0
4	1	0	0	0	D_3	D_2	D_1	D_0
5	0	0	0	D_4	D_3	D_2	D_1	D_0

Dn:Effective data bit

(3) D4 (Send Break/Abort)

This bit controls break or abort transmission. The function of this bit depends on the selected operation mode.

(a) Asynchronous or COP mode

This bit provides the Send Break function.

0: No Operation

If this bit is set to 1 and then reset to zero, the TxD pin is placed in the mark state ("H"). If the bit remains zero, it causes no operation.

1: Send Break

The TxD pin is placed in Send Break state (continuous "L" transmission), in which the Tx data in the Tx buffer becomes invalid. This bit remains valid even if transmission is disabled. The time interval in which this bit is kept at 1 must be longer than 1-character period.

If the Tx on Loop is specified (D4 and D1 of CR10 are 1) in the COP mode (D5, D4 of CR4 = 0, 0 or 0, 1 and D3, D2 = 0, 0), this bit, if set to 1, is automatically reset to zero immediately when synchronization is established between the receiver and transmitter (for details, refer to the description of the CR10).

(b) BOP mode

This bit enables/disables the Send Abort function.

0: No Operation

This bit causes no operation. If it is set to 1, it need no be reset to zero.

1: Send Abort
Starts a send abort sequence (8 consecutive 1s). Up to 13 ones may be sent depending on the preceding seial data pattern. The send abort operation invalidates the Tx data in the Tx buffer. The send abort sequence is followed by flag transmission. This bit is automatically reset to zero when the send abort sequence is completed.

(4) D3 (Transmit Enable)
This bit controls transmitter operations.
0: Disable
Disables transmitter function. If this bit is temporarily set to 1 to start transmission and then reset to zero, the transmitter enters the Mark state after completing the current data transmission. If the bit is reset during CRC character transmission, a SYNC character or flag is sent instead of the CRC character.
If this bit is reset in COP or BOP mode, the Tx Underrun/EOM bit (D6 of SR1) is set.
While the TxD pin is normally placed in the Mark state when the transmitter is disabled, it functions differently during the Echo Loop Test (D3 of CR14 = 1) or BOP Loop mode (D5, D4 of CR4 = 1, 0 and D4, D1 of CR10 = 1,1).
1: Enable
Enables the transmitter function to start trasmission. If the Auto Enable mode is selected (D5 of CR3 = 1), the signal applied to the $\overline{CTS}$ pin affects the transmit operation.

(5) D2 (CRC Polynominal; valid in COP or BOP mode)
This bit selects the polynominal used for CRC calculation. The initial state of the Tx/Rx CRC calculation differs depending on the value of bit D7 of CR10.

(a) COP mode

0: CRC-CCITT

In this case the polynominal expression is $X^{16} + X^{12} + X^{5} + 1$.

1: CRC-16

In this case the polynominal expression is $X^{16} + X^{15} + X^{2} + 1$.

(b) BOP mode

0: CRC-CCITT

Only the generation polynominal, $X^{16} + X^{12} + X^{5} + 1$, is selectable.

(6) D1 (RTS Control)

This bit controls the RTS pin. The function of this bit in Auto Enable mode of Async (D5 of CR3=1) differs from that of the other modes as shown in Table 3-5.

Table 3-5 RTS Control Bit and $\overline{RTS}$ Pin Status

Function Protocol	Auto Enable bit	RTS Cont bit	$\overline{RTS}$ pin status
Async	0	0	H
		1	L
	1	When 0 from the beginning	H
		When set to 1 and then to 0	Remains "L" while All Sent* = 0 and becomes. 1 when changed to "H"
		1	L
COP/BOP	X	0	H
		1	L

X:Don't Care

*:SR1:D2

(7) D0 (Transmit CRC Enable: valid in COP or BOP mode)

This bit determines whether or not the CRC calculation is to be done on Tx data.

0: Disable

CRC calculation is not done on Tx data.

1: Enable

CRC calculation is done on Tx data.

(a) COP mode
When a character written into the Tx buffer is transferred to the Tx shift register, the value of this bit is used to determine whether or not CRC calculation is to be done on that character. this bit must be manipulated before the Tx character is written in to the AMPSC.

(b) BOP mode
Since, in this mode, CRC calculation is done on all Tx data, this bit should normally be set to 1 (Tx CRC Enable).
The CRC character is sent if a Tx Underrun/EOM state occurs when TxCRC is enabled, provided the Tx Underrun/EOM bit (D6 of SR1) had been set to zero before. If the Tx Underrun/EOM bit is set to 1 before the Tx Underrun/EOM state occurs, the SYNC character (COP mode) or a flag (BOP mode) will be sent instead of the CRC character.

3.2.8 Control Register 6 (CR6)

		Bit	D7	D6	D5	D4	D3	D2	D1	D0
Operation protocol		Function	SYNC Character/Address							
Async										
COP	Mono-sync (Tx SYNC)	6	SYNC1	SYNC0	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
		8	SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
	Bi-sync	12	SYNC3	SYNC2	SYNC1	SYNC0	1			
		16	SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
	Ext. Sync	6	SYNC1	SYNC0	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
		8	SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
BOP	Address Search (Normal)		ADRS7-ADRS0							
	Address Search (Multicast)		ADRS7-ADRS4				Don't Care			

(1) D7-D0 (SYNC Caracter/Address; valid in COP or BOP mode)
These bits specify the SYNC character pattern or address value. The value of these bits differs depending on the selected operation mode.

(a) COP mode
In the Mono-sync or External Sync mode, the transmit SYNC character pattern is set in these bits. In the Bi-sync mode, a part of the SYNC character pattern is set in them.

(b) BOP mode
The value set in these bits differs depending on the Address Serch Mode bit (D2 of CR3) and Multicast bit (D1 of CR3) as shown in Table 3-6.

Table 3-6 CR6 Value in BOP Mode

Address Search Mode bit	Multicast bit	CR6 setting
0	0 or 1	Invalid
1	0	Address value (ADRS7-ADRS0)
	1	Higher 4 bits of address value (ADRS7-ADRS4, lower 4 bits settings are invalid)

3.2.9 Control Register 7 (CR7)

		Bit	D7	D6	D5	D4	D3	D2	D1	D0
Operation protocol		Function	SYNC Character/Flag							
Async										
COP	Mono-sync (Rx SYNC)	6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	Don't Care	
		8	SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
	Bi-sync	12	SYNC11	SYNC10	SYNC9	SYNC8	SYNC7	SYNC6	SYNC5	SYNC4
		16	SYNC15	SYNC14	SYNC13	SYNC12	SYNC11	SYNC10	SYNC9	SYNC8
	Ext. Sync									
BOP			Flag(01111110)							

(1) D7-D0 (SYNC Character/Flag; valid in COP or BOP mode)
These bits specify the SYNC character pattern or flag. The value to be set in these bits diffes depending on the selected operation mode.

(a) COP mode
In the Mono-sync mode, the receive SYNC chatacter pattern is set in these bits. In the Bi-sync mode, a part of the SYNC character is set in them. These bits are not used in the External Sync mode.

(b) BOP mode
The flag pattern $(\overset{MSB}{0}111111\overset{LSB}{0})$ is set in these bits.

3.2.10 Control Register 8 (CR8)

Bit function Operation protocol	D7	D6	D5	D4	D3	D2	D1	D0
	Tx Data Length L-Byte							
Async								
COP								
BOP	Tx Data Length Bit7-Bit0							

(1) D7-D0 (Transmit Data Length L-Byte; valid in BOP mode)
These bits specify the lower byte (bits 7-0) of the Tx data count. This register is used with CR9. It must be set before the Tx Data Length Counter Enable bit (D1 of CR13) and Tx Enable bit (D3 of CR5) are set.

3.2.11 Control Register (CR9)

Bit function Operation protocol	D7	D6	D5	D4	D3	D2	D1	D0
	Tx Data Length H-Byte							
Async								
COP								
BOP	Tx Data Length Bit15-Bit8							

(1) D7-D0 (Transmit Data Length H-Byte; valid in BOP mode)
These bits specify the higher byte (bits 15-8) of the Tx data count. This register is paired with CR8. Other than the setting of the higher byte, the rules are the same as those for CR8.

3.2.12 Control Register 10 (CR10)

Bit Function Operation protocol	D7	D6	D5	D4	D3	D2	D1	D0
	CRC Initialized Condition	Data Format		Auto Tx on Sync/ Tx on Loop	Idle Condition	Tx Condition on underrun	D4/Loop Enable	SYNC Character Bits
Async	0	0 0 NRZ 0 1 NRZ1 1 0 FM1 1 1 FM0		0			0	0
COP	0 All "0" 1 All "1"			"Auto Tx on Sync" 0 Disable 1 Enable	0	0	"D4 Enable" 0 Disable 1 Enable	0 8bits 1 6bits
BOP				"Tx on Loop" 0 Disable 1 Enable	0 Flag 1 Mark	0 Normal 1 Abort	"Loop Enable" 0 Disable 1 Enable	0

(1) D7 (CRC Initialized Condition; valid in COP or BOP mode)

This bit specifes the initial state of the CRC calculation circuit. The CRC calculation circuit is initialized to the state specified by this bit when the Initialize Tx/Rx CRC Calculator command (D7, D6 of CR0 = 1, 0/0, 1) is issued.

0: All zeros

The CRC calculation circuit is initialized into all zeros by the Initialize command.

1: All ones

The CRC calculation circuit is initialized into all 1s by the Initialize command.

(2) D6-D5 (Data Format)

These bits specify the serial data format. Setting these bits enables the corresponding encoder/decoder.

0 0: NRZ
0 1: NRZI
1 0: FM1
1 1: FM0

If these bits are set to 00 (NRZ), it is possible to decode Manchester encoded data by setting the DPLL mode to FM (D7, D6, D5 of CR14 = 1, 1, 0).

Fig. 3-7 shows the data formats and Table 3-7 shows the operation for each data format.

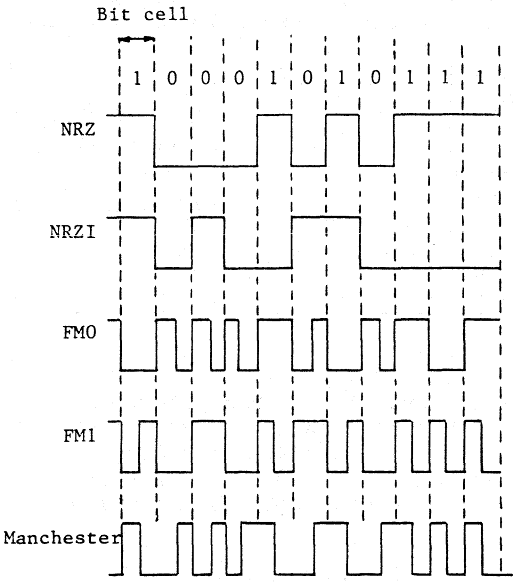


Fig. 3-7 Data Formats

Table 3-7 Operations for Each Data Format

Format	Operation
NRZI	All data bits which are zero are inverted into 1s, all data bits which are 1 are not affected.
FM0	Data value is always inverted at the beginning of a bit cell. If a data bit is zero, it is again inverted at the center of the bit cell; if the data bit is 1, it keeps the same status.
FM1	Data value is always inverted at the beginning of a bit cell. If a data bit is 1, it is again inverted at the center of the bit cell; if the data bit is zero, it keeps the same status.

Manchester	The first half of a bit cell takes on the same value as a data bit, and is inverted at the center of the bit cell. Therefore, the second half of the bit cell has a value which is an inverse of the data bit. This means that if the data bit is zero, the center of the bit cell rises, while if the data bit is one, the center of the bit cell falls.

(3) D4 (Auto Tx on Sync/Tx on Loop; valid in COP or BOP mode)

The function of this bit deffers depending on the operation mode. In the COP mode, it is used to synchronize the receiver with transmitter. In the BOP mode, it is used for SDLC loop operation control. This bit is valid only when the D4/Loop Enable state (D1 of CR10 = 1) is selected.

(a) COP mode

This bit provides the Auto Tx on Sync function to synchronize a receiver operation with a transmitter operation. It must be set before transmitter and receiver are enabled.

0: Disable

Disables the Auto Tx on Sync function (D1 of CR10 = 1).

1: Enable

If this bit is set ot 1 along with the D4 Enable bit, the transmitter is disabled, and the receiver enters the Hunt Phase. The receiver subsequently starts serching for the SYNC character set in CR6. When the SYNC character is detected, character synchronizaion is established, and the transmitter is enabled. Data transmission is thus enabled. Establishment of character synchronization can be acknowledged from the Tx Sync/on Loop bit (D1 of SR10) which is set.

Once sychronization is established after this bit is set to 1, resetting it to zero does not affect synchronization.

(b) BOP mode

This bit enables/disables the Tx on Loop secondary station function. It is used for data transmission during the SDLC loop operation.

0: Disable

Once the AMPSC forms a loop and starts transmission, this bit must be reset to zero. This allows the CRC and flag to be automatically transmitted if a Tx Underrun/EOM state occurred, and the AMPSC to be subsequently placed again in the Loop mode with 1-bit delay.

This bit must be reset before CRC transmission is completed after the first Tx data is written into the Tx buffer.

1: Enable

Setting this bit to 1 when the Loop Enable bit (D1 of CR10) is 1 selects the SDLC Loop Operatin mode, in which the RxD input is connected to the TxD output within the AMPSC to form a loop. At this point the data input to RxD is transferred directly to TxD without a 1-bit delay and GA (Go-Ahead) pattern ($\overset{LSB}{0}111111\overset{MSB}{1}$) detection is started. If the GA pattern is detected, a 1-bit delay is inserted, and GA pattern search is continued. While, at this point, the transmitter is still disabled and performs no transmit operation, the receiver can be controlled at the user's option.

If the GA pattern is subsequently detected again, the transmitter within the AMPSC is enabled. At this point the GA pattern is automatically transformed into a flag, so that the data in the Tx buffer, if any, may be transmitted following this flag.

Once transmission is started, this bit must be reset (see the description regarding Disable).

(4) D3 (Idle Condition; valid in BOP mode)

This bit determines the type of information to be transmitted following two Ending flags or an Abort.

0: Flag

Flags are transmitted.

1: Mark

A mark (consecutive 1s) is transmitted.

(5) D2 (Transmit Condition on Underrun; valid only in BOP mode)
This bit determines the type of information to be transmitted in the event of the Tx Underrun/EOM state.

0: Normal
If the Tx Underrun/EOM bit (D6 of SR1) is zero, the CRC and flags are consecutively transmitted. If the Tx Underrun/EOM bit is 1 or TxCRC is desabled (D0 of CR5 = 0), only the flags are transmitted.

1: Abort
Abort is transmitted regardless of the Tx Underrun/EOM bit status. The Abort transmission is followed by a flag transmission.

(6) D1 (D4/Loop Enable; valid in COP or BOP mode)
This bit determines whether the Auto Tx on Sync/Tx on Loop bit (D4 of CR10) is to be validated or not. The function of bit D4 is controlled by the status of this bit. The function of this bit differs depending on the operation mode. It must be set before transmitter and receiver are enabled.

(a) COP mode
Serves as the D4 Enable function.

0: Disable
Disables the Auto Tx on Sync function. Once synchronization is established, this bit does not affect the synchronization if reset to zero.

1: Enable
Enables the Auro Tx on Sync function. For the operation in this state, see the description of the Auto Tx on Sync bit.

(b) BOP mode
Serves as the Loop Enable function. This bit controls the SDLC Loop operation. Used with the Tx on Loop bit (D4 of CR10), this bit controls the loop.

0: Disable
Disables the loop function. If this bit is set to zero after the device entered a Loop operation, the TxD output is desconnected from the RxD input to open the SDLC Loop as soon as the GA pattern is detected. The TxD pin is subsequently set to "H".

1: Enable
Enables the SDLC Loop to be formed. Setting only this bit to 1 connects the RxD input to TxD output within the AMPSC to form a Loop. In this case a 1-bit delay is not inserted, so that the input data to the RxD input is directly output to the next station through TxD.
If the Tx on Loop bit is set in this state, the device starts GA pattern detection. For details of this operation, see the description of the Tx on Loop bit.

(7) D0 (SYNC Character Bits; valid in COP mode)
This bit determines the number of bits per SYNC character.

0: 8 bits
Specifies a SYNC character length of 8 bits. This means that the SYNC character length is 8 bits in the Mono-sync mode, and is 16 bits in the Bi-sync mode.

1: 6 bits
Specifies a SYNC character length of 6 bits. This means that the SYNC character length is 6 bits in the Mono-sync mode, and is 12 bits in the Bi-sync mode.

3.2.13 Control Register 11 (CR11)

Bit		D7	D6	D5	D4	D3	D2	D1	D0
Function Operation protocol		Break/ Abort/ GA IE	Tx Underrun/ EOM IE	CTS IE	Sync/ Hunt IE	DCD IE	All Sent IE	Idle Detect IE	BRG IE
Start-stop		"Break" 0 Disable 1 Enable	0	0 Disable 1 Enable	0 Disable 1 Enable	0 Disable 1 Enable	0 Disable 1 Enable	0	0 Disable 1 Enable
COP			0 Disable 1 Enable				0		
BOP	Normal	"Abort" 0 Disable 1 Enable					0 Disable 1 Enable	0 Disable 1 Enable	
	Loop	"Abort/ GA" 0 Disable 1 Enable							

Each bit of CR11 controls E/S interrupt requests which are generated by the E/S interrupt caused in the AMPSC. They are valid if the E/S INT is enabled (D0 of CR1 = 1). For details of the cause of interrupt assigned to each bit, refer to the description of SR1.

(1) D7 (Break/Abort Interrupt Enable)

(a) Async/COP mode

This bit serves as the Break IE function.

0: Disable

Disables E/S interrupt caused by detection/release of break state.

1: Enable

Enalbes E/S interrupt caused by detection/release of Break state.

(b) BOP mode

The function of the bit differs between the Normal BOP mode (HDLC/SDLC) and SDLC Loop mode:

(i) HDLC/SDLC mode

This bit serves as the Abort IE function.

0: Disable

Disables E/S interrupt caused by Abort detection/release.

1: Enable

Enables E/S interrupt caused by Abort detection/release.

(ii) Loop mode

This bit provides Abort/GA IE function.

0: Disable

Disables the E/S interrupt caused by detection/ release of an Abort or GA.

1: Enable

Enables the E/S interrupt caused by detection/ release of an Abort or GA.

(2) D6 (Transmit Underrun/End of Message Interrupt Enable; valid in COP or BOP mode)

0: Disable

Disables E/S interrupt caused by the Tx Underrun/EOM state.

1: Enable

Enables E/S interrupt caused by the Tx Underrun/EOM state.

(3) D5 (CTS Interrupt Enable)

0: Disable

Disables E/S interrupt caused by CTS pin status changes.

1: Enable
Enables E/S interrupt caused by $\overline{\text{CTS}}$ pin status changes.

(4) D4 (SYNC/Hunt Interrupt Enalbe)
0: Disable
Disables E/S interrupt caused by SYNC/Hunt status changes.
1: Enable
Enables E/S interrupt caused by SYNC/Hunt status changes.

(5) D3 (DCD Interrupt Enable)
0: Disable
Disables E/S interrupt caused by $\overline{\text{DCD}}$ pin status changes.
1: Enable
Enables E/S interrupt caused by $\overline{\text{DCD}}$ pin status changes.

(6) D2 (All Sent Interrupt Enable; valid in Async or BOP mode)
0: Disable
Disables E/S interrupt caused by All Sent state generation.
1: Enable
Enables E/S interrupt caused by All Sent state generation.

(7) D1 (Idle Detect Interrupt Enable; valid in BOP mode)
0: Disable
Disables E/S interrupt caused by Idle detection/release.
1: Enable
Enables E/S interrupt caused by Idle detection/release.

(8) D0 (BRG Interrupt Enable)
This bit is valid only if Tx/Rx BRG is enabled (D1, D0 of CR14 = "1, 1").It determines whether or not an E/S interrupt is to be requested when the BRG count reaches zero.

0: Disable
Disables E/S interrupt even if the BRG count reaches zero.

1: Enable
Enables E/S interrupt if the BRG count reaches zero.

In relation to this interrupt, see the descriptions of CR12, CR14, SR1, and SR3.

3.2.14 Control Register 12 (CR12)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	BRG Select for TRxC	BRG Select for DPLL			TxBRG IE	RxBRG IE	TxBRG Register Set	RxBRG Register Set
Contents	0 RxBRG 1 TxBRG	0 RxBRG 1 TxBRG	0		0 Disable 1 Enable	0 Disable 1 Enable	0 No Operation 1 TxBRG Register Set	0 No Operation 1 RxBRG Register Set

(1) D7 (BRG Select for $\overline{\text{TRxC}}$)
This bit selects the source of the BRG clock which is output at the $\overline{\text{TRxC}}$ pin of the device. It is valid only when the $\overline{\text{TRxC}}$ pin is specified for output (D2 of CR15 = 1) and the BRG is selected for the source of the clock at the $\overline{\text{TRxC}}$ pin (D1, D0 of CR15 = 1, 0).

0: RxBRG
The RxBRG clock is output at the $\overline{\text{TRxC}}$ pin.

1: TxBRG
The TxBRG clock is output at the $\overline{\text{TRxC}}$ pin.

(2) D6 (BRG Select for DPLL)

This bit selects the clock source for the AMPSC's on-chip DPLL circuit, from TxBRG and RxBRG. It is valid when the BRG clock is selected as the clock for the DPLL circuit (D7, D6, D5 CR14 = 1, 0, 0).

0: RxBRG

The RxBRG clock serves as the clock source for the DPLL.

1: TxBRG

The TxBRG clock serves as the clock source for the DPLL.

(3) D3 (Transmit BRG Interrupt Enable)

This bit determines whether or not an E/S interrupt is to be issued when the TxBRG count value reaches zero. It is valid when the BRG IE bit is set (D0 of CR11 = 1).

0: Disable

No E/S interrupt is issued even if the TxBRG count value reaches zero. The status of the BRG Zero Count bit (D0 of SR1) and TxBRG Zero Count bit (D4 of SR3) also do not change.

1: Enable

An E/S interrupt is issued when the TxBRG count value reaches zero, and the BRG Zero Count and TxBRG Zero Count bits are both set to one.

(4) D2 (Receive BRG Interrupt Enable)

This bit determines whether or not an E/S interrupt is to be issued when the RxBRG count value reaches zero. It is valid when the BRG IE bit is set (D0 of CR11 = 1).

0: Disable

No E/S interrupt occurs even if the RxBRG count value reaches zero. The BRG Zero Count bit and the RxBRG Zero Count bit (D3 of SR3) are both set to one.

1: Enable

An E/S interrupt occurs when the RxBRG count value reaches zero, and the BRG Zero Count and RxBRG Zero Count bits are both set to one.

(5) D1 (Transmit BRG Register Set)
This bit is used to set a count value into the TxBRG register.

0: No operation
No write cycle for the TxBRG register is provided.

1: TxBRG Register Set
The first two control data write cycles after this bit is set to one are used to write a count value into the TxBRG register. The lower byte is written in the first write cycle, and the upper byte is written in the second write cycle. This bit is automatically reset when a value is set into the registers.
Register setting should be done when the TxBRG down counter is not oprating, or, in other words, in the TxBRG Disable state (D0 of CR14 = 0). If a register write is attempted while the down counter is operating, the value will not be loaded into the counter until the counter value reaches zero.
Setting both this bit and bit D0 of CR12 (RxBRG Register Set) at the same time is not allowed.

(6) D0 (Receive BRG Register Set)
This bit is used to set a count value into the RxBRG register.

0: No operation
No write cycle for the RxBRG register is provided.

1: RxBRG Register Set
The first two control word write cycles available after this bit is set are used to write a count value into the RxBRG registers. The lower byte is Written in the first write cycle, and the upper byte is written in the second write cycle. This bit is automatically reset when a count value is set into the register. Register setting should be done when the RxBRG down counter is not oprating, or, in other words, in the RxBRG Disable state (D1 of CR14 = 0). If a write operation is attempted while the down counter is operating, the value will not be loaded

until the counter value reaches zero.
Setting both this bit and bit D1 of CR12 (TxBRG Register Set) at the same time is not allowed.

3.2.15 Control Register 13 (CR13)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function							TxDLC Enable	Stand-by Mode Set
Contents	0						0 Disable 1 Enable	0 No Operation 1 Stand-by Mode Set

(1) D1 (Transmit Data Length Counter Enable; valid in BOP mode)
This bit is used for TxDLC control. It enables or disables comparison of the Tx Data Length in CR8 and CR9 with the value of the TxDLC (SR8, SR9).
0: Disable
Disables the TxDLC function and comparison of its contents with the TxDL.
1: Enable
Enables the TxDLC function to start comparison of its contents with the TxDL. After initialization, this bit must be set before the transmitter is enabled (before transmission is started). If this bit is set, the TxDLC is incremented by one each time the TxINT/DMA request is activated. When the value of the TxDLC equals the TxDL value, the subsequent TxINT/DMA requests are masked, which causes counter increment operation to be stopped (counter value is preserved). If it is desired to activate the TxDLC in the next frame, this bit must be set up again (this set-up operation will clear the counter). The resetting of this bit chould be done after the current frame is completed (after at

least one Ending flag is output from the TxD pin). If the Tx Underrun/EOM state (D6 of SR1 = 1) occurs when the TxDLC is enabled, the value of the TxDL is compared with that of the TxDLC. If the two values are identical, the current frame wil be successfuly completed after a CRC and flag are output. If they do not match, an Send Abort operation will be automatically executed, with TxINT/DMA requests masked. In this case the TxDLC value is preserved and the Sending Abort bit (D1 of SR0) is set to indicate that an Abort was automatically sent. The mask on TxINT/DMA requests is cleared by enabling the TxDLC again, which clears the TxDLC.

(2) D0 (Standby Mode Set)

This bit is used to place the AMPSC in the Standby mode.

0: No Operation

This condition results in no operation. The Standby mode, if set, can not be cleared by manipulating this bit but by writing "00H" into CR0.

1: Standby Mode Set

Places the AMPSC in the Standby mode, in which no external clock input other than the system clock (CLK) is input. This bit is automatically reset when the Standby mode is cleared.

For the Status of CRs and SRs in the Standby mode, see Table 3-2; for the pin status in Standby mode, see Table 3-8.

Table 3-8 Pin Status in Standby Mode

Pin name	I/O	Pin status	Pin name	I/O	Pin status
$\overline{WR}$	I	—	TxDA, TxDB	O	Retains the current state
$\overline{RD}$	I		RxDA, RxDB	I	--
B/$\overline{A}$	I		$\overline{TRxCA}$, $\overline{TRxCB}$	I/O	High impedance
C/$\overline{D}$	I		$\overline{RTxCA}$, XI1A $\overline{RTxCB}$, XI1B	I	—
D7-D0	I/O	High impedance			
$\overline{INT}$	O	Retains the current state	XI2A/$\overline{SYNCA}$ XI2B/$\overline{SYNCB}$	I/O	High impedance
$\overline{INTAK}$	I	—			
$\overline{PRI}$	I		$\overline{RTSA}$, $\overline{RTSB}$	O	Retains the current state
$\overline{PRO}$	O	Depends on $\overline{PRI}$	$\overline{CTSA}$, $\overline{CTSB}$	I	—
DRQTxA	O	Retains the current state	$\overline{DCDA}$, $\overline{DCDB}$	I	
DRQRxA	O				
$\overline{DTRA}$/DRQTxB $\overline{DTRB}$/DRQRxB	O				

During the Standby mode, the input and input/output pins must be controlled as shown in Table 3-9.

Table 3-9 Input and Input/Output Pin Control During Standby Mode

Pin name	Control
$\overline{WR}$	Fixed to "H"
$\overline{RD}$	
XI2A/$\overline{SYNCA}$ XI2B/$\overline{SYNCB}$	Fixed to "H" or "L"
$\overline{CTSA}$, $\overline{CTSB}$	
$\overline{DCDA}$, $\overline{DCDB}$	
others	Don't Care

The Standby mode is cleared when "00" is written into the control register 0 ($\overline{WR}$= "L")

3.2.16 Control Register 14 (CR14)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	DPLL Command			Local Self Test	Echo Loop Test	BRG Source Select	RxBRG Enable	TxBRG Enable
Contents	0 0 0 No Operation 0 0 1 Enter Search 0 1 0 Reset Missing Clock 0 1 1 DPLL Disable 1 0 0 DPLL Source BRG Select 1 0 1 DPLL Source Xtal/STRxC Select 1 1 0 FM Mode 1 1 1 MRZI Mode			0 No Operation 1 Enable	0 No Operation 1 Enable	0 Xtal/ STRxC 1 System Clock	0 Disable 1 Enable	0 Disable 1 Enable

(1) D7, D6, D5 (DPLL Command)

These bits are used to control the DPLL circuit. When they are reset, the DPLL circuit is disabled, clock source input is assigned to the $\overline{\text{STRxC}}$ pin, and NRZI mode selected.

0 0 0: No operation

0 0 1: Enter Search

This command causes the DPLL circuit to start detection of edges in received data. It activates the DPLL circuit to start sampling the Rx data input. The DPLL circuit operation differs depending on the data format used.

0 1 0: Reset Missing Clock

This command is valid only when the FM mode is selected. It is used to reset the Clock Missing bits (D7 and D6 of SR10).

0 1 1: DPLL Disable

This command is used to stop the DPLL circuit operation. It also resets the Clock Missing bits (D7 and D6 of SR10).

1 0 0: DPLL Source BRG Select
This command allows the use of the BRG as the clock source for the DPLL. Whether the TxBRG of the RxBRG is to be used as the clock source is determined by the BRG Select for DPLL bit (D6 of CR12).

1 0 1: DPLL Source Xtal/$\overline{\text{STRxC}}$ Select
This command is issued when a crystal oscillator or a clock applied to the $\overline{\text{STRxC}}$ pin is to be used as a source clock for the DPLL. Selection between the crystal OSC and $\overline{\text{STRxC}}$ input is specified by the Xtal Select bit (D7 of CR15).

1 1 0: FM Mode
This command is issued when received serial data is to be treated in the FM data format.

1 1 1: NRZI Mode
This command is issued when received serial data is to be treated in the NRZI data format.

(2) D4 (Local Self Test)
This bit is used for local self test.

0: No operation

1: Enable
The transmitter output is directly connected to the receiver input within the AMPSC to loop back Tx data to the receiver (Fig. 3-8). External data applied to the RxD pin will be ignored. Once this test mode is selected, transmitter/receiver control from the $\overline{\text{CTS}}$/$\overline{\text{DCD}}$ inputs becomes unavailable.

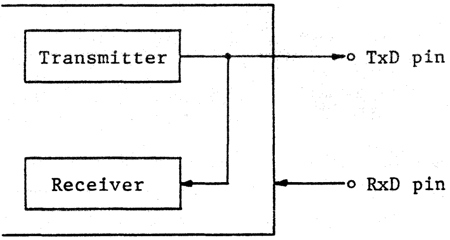


Fig. 3-8 Local Self/Test Mode

(3) D3 (Echo loop Test)
This bit is used to echo back received data directly to the remote sender. It is used for line tests.

0: No operation

1: Enable

The RxD input pin is connected to the TxD pin within the AMPSC so that received data is echoed back to the sender via the TxD pin (Fig. 3-9). In this case the AMPSC's transmitter function is invalidated, with transmitter control by $\overline{\text{CTS}}$ input disabled.

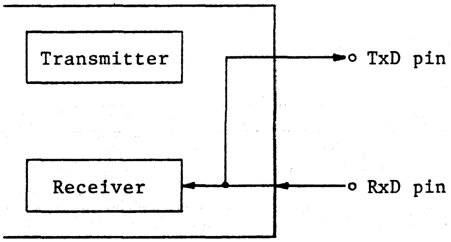


Fig. 3-9 Echo Loop Test Mode

(4) D2 (BRG Source Select)
This bit selects the source clock for the on-chip BRG. The selected clock source is shared for both the TxBRG and the RxBRG.

0: Xtal/$\overline{\text{STRxC}}$
If bit D7 of CR15 is set to one, the output of the crystal oscillator is used for the source clock for the BRG. If the same bit is set to zero, the clock applied to the $\overline{\text{STRxC}}$ pin of the device is used for the source clock.
1: System Clock
The system clock is used for the source clock for the BRG.

(5) D1 (Receive BRG Enable)
This bit is used for RxBRG control.
0: Disable
Disables the RxBRG, inhibiting count operation.
1: Enable
Enables the RxBRG to start a count operation. It takes 2 count periods after this bit is set before the receive clock is output.

(6) D0 (Transmit BRG Enable)
This bit is used for TxBRG control.
0: Disable
Disables the TxBRG, inhibiting count operation.
1: Enable
Enables the TxBRG to start count operation. It takes 2 count periods after this bit is set before the transmit clock is output.

3.2.17 Control Register 15 (CR15)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Xtal Select	RxCLK Select		TxCLK Select		$\overline{\text{TRxC}}$ Input/ Output	$\overline{\text{TRxC}}$ Source Select	
Contents	0 No Xtal	0 0 $\overline{\text{STRxC}}$ Input		0 0 $\overline{\text{STRxC}}$ Input		0 Input	0 0 Xtal	
		0 1 $\overline{\text{TRxC}}$ Input		0 1 $\overline{\text{TRxC}}$ Input		1 Output	0 1 TxCLK	
	1 Xtal	1 0 RxBRG Clock		1 0 TxBRG Clock			1 0 BRG Clock	
		1 1 DPLL Clock		1 1 DPLL Clock			1 1 DPLL Clock	

(1) D7 (Xtal Select)

This bit determines whether the on-chip Xtal oscillator is to be used or not.

0: No Xtal

The Xtal oscillator remains inoperative, so that no Xtal resonator can be connected to the device.

1: Xtal

Makes the on-chip Xtal OSC circuit available, so that a Xtal resonator can be connected across pins XI1 and XI2.

(2) D6, D5 (Receive Clock Select)

These bits select the clock source for the receive clock.

0 0: $\overline{\text{STRxC}}$ Input

The clock applied to the $\overline{\text{STRxC}}$ pin of the device is used as the receive clock.

0 1: $\overline{\text{TRxC}}$ Input

The clock applied to the $\overline{\text{TRxC}}$ pin is used as the receive clock.

If this clock is selected, bits D2, D1, and D0 of CR15 become invalid.

1 0: RxBRG Clock

The output clock of the RxBRG is used as the receive clock.

1 1: DPLL Clock

The output clock of the DPLL is used as the receive clock.

(3) D4, D3 (Transmit Clock Select)

These bits select the clock source for the transmit clock.

0 0: $\overline{\text{STRxC}}$ Input

The clock applied to the $\overline{\text{STRxC}}$ pin of the device is used as the transmit clock.

0 1: $\overline{\text{TRxC}}$ Input
The clock applied to the TRxC pin is used as the transmit clock. If this clock is selected, bits D2, D1, and D0 of CR15 become invalid.

1 0: TxBRG Clock
The output clock of the TxBRG is used as the transmit clock.

1 1: DPLL Clock
The output clock of the DPLL is used as the transmit clock.

(4) D2 ($\overline{\text{TRxC}}$ Input/Output)

This bit selects the $\overline{\text{TRxC}}$ pin function from the input and output functions.

0: Input
The $\overline{\text{TRxC}}$ pin functions as an input.

1: Output
The $\overline{\text{TRxC}}$ pin functions as an output. If the pin is specified as an input by D6, D5, D4, and D3 of CR15, the setting of this bit is invalid.

(5) D1, D0 ($\overline{\text{TRxC}}$ Source Select)

These bits select the source of the clock output at the $\overline{\text{TRxC}}$ pin of the device. They are valid only if the $\overline{\text{TRxC}}$ pin is specified for output (D2 of CR15 = 1). If, however, the $\overline{\text{TRxC}}$ pin is specified for input by D6, D5, D4, and D3 of CR15, the setting of these bits are invalid.

0 0: Xtal
The output of the on-chip Xtal oscillator appears at the $\overline{\text{TRxC}}$ pin. This setting is valid only if the Xtal oscillator is used.

0 1: TxCLK
The transmit clock appears at the $\overline{\text{TRxC}}$ pin.

1 0: BRG Clock
The BRG's output clock appears at the $\overline{\text{TRxC}}$ pin. Selection between TxBRG and RxBRG is done with D7 of CR12.

1 1: DPLL Clock
The output of the DPLL appears at the TRxC pin.

3.3 Status Registers (SR)

This section describes the status registers that are used to indicate the AMPSC's device status. Status registers hold information indicating the current device status of the AMPSC. The host processor can manage AMPSC operations appropriately by reading status information from the SRs.

3.3.1 Status Register 0 (SR0)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function / Operation protocol	End of Frame	CRC/ Framing Error	Rx Overrun Error	Parity Error	Short Frame Delect	Tx Buffer Empty	Sending Abort	Rx Data Available
Async	0	"Framing Error 0 No Error 1 Framing Error	0 No Error 1 Overrun Error	0 No Error 1 Parity Error	0	0 Tx Buffer Full 1 Tx Buffer Empty	0	0 Not Available 1 Available
COP		"CRC Error" 0 No Error 1 CRC Error				0 Tx Buffer Full (Including CRC Transmission) 1 Tx Buffer Empty		
BOP	0 Not EOF 1 EOF			0	0 Not Detect 1 Short Frame Detect		0 Not Sending Abort 1 Sending Abort	

(1) D7 (End of Frame; valid in BOP mode)
This bit indicates whether reception of a frame is completed or not.

0: Not EOF
Indicates that reception of a frame is not completed yet. This bit, after setting upon completion of one-frame reception, is reset to zero by an Error Reset command or reception of the first data of the next frame (except in the First Rx INT mode).

1: EOF
This bit is set when an Ending flag is received, to indicate the end of a frame reception. The CRC Error bit (D6 of SR0) and Residue Code (D2-D0 of SR3) are validated when this bit is set. The EOF condition causes a Special Rx Condition interrupt.

(2) D6 (CRC/Framing Error)

This bit indicates a CRC error or framing error. The function of this bit differs depending on the operation mode.

(a) Asynchronous mode

This bit indicates a framing error. It is set if a zero is detected at the stop-bit position. It causes a Special Rx Condition interrupt.

This bit is reset by an Error Reset command or reception of a normal data item (containing no framing error).

A framing error does not affect reception of the next data item.

(b) COP or BOP mode

This bit indicates the result of the CRC calculation performed on received data. If set to one, it indicates a CRC error; if remaining at zero, it indicates no CRC error.

The procedure in which this bit is to be read has the following restriction: In the COP mode, read the bit after elapsing 20-bit time from when the last bit of the second CRC byte is input to the RxD pin, or after elapsing 16-bit time from when the second CRC byte is transferred to the Rx buffer. In the BOP mode, read the bit when the End of Frame bit (D7 of SR0) is set to one.

A CRC error does not cause a Special Rx Condition interrupt. This bit is reset by the Error Reset command.

(3) D5 (Receive Overrun Error)

This bit indicates an Rx Overrun error. An Rx Overrun error occurs if 1 data items or more is transferred to the Rx buffer when it is full.

For example, assume that three data items, D0, D1, and D2, are already in the Rx buffer (see Fig. 3-10). When the next data item, D3, is assembled in the shift register and transferred to the Rx buffer, an Overrun error will occur because the buffer location is occupied by data item D2 (see Fig. 3-11).

An Rx Overrun error causes a Special Rx Condition interrupt. The timings at which the Rx Overrun Error bit is set and the resulting Special Rx Condition interrupt occurs differ depending on the setting of the Overrun Error INT bit (D6 of CR1). For more deteils, see the description of CR1.

The Rx Overrun Error bit is latched, and is reset by the Error Reset command.

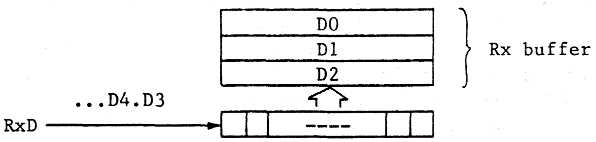


Fig. 3-10 Rx Buffer Full

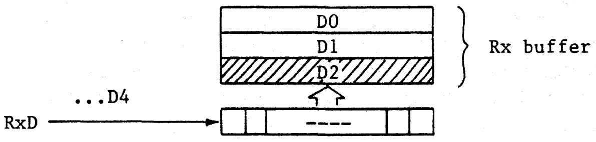


Fig. 3-11 Rx Overrun

(4) D4 (Parity Error; valid in Asynchronous or COP mode)
This bit indicates the result of parity check. It is valid when parity is enabled. If the received parity bit does not match the predetermined parity (parity error), this bit will be set. The Parity Error bit is latched and is reset by an Error Reset command.
When in the All Receive INT-1 mode (D4, D3 of CR1 = 1, 0), a parity error causes a Special Rx Condition interrupt.

(5) D3 (Short Frame Detect; valid in BOP mode)
This bit is valid when Short Frame Detect Enable is selected (D7 of CR1 = 1). It is set when a short frame (in which data between two flags has less than 32 bits) is received, and is reset by the Error Reset command. Detection of a short frame causes a Special Rx Condition interrupt.

(6) D2 (Transmit Buffer Empty)
This bit indicates AMPSC's Tx buffer status. It gives an indication of when the host system may transfer Tx data to the AMPSC.

0: Tx Buffer Full
Indicates that the Tx buffer is full with Tx data. This bit is also reset to zero when CRC transmission is busy in COP or BOP mode.
When this bit is at zero, it is not possible correctly to write Tx data into the AMPSC. However, when the PAD character is to be sent following the CRC in the COP mode, the character can be written into the device even if this bit is at zero.

1: Tx Buffer Empty
Indicates that the Tx buffer is empty. Tx data can be written into the AMPSC in this condition. This bit is, however, reset to zero when CRC transmission is busy in the COP or BOP mode, even if the Tx buffer is empty.

(7) D1 (Sending Abort; valid in BOP mode)

This bit indicates that the AMPSC entered the Sending Abort State. It is set to one when an Abort is sent in any of the following conditions:

- (a) The Send Abort bit (D4 of CR5) is set to one.
- (b) Tx Underrun occurred when the Tx Condition on Underrun bit (D2 of CR10) was set to one.
- (c) Tx Underrun occurred when the TxDLC Enable bit (D1 of CR13) was set to one, and the value of the Tx Data Length (CR8, CR9) did not match that of the Tx Data Length Counter (SR8, SR9).

This bit is latched, and is reset by the Error Reset command. Status changes of this bit do not cause a interrupt.

(8) D0 (Receive Data Available)

This bit indicates whether or not valid receive data exists in the Rx buffer within the AMPSC.

0: Not Available

Indicates that no valid receive data exists in the Rx buffer.

1: Available

Indicates that at least one valid receive data character exists in the Rx buffer.

3.3.2 Status Register 1 (SR1)

Bit		D7	D6	D5	D4	D3	D2	D1	D0
Function / Operation protocol		Break/ Abort/ GA Detect	Tx Underrun/ EOM	CTS	Sync/ Hunt	DCD	All Sent	Idle Detect	BRG Zero Count
Async		"Break Detect" 0 Not Detect 1 Break Detect	0	0 CTS="H" 1 CTS="L"	0 SYNC="H" 1 SYNC="L"	0 DCD="H" 1 DCD="L"	0 Not All Sent 1 All Sent	0	0 Not Zero 1 Zero
COP	Ext.	0	0 Not Tx Underrun/ EOM 1 Tx Underrun/ EOM				0		
	Int.				0 Exit Hunt Phase 1 Hunt Phase				
BOP	Normal	"Abort Detect" 0 Not Detect 1 Abort Detect					0 Not All Sent 1 All Sent	0 Not Idle 1 Idle Detect	
	Loop	"Abort GA Detect" 0 Not Detect 1 Abort /GA Detect							

Status Register 1 consists of E/S bits to indicate the causes of E/S interrupts. The AMPSC uses the E/S bits to indicate its internal status changes or input pin status changes to the host processor.
If E/S INT is enabled (D0 of CR1 = 1) and interrupt by an individual E/S bit is enabled, the changes in the pertinent E/S bit status are latched and cause an E/S interrupt.
If E/S interrupt is disabled, changes in E/S bit status will not be latched, simply indicating the changes in condition.

(1) D7 (Break/Abort/Go Ahead Detect; valid in Async or BOP mode)

The function of this bit differs depending on the operation mode:

(a) Asynchronous mode

This bit provides the Break (character in which the start, stop, and data bits are all zeros) Detect function.

0: Not Detect

Indicates that no break has occurred.

1: Break Detect

Indicates that a break was detected.

The changes in the Break Detect bit status are latched to cause an E/S interrupt. Data (null data) received during the Break Detect state is not available. Therefore, the Break condition cannot be read as a character.

(b) BOP mode

The function of this bit differs between the Normal BOP mode (HDLC/SDLC) and SDLC Loop mode:

(i) HDLC/SDLC mode

This bit provides the Abort (7 or more consecutive 1s) Detect function.

0: Not Detect

Indicates that no Abort has been detected or an Abort condition was cleared.

1: Abort Detect

Indicates that an Abort was detected.

The changes in the Abort Detect bit are latched to cause an E/S interrupt. The AMPSC's Abort Detect feature becomes valid when a start flag is detected after reception is enabled. When in the Address Search mode (D2 of CR3 = 1), the Abort Detect function remains invalid until an address field with the same value as that present in CR6 is detected. Once an address match has occurred, abort detection is subsequently done unconditionally. If it is desired to give the same function to subsequent frames, D2 of CR3 must be set to one again.

(ii) Loop mode

This bit serves the Abort/GA (Go-Ahead: $\overset{LSB}{0}111111\overset{MSB}{1}$) Detect function.

0: Not Detect

Indicates that no Abort/GA was detected or released (zero received after detection).

1: Abort/GA Detect

Indicates that a Abort/GA was detected.

The changes in the Abort/GA Detect bit status are latched to cause an E/S interrupt.

(2) D6 (Transmit Underrun/End of Message; valid in COP or BOP mode)

This bit indicates the Tx Underrun/EOM state (in which all Tx data has been transmitted and no Tx data remains in the Tx buffer). CRC transmission in the Tx Underrun/EOM state can be controlled by manipulating this bit.

0: Indicates that the device is not in the Tx Underrun/EOM state. If CRC transmission is desired in the Tx Underrun/EOM state, this bit must be reset to zero by the Reset Tx Underrun/EOM bit command (D7, D6 of CR0 = 1, 1). Before issuing this command, transmission must be enabled and at least one Tx data item must be transferred to the AMPSC.

In the BOP mode, however, this bit is automatically reset to zero when the first Tx data is transferred after transmission is enabled.

A status change from one to zero of this bit does not cause an E/S interrupt.

1: Indicates that the device entered the Tx Underrun/EOM state. This bit is reset before the device enters the Tx Underrun/EOM state. If it is set to one by the Tx Underrun/EOM state, the CRC will be transmitted if it is enabled, and an E/S interupt will occur.
This bit is set to one by a reset operation (system or channel reset), Send Abort, or transmission disable, and this change causes an E/S interrupt.

(3) D5 (Clear to Send)
This bit indicates the $\overline{\text{CTS}}$ pin status.
0: $\overline{\text{CTS}}$ pin = "H"
1: $\overline{\text{CTS}}$ pin = "L"
The status changes of this bit are latched to cause an E/S interrupt. However, if the CTS IE bit (D5 of CR11) is zero, this bit only indicates the $\overline{\text{CTS}}$ pin input status and is not latched.

(4) D4 (Sync/Hunt)
This bit indicates the $\overline{\text{SYNC}}$ pin status or AMPSC's synchronization establishment state. Its function differs depending on the operation mode:
(a) Async or External COP mode
This bit indicates the $\overline{\text{SYNC}}$ pin status.
0: $\overline{\text{SYNC}}$ pin = "H"
1: $\overline{\text{SYNC}}$ pin = "L"
The status changes of this bit are latched to cause an E/S interrupt, provided the crystal oscillator is not used (D7 of CR15 = 0).
(b) Internal COP mode or BOP mode
This bit indicates the AMPSC's synchronization establishment state.
0: Exit Hunt Phase
Indicates that synchronization is established.

1: Hunt Phase
Indicates the Hunt Phase (sync pattern detection busy or receive operation stop). This bit is set to one by the receiver disable or Enter Hunt Phase (D4 of CR3 = 1) operation, and reset to zero when a sync pattern is detected. The status changes of this bit are latched to cause an E/S interrupt.

(5) D3 (Data Carrier Detect)
This bit indicates the DCD pin status.
0: $\overline{\text{DCD}}$ pin = "H"
1: $\overline{\text{DCD}}$ pin = "L"
The status changes of this bit are latched to cause an E/S interrupt. However, if the DCD IE bit (D3 of CR11) is zero, this bit only indicates the $\overline{\text{DCD}}$ pin input status and will not be latched.

(6) D2 (All Sent; valid in Async or BOP mode)
This bit indicates that all Tx data within the AMPSC has been transmitted.
(a) Asynchronous mode
0: Not All Sent
Indicates that Tx data remains in the AMPSC. This bit is always zero when the Tx is disabled.
1: All Sent
Indicates that all Tx data within the AMPSC has been transmitted and no data remains in the Tx buffer or Tx shift register.
This bit is reset to zero when Tx data is transferred to the AMPSC. Zero-to-one transition of this bit is latched to cause an E/S interrupt.

(7) D1 (Idle Detect; valid in BOP mode)
This bit indicates detection of the Idle state (15 or more consecutive 1s).
0: Not Idle Detect
Indicates that no Idle state is detected or the Idle state has been finished.

1: Idle Detect
Indicates that the Idle state is detected.

Status changes of this bit are latched to cause an E/S interrupt.

(8) D0 (BRG Zero Count)

This bit indicates that the BRG count value reached zero.

0: Not Zero
Indicates that the BRG count value has not reached zero. This bit is always zero if the BRG IE bit (D0 of CR11) is zero.

1: Zero
Indicates that the BRG count value reached zero.

Zero-to-one transition of this bit is latched to cause an E/S interrupt. It is reset by the Reset E/S Bit Latch command.

Which BRG count value out of TxBRG and RxBRG can be determined from D4 and D3 of SR3.

3.3.3 Status Register 2B (SR2B)

Bit Function Vector type	D7	D6	D5	D4	D3	D2	D1	D0
	Vector Value							
Type A	V7	V6	V5	0 Ch-B 1 Ch-A	0 0 Tx Buffer Empty 0 1 External/ Status 1 0 Rx Data Available 1 1 Special Rx Condition		V1	V0
Type B	V7	V6	V5	V4	V3	0 Ch-B 1 Ch-A	0 0 Tx Buffer Empty 0 1 External/ Status 1 0 Rx Data Available 1 1 Special Rx Condition	

Status Register 2B indicates the value of an interrupt vector (cause of interrupt). The vector value indicated by this register differs depending on the value of D6 of CR2A (Status Affects Vector bit).

(a) If D6 of CR2A = 0
The value set in CR2B is derectly output.

(b) If d6 of CR2A = 1
The value of SR2B differs depending on the cause of interrupt. The bits of SR2B which are affected by the cause of interrupt depend on the Output Vector Type setting. V4, V3, and V2 are affected for Type A; V2, V1, and V0 are affected for Type B, and all other bits are directly output without being changed.

SR2B is normally used for analyzing the cause of interrupt by the AMPSC in an interrupt service routine in the Non-Vector mode (D7 of CR2A = 0).
When Tx/Rx data is to be transferred by using status polling, the interrupt vector set in SR2B can be treated as status by placing the AMPSC in the Non-Vector mode, Status Affects Vector (D6 of CR2A = 1), and Both Channel INT (D1, D0 of CR2A = 0, 0).

3.3.4 Status Register 3 (SR3)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function Operation protocol				TxBRG Zero Count	RxBRG Zero Count	Residue Code		
Async	Unknown			0 Not Zero 1 Zero	0 Not Zero 1 Zero	Unknown		
COP								
BOP						(when there is no remainder) 5 Bits/Char. 1 0 0 6 Bits/Char. 0 0 0 7 Bits/Char. 0 1 1 8 Bits/Char. 0 1 1		

(1) D4 (TxBRG Zero Count)
This bit is meaningful when the TxBRG is enabled (D0 of CR14 = 1).

0: Not Zero
Indicates that the TxBRG count value does not reach zero.

1: Zero
Indicates that the TxBRG count value reached zero.

Zero-to-one transtion of this bit is latched to cause an E/S interrupt. It is reset by the Reset E/S Bit Latch command. The BRG Zero Count bit (D0 of SR1) is one whenever this bit is set at one. The bit is zero whenever the BRG IE bit (D0 of CR11) is at zero.

(2) D3 (RxBRG Zero Count)
This bit is meaningful when RxBRG is enabled (D1 of CR14 = 1).

0: Not Zero
Indicates that the RxBRG count value is not zero.

1: Zero

Indicates that the RxBRG count value is zero. Zero-to-one transtion of this bit is latched to cause an E/S interrupt. It is reset by the Reset E/S Bit Latch command. The BRG Zero Count bit (D0 of SR1) is one whenever this bit is set at one. The bit is zero whenever the BRG IE bit (D0 of CR11) is at zero.

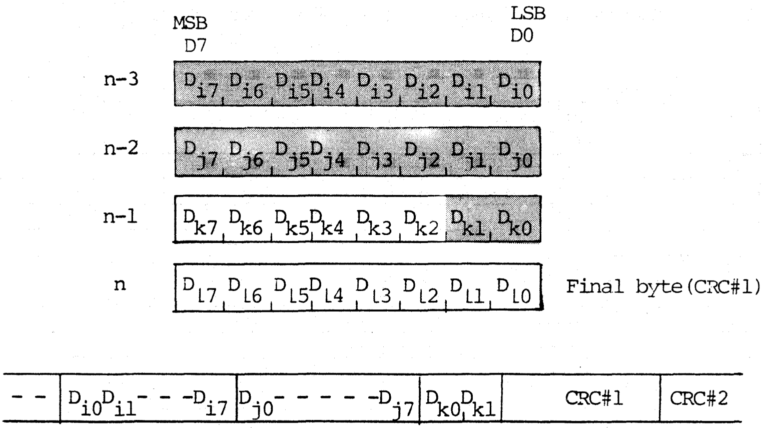
(3) D2-D0 (Residue Code; valid in BOP mode)
These bits indicate the valid range of data bits in the information field of a frame at the end of frame reception. The range of valid data bits can be determined by reading these bits when the AMPSC entered the EOF state. Table 3-10 shows the meanings of the Residue Code for different data lengths.

(Example) IF the Residue Code indicates "000" when data of 8-bits is received as shown in Fig. 3-12, the valid I field bits are ....Dj7, Dk0, Dk1.

Table 3-10 Meanings of Residue Code

	Residue Code			I field range (effective I field bit)			
Data length	D2	D1	D0	nth byte (final byte)	(n-1)th byte	(n-2)th byte	(n-3)th byte
5	0	0	0	—	—	D0-3	o
	0	0	1	—	—	D0-2	o
	0	1	0	—	—	D0	o
	1	0	0*	—	—	—	o
	1	1	0	—	—	D0-1	o
6	0	0	0*	—	—	o	o
	0	0	1	—	—	D0-3	o
	0	1	0	—	—	D0-1	o
	1	0	0	—	—	D0	o
	1	0	1	—	—	D0-4	o
	1	1	0	—	—	D0-2	o
7	0	0	0	—	D0	o	o
	0	0	1	—	—	D0-4	o
	0	1	0	—	—	D0-2	o
	0	1	1*	—	—	o	o
	1	0	0	—	—	D0-1	o
	1	0	1	—	—	D0-5	o
	1	1	0	—	—	D0-3	o
8	0	0	0		D0-1	o	o
	0	0	1	—	—	D0-5	o
	0	1	0	—	—	D0-3	o
	0	1	1*	—	—	o	o
	1	0	0	—	—	D0-2	o
	1	0	1	—	—	D0-6	o
	1	1	0	—	—	D0-4	o
	1	1	1	—	D0	o	o

Di: Indicates valid data bits
o : Indicates all bits (bits 5, 6, 7 and 8) are valid.
- : Indicates all bits (bits 5, 6, 7 and 8) are invalid (denotes CRC).
* : When there is no remainder. Denotes the case when the Boundary of the last receive data matches the boundary between the I field and CRC.



Recieved from the reft

▨ Bits in the enclosed portion become valid in I field.

Fig. 3-12 Valid Bits in I Field (Residue Code = "000")

3.3.5 Status Register 4A (SR4A)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function / Operation protocol	Ch-A Special Rx Condition INT Pending	Ch-B Special Rx Condition INT Pending	Ch-A Rx INT Pending	Ch-A Tx INT Pending	Ch-A E/S INT Pending	Ch-B Rx INT Pending	Ch-B Tx INT Pending	Ch-B E/S INT Pending
Async	0 Not Pending 1 Pending							
COP								
BOP								

Each bit of Status Register 4A indicates whether or not a corresponding cause of an interrupt exists within the AMPSC. it is set to one when the pertinent interrupt is pending, being serviced or existing another cause of interrupt with a higher priority.
It is set at zero in any other case.
While this register exists only on channel A, its function is shared for both channels.

(1) D7, D6 (Special Rx Condition INT Pending)
These bits indicate that the cause of a Special Rx Condition interrupt exists. Bits D7 and D6 indicate the presence of the cause of a Special Rx Condition interrupt on channel A and channel B, respectively.
These bits are set when a Special Rx Condition occurs, even if Rx INT is disabled (D4, D3 of CR1 = 0, 0).

(2) D5 (Channel A Rx INT Pending)
This bit indicates the presence of an Rx interrupt on channel A.

(3) D4 (Channel A Tx INT Pending)
This bit indicates the presence of a Tx interrupt on channel A.

(4) D3 (Channel A E/S INT Pending)
This bit indicates the presence of an E/S interrupt on channel A.

(5) D2 (Channel B Rx INT Pending)
This bit indicates the presence of an Rx interrupt on channel B.

(6) D1 (Channel B Tx INT Pending)
This bit indicates the presence of a Tx interrupt on channel B.

(7) D0 (Channel B E/S INT Pending)
This bit indicates the presence of an E/S interrupt on channel B.

3.3.6 Status Register 8 (SR8)

Bit Function Operation protocol	D7	D6	D5	D4	D3	D2	D1	D0
	Tx Data Length Counter L-Byte							
Async	Unknown							
COP								
BOP	Tx Data Length Counter Bit7-Bit0							

Status Register 8 (SR8) is used with SR9 to indicate the Tx interrupt/DMA request activation count. These registers are made valid if the Tx Data Length Counter is enabled (D1 of CR13 = 1).

(1) D7-D0 (Tx Data Length Counter L-Byte; valid in BOP mode)
These bits indicate the lower byte (bits 7-0) of the Tx interrupt request or Tx DMA request count. This register is usually used to judge whether the Tx Underrun/EOM state, if occurred, is legal or not. In this state the contents of SR8 and SR9 will be preserved.
If the value of CR8/CR9 does not match with that of SR8/SR9 when the Tx Underrun/EOM state occurred, the AMPSC automatically transmits an Abort. SR8 and SR9 are cleared in the following cases:
1) Reset
2) Tx Data Length Counter Enable bit is set to one.

3.3.7 Status Register 9 (SR9)

Bit / Function / Operation protocol	D7	D6	D5	D4	D3	D2	D1	D0
Function	Tx Data Length Counter H-Byte							
Async	Unknown							
COP								
BOP	Tx Data Length Counter Bit15-Bit8							

(1) D7-D0 (Transmit Data Length Counter H-Byte; valid in BOP mode)
These bits indicate the upper byte of the Tx interrupt request or Tx DMA request count. This register is used with SR8. The description of this resiter is identical to that for SR8, with the exception that SR9 indicates the upper byte of count data.

3.3.8 Status Register 10 (SR10)

Bit / Function / Operation protocol		D7	D6	D5	D4	D3	D2	D1	D0
Function		One Clock Missing	Two Clocks Missing		Sending on Loop			Tx Sync/ on Loop	
Async		0 Not Missing 1 Missing	0 Not Missing 1 Missing	Unknown	Unknown	Unknown		Unknown	Unknown
COP								"Tx Sync" 0 Not Tx Sync 1 Tx Sync	
BOP	Normal							Unknown	
	Loop				0 Not Sending 1 Sending			"on Loop" 0 Not on Loop 1 on Loop	

(1) D7 (one Clock Missing)

This bit indicates whether or not serial data contains edges. It is valid only when the FM data format is selected and the DPLL is in use.

When FM is used for the data format, an edge (rising or falling edge) occurs within a 1-bit time (at a bit boundary or center of a bit), unlike NRZ or NRZI. The DPLL uses this edge as a reference for clock generation. The DPLL identifies the position of edges in serial data, and generates a clock besed on this position. If no edge exists, clock generation by the DPLL may result in failure. To prevent this, the host processor uses this bit to identify whether or not an edge exists within the specified range of time.

The DPLL detects edges at every two bits.

0: Not Missing

Indicates that edges were detected in serial data.

1: Missing

Indicates an edge was not detected in serial data. This bit is latched, and is reset by the Reset Missing Clock command (D7, D6, D5 of CR14 = 0, 1, 0) or Enter Search command (D7, D6, D5 of CR14 = 0, 0, 1).

(2) D6 (Two Clocks Missing)

This bit indicates whether edges were detected in serial data. It is valid when FM is selected for the data format and the DPLL is in use. (The basic function of this bit is identical to that of the One Clock Missing bit).

0: Not Missing

Indicates that edges were detected in serial data. It remains zero if only one edge was not detected.

1: Missing

Indicates that edges were not detected in serial data two times consecutively.

(3) D4 (Sending on Loop; valid in BOP mode)

This bit indicates that the AMPSC formed an SDLC Loop and is busy for transmission. It is valid when the SDLC Loop mode is selected (D4, D1 of CR10 = 1, 1).

0: Not Sending
  Indicates that the AMPSC is not doing a loop transmission.

1: Sending
  Indicates that the AMPSC is doing a loop transmission.

(4) D1 (Tx Sync/on Loop; valid in COP or BOP mode)

The function of this bit differs depending on the operation mode.

(a) COP mode
  This bit indicates the Tx Sync state.

  0: Not Tx Sync
    Indicates that the receiver is not synchronized with the transmitter. This bit is also reset to zero if the Auto Tx on Sync bit (D4 of CR10) or D4 Enable bit (D1 of CR10) is reset.

  1: Tx Sync
    Indicates that synchronization between the transmitter and receiver is established (SYNC character detection on receiver completed) after both the Auto Tx on Sync and D4 Enable bits were set, and transmission is enabled for the device.

(b) BOP mode
  This bit serves the on Loop function during SDLC Loop operation, and indicates the repeat operation state of the AMSPC.

  0: Not on Loop
    Indicates that no GA pattern $(\overset{\text{LSB}}{0}111111\overset{\text{MSB}}{1})$ is detected even if the SDLC Loop is formed (D4, D1 of CR10 = 1, 1). In this state a 1-bit delay is not inserted. This bit is also reset to zero when the device releases the SDLC Loop (D1 of CR10 = 0).

1: on Loop
Indicates that a GA pattern was detected and a 1-bit delay was inserted between the RxD input and TxD output. This bit is maintained at one while the SLDC Loop is formed.

3.3.9 Status Register 11 (SR11)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Break/ Abort IE	Tx Underrun/ EOM IE	CTS IE	Sync/ Hunt IE	DCD IE	All Sent IE	Idle Detect IE	BRG IE

This register directly indicates the value set in CR11. The host processor can use this register to control interrupt enable for the AMPSC.

3.3.10 Status Register 12 (SR12)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Rx BRG Counter Constant L-Byte							

This register indicates the lower 8 bits (bits 7-0) of the value set in the Rx BRG.

3.3.11 Status Register 13(SR13)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Rx BRG Counter Constant H-Byte							

This register indicates the upper 8 bits (bits 15-8) of the value set in the Rx BRG.

3.3.12 Status Register 14 (SR14)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Tx BRG Counter Constant L-Byte							

This register indicates the lower 8 bits (bits 7-0) of the value set in the Tx BRG.

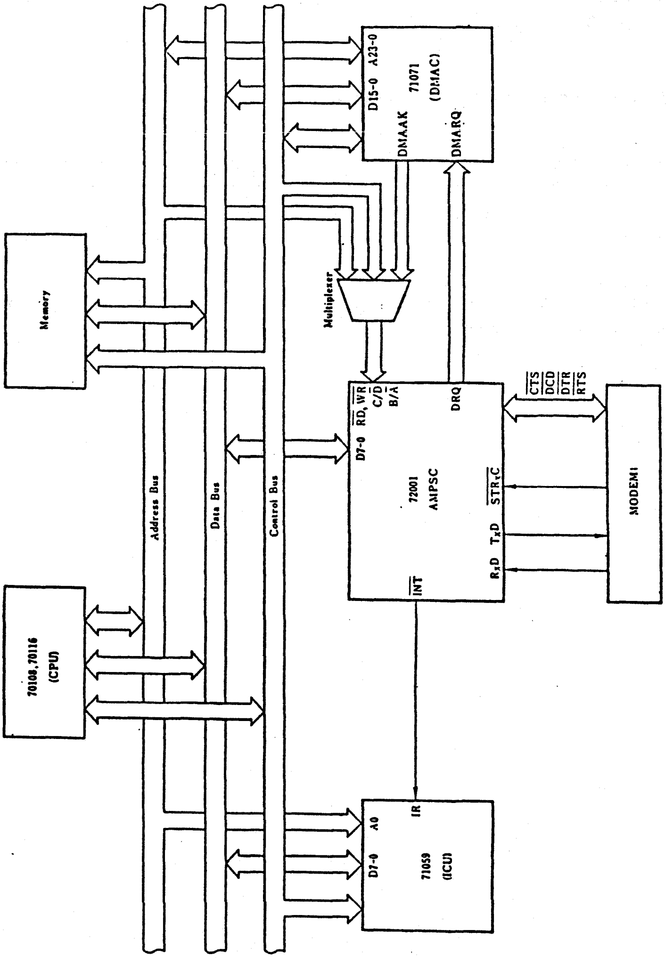
3.3.13 Status Register 15 (SR15)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Function	Tx BRG Counter Constant H-Byte							

This register indicates the upper 8 bits (bits 15-8) of the value set in the Tx BRG.

4. SYSTEM CONFIGURATION EXAMPLE

The following figure shows a system configuration example in which the uPD72001 is interfaced with a modem, with both channels placed in the DMA mode.



TARGET SPEC

µPD72001

ADVANCED MULTIPROTOCOL SERIAL CONTROLLER

3. Electrical Specifications

Absolute Maximum Ratings (Ta = 25°C)

Parameter	Symbol	Test Conditions	Ratings	Unit
Power Supply Voltage	V_{DD}		-0.5 to +7.0	V
Input Voltage	V_I		-0.5 to V_{DD}+0.5	V
Output Voltage	V_O		-0.5 to V_{DD}+0.5	V
Operating Temperature	T_{opt}		-10 to +70	°C
Storage Temperature	T_{stg}		-65 to +150	°C

DC Characteristics (Ta = -10 to +70°C, V_{DD} = 5V±10%)

Prameter	Symbol	Test Conditions	MIN.	TYP.	MAX.	Unit
Input Voltage High	V_{IHC}	Clock pin	+3.3		V_{DD}+0.5	V
	V_{IH}	Other pins	+2.2		V_{DD}+0.5	V
Input Voltage Low	V_{ILC}	Clock pin	-0.5		+0.6	V
	V_{IL}	Other pins	-0.5		+0.8	V
Output Voltage High	V_{OH}	I_{OH} = -400uA	0.7V_{DD}			V
Output Voltage Low	V_{OL}	I_{OL} = 2.0mA			+0.45	V
Input Leakage Current High	I_{LIH}	$V_I = V_{DD}$			+10	uA
Input Leakage Current Low	I_{LIL}	V_I = 0V			-10	uA
Output Leakage Current High	I_{LOH}	$V_O = V_{DD}$			+10	uA
Output Leakage Current Low	I_{LOL}	V_O = 0V			-10	uA
Supply Current	I_{DD}	8MHz Operation		20	40	mA

Capacitance (Ta = 25°C, V_{DD} = 0V)

Parameter	Symbol	Test Conditions	MIN.	MAX.	Unit
Input Capacitance	C_{IN}	f_C = 1MHz Unmeasured pins returned to 0V.		10	pF
I/O Capacitance	C_{IO}			20	pF

AC Characteristics (Ta = -10 to +70°C, V_{CC} = +5V ± 10%)

System Interface

Prameter	Symbol	Test Conditions	Standard value		Unit
			MIN.	MAX.	
Clock Cycle	t_{CYK}		125	2000	ns
Clock high-level width	t_{WKH}		50	1000	ns
Clock low-level width	t_{WKL}		50	1000	ns
Clock rise time	t_{KR}	1.5V to 3.0V		10	ns
Clock fall time	t_{KF}	3.0V to 1.5V		10	ns
Address setup time to $\overline{RD}$↓	t_{SAR}		0		ns
Address hold time from $\overline{RD}$↑	t_{HRA}		0		ns
$\overline{RD}$ pulse width	t_{WRL}		150		ns
Data output delay time from address	t_{DAD}			120	ns
Data output delay time from $\overline{RD}$	t_{DRD}			120	ns
Data float delay time from $\overline{RD}$	t_{FRD}		10	85	ns
Address setup time to $\overline{WR}$↓	t_{SAW}		0		ns
Address hold time from $\overline{WR}$↑	t_{HWA}		0		ns
$\overline{WR}$ pulse width	t_{WWL}		150		ns
Data setup time to $\overline{WR}$↑	t_{SDW}		120		ns
Data hold time from $\overline{WR}$↑	t_{HWD}		0		ns
$\overline{RD}$/ $\overline{WR}$ recovery time	t_{RV}		160		ns

Serial control

Parameter		Symbol	Test Conditions	Standard value		Unit
				MIN.	MAX.	
Transmit/Receive data cycle		t_{CYD}		5		t_{CYK}
$\overline{\text{STRxC}}$, $\overline{\text{TRxC}}$ input clock cycle		t_{CYC}		125		ns
$\overline{\text{STRxC}}$, $\overline{\text{TRxC}}$ input clock pulse width	High	t_{WCH}		50		ns
	Low	t_{WCL}		50		ns
TxD delay time from $\overline{\text{STRxC}}$, $\overline{\text{TRxC}}$ ↓		t_{DTCTD1}	x 1 mode		100	ns
		t_{DTCTD2}	x 16,32,64 mode		300	ns
RxD setup time to $\overline{\text{STRxC}}$, $\overline{\text{TRxC}}$ ↑		t_{SRDRC}		0		ns
RxD hold time from $\overline{\text{STRxC}}$, $\overline{\text{TRxC}}$ ↑		t_{HRCRD}		140		ns
$\overline{\text{INT}}$ delay time from TxD		t_{DTDIQ}	Tx INT mode	4	6	t_{CYK}
DRQTx delay time from TxD		t_{DTDDQ}	Tx DMA mode	4	6	t_{CYK}
$\overline{\text{INT}}$ delay time from $\overline{\text{RxC}}$* ↑		t_{DRCIQ}	Rx INT mode	7	11	t_{CYK}
DRQRx delay time from $\overline{\text{RxC}}$* ↑		t_{DRCDQ}	Rx DMA mode	7	11	t_{CYK}
DRQRx delay time from $\overline{\text{RD}}$ ↓		t_{DRDQ}			120	ns
DRQTx delay time from $\overline{\text{WR}}$ ↓		t_{DWDQ}			120	ns

*: $\overline{\text{STRxC}}$ or $\overline{\text{TRxC}}$, which is used for the receive clock.

Interrupt control

Parameter	Symbol	Test Conditions	Standard value		Unit
			MIN.	MAX.	
$\overline{INTAK}$ low-level width	t_{WIAL}		150		ns
$\overline{PRO}$ delay time from $\overline{PRI}$	t_{DPIPO}			50	ns
$\overline{PRI}$ setup time to $\overline{INTAK}$↓	t_{SPIIA}	When vector output is permitted.	0		ns
$\overline{PRI}$ hold time from $\overline{INTAK}$↑	t_{HIAPI}	When vector output is permitted.	20		ns
Data output delay time from $\overline{INTAK}$	t_{DIAD}			120	ns
Data float delay time from $\overline{INTAK}$	t_{FIAD}		10	85	ns

Modem control

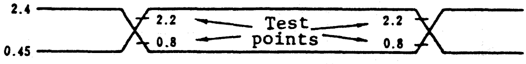
Parameter		Symbol	Test Conditions	Standard value		Unit
				MIN.	MAX.	
$\overline{CTS}$, $\overline{DCD}$, $\overline{SYNC}$ pulse width	High	t_{WMH}		2		t_{CYK}
	Low	t_{WML}		2		t_{CYK}
$\overline{INT}$ delay time from $\overline{CTS}$, $\overline{DCD}$, $\overline{SYNC}$		t_{DMIQ}			2	t_{CYK}
$\overline{SYNC}$ delay time from $\overline{STRxC}$, $\overline{TRxC}$↑		t_{DTRKSY}	COP external synchronization	0	2	t_{CYK}

Crystal oscillation and reset

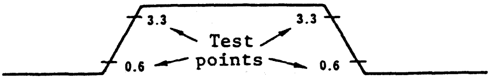
Parameter	Symbol	Test Conditions	Standard value		Unit
			MIN.	MAX.	
XI1 Input cycle time	t_{CYX}		125	1000	ns
$\overline{\text{RESET}}$ pulse width	t_{WRSL}		2		t_{CYK}

Note: At all modes, system clock cycle must be more than five times of data rate.

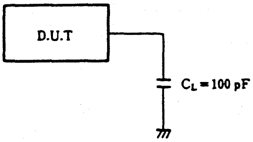
AC test I/O wave (except clock)



AC test clock input wave



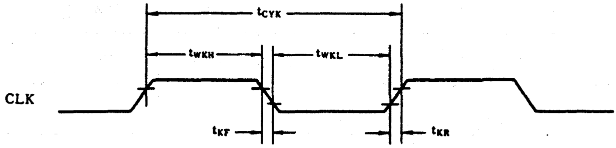
Load circuit for AC test



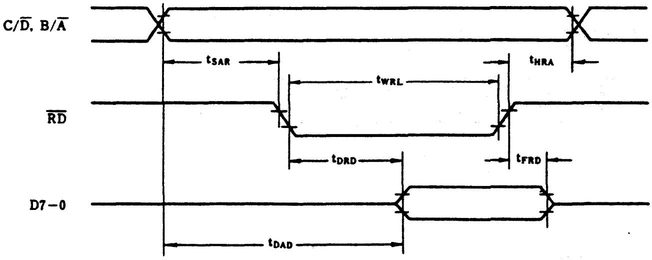
C_L includes the jig.

Timing Wave

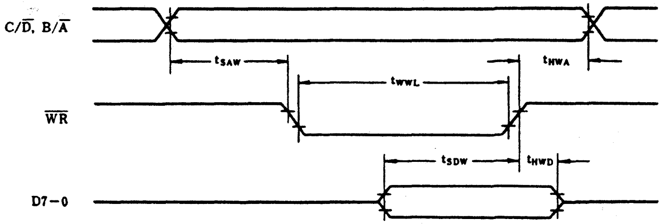
Clock timing



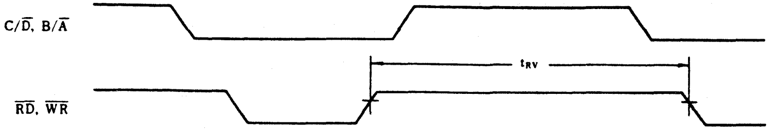
Read cycle timing



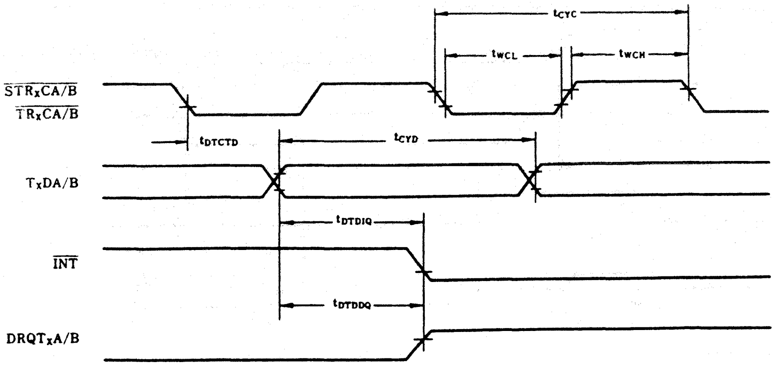
Write cycle timing



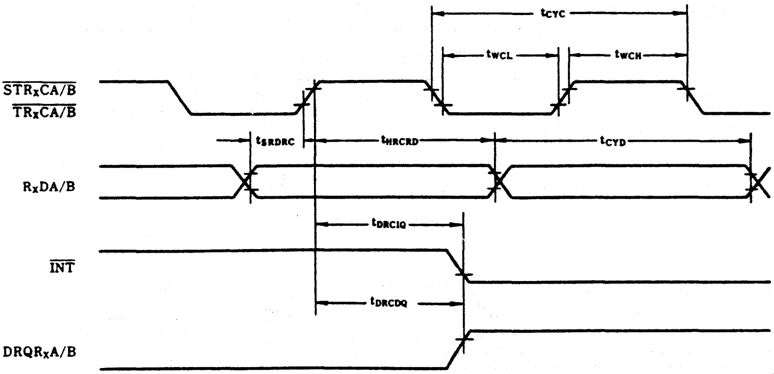
Read/Write cycle timing (data transfer by software)



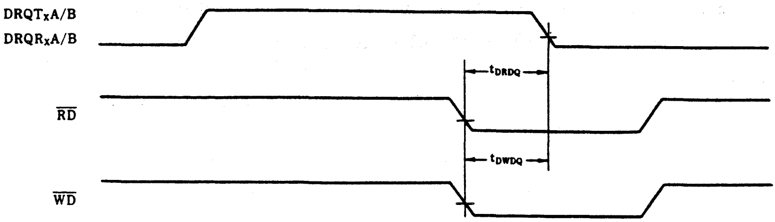
Transmit cycle timing



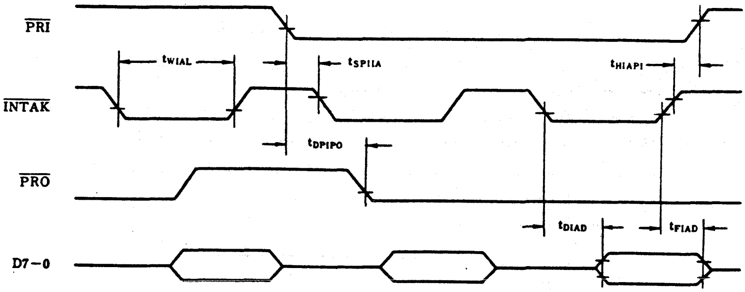
Receive cycle timing



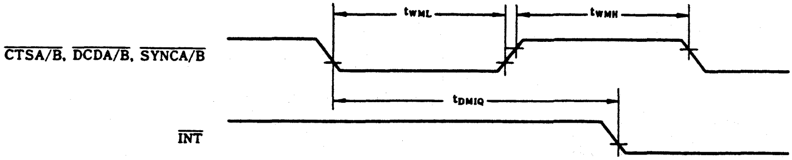
DMA request reset timing



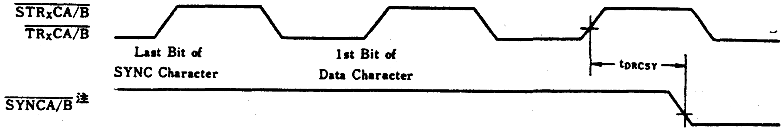
$\overline{\text{INTAK}}$ cycle timing



E/S timing

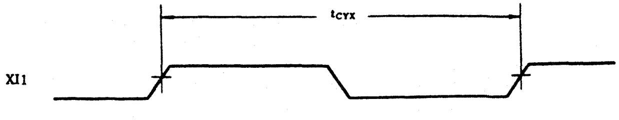


SYNC input timing (external synchronization mode)

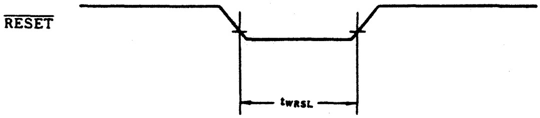


*: $\overline{\text{SYNCA/B}}$ input must be set to 0 at the rising edge of $\overline{R_XC}$ after 2 clock cycles following the last bit of SYNC character.

XI1 input timing



$\overline{\text{RESET}}$ pulse



µPD7210

GPIB-CONTROLLER

CHAPTER 1

INTRODUCTION

The uPD7210 GPIB-IFC is a general-purpose interface bus interface controller. It interfaces between the interface bus specified in IEEE Std 488-1978 and a microcomputer system under the control of a microprocessor.

Features

☐ Interface capability that meets IEEE Std 488-1978

☐ SH1, AH1, T5/TE5, L3/LE3, SR1, RL1, PP1/PP2, DC1, DT1, C1, C2, C3, C4, C5

☐ Programmable data transfer rate

☐ Sixteen registers
  eight read register and eight write registers
  message transmission and reception, interface function control, and status information

☐ Address registers
  detection of MTA, MLA, and MSA
  two device addresses

☐ Automatic EOS message detection

☐ Automatic command processing and undefined command read capability

☐ DMA capability

☐ Programmable bus transceiver I/O specification

☐ 1 to 8MHz clock range

☐ TTL-compatible

☐ N-channel MOS

☐ +5V single power supply

☐ 40-pin plastic DIP

Names of states and messages are described using the method specified in IEEE Std. 488-1978. Their explanations are omitted. For detailed explanations, refer to the IEEE paper:

> IEEE Std. 488-1978 "IEEE Standard Digital Interface for Programmable Instrumentation"

The uPD7210 operates on positive logic; IEEE Std. 488-1978 is based on negative logic. A term or state name in a logic equation equals one when the uPD7210 is in that state, and zero otherwise.

GPIB:	an interface bus which meets IEEE Std. 488-1978
Command:	Multiline remote interface message
Data:	Multiline remote device-dependent message

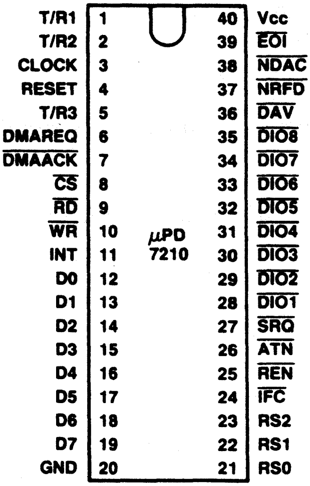


Figure 1.1 Pin Configuration

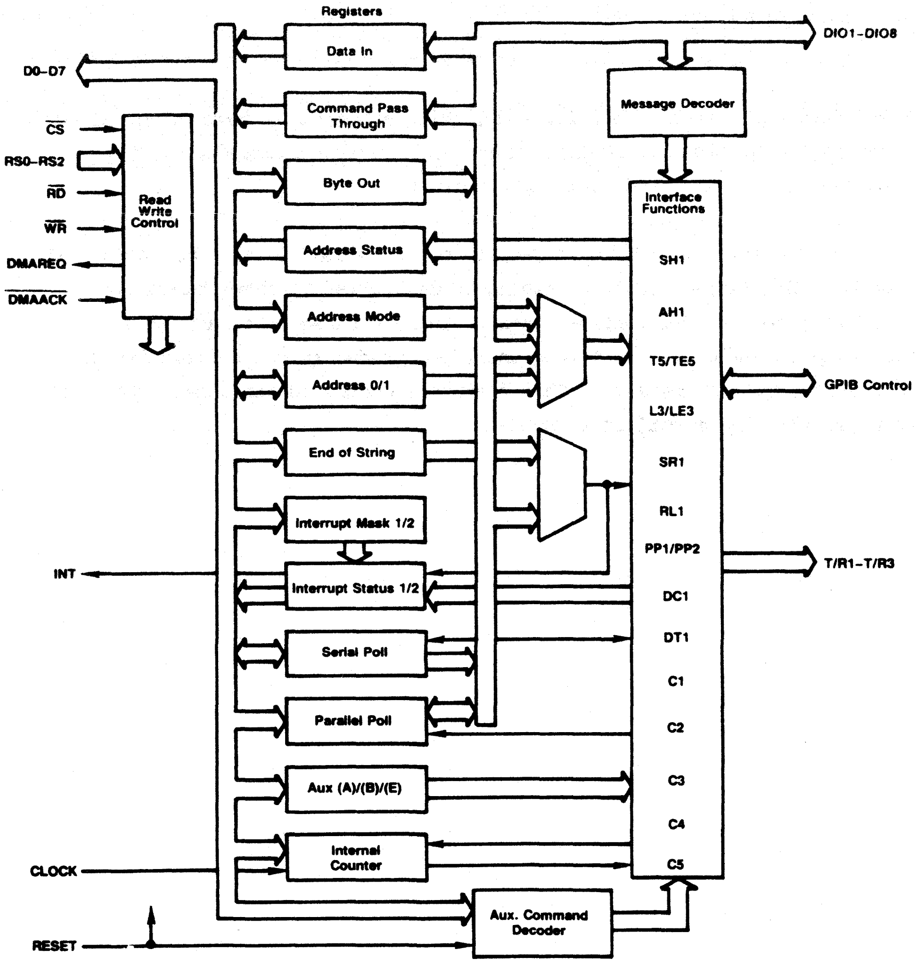


Figure 1.2 Block Diagram

1.1 General Description

The uPD7210 is designed to conform to IEEE Std 488-1978. It provides an interface between a microcomputer system and the GPIB. This uPD7210 is connected to a GPIB via non-inverting bus transceivers that meet the electrical specifications in IEEE Std 488-1978.

1.2 IEEE Std 488-1978

This standard outlines a method for simplifying a system by connecting various peripherals to the same bus (GPIB). This requires standardized control signals and data flow to and from each peripheral.

Data transfer using the uPD7210 is carried out in bit-parallel and byte-serial fashion over DIO lines. Transfer timing is controlled by three handshake lines (data byte transfer control lines). In this three-line handshake system, a byte cannot be transferred until one handshake cycle (transfer of the previous byte) is complete. This feature enhances reliable data transfer between peripherals of different speeds.

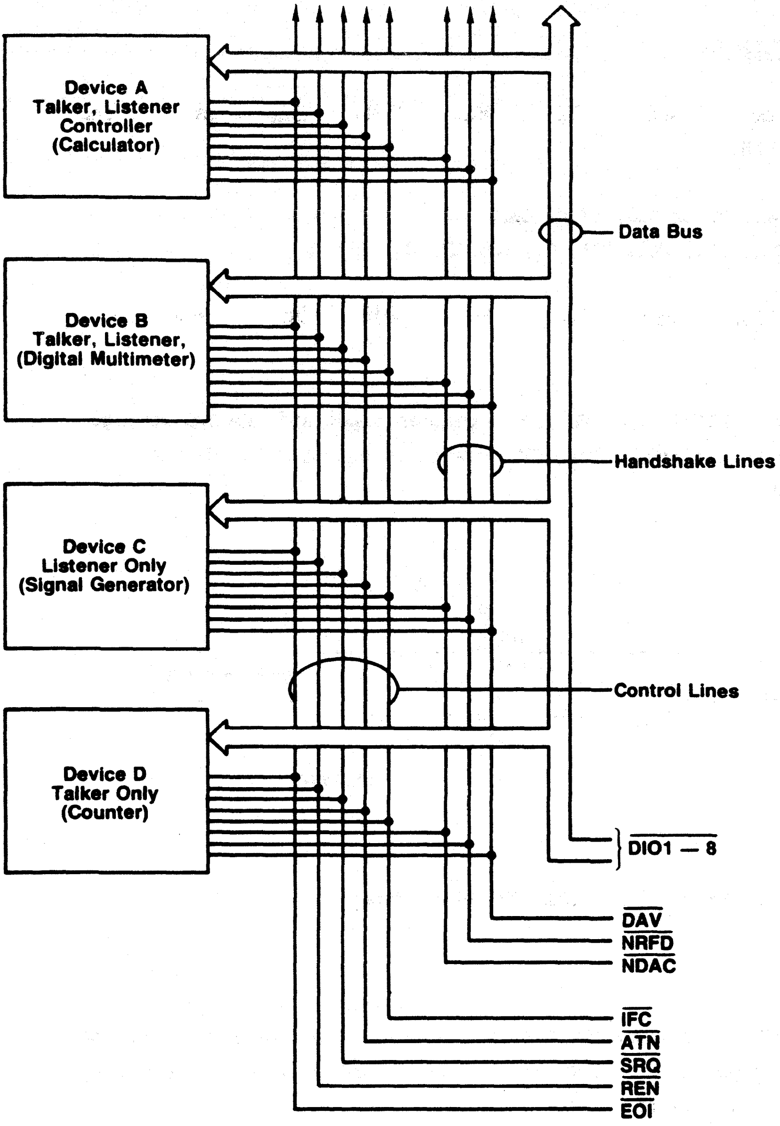


Figure 1.3 Interface Functions and Bus Configuration

CHAPTER 2

FUNCTIONAL DESCRIPTION

The uPD7210 meets the functional requirements of the following as specified in IEEE Std. 488-1978.

SH1, AH1, T5 or TE5, L3 or LE3,
SR1, RL1, PP1, DC1, DT1, C1, C2, C3, C4, C5

When the controller is active, i.e., CIC=1 (CIC=$\overline{\text{CIDS+CADS}}$), there are two exceptions.

☐ When CIC·SRQS=1, the SRQ pin becomes an input and the SRQ message is not transmitted. This SRQ message is detected inside the uPD7210.

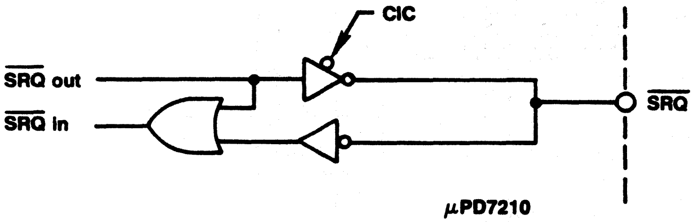


Figure 2.1 IEEE 488-1978: Exception 1

☐ When CIC·PPAS=1, the DIO pins (DIO1 to DIO8) become inputs and the PPR message is not transmitted. You can detect the PPR message by reading the CPT register.

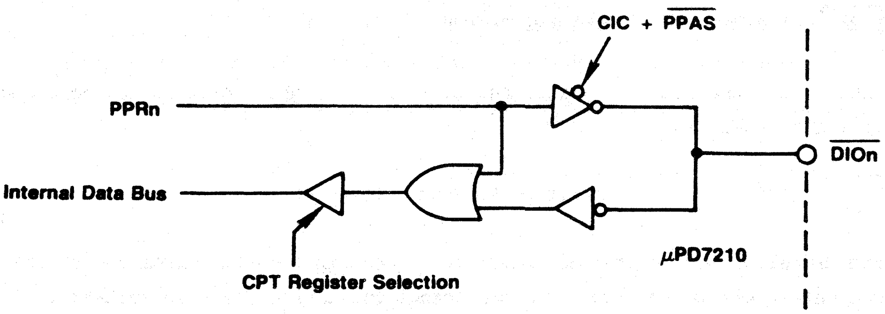


Figure 2.2 IEEE 488-1978: Exception 2

2.1 Pin Description

4 RESET Reset (input, active high)

The uPD7210 goes to an idle state when this pin is high.

3 CLOCK Clock (input)

This is a reference clock input that generates the state change prohibit times (T_1, T_6, T_7, T_9) as specified in the IEEE std. 488-1978.

12-19 D_0 to D_7 Data Bus (input/output)

This is an 8-bit bidirectional data bus which is connected to the data bus of the microcomputer system.

8 $\overline{CS}$ Chip Select (input, active low)

This enables access to the register selected by RS0-RS2.

9 $\overline{RD}$ Read (input, active low)

This reads the contents of the read register specified by RS0-RS2 on D0-D7.

10 $\overline{WR}$ Write (input, active low)

This writes the data in D0-D7 to the write register specified by RS0-RS2.

21-23 RS0-RS2 Register Select (input)

This selects one read (or write) register of the eight registers during a read (or write) operation.

11 INT Interrupt Request (output)

This signal is active when an interrupt request is generated from one of the internal interrupt factors. You can change the active level in software.

Active high: when $B_3=0$ ($B_3=0$ on reset)
Active low: when $B_3=1$

B_3: Bit 3 of auxiliary B register

6 DMA REQ DMA Request (output, active high)

This signal indicates that a request for DMA has been made. It goes low when a DMA acknowledge signal is input.

7 $\overline{\text{DMA ACK}}$ DMA Acknowledge (input, active low)

This signal connects the data bus of the microcomputer system to the data register. When this signal is low, the contents of the Data In register is output into D0-D7 with a read signal, and the data in D0-D7 is output to the Byte Out register with a write signal.

$\overline{\text{CS}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{DMA ACK}}$	RS0-RS2	Function
0	0	1	1	any	read registers 0R-7R
0	1	0	1	any	write to registers 0W-7W
X	0	1	0	XXX	read to Data In register (0R)
X	1	0	0	XXX	write to Byte Out register (0W)
1	X	X	1	XXX	register is not selected (D0-D7=Hi-Z)
X	1	1	X	XXX	
0	0	0	X	XXX	prohibit (operation is not guaranteed)
X	0	0	0	XXX	

1 T/R1 Transmit/Receive Control 1 (output)

This line is the input and output control signals of the GPIB bus transceivers.

T/R1	$\overline{DIO1}$-$\overline{DIO8}$	$\overline{DAV}$	$\overline{NRFD}$	$\overline{NDAC}$
1	output	output	input	input
0	input	input	output	output

2,5 T/R2, T/R3 Transmit/Receive Control 2,3 (output)

The function of the T/R2 and T/R3 signals is determined by the TRM0 and TRM1 values of the address mode register as shown below.

T/R2	T/R3	TRM0	TRM1
EOIOE	TRIG	0	0
CIC	TRIG	1	0
CIC	EOIOE	0	1
CIC	PE	1	1

EOIOE=TACS+SPAS+CIC.$\overline{CSBS}$

When EOIOE=1, the $\overline{EOI}$ pin is an output; when EOIOE=0, EOI is an input.

CIC=$\overline{CIDS+CADS}$

When CIC=1, $\overline{ATN}$ is an output and $\overline{SRQ}$ is an input. When CIC=0, $\overline{ATN}$ is an input and $\overline{SRQ}$ is an output.

PE=CIC.$\overline{PPAS}$

When PE=1, three-state bus transceivers are used in DIO1-DIO8 and the DAV lines. When PE=0, open-collector transceivers are used.

TRIG

When DTAS=1 or when a Trigger auxiliary command is issued, a high pulse is generated.

The following pins are connected to the corresponding lines of the GPIB through non-inverting bus transceivers.

28-35 $\overline{\text{DIO1}}$-$\overline{\text{DIO8}}$ Data Input/Output (input/output)

This is an 8-bit bidirectional data bus used for transferring remote multiline messages.

36 $\overline{\text{DAV}}$ Data Valid (input/output)

This is a handshake line which indicates that the data on the DIO lines is valid.

37 $\overline{\text{NRFD}}$ Not Ready for Data (input/output)

This is a handshake line which indicates whether the GPIB is prepared to receive messages.

38 $\overline{\text{NDAC}}$ Not Data Accepted (input/output)

This is a handshake line which indicates that the message has been received.

26 $\overline{\text{ATN}}$ Attention (input/output)

This is a control line which indicates that data on the DIO lines is an interface message or a device-dependent message.

24 $\overline{\text{IFC}}$ Interface Clear (input/output)

This is a control signal that clears the interface function.

27 $\overline{\text{SRQ}}$ Service Request (input/output)

This control line asks the controller for service.

25 $\overline{\text{REN}}$ Remote Enable (input/output)

This control line selects remote or local control for a device.

39 $\overline{\text{EOI}}$ End of Identify (input/output)

This control line indicates the end of a transfer of multiple bytes or, with ATN, executes a parallel poll.

CHAPTER 3

INTERNAL REGISTERS

The uPD7210 contains 16 registers; eight read registers and eight write registers.

RS2	RS1	RS0	Read Registers								
0	0	0	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Data In [0R]
0	0	1	CPT	APT	DET	END	DEC	ERR	DO	DI	Interrupt Status 1 [1R]
0	1	0	INT	SRQI	LOK	REM	CO	LOKC	REMC	ADSC	Interrupt Status 2 [2R]
0	1	1	S8	PEND	S6	S5	S4	S3	S2	S1	Serial Poll Status [3R]
1	0	0	CIC	$\overline{\text{ATN}}$	SPMS	LPAS	TPAS	LA	TA	MJMN	Address Status [4R]
1	0	1	CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0	Command Pass Through [5R]
1	1	0	X	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0	Address 0 [6R]
1	1	1	EOI	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1	Address 1 [7R]

Figure 3.1 Read Registers

			Write Registers								
0	0	0	BO7	BO6	BO5	BO4	BO3	BO2	BO1	BO0	Byte Out [0W]
0	0	1	CPT	APT	DET	END	DEC	ERR	DO	DI	Interrupt Mask 1 [1W]
0	1	0	0	SRQI	DMAO	DMAI	CO	LOKC	REMC	ADSC	Interrupt Mask 2 [2W]
0	1	1	S8	rsv	S6	S5	S4	S3	S2	S1	Serial Poll Mode [3W]
1	0	0	ton	lon	TRM1	TRM0	0	0	ADM1	ADM0	Address Mode [4W]
1	0	1	CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0	Auxiliary Mode [5W]
1	1	0	ARS	DT	DL	AD5	AD4	AD3	AD2	AD1	Address 0/1 [6W]
1	1	1	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0	End of String [7W]

Figure 3.2 Write Registers

3.1 Data Registers

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	(0R) Data In register
BO7	BO6	BO5	BO4	BO3	BO2	BO1	BO0	(0W) Byte Out register

These data registers are used for transferring commands and data between the GPIB and the microcomputer system. The Data In register holds data sent from the talker over the GPIB when the uPD7210 is designated as the listener. Data is output over the data bus with a read operation. The contents of the Data In register are held until the next eight bits of data are received.

The Byte Out register holds data or a command written to it by a write operation and sends the data or command to the GPIB. The contents of the Byte Out register are updated at the trailing edge of the write strobe.

3.2 Interrupt Registers

CPT	APT	DET	END	DEC	ERR	DO	DI	(1R) Interrupt Status Register 1
INT	SRQI	LOK	REM	CO	LOKC	REMC	ADSC	(2R) Interrupt Status Register 2
CPT	APT	DET	END	DEC	ERR	DO	DI	(1W) Interrupt Mask Register 1
0	SRQI	DMAO	DMAI	CO	LOKC	REMC	ADSC	(2W) Interrupt Mask Register 2

The interrupt registers are composed of interrupt status bits, interrupt mask bits, and other bits not associated with interrupts.

3.2.1 Interrupt Bits

There are thirteen possible interrupt conditions. Each possible interrupt condition has an interrupt status bit and an interrupt mask bit associated with it.

Table 3.1 shows the conditions under which an interrupt status bit is set or reset (bit=1=set). The interrupt mask bit enables or disables the corresponding interrupt condition. When the interrupt mask bit is set, the corresponding interrupt condition is enabled.

Bit	Set Conditions	Reset Conditions
CPT	[UCG+ACG·(TADS+LADS)]·undefined·ACDS·B_0*+UDPCF·SCG·ACDS·B_0	pon+(Read INT Status 1 Reg.)* ↓
UDPCF	[UCG+ACG·(TADS+LADS)]·undefined·ACDS·B_0	[(UCG+ACG)·defined+TAG+LAG]·ACDS+$\overline{B_0}$+pon
APT	ADM 1*·ADM 0*·(TPAS+LPAS)·SCG·ACDS	pon+(Read INT Status 1 Reg.) ↓
DET	DTAS	
END	LACS·(EOI+EOS·A_2)*·ACDS	
DEC	DCAS	
ERR	SDYS·DAC·RFD+SIDS·(Write Byte Out Reg.)*+(SDYS→SIDS)*	
DO	(TACS·SGNS) ↑	(Read INT Status 1 Reg.) ↓ + $\overline{\text{TACS}}$+$\overline{\text{SGNS}}$
DI	LACS·ACDS·$\overline{\text{Continuous Mode}}$	pon+(Read INT Status 1 Reg.) ↓ +(Finish Handshake)*·(Holdoff Mode)* +(Read Data In Reg.)*
SRQI	(CIC*·SRQ·$\overline{\text{RQS}\cdot\text{DAV}}$) ↑	pon+(Read INT Status 2 Reg.)* ↓
LOKC	LOK* ↑ +LOK ↓	
REMC	REM* ↑ +REM ↓	
ADSC	(TA* ↑ +TA ↓ +LA* ↑ +LA ↓ +$\overline{\text{CIC}}$ ↑ +CIC ↓ + +MJMN* ↑ +MJMN ↓)·$\overline{\text{lon+ton}}$	
CO	(CACS·SGNS) ↑	(Read INT Status 2 Reg.) ↓ + $\overline{\text{CACS}}$+$\overline{\text{SGNS}}$

Table 3.1 Interrupt Status Bits

Notes:

A_2:	auxiliary register A
B_0:	auxiliary register B
ADM0,1:	address mode register bits
LOK, REM:	interrupt status register 2 bits
TA, LA, CIC, MJMN:	address status register bits
(finish handshake):	Finish Handshake auxiliary command issued
(holdoff mode):	RFD holdoff state
SDYS->SIDS:	transition from SDYS to SIDS

The INT bit (interrupt status register 2) is the logical OR of the enabled interrupt status bits. When any unmasked interrupt status bit is set, the INT bit=1. This makes the INT pin active.

When the CPU receives an interrupt, it can tell which condition triggered the interrupt by reading the interrupt status register. All the bits of the interrupt status register are cleared after a read. If an interrupt occurs during a read, the interrupt request is held until after the register is cleared, then it is placed in the register.

CPT Command Pass Through

When B_0=1, this bit indicates that an undefined command has been received over the GPIB or that a secondary command has been received just after an undefined command.

When this bit is set, the DAC message is held and the handshake stops until the Valid auxiliary command is issued. The undefined command can be read from the command pass through register.

The UDPCF also indicates that an undefined primary command has been received.

APT Address Pass Through

This indicates that the secondary address (which the CPU is required to check in address mode 3) has been received.

When this bit is set, the DAC message is held and the handshake stops until either the Valid or Non-valid auxiliary command is issued. The secondary address can be read from the command pass through register.

DET Device Trigger

This indicates that the device has been in DTAS. A high pulse is output when T/R3 is used as the TRIG pin.

END

This indicates that the transfer of a data block is complete. This bit is set when either the END message (EOI) or EOS message (when the contents of the EOS register and the Data In register are the same) is received.

DEC Device Clear

This indicates that the device is in DCAS.

ERR Error

This bit indicates that the contents of the Byte Out register have been lost. This bit is set when data is sent over the GPIB without a specified listener or when a byte is written to the Byte Out register during SIDS or during the SDYS->SIDS transition.

DO, DI Data Out, Data In

The DO bit indicates a data write request to the Byte Out register.

The DI bit indicates that a data byte has been written to the Data In register from the GPIB and asks the CPU to read the Data In register.

In continuous mode, the DI bit is not set by a write to the Data In register.

When you are not using DMA, you can use the DMAREQ pin as the DO/DI interrupt pin. This causes the DMAO and DMAI bits to function as mask bits.

SRQI Service Request Input

This indicates that an SRQ message has been received while the controller is active (CIC=1).

When a service request comes from several devices, the RQS message is detected on the DIO line and the SRQI bit is set again.

LOKC, REMC Lockout Change, Remote Change

LOKC indicates a change in the value of the LOK bit (RWLS+LWLS). REMC indicates a change in the value of the REM bit (REMS+RELS).

ADSC Address Status Change

This indicates that a change occurred in one of the four bits (TA, LA, CIC, MJMN) of the address status register. You can find the values of these bits by reading the address status register.

CO Command Output

This indicates a request that a command be written to the Byte Out register so that it can be transmitted to the GPIB.

3.2.2 Non-interrupt Bits

LOK, REM Lockout, Remote

These indicate the status of the RL interface functions. The LOK bit indicates that the device is in LWLS or RWLS. The REM bit indicates that the device is in REMS or RWLS.

DMAO, DMAI DMA Output, DMA Input

These bits enable and disable DMA transfers between memory and data registers.

When DMAO=1 and the uPD7210 is in data transmission enable mode, a DMA request that asks for a data byte to be written to the byte out register is generated. When DMAI=1, a DMA request that asks for data to be written from the GPIB to the Data In register is generated.

3.3 Serial Poll Registers

S8	PEND	S6	S5	S4	S3	S2	S1

(3R) Serial Poll Status Register

S8	rsv	S6	S5	S4	S3	S2	S1

(3W) Serial Poll Mode Register

The serial poll mode register holds the status byte (S1-S6, S8; sent over the GPIB) and the local message, rsv.

When the CPU sets rsv=1 (rsv message is issued), the state of the SR interface function becomes SRQS when the controller is not serial polling the device.

When the polling of the controller puts the T/TE interface in SPAS, the contents of the serial poll mode register are sent over the DIO lines as STB and RQS messages. The rsv bit is cleared when the SR interface function is in APRS.

You can read STB in the serial poll mode register from the serial poll status register. The PEND bit is set when $\overline{\text{rsv}}$=1 and cleared when NPRS•$\overline{\text{rsv}}$=1. You can confirm that a request was accepted and that the STB bit was transmitted by reading the status of the PEND bit.

You can clear all the bits of the serial poll mode register with a reset pulse or with the Chip Reset auxiliary command.

3.4 Address Mode Status Registers

CIC	$\overline{\text{ATN}}$	SPMS	LAPS	TPAS	LA	TA	MJMN

(4R) Address Status Register

ton	lon	TRM1	TRM0	0	0	ADM1	ADM0

(4W) Address Mode Register

The address mode register selects the functions of the T/R2 and T/R3 pins and selects the address mode.

3.4.1 Selecting T/R2 and T/R3 Pin Function

The T/R2 and T/R3 pin functions are selected as follows:

TRM1	TRM0	T/R2	T/R3
0	0	EOIOE	TRIG
0	1	CIC	TRIG
1	0	CIC	EOIOE
1	1	CIC	PE

Table 3.2 Pin Function Select

3.4.2 Selecting Address Mode

ton	lon	ADM1	ADM0	Address Mode	Contents of Adr. reg. 0	Contents of Adr. reg. 1
1	0	0	0	talk only	not used	not used
0	1	0	0	listen only	not used	not used
0	0	0	1	Address mode 1	major talk or major listen addr.	minor talk or minor listen address
0	0	1	0	Address mode 2	primary addr. talk or listen	secondary address talk or listen
0	0	1	1	Address mode 3	primary addr. major talk or major listen	primary address minor talk or minor listen

All other combinations are prohibited.

Table 3.3 Address Mode Select

The uPD7210 automatically detects two types of addresses. These are held in address registers 0 and 1.

☐ **Address Mode 1**

Address mode 1 includes two types of device addresses: major and minor. MTA or MLA reception is indicated when either address equals the received address. The interface function is either T or L.

☐ **Address Mode 2**

Address register 0 holds the primary address and address register 1 holds the secondary address. The interface function is either TE or LE.

☐ **Address Mode 3**

Address mode 3 provides major and minor primary addresses. The CPU must identify the secondary address by reading the command pass through register. The interface function is either TE or LE.

☐ **Talk Only and Listen Only Modes**

Address identification is not necessary in these modes. No address register is used.

Bit Name	Definition
$\overline{ATN}$	$\overline{ATN}$
LPAS	LPAS
TPAS	TPAS
CIC	$\overline{CIDS+CADS}$
LA	LACS+LADS=$\overline{LIDS}$
TA	TACS+SPAS+TADS=$\overline{TIDS}$
MJMN	Set: receipt of minor talk or minor listen address Reset: receipt of major talk or major listen address; ADM0=0 or pon=1 or IFC=1
SPMS	SPMS

Table 3.4 Address Status Register Bits

The $\overline{ATN}$ bit confirms that the device has entered CSBS after the Go To Standby auxiliary command has been issued.

LPAS and TPAS determine whether the received secondary address is the talk or listen address after an APT interrupt.

CIC, LA, TA, and MJMN are used when you must know their values when an ADSC interrupt is generated.

3.5 Address Registers

x	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0	(6R) Address Register 0
EOI	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1	(7R) Address Register 1
ARS	DT	DL	AD5	AD4	AD3	AD2	AD1	(6W) Address Register 0/1

You set device address by writing the address to address register 0/1. Each bit of address register 0/1 is as follows:

Bit Name	Function		
ARS	0 1	Address register 0 Address register 1	selects the address register to which the low-order bits are written (AD1 to AD5)
DT	0 1	permitted prohibited	permits or prohibits the set address (AD1 to AD5) detected as a talk address. This bit corresponds to DT1 or DT0 of the address registers.
DL	0 1	permitted prohibited	permits or prohibits the set address (AD1 to AD5) detected as a listen address. This bit corresponds to DL1 or DL0 of the address registers.
AD1-AD5			these bits indicate device addresses and correspond to AD5-0 to AD1-0 and AD5-1 to AD1-1.

Table 3.5 Address Register 0/1 Bits

For example, when the following codes are written into address register 0/1 in address mode 1:

0 0 X A A A A A	AAAAA: major talk address
1 0 X B B B B B	BBBBB: minor talk address

the device has both the major and minor talk addresses. The uPD7210 operates as if the MTA has been received when the talk address of either AAAAA or BBBBB is received.

You can read the adddress (AD1-AD5) and the DT and DL bits written in address register 0/1 in either address register 0 or 1, according to the value of the ARS bit. However, the value of bit 7 of address register 0 is unknown. Bit 7 (EOI) indicates the value of the EOI line latched when a data byte is received.

3.6 Command Pass Through Register

CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

(5R) Command Pass Through Register

The CPU reads the data on the DIO lines through the command pass through register. You can read the following three types of registers from the CPT register:

☐ CPT=1, B_0=1
This indicates an undefined command (command not defined in IEEE Std. 488-1978) or a secondary command received after an undefined primary command.

☐ APT=1, address mode 3
This indicates a secondary address.

☐ After a parallel poll
This indicates the PPR message in the parallel poll. If the PPRn message to be output by the uPD7210 is true during the parallel poll execution in PPAS, the PPR message is latched to the CPT register instead of being output to the DIO line.

In the first two cases, the CPT contains the data on the DIO line. In the last case, the PPR message is latched into the CPT register when CPPS=1 until CIDS=1 or a command byte is sent over the GPIB.

3.7 End of String (EOS) Register

EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0

(7W) End of string register

This register holds the seven- or eight-bit EOS message byte used by the GPIB to detect the end of a data block transfer. The length of the EOS byte is selected by A_4 (bit 4 of auxiliary register A).

When data is being received with A_2 set, end of string is detected when the received data and the contents of EOS are equal. This causes the END bit to be set.

When data is being transmitted with A_3 set, the END message is sent at the same time as the transmitted data if the transmitted data and EOS register are equal.

3.8 Auxiliary Mode Register

CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0

(5W) Auxiliary mode register

A write to this register generates one of the following operations according to the value of the CNT bits (CNT0-2):

☐ a write to the auxiliary register
☐ an auxiliary command is issued
☐ state change prohibit time is set
☐ a write to the parallel poll register

CNT 2 1 0	COM 4 3 2 1 0	Operation
0 0 0	C_4 C_3 C_2 C_1 C_0	issues an auxiliary command specified by C_0-C_4
0 0 1	0 F_3 F_2 F_1 F_0	specifies the reference clock frequency and determines T_1, T_6, T_7, and T_9
0 1 1	U S P_3 P_2 P_1	writes to parallel poll register
1 0 0	A_4 A_3 A_2 A_1 A_0	writes to auxiliary register A
1 0 1	B_4 B_3 B_2 B_1 B_0	writes to auxiliary register B
1 1 0	0 0 0 E_1 E_0	writes to auxiliary register E

Table 3.6 Auxiliary Register Bits

3.8.1 Auxiliary Commands

You issue the following auxiliary commands by writing $000C_4C_3C_2C_1C_0$ to the auxiliary mode register.

C_4	C_3	C_2	C_1	C_0
0	0	0	0	0

Immediate Execute pon

A local message "pon" is generated that places the following interface functions into their idle states:

SIDS, AIDS, TIDS, SPIS, TPIS, LIDS, LPIS, NPRS, LOCS, PPIS,
PUCS, CIDS, SRIS, SIIS

If you issue this command while a pon local message is already active (by either an external reset pulse or the Chip Reset auxiliary command) the pon local message becomes false.

0	0	0	1	0

Chip Reset

This command performs the same function as an external reset pulse. The uPD7210 is reset to the following state:

- ☐ local message pon is set and the interface functions are placed in
- ☐ their idle states;
- ☐ all bits of the serial poll mode register are cleared;
- ☐ EOI bit is cleared;
- ☐ all bits of the auxiliary A, B, and E registers are cleared;
- ☐ the Parallel Poll flag and RSC local message are cleared;
- ☐ sets $N_F=8$ $(F_3=1, F_2=F_1=F_0=0)$;
- ☐ clears the TRM0 bit and the TRM1 bit;

0	0	0	1	1

Finish Handshake

This command ends the handshake by releasing the RFD message transmission from the holdoff state.

0	0	1	0	0	**Trigger**

This command generates a high pulse in the TRIG pin (T/R3 pin when TRM1=0). This auxiliary command performs the same function as if the DET bit (interrupt status register 1) were set. The DET bit is not set by this command.

0	X	1	0	1	**Return to Local**

When C_3=0, this command generates the local message "rtl" in the form of pulses. If rtl is already set, this command clears it.

When C_3=1, this command sets the local message rtl.

0	0	1	1	0	**Send EOI**

This command sends the END message with the next data byte. It is valid only for TACS.

0	0	1	1	1	**Non-valid**

This command releases the DAC message heldoff by the address pass through. The uPD7210 is allowed to operate as if an OSA message has been received.

0	1	1	1	1	**Valid**

This command releases the DAC message heldoff by address pass through and functions as if an MSA message had been received. The DAC message is released at the time of command pass through. DAC is also released if DCAS or DTAS is in holdoff state.

0	X	0	0	1	**Set Parallel Poll Flag**

This command sets the Parallel Poll Flag to the value of C_3. The value of the Parallel Poll flag is used as the local message ist when B_4=0; the value of SRQS is used as the ist when B_4=1.

1	0	0	0	0

Go to Standby

This command sets the local message gts at the time of CACS. gts is cleared when CACS goes low.

1	0	0	0	1

Take Control Asynchronously

This generates the local message tca in the form of pulses.

1	0	0	1	0

Take Control Synchronously

This command sets the local message tcs. tcs is effective only when CSBS+CSWS=1. tcs is cleared at the leading edge of CACS.

1	1	0	1	0

Take Control Synchronously on End

This command sets the local message tcs when the data block transfer end (END=1) is generated at CSBS. tcs is cleared at the leading edge of CACS.

1	0	0	1	1

Listen

This command generates the local message ltn in the form of a pulse.

1	1	0	1	1

Listen in Continuous Mode

This command generates the local message ltn in the form of a pulse and places the device in continuous mode.

In continuous mode, the local message rdy is issued when ANRS is initiated unless data block transfer end is detected (END=1). When the end is detected, the device is placed in the RFD holdoff state, preventing generation of the rdy message. In continuous mode, the DI bit is not set when a data byte is received. The continuous mode caused by this command is released when the Listen auxiliary command is issued or LIDS is initiated.

1	1	1	0	0

Local Unlisten

This command generates the local message lun in the form of a pulse.

1	1	1	0	1

Execute Parallel Poll

This command sets the local message rpp. rpp is cleared when CPPS+CIDS=1. The transition of the C interface function is not guaranteed if the local messages rpp and gts are issued simultaneously when $CACS \cdot \overline{STRS} \cdot \overline{SDYS}=1$.

1	X	1	1	0

Set/Clear IFC

This command generates the local message rsc and sets IFC to the value of C_3. In order to meet IEEE Std. 488-1978, you must not issue the Clear IFC command until IFC has been held true for at least 100us.

$C_3=1=IFC$ $\qquad$ $C_3=0=\overline{IFC}$

1	X	1	1	1

Set/Clear REN

This command generates the local message rsc and sets REN to the value of C_3. In order to meet IEEE Std. 488-1978, you must not issue the Set REN command until REN has been held false for at least 100us.

$C_3=1=REN$ $\qquad$ $C_3=0=\overline{REN}$

1	0	1	0	0

Disable System Control

This command clears the local message rsc.

3.8.2 Internal Counter

The internal counter generates the state change prohibit times (T_1, T_6, T_7, T_9) specified in IEEE Std. 488-1978.

$$T_1 \text{ (low speed)}=T_6=T_7=T_9=\frac{2N_F}{fc}+t_{sync}$$

$$T_1 \text{ (high speed)} = \frac{N_F}{2fc} + t_{sync}$$

Where N_F=integer represented by binary F_0-F_3. $1 \leq N_F \leq 8$
fc=reference clock frequency (clock input)
$0 \leq t$ sync $\leq$ max (reference clock high or low period)

T_1 (high speed) is used for all bytes following the first byte sent after each false transition of ATN if B_2=1. You should use this T_1 if you are using three-state bus drivers on the DIO, DAV, and EOI lines. In other cases, use T_1 (low speed).

When N_F(MHz)=fc, then:

T_1 (low speed)$=T_6=T_7=T_9=$2us+t sync
T_1 (high speed)=500ns+t sync

When N_F(MHz)<fc, IEEE Std. 488-1978 is not satisfied.

t_{sync} is a synchronization error greater than zero but less than the larger of the reference clock high and low.

$$0 \leq t_{sync} \leq \text{max reference clock high or low period}$$

For a 50% duty clock:

$$0 \leq t_{sync} \leq 1/2 \text{ reference clock period}$$

3.8.3 Auxiliary Register A

You can write to auxiliary register A by writing $100A_4A_3A_2A_1A_0$ to the auxiliary mode register. The contents of auxiliary register A control the messages (holdoff, EOS/END) associated with data transfer.

A_1	A_0	Data Receiving Mode
0	0	normal handshake mode
0	1	RFD Holdoff on All data mode
1	0	RFD Holdoff on End mode
1	1	Continuous mode

Table 3.7 Data Receiving Modes

☐ In Normal handshake mode, the local message $\overline{rdy}$ is generated when data is received. When the received data is read from the Data In register, rdy is generated in ANRS. This causes the RFD message to be transmitted and the handshake continues.

☐ In RFD Holdoff on All Data mode, RFD is not sent true after data is received until the microprocessor issues the Finish Handshake auxiliary command. Unlike normal handshake mode, this mode does not generate the rdy message even if the received data is read through the Data In register (that is, the RFD message is not generated).

☐ In RFD Holdoff on End mode, operation is the same as the previous mode when the end of the data block (EOS or END message) is detected. Handshake holdoff is released by the Finish Handshake auxiliary command.

☐ In continuous mode, the rdy message is generated when in ANRS util the end of the data block is detected. A holdoff is generated at the end of a data block. The Finish Handshake command must be issued to release the holdoff. This mode is useful for monitoring the data block transfer without data reception. In this mode, the DI bit is not set by the reception of a data byte.

Bit Name	Function		
A_2	0	prohibit	permits or prohibits setting the END bit at reception of the EOS message
	1	permit	
A_3	0	prohibit	permits or prohibits automatic transmission of the END message at the same time as the EOS message in TACS
	1	permit	
A_4	0	7-bit EOS	selects seven or eight bits as the valid length of the EOS message
	1	8-bit EOS	

Table 3.8 Functions of Auxiliary Register A

3.8.4 Auxiliary Register B

You can write to auxiliary register B by writing 101$B_4B_3B_2B_1B_0$ to the auxiliary mode register.

Bit Name	Function		
B_0	1	permit	permits or prohibits the detection of undefined commands; permits or prohibits the setting of the CPT bits on receipt of an undefined command
	0	prohibit	
B_1	1	permit	permits or prohibits the transmission of the END message in SPAS.
	0	prohibit	
B_2	1	T_1 (high spd)	sets T_1 (high speed) as T_1 of handshake after transmission of second byte following data transmission
	0	T_1 (low spd)	sets T_1 (low speed) as T_1 in all cases
B_3	1	INT	specifies the active level of the INT pin
	0	INT	
B_4	1	ist=SRQS	SRQS indicates the value of the ist local message (the Parallel Poll flag is ignored) SQRS=ist=1; SQRS=ist=0
	0	ist=Parallel Poll flag	the value of the Parallel Poll flag is taken as the ist local message

Table 3.9 Functions of Auxiliary Register B

3.8.5 Auxiliary Register E

You can write to auxiliary register E by writing 110000E_1E_0 to the auxiliary mode register.

Bit Name	Function		
E_0	1	enables	enables or disables DAC holdoff by initiating DCAS
	0	disables	
E1	1	enables	enables or disables DAC holdoff by initiating DTAS
	0	disables	

Table 3.10 Functions of Auxiliary Register E

3.8.6 Parallel Poll Register

You can write to the parallel poll register by writing 011USP$_3$P$_2$P$_1$ to the auxiliary mode register.

☐ When you use the subset PP1 as the PP interface function, you should not write to this register. The parallel poll response (PPR$_N$) is automatically sent out according to the PPE message issued by the GPIB controller. For example, when the values of S and ist are equal, the PPR$_N$ message is sent out true according to the specification of P$_3$P$_2$P$_1$ (=N-1).

☐ When you use the subset PP2, you must write to this register in advance. The U bit implies the local message $\overline{\text{lpe}}$. When U=0, S and P$_3$P$_2$P$_1$ mean the same as the bit of the same name in the PPE message and the write operation is the same as the receipt of the PPE message. When U=1, S and P$_3$P$_2$P$_1$ bits do not carry any meaning, but they should be reset to zero.

CHAPTER 4
USING THE uPD7210

4.1 Transmitting Commands

A command is transmitted by writing the code to the Byte Out register when a request for command transmission is received (CO=1). This is repeated when you send several command bytes.

4.2 Processing Undefined Commands

When B_0=1, the DAC message is held false and the CPT bit is set when an undefined command is received. The CPU reads the undefined code via the CPT register. The handshake that stopped is completed when the Valid auxiliary command is issued.

When B_0=0, the handshake is completed, just as when a defined command is received, and the CPT bit remains 0. The received code is ignored.

A secondary command received immediately after an undefined primary command is handled as an undefined command.

4.3 Processing Address Pass Through

The APT bit is set when the secondary address is received. This is the case where a secondary command is received in address mode 3 and LPAS+TPAS=1.

☐ Address Mode 3 (ton=lon=0 and AIM1=AIM0=1)
In this mode, the TE and LE interface functions are used as the talker and the listener, respectively. Address register 0 holds the major primary address and address register 1 holds the minor primary address.

☐ LPAS+TPAS=1
This condition is satisfied when either the MTA (My Talk Address) or the MLA (My Listen Address) has been received.

When the APT bit is set, the handshake stops with the DAC message held false just as the CPT bit is set. The CPU must then perform the following:

☐ determine whether the secondary command just received is a listen, talk, major, or minor address by reading the LPAS, TDAS, and MJMN bits of the address status register

☐ determine whether the secondary command read through the CPT register is my address. If it is my address, the Valid auxiliary command is issued. If it is not my address, the Non-valid auxiliary command is issued.

When the Valid auxiliary command is received, the uPD7210 assumes that the MSA (My Secondary Address) message has been received. When this command is issued, LADS=TIDS=1 if LPAS=1 or TADS=LIDS=1 if TPAS=1, the DAC message is sent true, and the handshake is finished.

When the Non-valid auxiliary command is received, the uPD7210 assumes that the OSA (Other Secondary Address) message has been received. When this command is issued, TIDS=1 if TPAS=1, the DAC message is sent true, and the handshake is finished.

4.4 Beginning Data Transfer

After specifying the talker and the listener, issue the Go To Standby auxiliary command. Data transfer begins when the last command written to the Byte Out register has been transmitted.

4.5 Transmitting Data

When a data request is received (DO=1), one byte of data is transmitted by a write to the Byte Out register. This process is repeated to send several bytes. The DO bit is cleared when read or when a write is made to the Byte Out register.

You can use DMA for sending or receiving data. When a request to send data is received and DMAO=1, or when a device asks to received data and DMAI=1, a DMA request is generated (DMAREQ=1).

4.6 Receiving Data

When the CPU receives a data receive request, the contents of the Data In register are read. Data is received in the four modes below.

4.6.1 Normal Handshake Mode (A_0=A_1=0)

When the device receives data as a listener, a data receive request (DI=1) is made and the RFD message is sent false. As the receive data is read from the Data In register, the RFD message is sent true, informing the talker that the listener is read for the next data.

4.6.2 RFD Holdoff on All Data Mode (A_0=1, A_1=0)

When the device receives data as a listener, a data receive request (DI=1) is made and the RFD message is sent false, just as in normal handshake mode. However, even if the CPU reads the received data from the Data In register, the RFD message is held false until the Finish Handshake auxiliary command is issued. While the RFD message is being held false, the next data is not received and the contents of the Data In register are not updated. Therefore, the CPU can read the same data several times, if desired.

4.6.3 RFD Holdoff on End Mode (A_0=0, A_1=1)

In this mode, the RFD message is sent and held false when the data is received with the END bit set. Issue the Finish Handshake auxiliary command to send the RFD message true. The function is the same as in normal handshake mode unless the END bit is set.

4.6.4 Continuous Mode ($A_0=A_1=1$)

In this mode, the RFD message is returned true in response to the DAV message unless the END bit is set. When the END bit is set, operation proceeds as in RFD Holdoff on End mode. This mode is used for data block end detection or interrupt synchronized with the handshake by the tcs local message. When the Listen with Continuous Mode auxiliary command is issued at CACS, operation is the same as in this mode.

4.7 Completing Data Block Transfer

In accordance with IEEE STd. 488-1978, the following two methods are provided for detecting the end of the data block.

4.7.1 Placing EOS Byte After Data Block

The EOS code may be user-defined, but it is not possible to identify the EOS message if a code appearing in the data byte is used. You may not use it if the data byte is a full eight bits. In order to use ASCII code, use the LF code as EOS and the others as data bytes.

4.7.2 Using the EOI Line

In this method, the END message (EOI=1) is sent out with the last byte of the data block.

4.7.3 Transmission of the EOS Message

This transmission works the same as transmission of a data byte. When DO=1, the EOS message can be transmitted by writing the EOS code to the Byte Out register.

4.7.4 Transmission of the END Message

After the Send EOI auxiliary command is issued to the uPD7210, the END message is transmitted with the data byte when the next data byte is written to the Byte Out register.

When A_3=1 (Output EOI on EOS Sent), the END message is sent when the EOS message is sent (based on the contents of the Byte Out register, the EOS register, and the value of A_4).

4.7.5 Detecting the EOS Message

When A_2=1 (End on EOS Received), the END bit is set when the EOS message is received. Receipt of the EOS message is determined by the contents of the Data In register, the EOS register, and the value of A_4.

When the STB (Status Byte) code in the serial poll register and the EOS code agree, this is not evidence of receipt of the EOS message.

4.7.6 Detecting the END Message

The END bit is set when the END message is received in LACS. You can identify the END or EOS message because the value of the EOI line is latched in the EOI bit when data is received and data itself is latched in the Data In register.

4.8 Discontinuing Data Transfer

There are three methods of discontinuing data transfer, as decribed below.

4.8.1 Using the Take Control Asynchronously Command

When the Take Control Asynchronously command is issued, ATN is set equal to one and the data transfer stops. You must be careful when using this method to stop data transfer because data on the DIO line might be taken as a command.

4.8.2 Using the Take Control Synchronously Command

When the Take Control Synchronously command is issued, ATN is set equal to one and the data transfer stops at the end of the handshake (ANRS). You must specify the controller as the listener before using this method.

To specify the controller as listener, issue the Listen or Listen with Continuous Mode auxiliary command before issuing the Go to Standby auxiliary command. When you use the Listen command, the listener functions in the mode specified by the A_0 and A_1 bits. When you use the Listen with Continuous Mode command, the listener functions in continuous mode, whatever the values of A_0 and A_1.

4.8.3 Using the Take Control Synchronously on End Command

When this command is issued, ATN is set equal to one at the end of the current data transfer and transfer stops.

4.9 Serial Polling

To request service, you must confirm that there is no pending service request (PEND=0). Write the STB into the serial poll mode register with the local message rsv=1. If the device is not in SPAS, the SRQ message is sent true as soon as the rsv message is set. If the device is in SPAS, the SRQ message remains false until the serial polling is complete (SPAS=0). The PEND bit indicates whether a service request is accepted or left pending. It is set when rsv=1 and cleared when the STB is read out by the controller-in-charge (SPAS=0), or when the local message rsv is cleared before SPAS=1.

The STB set to the serial poll mode register is sent out when the STB is asked to send. The STB is sent only once even if the controller does not assert ATN after the first transfer. The END message is sent out if B_1=1.

4.10 Parallel Polling

When the Execute Parallel Poll auxiliary command is issued and the local message rpp is set to 1, the parallel poll is executed as soon as the C interface function is placed in the proper state (CAWS or CACS). The PPR (Parallel Poll Response) is automatically taken in the CPT register and rpp is cleared to 0. The CPU knows that this operation is complete based on the condition of CO=1. The PPR can be obtained by reading the contents of the CPT register, which are held until a command is transmitted or the controller becomes inactive.

4.11 Parallel Poll Protocol

Before a parallel poll is executed, you must specify to which line of DIO1 to DIO8 the one-bit status (ist:individual status) should be output and which polarity should be used. The following two methods are provided for this.

☐ Remote Configuration (PP1)
In this method, the specifications are made by either PPE or PPD messages sent from the controller.

☐ Local Configuration (PP2)
In this method, the specifications are made from the device.

Specifications by the CPU are not required in the Remote Configuration. In the Local Configuration, you must write the following values to the auxiliary mode register.

0	1	1	U	S	P_3	P_2	P_1

status bit output line (DIO1 to DIO8)

status bit polarity S=1: in phase
S=0: in reverse phase

response to parallel poll U=1: no response
U=0: response made

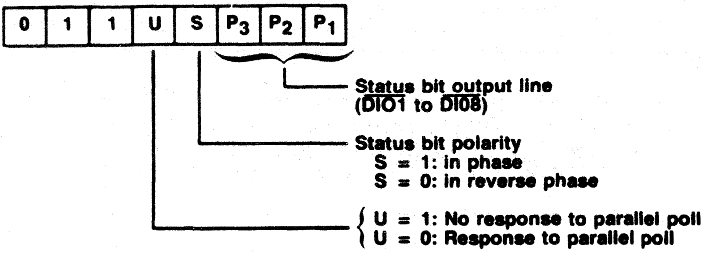


Figure 4.1 Local Configuration for Parallel Poll

CHAPTER 5

GPIB-INTERFACE USING uPD7210

This chapter describes how to set up a GPIB-Interface using the uPD7210. Complete hardware schematics as well as a software example will be given, to show you how easy an implementation of a GPIB-Interface with the uPD7210 may be.

5.1 Hardware

Figure 1 shows one possible way to implement a minimal uPD8085 system including a GPIB-Interface. As the uPD7210 is directly bus-compatible with the uPD8080,8085,8088 ... processor series, no additional hardware is needed to connect the uPD7210 to the uPD8085.

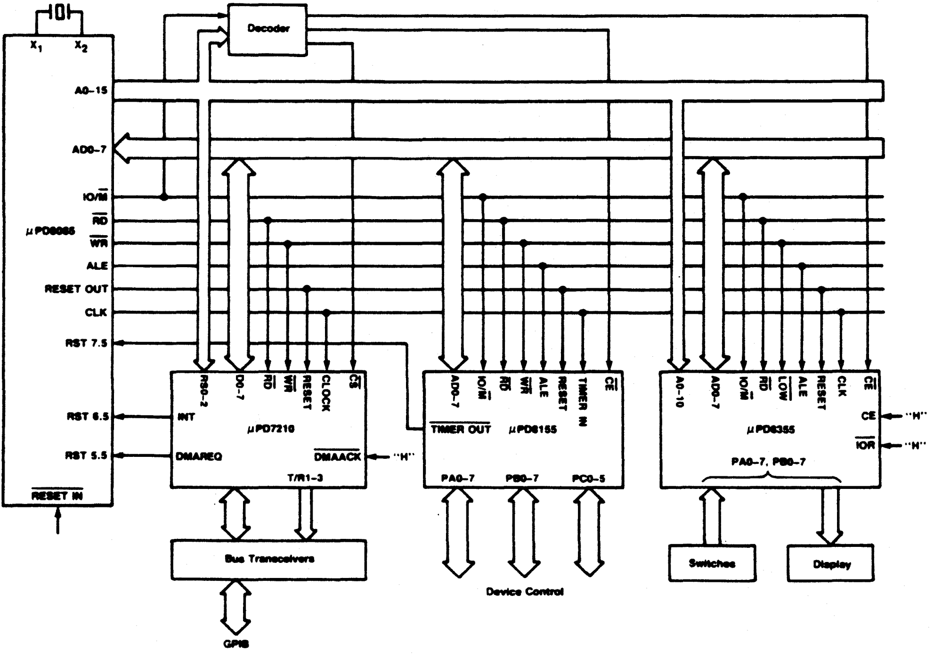
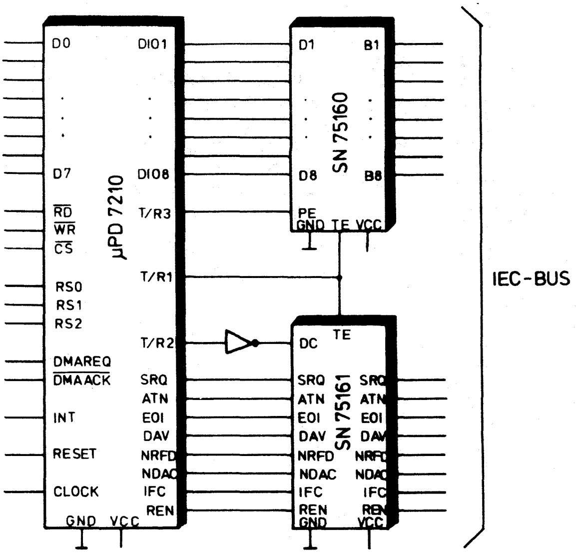


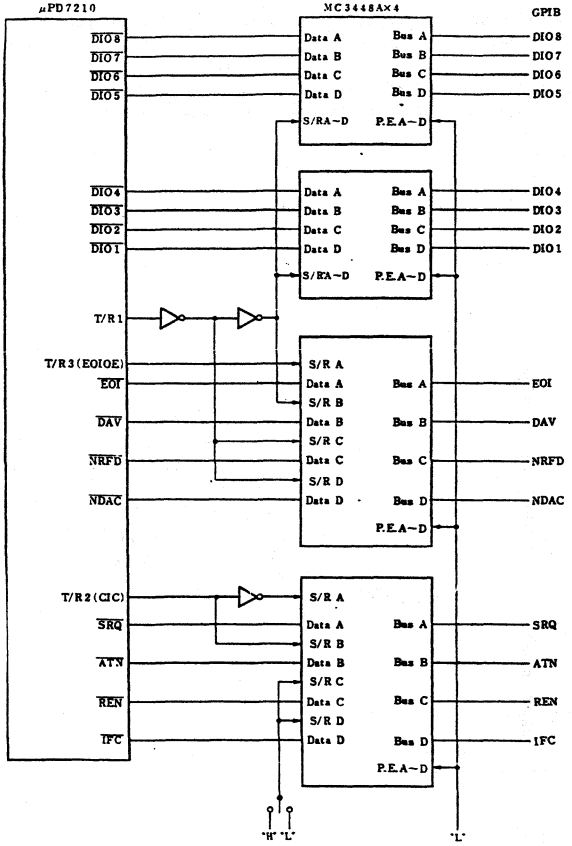
Figure 1. Minimum 8085 System using uPD7210

The uPD7210 is attached to the GPIB-Bus via appropriate bus drivers, which meet the IEEE-488 bus specifications (Figures 2,3).
The uPD7210 is able to operate with almost any IEEE 488 bus drivers, which are controlled by the T/R1,T/R2 and T/R3 lines. These lines are programmed by software depending on the type of the bus drivers used.



Note:
In the case of low-speed data transfer (B_2=0), the T/R3 pin can be used as a TRIG output. The PE input of the SN75160 should be set to "0".

Figure 2. Using SN75160 and SN75161 Bus drivers (T/R mode 3: TRM1=TRM0=1)



Note:
In this example, no high-speed data transfer can be made because the bus transceiver are open collector type (set B_2=0).

Figure 3. Using MC3448A Bus drivers (T/R mode 2: TRM1=1,TRM0=0)

5.2 SOFTWARE

In this software example the uPD7210 operates as CONTROLLER/TALKER on the GPIB-Bus. After specifying a LISTENER the uPD7210 will send a short stream of data to the GPIB-Bus. Figure 4 shows the command and data sequence put on the GPIB-Bus by the uPD7210.

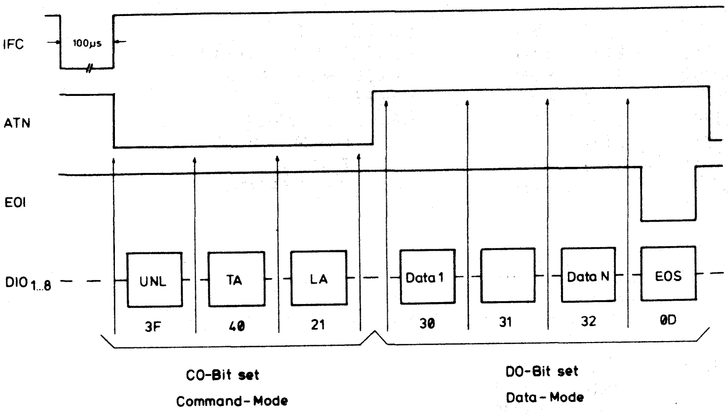


Figure 4. Data Transmission with the GPIB-Bus

During initialisation the IFC (interface clear) line is activated to reset all devices attached to the bus. Before starting the data transfer, a TALKER and at least one LISTENER has to be specified by the CONTROLLER. For this reason the CONTROLLER places the command UNLISTEN, the TALKER-ADDRESS (his own one) and a LISTENER-ADDRESS on the bus. As soon as the CONTROLLER releases the ATN line (end of command mode), the TALKER starts the transmission of data (data mode). With the last character transmitted (carriage return) the TALKER activates the EOI (end or identify) line indicating that the data transfer is finished. Now the CONTROLLER again takes the bus control to initiate a new data transfer.

The software listing shows the implementation of the procedure discussed above. This listing is commented extensively and should be self-explanatory.

```
                ;**************************************************
                ;*
                ;*        uPD7210 Software Example
                ;*        ========================
                ;*
                ;* In this example the uPD7210 works as
                ;* CONTROLLER/TALKER and sends the data stream
                ;* '0' '1' '2' 'CR' to a LISTENER.
                ;*
                ;**************************************************
                ;
                ;
0080 =          BOUT      EQU  80H        ;BYTE OUT
0081 =          INTM1     EQU  81H        ;INTERRUPT MASK1
0082 =          INTM2     EQU  82H        ;INTERRUPT MASK2
0083 =          SPM       EQU  83H        ;SERIAL MODE
0084 =          ADRM      EQU  84H        ;ADDRESS MODE
0085 =          AUXM      EQU  85H        ;AUXILIARY MODE
0086 =          ADR01     EQU  86H        ;ADDRESS 0/1
0087 =          EOS       EQU  87H        ;END OF STRING
                ;
                ;
                ;*
                ;* JMP Table
                ;* =========
                ;*
                ;
0000 C38500     MAIN:     JMP  XMAIN
0003 C30900     INIT:     JMP  XINIT
0006 C33400     SEND:     JMP  XSEND
                ;
                ;
                ;*
                ;* INIT Routine
                ;* ============
                ;*
                ;* - initialise uPD7210
                ;*
                ;
0009 3E02       XINIT:    MVI  A,02H      ;Chip reset
000B D385                 OUT  AUXM
                ;
000D 3E00                 MVI  A,00H
000F D381                 OUT  INTM1      ;disable interrupt1
0011 D382                 OUT  INTM2      ;disable interrupt2
                ;
0013 3E31                 MVI  A,31H      ;Address mode 1
0015 D384                 OUT  ADRM       ;T/R mode 3 : 75160/1 drivers
                ;
0017 3E00                 MVI  A,00H      ;my address=0
0019 D386                 OUT  ADR01      ;(address 0 register)
                ;
001B 3EE0                 MVI  A,0E0H     ;disable address 1 register
```

```
001D D386               OUT  ADR01
               ;
001F 3E0D               MVI  A,0DH     ;EOS = CR
0021 D387               OUT  EOS       ;end of string = carriage return
               ;
0023 3EA4               MVI  A,0A4H    ;high speed transfer (tri state driver)
0025 D385               OUT  AUXM      ;INT pin is active high
;                                      ;ist = parallel poll flag
0027 3E24               MVI  A,24H     ;NF = 4
0029 D385               OUT  AUXM      ;if NF = 4Mhz , T1 = 500ns
               ;
002B 3E8C               MVI  A,8CH     ;normal handshake mode
002D D385               OUT  AUXM      ;7-bit word EOS
               ;
002F 3E00               MVI  A,00H      ;immediate execute power on
0031 D385               OUT  AUXM
               ;
0033 C9                 RET            ;end of initialisation
               ;
               ;*
               ;* SEND Routine
               ;* ============
               ;*
               ;* - send the data in registers B,C,D,E to the GPIB-Bus
               ;*
               ;
0034 3E8B      XSEND:   MVI  A,8BH     ;continuous mode
0036 D385               OUT  AUXM      ;output EOI on EOS sent
               ;
0038 CD7000             CALL COCHECK   ;CO Bit check (Command Output)
003B 3E3F               MVI  A,3FH     ;send UNLISTEN
003D D380               OUT  BOUT
               ;
003F CD7000             CALL COCHECK
0042 3E40               MVI  A,40H     ;send TALKER Address
0044 D380               OUT  BOUT
               ;
0046 CD7000             CALL COCHECK
0049 3E21               MVI  A,21H     ;send LISTENER Address
004B D380               OUT  BOUT
               ;
004D 3E1B               MVI  A,1BH     ;local message: ltn continuous
004F D385               OUT  AUXM      ;(listen)
               ;
0051 3E1A               MVI  A,1AH     ;local message: tcs
0053 D385               OUT  AUXM      ;(take control synchronously on end)

               ;        MVI  A,10H     ;local message: gts
0055 D385               OUT  AUXM      ;(go to standby)
               ;
               ;
0057 CD7800             CALL DOCHECK   ;DO Bit check (Data Output)
005A 78                 MOV  A,B       ;send contents of register B
005B D380               OUT  BOUT
```

```
                    ;
005D CD7800                 CALL DOCHECK
0060 79                     MOV  A,C         ;send contents of register C
0061 D380                   OUT  BOUT
                    ;
0063 CD7800                 CALL DOCHECK
0066 7A                     MOV  A,D         ;send contents of register D
0067 D380                   OUT  BOUT
                    ;
0069 CD7800                 CALL DOCHECK
006C 7B                     MOV  A,E         ;send contents of register E
006D D380                   OUT  BOUT        ;end of SEND-Routine

006F C9                     RET
                    ;
                    ;
0070 DB82           COCHECK: IN   INTM2      ;CO Bit check (Command Output)
0072 E608                   ANI  08H
0074 CA7000                 JZ   COCHECK
0077 C9                     RET
                    ;
                    ;
0078 DB81           DOCHECK: IN   INTM1      ;DO Bit check  (Data Output)
007A E606                   ANI  06H
007C CA7800                 JZ   DOCHECK
007F E604                   ANI  04H
0081 C28500                 JNZ  ERR
0084 C9                     RET
                    ;
                    ;
                    ERR:                     ;put in your individual
                                             ;ERROR Routine here

                    ;
                    ;
                    ;*
                    ;* MAIN Routine
                    ;* ============
                    ;*
                    ;* - initialise uPD7210
                    ;* - send IFC (interface clear)
                    ;* - load registers B,C,D,E
                    ;* - call SEND Routine
                    ;*
                    ;
0085 CD0300         XMAIN:   CALL INIT       ;set IFC (interface clear)
0088 3E1E                   MVI  A,1EH
008A D385                   OUT  AUXM
008C 0E20                   MVI  C,20H
008E 0D             IFCWAIT: DCR  C          ;wait 100us
008F C28E00                 JNZ  IFCWAIT
0092 3E16                   MVI  A,16H       ;clear IFC
0094 D385                   OUT  AUXM
                    ;
0096 0630                   MVI  B,30H       ;load registers
```

```
0098 0E31                 MVI   C.31H
009A 1632                 MVI   D.32H
009C 1E0D                 MVI   E.0DH       :carriage return CR
               :
009E CD0600               CALL  SEND        :send contents of the registers
               :
               :
00A1 DB81      ENDCHECK:IN    INTM1       :test END bit
00A3 E610                 ANI   10H
00A5 CAA100               JZ    ENDCHECK
00A8 3E03                 MVI   A.03        :send finish handshaking command
00AA D385                 OUT   AUXM
               :
               :
00AC                      END               :end of MAIN routine
```

μPD7210

ELECTRICAL SPECIFICATION

ABSOLUTE MAXIMUM RATINGS

T_a = 25°C

PARAMETER	SYMBOL	TEST CONDITIONS	RATINGS	UNITS
Supply Voltage	V_{CC}		−0.5 ∿ +7.0	V
Input Voltage	V_I		−0.5 ∿ +7.0	V
Output Voltage	V_O		−0.5 ∿ +7.0	V
Operating Temperature	T_{opt}		0 ∿ +70	°C
Strage Temperature	T_{stg}		−65 ∿ + 125	°C

DC CHARACTERISTICS

T_a = 0 ∿ +70°C, V_{CC} = 5V ± 10%

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Low Voltage	V_{IL}		−0.5		+0.8	V
Input High Voltage	V_{IH}		+2.0		V_{CC} +0.5	V
Low Level Output Voltage	V_{OL}	I_{OL} = 2mA (4mA : T/R1 Pin)			+0.45	V
High Level Output Voltage	V_{OH1}	I_{OH} = −400uA, Except INT	+2.4			V
	V_{OH2}	I_{OH} = −400uA INT Pin	+2.4			V
		I_{OH} = −50uA	+3.5			V
Input Leakage Current	I_{IL}	I_{IN} = 0V ∿ V_{CC}	−10		+10	uA
Output Leakage Current	I_{OL}	I_{OUT} = 0.45V ∿ V_{CC}	−10		+10	uA
Supply Current	I_{CC}				+180	mA

CAPACITANCE

T_a = 25°C, V_{CC} = GND = 0V

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Capacitance	C_{IN}	f = 1 MHz All Pins Except Pin Under Test Tied to AC Ground			10	pF
Output Capacitance	C_{OUT}				15	pF
I/O Capacitance	$C_{I/O}$				20	pF

AC CHARACTERISTICS

T_a = 0 ∿ +70°C, V_{CC} = 5V ± 10%

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Address Setup to $\overline{RD}$	t_{AR}	$RS_{2\text{-}0}$	85			ns
		$\overline{CS}$	0			
Address Hold from $\overline{RD}$	t_{RA}		0			ns
$\overline{RD}$ Pulse Width	t_{RR}		170			ns
Data Delay from Address	t_{AD}				250	ns
Data Delay from $\overline{RD}$ ↓	t_{RD}				150	ns
Output Float Delay from $\overline{RD}$ ↑	t_{DF}		0		80	ns
$\overline{RD}$ Recovery Time	t_{RV}		250			ns
Address Setup to $\overline{WR}$	t_{AW}		0			ns
Address Hold from $\overline{WR}$	t_{WA}		0			ns
$\overline{WR}$ Pulse Width	t_{WW}		170			ns
Data Setup to $\overline{WR}$	t_{DW}		150			ns
Data Hold from $\overline{WR}$	t_{WD}		0			ns
$\overline{WR}$ Recovery Time	t_{RV}		250			ns
DMAREQ ↓ Delay from $\overline{DMAACK}$	t_{AKRQ}				130	ns
Data Delay from $\overline{DMAACK}$	t_{AKD}				200	ns

AC CHARACTERISTICS

$T_a = 0 \sim +70°C, V_{CC} = 5V \pm 10\%$

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
$\overline{\text{EOI}}$ ↓ → $\overline{\text{DIO}}$	t_{EODI}	PPSS → PPAS, ATN = True			250	ns
$\overline{\text{EOI}}$ ↓ → T/R1 ↑	t_{EOT11}	PPSS → PPAS, ATN = True			155	ns
$\overline{\text{EOI}}$ ↑ → T/R1 ↓	t_{EOT12}	PPAS → PPSS, ATN = False			200	ns
$\overline{\text{ATN}}$ ↓ → $\overline{\text{NDAC}}$ ↓	t_{ATND}	AIDS → ANRS, LIDS			155	ns
$\overline{\text{ATN}}$ ↓ → T/R1 ↓	t_{ATT1}	TACS + SPAS → TADS, CIDS			155	ns
$\overline{\text{ATN}}$ ↓ → T/R2 ↓	t_{ATT2}	TACS + SPAS → TADS, CIDS			200	ns
$\overline{\text{DAV}}$ ↓ → DMAREQ ↑	t_{DVRQ}	ACRS → ACDS, LACS			600	ns
$\overline{\text{DAV}}$ ↓ → $\overline{\text{NRFD}}$ ↓	t_{DVNR1}	ACRS → ACDS			350	ns
$\overline{\text{DAV}}$ ↓ → $\overline{\text{NDAC}}$ ↑	t_{DVND1}	ACRS → ACDS → AWNS			650	ns
$\overline{\text{DAV}}$ ↑ → $\overline{\text{NDAC}}$ ↓	t_{DVND2}	AWNS → ANRS			350	ns
$\overline{\text{DAV}}$ ↑ → $\overline{\text{NRFD}}$ ↑	t_{DVNR2}	AWNS → ANRS → ACRS			350	ns
$\overline{\text{RD}}$ ↓ → $\overline{\text{NRFD}}$ ↑	t_{RNR}	ANRS → ACRS LACS, DI reg. selected			500	ns
$\overline{\text{NDAC}}$ ↑ → DMAREQ ↑	t_{NDRQ}	STRS → SWNS → SGNS, TACS			400	ns
$\overline{\text{NDAC}}$ ↑ → $\overline{\text{DAV}}$ ↑	t_{NDDV}	STRS → SWNS → SGNS			350	ns
$\overline{\text{WR}}$ ↑ → $\overline{\text{DIO}}$	t_{WDI}	SGNS → SDYS, BO reg. selected			250	ns
$\overline{\text{NRFD}}$ ↑ → $\overline{\text{DAV}}$ ↓	t_{NRDV}	SDYS → STRS, T_1 = True			350	ns
$\overline{\text{WR}}$ ↑ → $\overline{\text{DAV}}$ ↓	t_{WDV}	SGNS → SDYS → STRS BO reg. selected, RFD = True $N_F = f_c$ = 8 MHz, T_1 (High Speed)			830 $+t_{SYNC}$	ns
TRIG Pulse Width	t_{TRIG}		50			ns

EXTENDED TEMPERATURE RANGE

ABSOLUTE MAXIMUM RATINGS

$T_a = 25°C$

PARAMETER	SYMBOL	TEST CONDITIONS	RATINGS	UNITS
Supply Voltage	V_{CC}		–0.5 ∿ +7.0	V
Input Voltage	V_I		–0.5 ∿ +7.0	V
Output Voltage	V_O		–0.5 ∿ +7.0	V
Operating Temperature	T_{opt}		-40 – +85	°C
Strage Temperature	T_{stg}		–65 ∿ + 125	°C

DC CHARACTERISTICS

T_a=–40°C...+85°C, V_{CC}=+5V$\pm$10%

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Low Voltage	V_{IL}		–0.5		+0.6	V
Input High Voltage	V_{IH}		+2.2		V_{CC} +0.5	V
Low Level Output Voltage	V_{OL}	I_{OL} = 2mA (4mA : T/R1 Pin)			+0.45	V
High Level Output Voltage	V_{OH1}	I_{OH} = –400uA, Except INT	+2.3			V
	V_{OH2}	I_{OH} = –400uA INT Pin	+2.3			V
		I_{OH} = –50uA INT Pin	+3.4			V
Input Leakage Current	I_{IL}	I_{IN} = 0V ∿ V_{CC}	–10		+10	uA
Output Leakage Current	I_{OL}	I_{OUT} = 0.45V ∿ V_{CC}	–10		+10	uA
Supply Current	I_{CC}				+220	mA

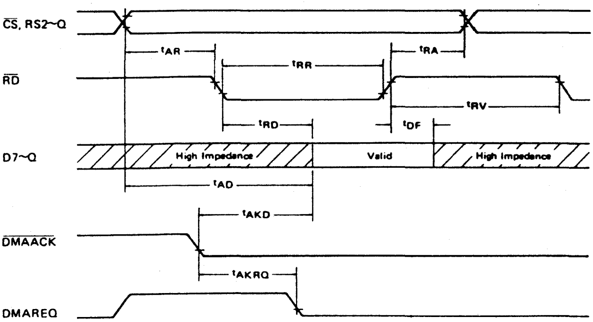
CAPACITANCE

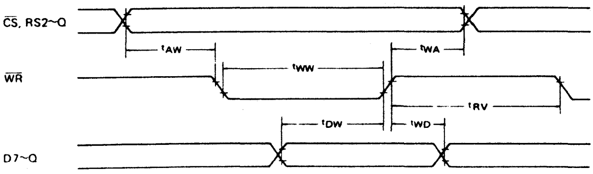
$T_a = 25°C, V_{CC} = GND = 0V$

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Input Capacitance	C_{IN}	f = 1 MHz All Pins Except Pin Under Test Tied to AC Ground			10	pF
Output Capacitance	C_{OUT}				15	pF
I/O Capacitance	$C_{I/O}$				20	pF

T_a=-40°C-+85°C, V_{cc}=+5V±10 %

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS			UNITS
			MIN	TYP	MAX	
Address Setup to $\overline{RD}$	t_{AR}	$RS_{2\text{-}0}$	85			ns
		$\overline{CS}$	+20			
Address Hold from $\overline{RD}$	t_{RA}		0			ns
$\overline{RD}$ Pulse Width	t_{RR}		170			ns
Data Delay from Address	t_{AD}				275	ns
Data Delay from $\overline{RD}$ ↓	t_{RD}				150	ns
Output Float Delay from $\overline{RD}$ ↑	t_{DF}		0		80	ns
$\overline{RD}$ Recovery Time	t_{RV}		250			ns
DMAREQ ↓ Delay from $\overline{DMAACK}$	t_{AKRQ}				160	ns
Data Delay from $\overline{DMAACK}$	t_{AKD}				240	ns
Address Setup to $\overline{WR}$	t_{AW}		0			ns
Address Hold from $\overline{WR}$	t_{WA}		+20			ns
$\overline{WR}$ Pulse Width	t_{WW}		170			ns
Data Setup to $\overline{WR}$	t_{DW}		150			ns
Data Hold from $\overline{WR}$	t_{WD}		+20			ns
$\overline{WR}$ Recovery Time	t_{RV}		250			ns
$\overline{EOI}$ ↓ → $\overline{DIO}$	t_{EODI}	PPSS → PPAS, ATN = True			300	ns
$\overline{EOI}$ ↓ → T/R1 ↑	t_{EOT11}	PPSS → PPAS, ATN = True			202	ns
$\overline{EOI}$ ↑ → T/R1 ↓	t_{EOT12}	PPAS → PPSS, ATN = False			260	ns
$\overline{ATN}$ ↓ → $\overline{NDAC}$ ↓	t_{ATND}	AIDS → ANRS, LIDS			202	ns
$\overline{ATN}$ ↓ → T/R1 ↓	t_{ATT1}	TACS + SPAS → TADS, CIDS			202	ns
$\overline{ATN}$ ↓ → T/R2 ↓	t_{ATT2}	TACS + SPAS → TADS, CIDS			260	ns
$\overline{DAV}$ ↓ → DMAREQ ↑	t_{DVRQ}	ACRS → ACDS, LACS			720	ns
$\overline{DAV}$ ↓ → $\overline{NRFD}$ ↓	t_{DVNR1}	ACRS → ACDS			420	ns
$\overline{DAV}$ ↓ → $\overline{NDAC}$ ↑	t_{DVND1}	ACRS → ACDS → AWNS			780	ns
$\overline{DAV}$ ↑ → $\overline{NDAC}$ ↓	t_{DVND2}	AWNS → ANRS			420	ns
$\overline{DAV}$ ↑ → $\overline{NRFD}$ ↑	t_{DVNR2}	AWNS → ANRS → ACRS			420	ns
$\overline{RD}$ ↓ → $\overline{NRFD}$ ↑	t_{RNR}	ANRS → ACRS LACS, DI reg. selected			600	ns
$\overline{NDAC}$ ↑ → DMAREQ ↑	t_{NDRQ}	STRS → SWNS → SGNS, TACS			480	ns
$\overline{NDAC}$ ↑ → $\overline{DAV}$ ↑	t_{NDDV}	STRS → SWNS → SGNS			420	ns
$\overline{WR}$ ↑ → $\overline{DIO}$	t_{WDI}	SGNS → SDYS, BO reg. selected			300	ns
$\overline{NRFD}$ ↑ → $\overline{DAV}$ ↓	t_{NRDV}	SDYS → STRS, T_1 = True			420	ns
$\overline{WR}$ ↑ → $\overline{DAV}$ ↓	t_{WDV}	SGNS → SDYS → STRS BO reg. selected, RFD = True $N_F = f_c$ = 8 MHz, T_1 (High Speed)			860 +t_{SYNC}	ns
TRIG Pulse Width	t_{TRIG}		45			ns



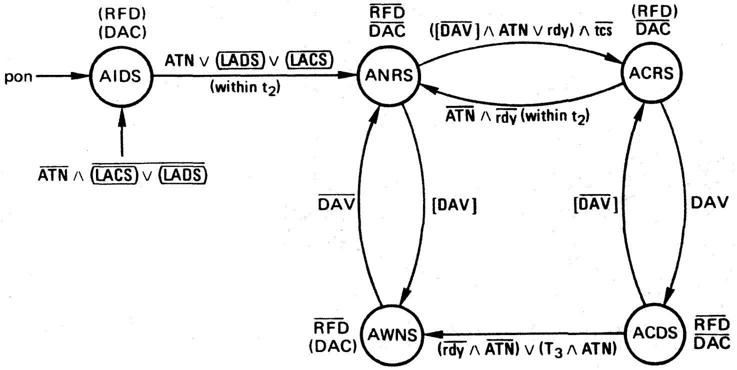


APPENDIX

APPENDIX A

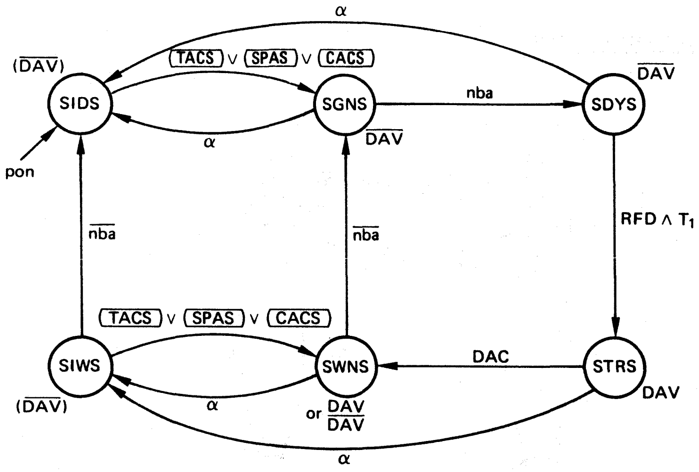
This Chapter contains all Status Diagrams implemented in the uPD7210.

1. ACCEPTOR HANDSHAKE Function (AH)



local messages	
pon rdy tcs	power on ready take control synchronously
external messages	
ATN DAC DAV RFD	Attention Data Accepted Data Valid Ready For Data
AH-States	
ACDS AIDS ANRS ACRS AWNS	Accept Data State Acceptor Idle State Acceptor Not Ready State Acceptor Ready State Acceptor Wait For New Cycle State

2. SOURCE HANDSHAKE Function (SH)



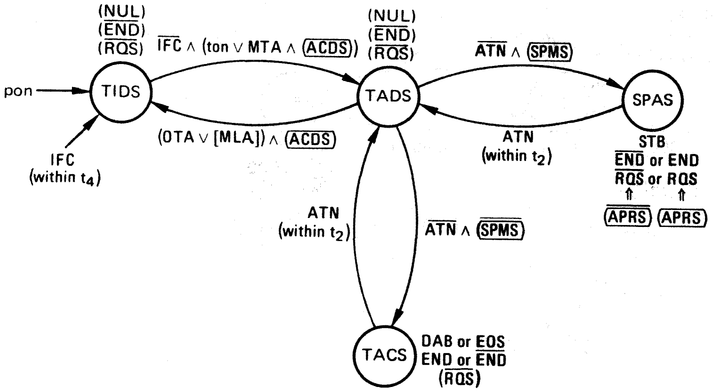
$$\alpha = (\text{ATN} \wedge \overline{\boxed{\text{CACS}} \vee \boxed{\text{CTRS}}}) \vee (\overline{\text{ATN}} \wedge \overline{\boxed{\text{TACS}} \vee \boxed{\text{SPAS}}})$$

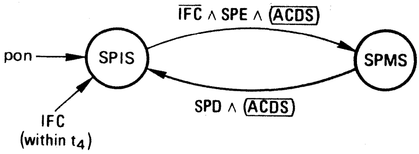
(within t_2)

local messages	
pon	power on
nba	new byte available
external messages	
ATN	Attention
DAC	Data Accepted
DAV	Data Valid
RFD	Ready For Data
SH-States	
SDYS	Source Delay State
SGNS	Source Generator State
SIDS	Source Idle State
SIWS	Source Idle Wait State
STRS	Source Transfer State
SWNS	Source Wait For New Cycle State

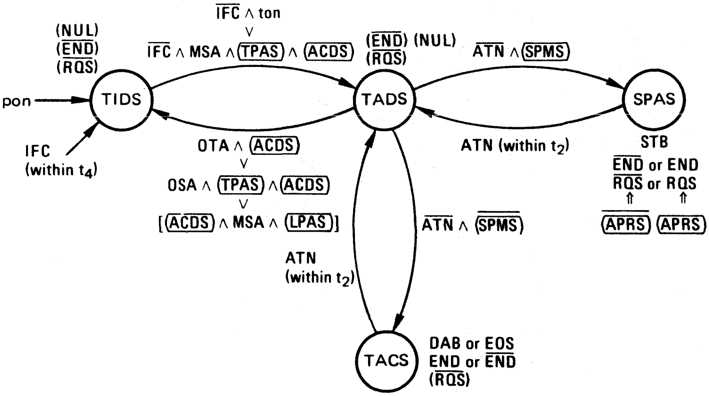
3. TALKER Functions (T,TE)

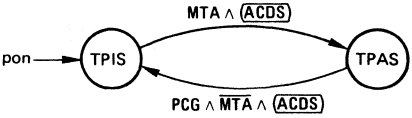
3.1 TALKER Function

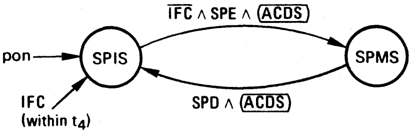




3.2 TALKER EXTENDED Function



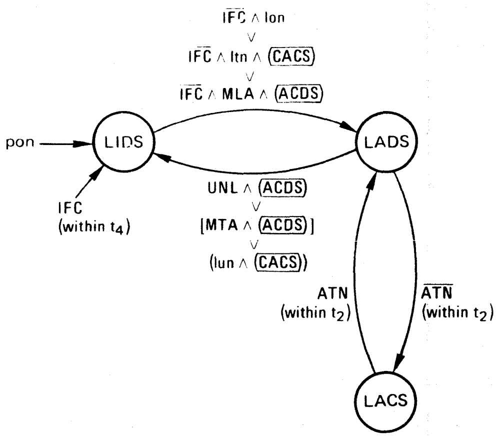




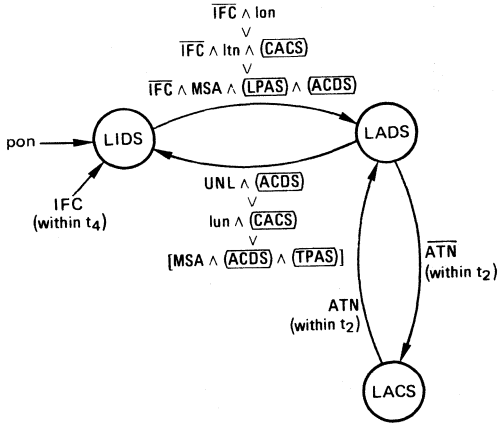
local messages	
pon	power on
ton	talk only
external messages	
ATN	Attention
IFC	Interface Clear
MLA	My Listen Address
MTA	My Talk Address
MSA	My secondary Address
OTA	Other Talk Address
OSA	Other Secondary Address
SPE	Serial Poll Enable
SPD	Serial Poll Disable
RQS	Request Service
PCG	Primary Command Group
DAB	Data Byte
END	End
STB	Status Byte
T,TE-States	
TACS	Talker Active State
TADS	Talker Addressed State
TIDS	Talker Idle State
TPAS	Talker Primary Addressed State
TPIS	Talker Primary Idle State
SPAS	Serial Poll Active State
SPIS	Serial Poll Idle State
SPMS	Serial Poll Mode State

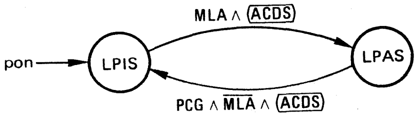
4. LISTENER Functions (L,LE)

4.1 LISTENER Function



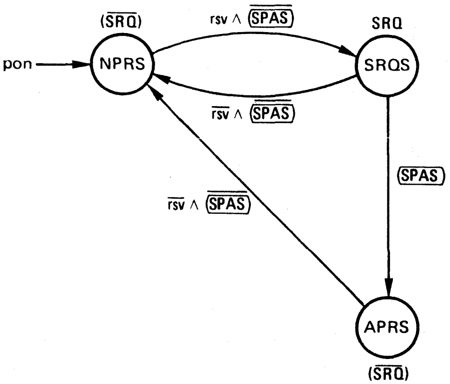
4.2 LISTENER EXTENTED





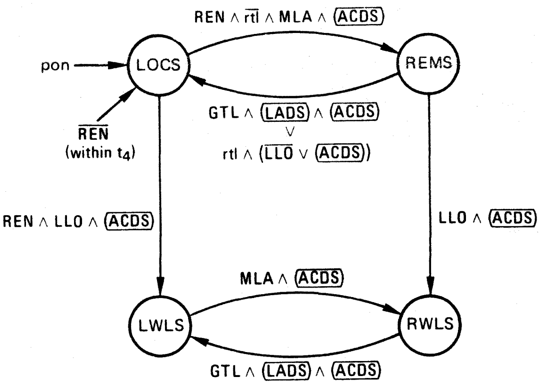
local messages	
pon	power on
lon	listen only
ltn	listen
lun	local unlisten
external messages	
ATN	Attention
IFC	Interface Clear
UNL	Unlisten
MLA	My Listen Address
MTA	My Talk Address
MSA	My Secondary Address
L,LE-States	
LACS	Listener Active State
LADS	Listener Addressed State
LIDS	Listener Idle State
LPAS	Listener Primary Addressed State
LPIS	Listener Primary Idle State

5. SERVICE REQUEST Function (SR)



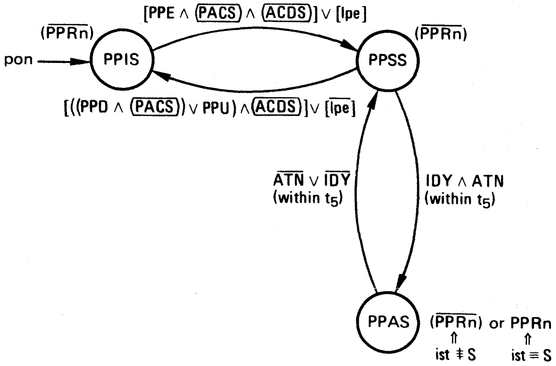
local messages	
pon	power on
rsv	request service
external messages	
SRQ	Service Request
SR-States	
APRS	Affirmative Poll Reponse State
NPRS	Negative Poll Response State
SRQS	Service Request State

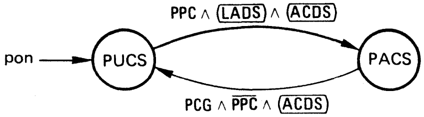
6. REMOTE/LOCAL Function (RL)



local messages	
pon	power on
rtl	return to local
external messages	
REN	Remote Enable
LLO	Local Lockout
GTL	Go To Local
MLA	My Listen Address
MSA	My Secondary Address
RL-States	
LWLS	Local With Lockout State
LOCS	Local State
REMS	Remote State
RWLS	Remote With Lockout State

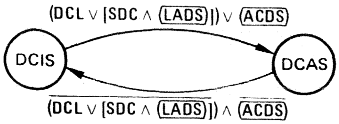
7. PARALLEL POLL Function (PP)





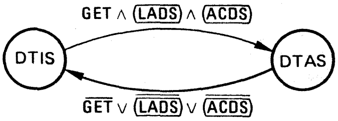
local messages	
pon	power on
ist	individual status
lpe	local poll enable
external messages	
ATN	Attention
IDY	Identify
PPE	Parallel Poll Enable
PPD	Parallel Poll Disable
PPC	Parallel Poll Configure
PPU	Parallel Poll Unconfigure
PCG	Primary Command Group
PPRn	Parallel Poll Response
PP-States	
PPAS	Parallel Poll Active State
PACS	Parallel Poll Addressed To Configure State
PPIS	Parallel Poll Idle State
PPSS	Parallel Poll Standby State
PUCS	Parallel Poll Unaddressed To Configure State

8. DEVICE CLEAR Function (DC)



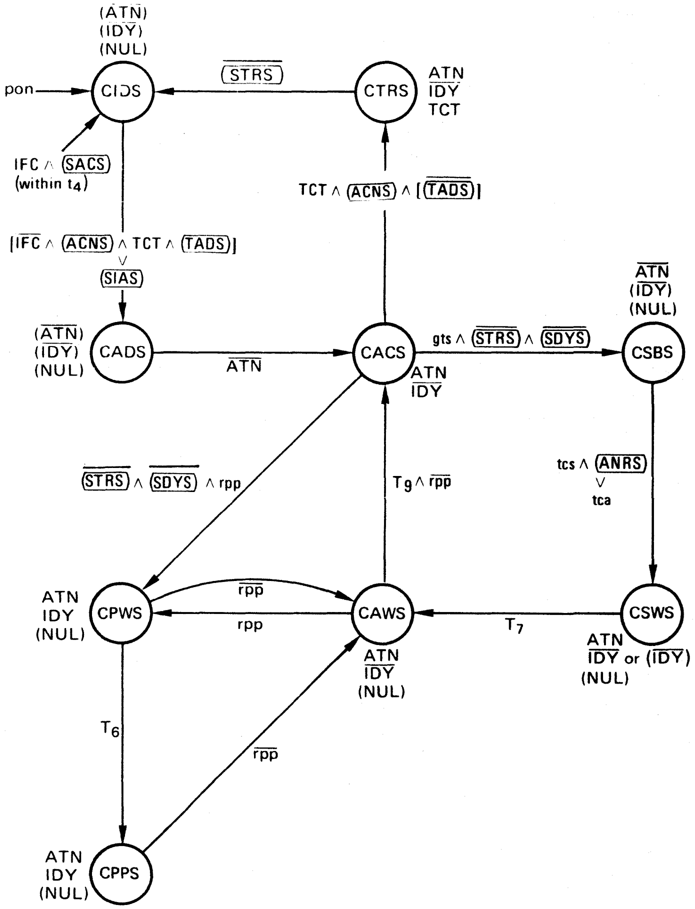
external messages	
DCL	Device Clear
SDC	Selected Device Clear
DC-States	
DCAS	Device Clear Active State
DCIS	Device Clear Idle State

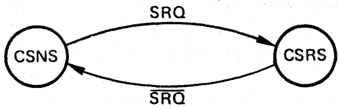
9. DEVICE TRIGGER Function

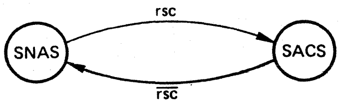


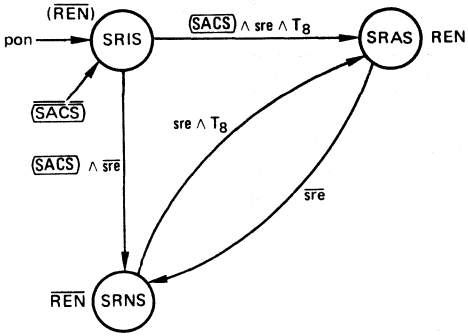
external messages	
GET	Group Execute Trigger
DT-States	
DTIS	Device Trigger Idle State
DTAS	Device Trigger Active State

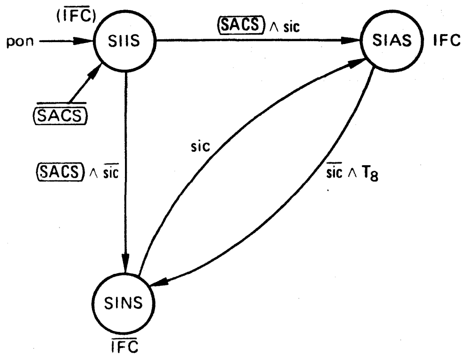
10. CONTROLLER Function (C)




SRQ
CSNS
CSRS
$\overline{SRQ}$


rsc
SNAS
SACS
$\overline{rsc}$


($\overline{REN}$)
pon
SRIS
$\boxed{SACS} \wedge sre \wedge T_8$
SRAS
REN
$\overline{\boxed{SACS}}$
$sre \wedge T_8$
$\boxed{SACS} \wedge \overline{sre}$
$\overline{sre}$
$\overline{REN}$
SRNS


($\overline{IFC}$)
pon
SIIS
$\boxed{SACS} \wedge sic$
SIAS
IFC
$\overline{\boxed{SACS}}$
sic
$\boxed{SACS} \wedge \overline{sic}$
$\overline{sic} \wedge T_8$
SINS
$\overline{IFC}$

local messages	
pon	power on
gts	go to standby
rpp	request parallel poll
rsc	request system control
sic	send interface clear
sre	send remote enable
tca	take control ansynchronously
tcs	take control synchronously
external messages	
ATN	Attention
IFC	Interface Clear
TCT	Take Control
IDY	Identify
REN	Remote Enable
C-States	
CACS	Controller Active State
CAWS	Controller Active Wait State
CADS	Controller Addressed State
CIDS	Controller Idle State
CPPS	Controller Parallel Poll State
CPWS	Controller Parallel Poll Wait State
CSNS	Controller Service Not Requested State
CSRS	Controller Service Requested State
CSBS	Controller Standby State
CSWS	Controller Synchronous Wait State
CTRS	Controller Transfer State
SACS	System Control Active State
SIAS	System Control Interface Clear Active State
SINS	System Control Interface Clear Not Active State
SIIS	System Control Interface Clear Idle State
SNAS	System Control Not Active State
SRAS	System Control Remote Enable Active State
SRIS	System Control Remote Enable Idle State
SRNS	System Control Remote Enable Not Active State

Timing values for State Transitions

$T_1 \geq$ 2us for open collector drivers
$\geq$ 1100ns if DIO,DAV,EOI driven by tri-state drivers
$\geq$ 700ns if DIO,DAV,EOI,ATN driven by tri-state drivers
$\geq$ 500ns for all bytes following ATN=false
$\geq$ 350ns " " " " " ,and short connections
$t_2 \leq$ 200ns
$t_4 <$ 100us
$T_6 \geq$ 2us
$T_7 >$ 500ns
$T_8 >$ 100us
$T_9 \geq$ 1.5us
$\geq$ 600ns for tri-state drivers

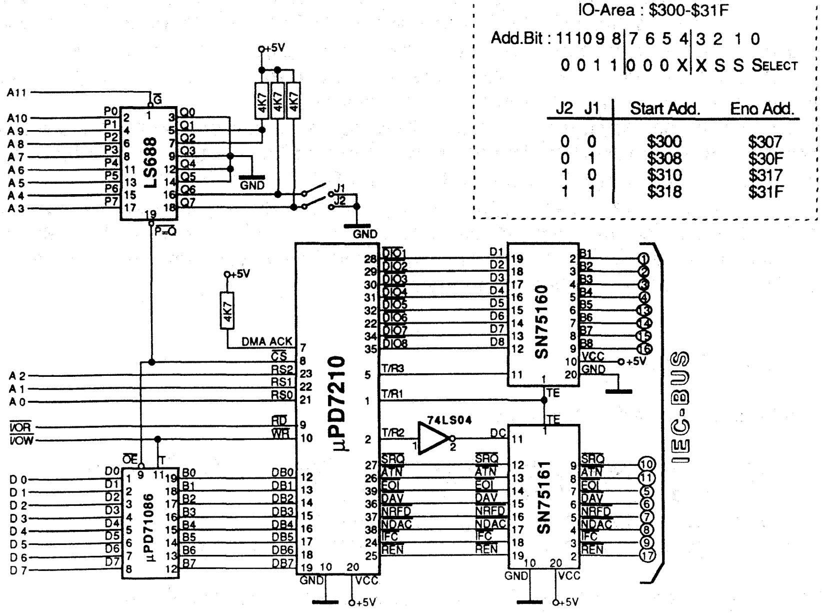
APPLICATION NOTE

Programming the GPIB Controller
μPD7210

1 Introduction

The µPD7210 allows to implement a fully IEEE-488 (1978) compatible general purpose interface bus (GPIB) with only a minimum amount of hardware. In Europe the IEEE-488 specification is adapted to IEC-625 standard with the same electrical features but different cable and connector. The complete schematic on an IEEE-488 controller board is shown below.

µPD7210 Intelligent GPIB Interface Controller for IBM PC



IO-Area : $300-$31F

Add.Bit : 11 10 9 8 | 7 6 5 4 | 3 2 1 0
0 0 1 1 | 0 0 0 X | X S S Select

J2	J1	Start Add.	Eno Add.
0	0	$300	$307
0	1	$308	$30F
1	0	$310	$317
1	1	$318	$31F

On the other hand the GPIB bus is a very powerful bus by means of communication speed and control protocols.This application note provides a set of routines that allows to handle these protocols. All routines are written in µPD8085 assembly language and can be transfered to any other programming language without problems. Each piece of sofware is well documented so the user can easily extract those routines which are needed for his application.

2 Software Considerations

Before any software should be written for a GPIB, the user should decide what IEEE functions he desires, which will define what processor routines are necessary. Second, you should develop the necessary interrupt and software protocols to suit your microcomputer system. Lastly, you should write the necessary software.

At the end of this application note are program listings with description that should help you to understand what functions are used in this example, and the details of how these functions are implemented. The program flow for the routines included are as follows.

The µPD7210 (Talker/Listener/Controller) interface is activated by calling the INIT routine, thus initializing the µPD7210 and the GPIB. The INT1 routine should be jumped to by any interrupt generated because of changing bus status and polling requests. This routine reads the µPD7210 interrupt status and stores it in memory so that your software can check and respond when available.

The INTO routine is used to speed data and polling transfers, and should be vectored to by a different intrerrupt than INT1. The µPD7210 is set up for this type of operation as part of routines for that can use it , such as SEND and RECV.

Now your main program can respond to changing status and initiate desired functions by calling a particular routine. You should note from the routine descriptions that for some routines it is necessary for the main program to initialize its own processor registers with the necessary routine parameters.

3 Description of Routines

The subroutines described below allow you to

- configure your system (i. g. , talkers, listeners, remote, local, IFC, etc)
- transfer data between devices or the controller
- service device requests in a serial or parallel format
- finally, trigger synchronous operation of all devices on the bus.

The NEC µPD7210 can make your GPIB interface easier, more versatile, and more powerful than by using first generation controllers, and in most systems, second generation controller also.

ROUTINE LIST

INIT	Initializaition
INTO	Interrupt 0

INT1	Interrupt 1
SEND	Send Data
RECV	Receive Data
XFER	Transfer Data
TRIG	Device Trigger
CLEAR	Device Clear
EXSP	Execute Serial Polling
EXPP	Execute Parallel Polling
PPC	Parallel Poll Configure
IFC	Interface Clear
REM	Remote
LOC	Local

3.1 Initializaition

The operating mode of the μPD7210 is set in this routine. The first function of the routine is to issue a chip reset, thereby disabling the GPIB interface functions. The next operation is to set the operating modes of the μPD7210, which includes interrupt masks, address mode, device address, EOS code. To finalize this routine, the Immediate Execute pon command is issued followed by the system interface clear command.

3.2 Interrupt 0

This routine is initiated by the DMA REQ signal which requests a data transfer when sending data, receiving data, or when executing a serial polling system command. Before executing these system commands, source or destination address of the first data byte is set to the HL register pair (data pointer) of the CPU. In case of the Send Data Command, the BC register pair is used to hold the number of data bytes to be send (data counter).

The subroutine's flow is as follows:

The TA (Talker Active) bit in the Address Status register is checked first to decide on the direction of the data transfer. If it is cleared (data input mode), GPIB data is available in the Data In register and is transferred to the memory location pointed by HL. The data pointer is then incremented preparing for the next byte.

If the TA bit is set (data output mode), the data byte pointed by HL is transferred to the

Byte Out register, sending the data byte onto the GPIB. The data pointer is then incremented and the data counter (BC) is decremented. If the contents of the data counter (BC) becomes 1, a send EOI command issued, thereby alerting the µPD7210 to send END message with the next byte (Last byte). After the last byte is sent (BC = 0), the DMAO bit of the µPD7210 is reset inhibiting any further data transfer. Finally, a subroutine return is implemented.

3.3 Interrupt 1

This routine is initiated by the INT signal, caused by a SRQI (service request interrupt) or the CO (command output request) bit being set. The interrupt status is read from the Interrupt Status 2 register of the µPD7210 and is kept in memory because it is automatically cleared by the read. The interrupt status byte in memory will be read and cleared, bit by bit, by the routine which uses it.

3.4 Send Data

This routine sends data from the microcomputer system to one or more devices via the GPIB using the HL,BC,and DE register pairs of the CPU as data pointer, data counter and device address pointer, respectively. Each device with its device address listed in an address table pointed by the DE register pair is assigned to receive data (a listener). Then all of the data in the data buffer specified by the HL and BC register pairs is transferred.

This routine assumes that device address of the microcomputer system is not involved in the address table. The last device address in the table is assumed to be followed by a delimiter having a value which is greater than 30.

3.5 Receive Data

This routine is used to input data from a device whose device address is in the B register of the CPU. The system is configured such that the microcomputer is the only listener and receives data until an END or EOS (End of String) message is received. The data is stored in the data buffer pointed by the HL register pair. The talker device is assumed not to be the microcomputer system itself.

3.6 Transfer Data

This routine is used to transfer data from a device whose device address is in the B

register to one or more devices whose device address(es) is (are) listed in the address table pointed by the DE register pair.
The microcomputer system does not receive the data but lets itself be a listener, so it can detect an END or EOS message from talker. On detection of them, data transfer is terminated. The microcomputer system does not participate in the data transfer itself.

3.7 Device Trigger

This routine issues a GET (Group Execute Trigger) message onto the GPIB after addressing devices listed in the address table pointed by the DE register pair. This routine is intended to start operation in the listed devices.

3.8 Device Clear

This routine issues a SDC (Selected Device Clear) message onto the GPIB after addressing devices listed in the address table pointed by the DE register pair . This routine is intended to clear or initialize these devices. If there is no device address listed, a DCL (Device Clear) message is issued instead of a SDC message, thus clearing all devices.

3.9 Execute Serial Polling

This routine executes serial polling according to the sequence of device addresses in the address table pointed by the DE register pair. One byte of status from each device is received and stored in the data buffer pointed by the HL register pair in the same order as in the address table.
The device address of the microcomputer system is assumed not to be involved in the address table .

3.10 Execute Parallel Polling

This routine executes parallel polling and returns one byte of status, Parallel Poll Response, in the A register of the CPU.

3.11 Parallel Poll Configure

This routine configures one or more devices to respond to parallel polling assuming each device can implement the PP1 interface function. The device address are listed in the address table pointed by the DE register pair and the configuration information is stored in a buffer pointed by the HL register pair in the same order as in the adress table.

The least significant five bits of the configuration byte are assumed to have the same format as PPE (Parallel Poll Enable) or the PPD (Parallel Poll Disable) message defined by the IEEE Standard 488.

3.12 Interface Clear

This routine activates the GPIB's IFC line for 100 microseconds, causing the interface functions of all devices to go to known state. This routine is also executed at the end of the initialization routine.

3.13 Local

This routine issues a GTL (Go To Local) message after addressing the devices listed in the address table pointed by the DE register pair. This causes the addressed devices to go to the local state. If there are no device addresses listed in the address table, the GPIB's REN line is inactivated, letting all of the devices go to local state. This routine is intended to enable the devices to receive local data.

3.14 Remote

This routine activates the GPIB's REN line enabling each device to go to remote state when addressed to listen.

Appendix Software Listing

```
;
;*************************************************************************
;*
;* UTILILTY ROULTINES FOR uPD7210 GPIB CONTROLLER
;*
;*************************************************************************
;
; Write Registers
; ===============
;
BO&REG          EQU     00H
INT&M1          EQU     01H
INT&M2          EQU     02H
ADR&MODE        EQU     04H
AUX&MODE        EQU     05H
ADR&REG         EQU     06H
EOS&REG         EQU     07H
;
; Read Registers
; ==============
;
DI&REG          EQU     00H
INT&ST1         EQU     01H
INT&ST2         EQU     02H
ADR&ST          EQU     04H
CPT&REG         EQU     05H
;
; GENERAL EQUATES
; ===============
;
MY&ADR          EQU     04H
EOS&CODE        EQU     0DH
FREQ            EQU     08H
AUX&A           EQU     00H
;
; INITIALISATION
; ==============
;
INIT:   MVI     A,02H           ;CHIP RESET
        OUT     AUX&MODE
        XRA     A
        OUT     INT&M1          ;DISABLE INTERRUPTS OTHER THAN SRQI & CO
        MVI     A, 58H
        OUT     INT&M2          ;ENABLE DMAI,DISABLE DMAO
        MVI     A,31H
        OUT     ADR&MODE        ;SET T/R MODE 3,SET ADDRESS MODE1
        LDA     MY&ADR
        OUT     ADR&REG         ;SET MY ADDRESS 0-30 TO ADDRESS 0 REGISTER
        MVI     A,0E0H
        OUT     ADR&REG         ;DISABLE ADDRESS 1 REGISTER
        MVI     A,EOS&CODE
        OUT     EOS&REG         ;SET EOS CODE
        MVI     A, FREQ OR 20H
        OUT     AUX&MODE        ;SET FREQUENCY OF CLOCK INPUT 0-8 MHz
```

```
        MVI      A, AUX&A OR 80H
        OUT      AUX&MODE          ;SET AUXILIARY MODE A
        MVI      A,0A6H
        OUT      AUX&MODE          ;SET AUX.MODE B
                                   ;(SEND E01 IN SPAS,HIGH SPEED T1)
        XRA      A
        STA      INT&ST1           ;CLEAR WORKING AREA FOR INTERRUPT STATUS
        OUT      AUX&MODE          ;IMMEDIATE EXECUTE PON
        CALL     IFC
        JMP      WAIT&CO           ;RETURN ON ENTERING INTO CACS
;
;INTERRUPT 0
;===========
;
INT0:   PUSH     PSW
        IN       ADR&ST
        ANI      02H               ;TA = 1?
        JNZ      DATA&OUT
        IN       DI&REG            ;DATA IN
        MOV      M,A               ;STORE GPIB DATA
        INX      H                 ;INCREMENT DATA POINTER
RETURN: POP      PSW
        EI
        RET
DATA&OUT: MOV    A,M               ;DATA OUT
        OUT      BO&REG            ;LOAD GIPB DATA
        INX      H                 ;INCREMENT DATA POINTER
        DCX      B                 ;DECREMENT DATA COUNTER
        XRA      A
        ORA      B
        JNZ      RETURN            ;RETURN IF (BC) IS GREATER THAN 2
        INR      A
        CMP      C
        JC       RETURN            ;RETURN IF (BC)IS GREATER THAN 2
        MVI      A,6H              ;(BC) = 0 OR 1
        OUT      AUX&MODE          ;SEND EOI WITH THE NEXT BYTE
        JZ       RETURN            ;RETURN IF (BC) = 1
        MVI      A,58H             ;(BC)= 0
        OUT      INT&M2            ;DISABLE DMAO INTERRUPT
LOOP:   IN       INT&ST1
        ANI      2H                ;DO = 1?
        JZ       LOOP              ;WAIT UNTIL HANDSHAKE IS FINISHED
        MVI      A,11H             ;TCA (TAKE CONTROL ASYNCHRONOUSLY) CODE
        OUT      AUX&MODE          ;ISSUE TCA TO 7210
        POP      PSW
        EI
        RET
;
;INTERRUPT 1
;===========
;
INT1:   PUSH     PSW
        PUSH     H
        IN       INT&ST2           ;READ INTERRUPT STATUS 2 REGISTER
```

```
        LXI      H,INT&ST1
        ORA      M
        MOV      M,A                 ;SET INTERRUPT STATUS BYTE IN MEMORY
        POP      H
        POP      PSW
        EI
        RET
;
;UTILITY SUBROUTINES
;===================
;
WAIT&CO: PUSH    H
        LXI      H,INT&ST1
        MOV      A,M                 ;LOAD INTERRUPT STATUS BYTE
        ANI      8H                  ;CO = 1?
        JZ       WAIT&CO+1           ;WAIT UNTIL CO BIT IS SET
        DI
        XRA      M
        MOV      M,A                 ;CLEAR CO BIT
        POP      H
        EI
        RET
;
;UNLISTEN
;========
;
UNLTN:  MVI      A,3FH               ;UNL (UNLISTEN)CODE
        OUT      BO&REG              ;ISSUE UNL ONTO GPIB
        JMP      WAIT&CO             ;RETURN WHEN HANDSHAKE IS FINISHED
;
;ADDRESS LISTENERS (DEVICE ADDRESS POINTER = DE)
;===============================================
;
ADR&L:  CALL     UNLTN               ;ISSUE UNL ONTO GPIB
LOOP1:  LDAX     D                   ;LOAD DEVICE ADDRESS
        CPI      31                  ;DELIMITER?
        RNC                          ;RETURN IF DELIMITER
        ORI      20H                 ;FROM LISTEN ADDRESS
        OUT      BO&REG              ;ISSUE LISTEN ADDRESS ONTO GPIB
        INX      D                   ;INCREMENT LISTEN ADDRESS POINTER
        CALL     WAIT&CO             ;WAIT UNTIL HANDSHAKE IS FINISHED
        JMP      LOOP1               ;REPEAT
;
;ADDRESS TALKER (TALK ADDRESS = B)
;=================================
;
ADR&T:  MOV      A,B
ADR&T1: ORI      40H                 ;FROM TALK ADDRESS
        OUT      BO&REG              ;ISSUE TALK ADDRESS ONTO GPIB
        JMP      WAIT&CO             ;RETURN WHEN HANDSHAKE IS FINISHED
```

```
;
;SEND DATA (DATA POINTER = HL,DATA COUNTER=BC,DEVICE ADDRESSPOINTER=DE)
;======================================================================
;
SEND:   CALL    ADR&L           ;ADDRESS LISTENERS
        IN      ADR&REG         ;READ MY ADDRESS
        ORI     40H             ;FROM MY TALK ADDRESS
        OUT     BO&REG          ;ISSUE MTA ONTO GPIB
        CALL    WAIT&CO         ;WAIT UNTIL HANDSHAKE IS FINISHED
        MVI     A,78H           ;INTERRUPT MASK 2
        OUT     INT&M2          ;ENABLE DMAO INTERRUPT
        MVI     A,10H           ;GTS (GO TO STANDBY)CODE
        OUT     AUX&MODE        ;ISSUE GTS TO 7210
        JMP     WAIT&CO         ;RETURN ON DATA CYCLE TERMINATION
;
;RECIEVE DATA(DATA POINTER = HL,TALK ADDESS = B)
;==============================================
;
RECV:   CALL    ADR&T           ;ADDRESS TALKER
        CALL    UNLTN           ;ISSUE UNL (UNLISTEN) ONTO GPIB
        MVI     A,13H           ;LTN (LISTEN) CODE
RECV1:  OUT     AUX&MODE        ;ISSUE LTN OR LTNC TO 7210
        MVI     A,1AH           ;TCSE(TAKE CONTROL SYNCHRONOUSLY ON END)COD
RECV2:  OUT     AUX&MODE        ;ISSUE TCSE OR TCS TO 7210
        MVI     A,10H           ;GTS (GO TO STANDBY) CODE
        OUT     AUX&MODE        ;ISSUE GET TO 7210 TO INITIATE DATA CYCLE
        JMP     WAIT&CO         ;RETURN ON DATA CYCLE TERMINATION
;
;TRANSFER DATA (DEVICE ADDRESS POINTER = DE,TALK TALK ADDRESS = B)
;=================================================================
;
XFER:   CALL    ADR&L           ;ADDRESS LISTENERS
        CALL    ADR&T           ;ADDRESS TALKER
        MVI     A,1BH           ;LTNC(LISTEN WITH CONTINUOUS MODE)CODE
        JMP     RECV1           ;ISSUE LTNC TO 7210,START DATA TRANSFER
                                ;AND RETURN ON DATA TRANSFER TERMINATION
;
;DEVICE TRIGGER (DEVICE ADDRESS POINTER = DE)
;===========================================
;
TRIG:   CALL    ADR&L           ;ADDRESS LISTENERS
        MVI     A,8H            ;GET(GROUP EXECUTE TRIGGER) CODE
        OUT     BO&REG          ;ISSUE GET ONTO GPIB
        JMP     WAIT&CO         ;RETURN WHEN HANDSHAKE IS FINISHED
;
;DEVICE CLEAR (DEVICE ADDRESS POINTER = DE)
;=========================================
;
CLEAR:  LDAX    D               ;LOAD DEVICE ADDRESS
        CPI     31              ;NO DEVICE ADDRESS?
        MVI     A,14H           ;DCL(DEVICE CLEAR)CODE
        JC      CLEAR1
        CALL    ADR&L           ;ADDRESS LISTENERS
        MVI     A,4H            ;SDC(SELECTED DEVICE CLEAR)CODE
```

```
CLEAR1: OUT     BO&REG          ;ISSUE DCL OR SDC ONTO GPIB
        JMP     WAIT&CO         ;RETURN WHEN HANDSHAKE IS FINISHED
;
;EXECUTE SERIAL POLLING(DATA POINTER = HL,DEVICE ADDRESS POINTER = DE)
;=====================================================================
;
EXSP:   MVI     A,18H           ;SPE(SERIAL POLL ENABEL)CODE
        OUT     BO&REG          ;ISSUE SPE ONTO GPIB
        CALL    WAIT&CO         ;WAIT UNTIL HANDSHAKE IS FINISHED
        CALL    UNLTN           ;ISSUE UNL (UNLISTEN) ONTO GPIB
        MVI     A,13H           ;LTN (LISTEN) CODE
        OUT     AUX&MODE        ;ISSUE LTN TO 7210
LOOP2:  LDAX    D               ;LOAD DEVICE ADDRESS
        CPI     31              ;DELIMITER?
        JC      CONTN           ;CONTINUE IF NOT DELIMITER
        MVI     A,19H           ;SPD(SERIAL POLL DISABLE)CODE
        OUT     BO&REG          ;ISSUE SPD ONTO GPIB
        JMP     WAIT&CO         ;RETURN WHEN HANDSHAKE IS FINISHED
CONTN:  CALL    ADR&T1          ;ADDRESS TALKER
        MVI     A,12H           ;TCS(TAKE ONTROL SYNCHRONUOUSLY)CODE
        CALL    RECV2           ;ISSUE TCS AND GTS TO 7210
                                ;INITAITING STB TRANSFER
                                ;RETURN ON RECEPTION OF STB
        INX     D               ;INCREMENT TALK ADDRESS POINTER
        JMP     LOOP2           ;REPEAT
;
;EXECUTE PARALLEL POLLING (PARALLEL POLL RESPONSE = A)
;=====================================================
;
EXPP:   MVI     A,1DH           ;EPP(EXECUTE PARALLEL POLLING)CODE
        OUT     AUX&MODE        ;ISSUE EPP TO 7210
        CALL    WAIT&CO         ;WAIT UNTIL LPARALLEL POLLING IS COMPLETED
        IN      CPT&REG         ;READ PPR FROM 7210
        RET
;
;PARALLEL POLL CONFIGURE (DEVICE ADDRESS POINTER=DE
;=======================  SECONDARY COMMAND POINTER = HL)
;
PPC:    LDAX    D               ;LOAD DEVICE ADDRESS
        CPI     31              ;DELIMITRE?
        RNC                     ;RETURN IF DELIMITER
        CALL    UNLTN           ;ISSUE UNL(UNLISTEN)ONTO GPIB
        LDAX    D               ;RELOAD DEVICE ADDRESS
        ORI     20H             ;FROM LISTEN ADDRESS
        OUT     BO&REG          ;ISSUE LISTEN ADDRESS ONTO GPIB
        CALL    WAIT&CO         ;WAIT UNTIL HANDSHAKE IS FINISHED
        MVI     A,5             ;PPC(PARALLEL POLL CONFIGURE)CODE
        OUT     BO&REG          ;ISSUE PPC ONTO GPIB
        CALL    WAIT&CO
        MOV     A,M             ;LOAD SECONDARY COMMAND
        ORI     60H             ;FORM SECONDARY COMMAND
        OUT     BO&REG          ;ISSUE PPE OR PPD ONTO GPIB
        INX     D               ;INCREMENT POINTERS
        INX     H
```

```
        CALL    WAIT&CO
        JMP     PPC             ;REPEAT
;
;INTERFACE CLEAR
;===============
;
IFC:    MVI     A,1EH           ;SIFC(SET IFC) CODE
        OUT     AUX&MODE        ;ISSUE SIFC TO 7210 ACTIVATING IFC GPIB LINE
        CALL    WAIT&100        ;WAIT 100 MICROSECONDS
        MVI     A,16H           ;RIFC(RESET IFC)CODE
        OUT     AUX&MODE        ;ISSUE RIFC TO 7210 INACTIVATING IFC
        RET
;
;LOCAL(DEVICE ADDRESS POINTER = DE)
;==================================
;
LOC:    LDAX    D               ;LOAD ADDRESS
        CPI     31              ;NO ADDRESS?
        JC      LOC1
        MVI     A,17H           ;RREN(RESET REN)CODE
        OUT     AUX&MODE        ;ISSUE RREN TO 7210 INACTIVATING GPIB LINE
        RET
LOC1:   CALL    ADR&L           ;ADDRESS LISTENERS
        MVI     A,1             ;GTL(GO TO LOCAL)CODE
        OUT     BO&REG          ;ISSUE GTL ONTO GPIB
        JMP     WAIT&CO         ;RETURN WHEN HANDSHAKE IS FINISHED
;
;REMOTE
;======
;
REM:    MVI     A,1FH           ;SREN(SET REN)CODE
        OUT     AUX&MODE        ;ISSUE SREN TO 7210 ACTIVATING REN GPIB LINE
        RET
;
WAIT&100: RET                   ; DUMMY DELAY ROUTINE
;
        END
```

PRELIMINARY

μPD72105

LOCAL AREA NETWORK CONTROLLER

INTRODUCTION

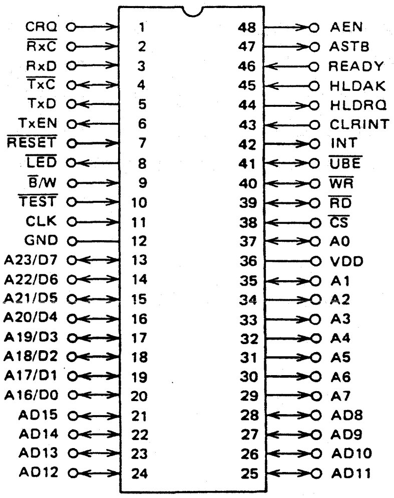
The uPD72105 CSMAC (CSMA Controller) is a network controller that supports the Omninet LAN Protocol developed by Corvus Systems Inc. The on-chip DMA capability of the uPD72105 enables the host system to communicate with other nodes in the network by writing the command bytes and reading results directly to and from memory.

Features:

° Supports Omninet LAN Protocol I and II
° Transfer rates: 0.5 to 4 Mbps
  Omninet LAN Protocol I : 1 Mbps
  Omninet LAN Protocol II: 4 Mbps
° Command chain function
° On-chip DMA controller
° 8/16-bit system bus compatible
° Memory space : 16M bytes (2^{24})
° Transmit FIFO : 12 bytes
  Receive FIFO : 20 bytes
° 16/32-bit CRC
° On-chip DPLL (40 MHz clock input)
° LOOP BACK test capability
° 8-MHz system clock (independent from the serial clock)
° CMOS technology
° 48-pin DIP

Note: The use of this device requires a licensing agreement with Corvus Systems Inc.

Pin Connection (Top View)



CRQ : Command Request

$\overline{\text{RxC}}$: Receive Clock

RxD : Receive Data

$\overline{\text{TxC}}$: Transmit Clock

TxD : Transmit Data

TxEN : Transmit Enable

$\overline{\text{RESET}}$: Reset

$\overline{\text{LED}}$: LED Drive

$\overline{\text{B}}$/W : Byte/Word

$\overline{\text{TEST}}$: Test

CLK : Clock

A23/D7-A16/D0, : Address Bus,
AD15-AD8, Data Bus
A7-A0

AEN : Address Enable

ASTB : Address Strobe

READY : Ready

HLDAK : Hold Acknowledge

HLDRQ : Hold Request

CLRINT : Clear Interrupt Request

INT : Interrupt Request

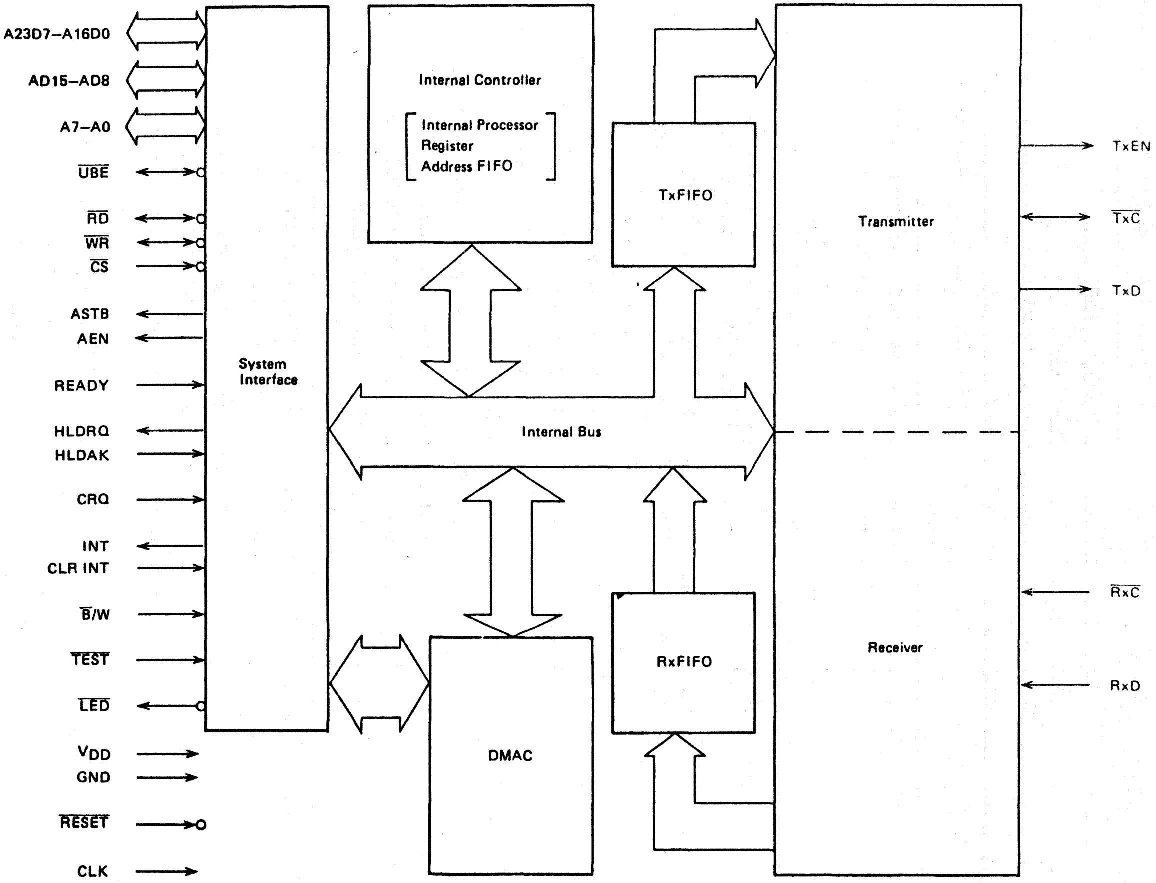
$\overline{\text{UBE}}$: Upper Byte Enable

$\overline{\text{WR}}$: Write

$\overline{\text{RD}}$: Read

$\overline{\text{CS}}$: Chip Select

Internal Block Diagram



1. PIN FUNCTIONS

Pin Name	Pin Number	I/O	Active Level	Function
VDD	36	-	-	+5 power supply pin
GND	12	-	-	Ground potential (0V) pin
CLK	11	I	-	Inputs a single-phase system clock.
$\overline{\text{TEST}}$	10	I	L	Tests the uPD72105 and is normally held high.
$\overline{\text{RESET}}$	7	I	L	Resets the internal circuitry of the chip. The uPD72105 serves as a bus slave after resetting. The minimum pulse width is as 7 clocks.
$\overline{\text{CS}}$	38	I	L	Enables $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals when the chip operates as a bus slave.
$\overline{\text{RD}}$	39	I/O	L	Three-state control signal. It acts as an input pin when the chip operates as a bus slave, allowing an external processor to read the contents of the internal registers of the uPD72105. With the uPD72105 operating as a bus master, this pin acts as an output pin allowing the uPD72105 to read the external memory contents.
$\overline{\text{WR}}$	40	I/O	L	Three-state control signal. It acts as an input when the chip operates as a bus slave, allowing an external processor to write data to the internal registers of the uPD72105. With the uPD72105 operating as a bus master, this pin acts as an output pin allowing the uPD72105 to write data into external memory.

Pin Name	Pin Number	I/O	Active Level	Function
$\overline{B}$/W	9	I	L/H	Specifies a data bus to be accessed by the chip operating as a bus master. $\overline{B}$/W=0 Byte (8-bit) transfers $\overline{B}$/W=1 Word (16-bit) transfers Do not change the state of this pin after power-up. When word access is performed, the lower bits of the data bus access even-numbered addresses
HLDRQ	44	O	H	Outputs the hold request signal to the host processor. The uPD72105, when performing DMA within the chip, activates this signal to request the right to control the system bus to start operation as a bus master.
HLDAK	45	I	H	Inputs the hold acknowledge signal from the host processor. When this signal becomes high, the uPD72105 operates as a bus master and starts DMA.
AEN	48	O	H	Enables a higher address that is latched when the chip operates as a bus master for outputting to the system address bus. This pin is also used to disable the other system bus drivers.
$\overline{UBE}$	41	I/O	L	Three-state control signal. It acts as an input pin and indicates whether valid data are present on the A16/D0-A23/D7 or AD8-AD15 when the chip operates as a bus slave.

<table>
<tr><th>Pin Name</th><th>Pin Number</th><th>I/O</th><th>Active Level</th><th>Function</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th>$\overline{\text{UBE}}$</th><th>A0</th><th>A16/D0-A23/D7</th><th>AD8-AD15</th></tr>
<tr><td>0</td><td>0</td><td>o</td><td>x</td></tr>
<tr><td>0</td><td>1</td><td>x</td><td>o</td></tr>
<tr><td>1</td><td>0</td><td>o</td><td>x</td></tr>
<tr><td>1</td><td>1</td><td>o</td><td>x</td></tr>
</table>
The state of this pin changes depending on the state of the $\overline{\text{B}}$/W pin when the chip operates as a bus master. It is always in a high-impedance state in the byte transfer mode (when $\overline{\text{B}}$/W=0), whereas it acts as an output pin in the word transfer mode (when $\overline{\text{B}}$/W=1), indicating whether Valid data is on A16/D0-A23/D7 or AD8-AD15.
<table>
<tr><th>$\overline{\text{UBE}}$</th><th>A0</th><th>A16/D0-A23/D7</th><th>AD8-AD15</th></tr>
<tr><td>0</td><td>0</td><td>o</td><td>o</td></tr>
<tr><td>0</td><td>1</td><td>x</td><td>o</td></tr>
<tr><td>1</td><td>0</td><td>o</td><td>x</td></tr>
<tr><td>1</td><td>1</td><td>x</td><td>x</td></tr>
</table>
</td></tr>
<tr><td>ASTB</td><td>47</td><td>O</td><td>H</td><td>Strobes the higher address to an external latch.</td></tr>
<tr><td>READY</td><td>46</td><td>I</td><td>H</td><td>Indicates when the memory is in the ready state. This signal is inactivated during the wait cycle in which the width of the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals output by the uPD72105 are extended so that slow-speed memories are accessed.</td></tr>
</table>

Pin Name	Pin Number	I/O	Active Level	Function
CRQ	1	I	H	Signal with which the host system requests the uPD72105 to execute commands. The uPD72105 receives a command at the rising edge of this signal and starts executing it.
INT	42	O	H	Indicates that the uPD72105 has completed command execution or received a packet.
CLRINT	43	I	H	Inactivated the INT signal output by the uPD72105 . The INT signal is inactivated at the rising edge of this signal.
A0,A1	37,35	I/O	-	Bidirectional three-state address lines. These pins act as output pins to indicate the lowest 2 bits of a memory address when the chip operates as a bus master. With the chip operating as a bus slave, they act as input pins to indicate the lowest 2 bits of an I/O address with which an external processor accesses the uPD72105.
A2-A7	34-29	O	-	Three-state address lines. These pins act as output pins to indicate bits 2 to 7 of a memory address when the chip operates as a bus master. They are in a high-impedance state when the chip operates as a bus slave.
AD8-AD15	28-21	I/O	-	Provide a bidirectional three-state address/data bus. The middle bits (8 to 15) of an address are multiplexed with the higher bits (8 to 15) of data.

Pin Name	Pin Number	I/O	Active Level	Function
A16/ D0- A23/ D7	20-13	I/O	-	Provide a bidirectional three-state address/data bus. The higher bits (16 to 23) of an address are multiplexed with the lower bits (0 to 7) of data.
$\overline{\text{LED}}$	8	O	L	Indicates the normal uPD72105 operation by an LED connected externally to this pin. It normally outputs a high signal but becomes low immediately after resetting or when the uPD72105 transmits data to the serial bus.
TxEN	6	O	H	Enables the serial transmitter driver. It becomes high when the uPD72105 transmits data to the serial bus.
$\overline{\text{TxC}}$	4	I/O	-	Outputs the transmit clock generated within the uPD72105 when the DPLL mode is selected; otherwise inputs the transmit clock from an external device.
TxD	5	O	-	Outputs serial transmit data.
$\overline{\text{RxC}}$	2	I	-	Inputs the clock for the DPLL when DPLL mode is selected. If the internal DPLL is not selected, it inputs the receive clock, in which case the clock is generated by an external PLL.
RxD	3	I	-	Inputs serial receive data.

2. INTERNAL CONFIGURATION

The uPD72105 has a transmitter and a receiver that transmits/receives serial data as well as TxFIFO and RxFIFO buffers that respectively hold parallel data for the transmitter and receiver. In addition, it has a system interface to transfer data to and from the the host system, a DMA controller to control data transfer, and an internal controller to control operation of each block.

2.1 Internal Controller

The internal controller interprets the commands sent from the host processor and controls the serial section and DMA controller. It also generated results which are to be transmitted to the host system after command execution has been completed.

2.2 System Interface

This is hardware for interfacing with the host system to process the I/O access and DMA transfer initiated by the host system.

2.3 DMA controller

The DMA controller processes the information (such as commands, data, and results) transferred between the host system and uPD72105 on a memory basis, using DMA. The transfer is performed in units of 4-byte blocks. Each time the transmitter or receiver processes the 4-byte data, the DMA controller issues a HLDRQ signal.

2.4 RxFIFO

This is a 20-byte receive buffer.

2.5 TxFIFO

This is a 12-byte transmit buffer.

2.6 Receiver

The receiver receives packets via the RxD pin and stores them in RxFIFO.

2.7 Transmitter

The transmitter transmits the contents of TxFIFO via the TxD pin.

3. INTERFACING WITH HOST SYSTEM

The host system can access three registers in the uPD72105: the control, status, and address FIFO registers. The control and status registers are used to control the uPD72105 without using the CRQ, INT, and CLRINT pins. The address FIFO is used to provide command addresses to the uPD72105.

Table 3.1 Register Selection (x : Don't Care)

$\overline{CS}$	$\overline{WR}$	$\overline{RD}$	A1	A0	Operation
0	0	1	0	0	Writes to the control register
	1	0			Reads from the status register
	0	1	0	1	Specifies the address FIFO
	0	1	1	1	Writes to the address FIFO
	All others				Use prohibited
1	x				Not selected

3.1 Control Register

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	MIRQ	CIRQ	0	0	CMDRQ

(X: Don't Care)

CMDRQ (Command Request):
This bit has the same functions as the CRQ pin. The host system can issue a command request to the uPD72105 by setting this bit to 1. This bit is automatically reset when the command is executed.

CIRQ (Clear Interrupt Request):
This bit has the same functions as the CLRINT pin. The host system can clear the INT signal of the uPD72105 by setting this bit to 1. When the INT signal is cleared, this bit is automatically reset.

3.2 Status Register

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	IRQ	–	–	WNR	–

(–: Undefined)

WNR (Write Not Ready):
This bit indicates that the uPD72105 is not ready when the host system is preparing to write a command address to the address FIFO. The host system must wait for this bit to become 0 before writing the next byte.

IRQ (Interrupt Request):
This bit indicates the status of the INT pin. The host system can detect an interrupt from the uPD72105 by polling this bit.

3.3 Address FIFO

This is a register that receives the memory address to which the host system has written a command and then issues a command request. The uPD72105 then receives a command from the memory area indicated by the address in the address FIFO and executes it.
When the host system issues a command request without writing an address in the address FIFO, the uPD72105 fetches a command from the memory area indicated by the default address (see NEW DEFLT ADDR command) and executes it.

4. COMMANDS

4.1 Types of Commands

The uPD72105 is provided with the following commands:

Command name	Description
INIT (Initialize)	Initializes the uPD72105.
SEND (Send)	Transmits a message to the other nodes through the network.
SETUP RCV (Setup Receive)	Activates a socket for receive.
END RCV (End Receive)	Inactivates the set-up socket.
WAIT RCV (Wait receive)	Directs a particular socket to await reception for a specific period of time.
RCV LIST (Receive List)	Sets up a list of the receive buffers.
ECHO (Echo)	Transmits an ECHO packet to the other transporter on the network.
LOOP BACK (Loop Back)	Transfers data stored in a particular buffer in the node to an other buffer via the uPD72105.
INIT MONIT (Initialize Monitor)	Starts the network monitor.
MONIT OFF (Monitor OFF)	Suspends transfer during network monitoring.
MONIT ON (Monitor ON)	Resumes suspended transfer during network monitoring.
SET PARM (Set Parameters)	Sets the internal parameters of the uPD72105.
GET PARM (Get Parameters)	Reads the internal parameters of uPD72105.
NEW CHAIN (New Chain)	Specifies a command address to execute the command stored there.
NEW DEFLT ADDR (New Default Address)	Sets the default value of a command address to execute the command stored there.

Command name	Description
CLR STAT (clear Statistics)	Clears the Tx/Rx statistics of the uPD72105.
GET STAT (Get statistics)	Reads the Tx/Rx statistics of the uPD72105.

Note: INIT MONIT, MONIT OFF and MONIT ON commands don't start operation without setting the uPD72105 in a state. The setting way will be informed to the licensees of corvus systems Inc.

4.2 Command Functions

We will now discuss the function of each command in detail. The uPD72105 fetches 12-byte data as a command and processes valid data only. Command codes are even values and when they are changed into odd values by setting to D0 bit to 1, the "command chaining function" of the uPD72105 is enabled. This function allows the uPD72105 to execute a sequence of commands on memory when the host system issues a single command request. When the uPD72105 receives a command whose command code has the D0 bit set to 1, it executes the command then the next one automatically. In this case, however, commands must continuously exist in every 12-byte area in memory. (See Fig. 4-1)

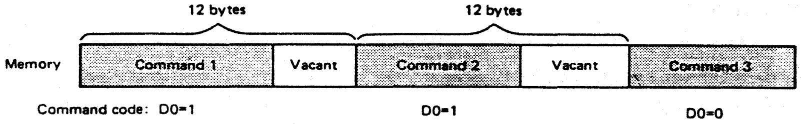
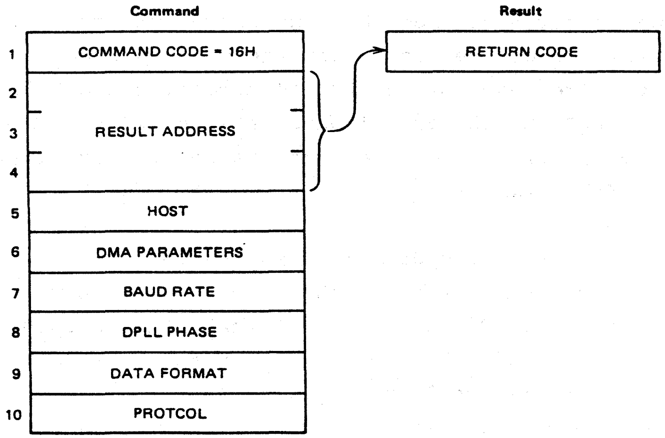


Fig. 4-1 Example of Command Chaining (3 commands)

4.2.1 INIT



The INIT command initializes the uPD72105.

(1) Command

° RESULT ADDRESS

This parameter specifies an address where the uPD72105 stores the result of a command after it has been executed.

° HOST

This parameter specifies a node number and is a value of 00H to 3FH. The uPD72105 checks if the specified number has already been used by another node and indicates the result by RETURN CODE. When HOST is specified to be FFH, the uPD72105 indicates by RETURN CODE the maximum value of up to 3FH of the node number which is not used in the newtork (the minimum value of the node number corresponds to the value of FIXED NODES).

° DMA PARAMETER

This parameter specifies the number of ready cycles for the on-chip DMA controller. One ready cycle corresponds to 1 clock cycle.

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	0	0	READY CYCLES		

° BAUD RATE

This parameter specifies the frequency-dividing ratio for the transmit/receive clock when the on-chip DPLL is used. Where the frequency of the transmit/receive clock is x and BAUD RATE=N, the actual baud rate is x/(N+1). When the on-chip DPLL is not used, the value of this parameter has no effect.

° DPLL PHASE

Bit D7 of this parameter specifies whether the on-chip DPLL is used; when it is 0, the on-chip DPLL is used, whereas when bit D7 is 1, the external DPLL is used. Also, when bit D7 is 0, the lower 7 bits are used to specify an operational phase of the DPLL in relation to the data received. These 7 bits are normally set to 02H. When bit D7 is 1, the value of these bits have no effect.

° DATA FORMAT

When commands or results are transferred between the host system and the uPD72105, this byte specifies which byte of a value (such as an address or data length) comprising 2 or more bytes should be processed first, the lower or higher byte. The default value immediately after resetting is 0; however, the RESULT ADDRESS of the first INIT command itself after reset will be defined by this parameter.

	DATA FORMAT=0	DATA FORMAT≠0
2-byte data	Form lower to higher byte	From higher to lower byte
3-byte data	From lower, middle to higher byte	From higher, middle to lower byte

° PROTOCOL

This parameter specifies the protocol according to which packets are transmitted. When the value of this parameter is 01H, the Omninet I protocol is specified, when it is 02H, the Omninet II protocol is specified. Setting to the Omninet I protocol, 16-bit CRC is selected. Setting to the Omninet II protocol, 32-bit CRC is selected.

° FIXED NODES

This parameter specifies the number of nodes using fixed host addresses. It is valid when HOST=FFH. For example, when HOST=FFH and FIXED NODES=0FH, 0 to 14 are assined to the fixed addresses. The node number is then determined between 15 to 63.

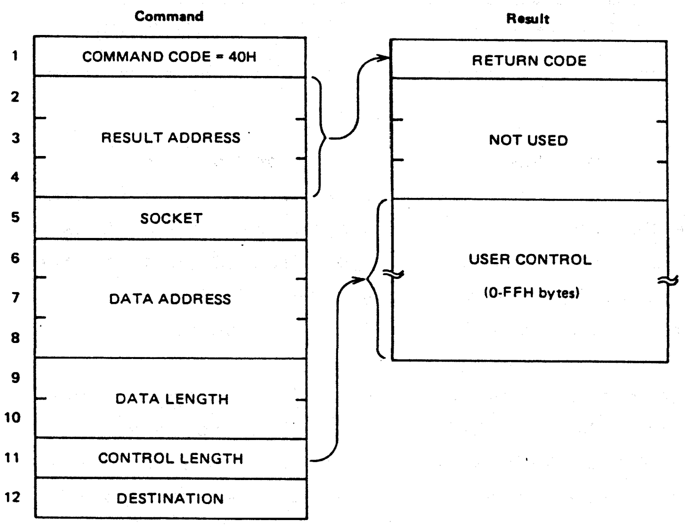
The fixed host address can be used as an address of common node such as a disk server.

(2) Result

° RETURN CODE

00H to 3FH:	When the RETURN CODE parameter has any of these values, it indicates that initialization has been successfully accomplished and a code is assigned as a node number.
86H	: This value indicates that the value of the parameter HOST is invalid (40H to FEH).
89H	: This value indicates that the node number specified by the parameter HOST has already been used by the other nodes.

4.2.2 SEND



The SEND command transmits messages from memory to the other nodes through the network.

(1) Command

° SOCKET

This parameter specifies the socket number of the destination. As a socket number, 80H, 90H, A0H, or B0H can be specified.

- DATA ADDRESS

  This parameter specifies the first address of memory that holds the data to be transmitted.

- DATA LENGTH

  This parameter specifies the number of bytes in the data to be transmitted.

  Omninet I : 0 to 7FFH
  Omninet II: 0 to 800H

- CONTROL LENGTH

  This parameter specifies the number of bytes of user control the host system created in the user message. Where the value of SOCKET is 80H or 90H, the number of bytes is 0.

- DESTINATION

  This parameter specifies the node number of the destination. 00H to 3FH are the regular values, and FFH indicates the broadcast message. All other transporters receive a message addressed DESTINATION=FFH, but do not return acknowledgement.

(2) Result

- RETURN CODE

00H	Indicates that transmission has been successfully accomplished without retransmission.
01H to 0AH	Indicates the number of retransmission attempts made before successful transmission was accomplished.
80H	Indicates that retransmission was attempted the maximum allowable number of times but no response was obtained.
81H	Indicates that the transfer data length was longer than the buffer length provided in the destination.
82H	Indicates that the socket has not been set up at the destination.
83H	Indicates that the value of CONTROL LENGTH specified did not agree with the value at the destination.

84H : Indicates that the parameter SOCKET was invalid, i.e., the value of the parameter SOCKET was other than 80H, 90H, A0H, and B0H).

86H : Indicates that the parameter DESTINATION was invalid, i.e., the value of the parameter was other than 00H to 3FH and FFH).

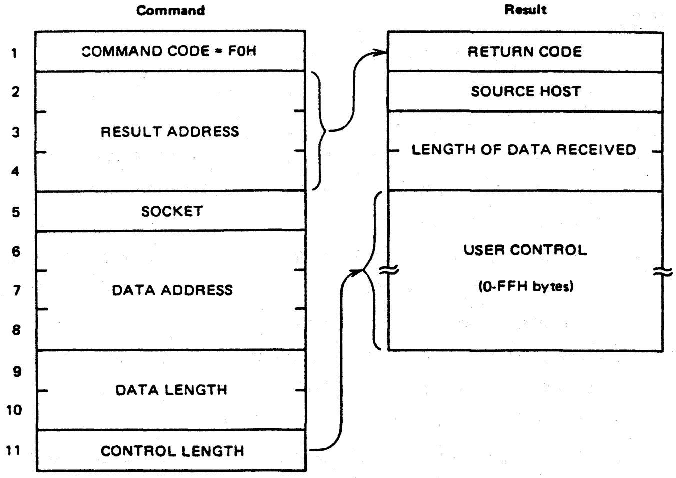
87H : Indicates that the "no empty message" buffer is available for the socket at the destination.

8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

° USER CONTROL

The user control, which the host system provides in packet information, should be written in addresses higher than (RESULT ADDRESS+4) prior to transmission.

4.2.3 SETUP RCV



The SETUP RCV command readies a socket to receive message packets.

(1) Command

° SOCKET

This parameter specifies the socket number that enables reception.

° DATA ADDRESS

This parameter specifies the first address of a buffer in memory in which the rceived data are to be stored.

° DATA LENGTH

This parameter specifies the number of bytes in the receive buffer to be provided in the addresses higher than DATA ADDRESS:

Omninet I : 0 to 7FFH
Omninet II: 0 to 800H

° CONTROL LENGTH

This parameter specifies the number of bytes in the user control to be received. When the value of SOCKET is 80H or 90H, it is 00H.

(2) Result

° RETURN CODE

FEH : Indicates that the receive socket has been successfuly set up.

84H : Indicates that the parameter SOCKET was invalid.

85H : Indicates that the socket specified by the parameter SOCKET is being used.

8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

00H : Indicates that the data has been successfuly received.

The SETUP RCV command first sets a value to RETURN CODE immediately after the command has been issued to indicate whether the socket has been set up. If the socket has been successfully set up, the command sets the value 00H to RETURN ODE when a message packet is received later. The uPD72105 generates an interrupt whenever it sets a value to RETURN CODE.

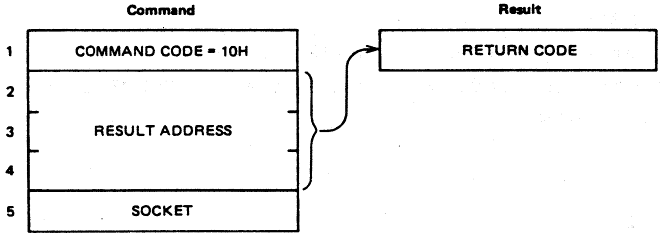
- SOURCE HOST

  When a message packet is received (RETURN CODE=00H), this parameter is set as a result along with LENGTH OF DATA RECEIVED and USER CONTROL to indicate the number of the node that transmitted the packet.
- LENGTH OF DATA RECEIVED

  This parameter indicates the number of bytes in the user data of the received packet.
- USER CONTROL

  This area stores the user control of the received packet.

4.2.4 END RCV



The END RCV command causes the socket that has been set up by the SETUP RCV or RCV LIST command to return to the inactive state.

(1) Command

- SOCKET

  Specifies the number of the socket that is to be inactivated.

(2) Result

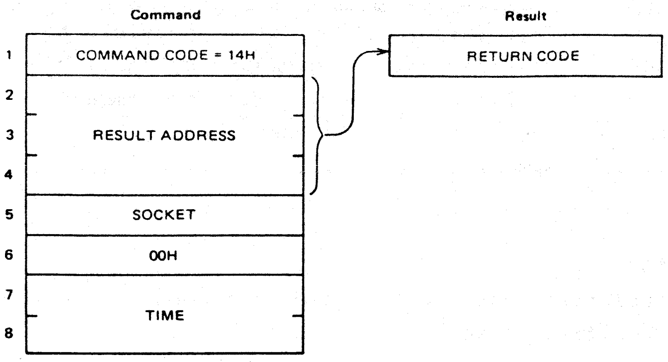
- RETURN CODE

  84H : Indicates that the parameter SOCKET was invalid.

  00H : Indicates that the socket has been inactivated.

  8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

4.2.5 WAIT RCV



The WAIT RCV command specifies the time during which the socket set up by the SETUP RCV command is to await reception of a message packet.

(1) Command

- SOCKET

  This parameter specifies the number of the socket for which the wait time for receiving a packet is specified.

- TIME

  This parameter specifies to socket a wait time for receiving a packet.

TIME	Wait time [ms] (system clock: 8 MHz)
01	1x100
02	2x100
03	3x100
:	:
FF	25x100

Where the value of TIME is 00H, the uPD72105 enters the busy wait state and processes no command until it receives a message packet. With this function, it is possible to suspend execution of a sequence of commands by specifying the command chaining function with a series of commands following the WAIT RCV command, and executing the WAIT RCV command with TIME specified to be 00H.

(2) Result

° RETURN CODE

00H	: Indicates that the wait time has been set to the specified socket.
84H	: Indicates that the parameter SOCKET was invalid.
88H	: Indicates that the specified socket has not been set up.
8AH	: Indicates that the command was not accepted, because the chip has been in the network monitor mode.
92H	: Indicates that the RCV LIST command is being executed on the specified socket. The WAIT RCV command cannot be executed on a socket that has been set up by the RCV LIST command.

The subsequent RETURN CODE is set to the address specified by RESULT ADDRESS of the SETUP RCV command that has set up the socket.

00H	: Indicates that a packet has been received within the time specified by TIME.
90H	: Indicates that a packet has not been received within the time specified by TIME (i.e., time-out). In this case, the specified socket becomes inactive.

4.2.6 RCV LIST

Command

1	COMMAND CODE = 12H
2	MESSAGE LIST ADDRESS
3	
4	
5	SOCKET
6	NOT USED
7	BUFFER LENGTH
8	

The RCV LIST command sets up a list of message buffers (message descriptor) to allow reception by sockets 80H and 90H. Only socket numbers 80H and 90H are supported by this command. The list provided by the host system when this command is executed is shown in Fig. 4-2. The message descriptor consists of a message list indicating the state of the buffer in which the received message is to be stored and control records which manage the buffer.

(1) Command

- ° MESSAGE LIST ADDRESS
  This parameter specifies the first address of the message descriptor.
- ° SOCKET
  This parameter specifies the number of the socket that is set up to receive packets.
- ° BUFFER LENGTH
  This parameter specifies the number of bytes in each message buffer.

(2) Control Record

- ° SOCKET STATE
  Immediately after a command has been executed, RETURN CODE is first set. Afterward, this parameter indicates the presence of a buffer storing unprocessed received messages.

- RETURN CODE

FEH	: Indicates that the receive list has been set up and that reception is enabled.
84H	: Indicates that parameter SOCKET is invalid (i.e., the value of the parameter is other than 80H and 90H).
85H	: Indicates that the specified socket is being used.
8AH	: Indicates that the command was not accepted because the chip has been in the network monitor mode.

- Buffer state

00H	: Indicates that the received message has been stored after the last interrupt was generated.
FDH	: Indicates that the received messages are stored in all the buffers.
FCH	: Indicates that the host system has completed processing of all the messages.

° FIRST FREE AND LAST FREE

These parameters indicate the first and last indexes of FREE LIST, respectively. The "FREE LIST" is a message list of an empty message buffer.

° FIRST RECEIVED and LAST RECEIVED

These parameters indicate the first and last indexes of RECEIVED LIST, respectively. When the value of LAST RECEIVED is 00H, it indicates that no message has been received yet. When the value of FIRST RECEIVED equals that of FIRST FREE, RECEIVED LIST does not exist. "RECEIVED LIST" is a message list of the buffers storing unprocessed received messages.

(3) Message List

° NEXT

This parameter indicates the index of the next message list. When its value is 00H, it indicates the last list.

° DATA BUFFER ADDRESS

This parameter indicates the first address of the message buffer each message list supports.

° RECORD CODE

FFH : Indicates that the message buffer is empty.

00H : Indicates that the received message is stored in the message buffer.

The host system rewrites FFH to RECORD CODE when a command is issued or when the received message in the message buffer is processed.

° SOURCE

This parameter indicates the number of the node that transmitted the received packet. This node number is not set by the host system during command execution. It is written by the uPD72105 after the packet has been received.

° ACTUAL LENGTH

This parameter indicates the number of bytes in a received message. It is set by the uPD72105 after the packet has been received.

When the RCV LIST command is issued, the first message list is set up. The uPD72105 then removes the set up of a message list one by one each time receiving a packet, and stores the received message to a buffer according to the list. An interrupt is requested each time a packet is received.

On receiving a packet, the uPD72105 stores a message in the buffer of the message list indicated by FIRST FREE, sets the RECORD CODE in the message list to 00H, and also sets the SOURCE and ACTUAL LENGTH CODE to specific values. It then sets FIRST FREE to an index of the next message list and LAST RECEIVED to that of the current message list. The uPD72105 also sets the SOCKET STATE to 00H and generates an interrupt. Figs. 4-3 and 4-4 show updating on the message descriptor when the first and i-th packets are received, respectively.

If a message is stored in the last buffer (NEXT=00H) of FREE LIST, the uPD72105 sets the SOCKET STATE to FDH and FIRST FREE to 00H.

The host system, on the other hand, checks the SOCKET STATE on acknowledging an interrupt and processes received messages if it is either 00H or FDH. The host system first checks that FIRST RECEIVED does equal FIRST FREE and that FIRST RECEIVED does not equal 00H (RECEIVED LIST is present). It then fetches received message from the buffer indicated by FIRST RECEIVED.

After that, the host system rewrites FIRST RECEIVED to the NEXT value in the current message list and sets the NEXT to 00H and the RECORD CODE to FFH. It then checks FIRST FREE further. If FIRST FREE does not equal 00H (FREE LIST is present), the host system sets NEXT and LAST FREE of the message list, indicated by LAST FREE to the index of the current message list. If FIRST FREE equals 00H (FREE LIST is not present), the host system sets FIRST FREE and LAST FREE to the index of the current message list. These procedures mean to add an empty buffer to the end of the FREE LIST. Fig. 4-5 shows updating on the list when the first message is processed after the i-th packet has been received.

The host system continues these processes until RECEIVED LIST runs out (FIRST RECEIVED does not equal FIRST FREE) and sets the SOCKET STATE to FCH.

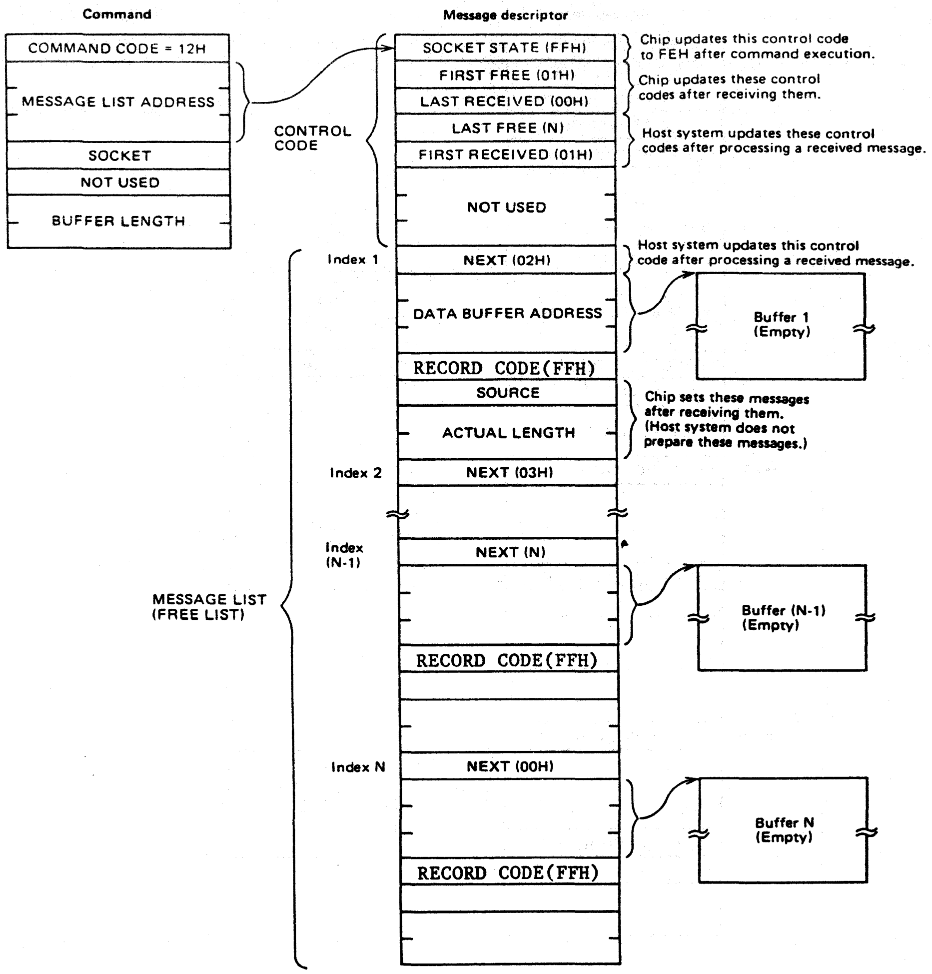


Fig. 4-2 RCV LIST Command List (created by the host system)

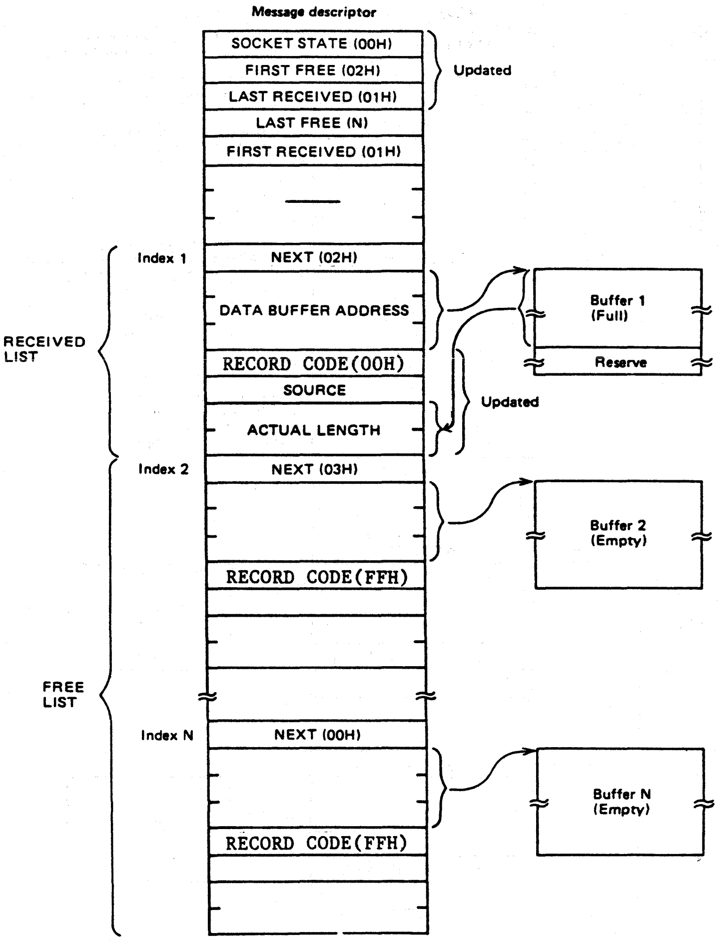


Fig. 4-3 Updated List on Receiving the First Packet
(by the chip)

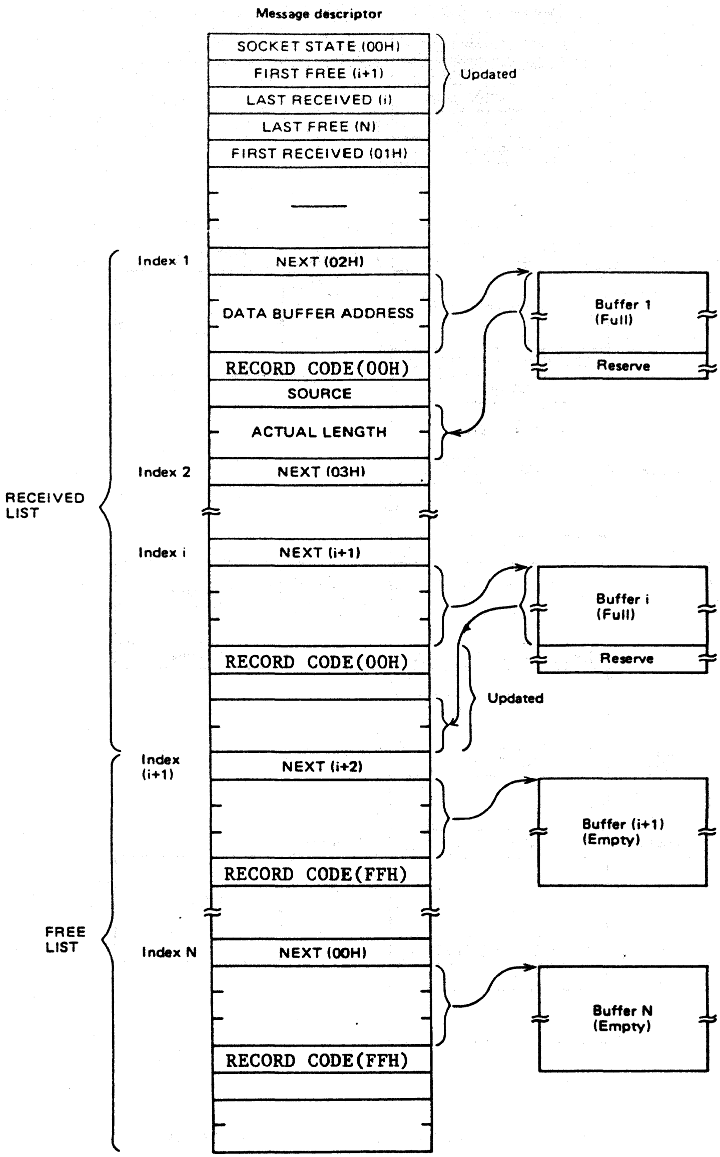
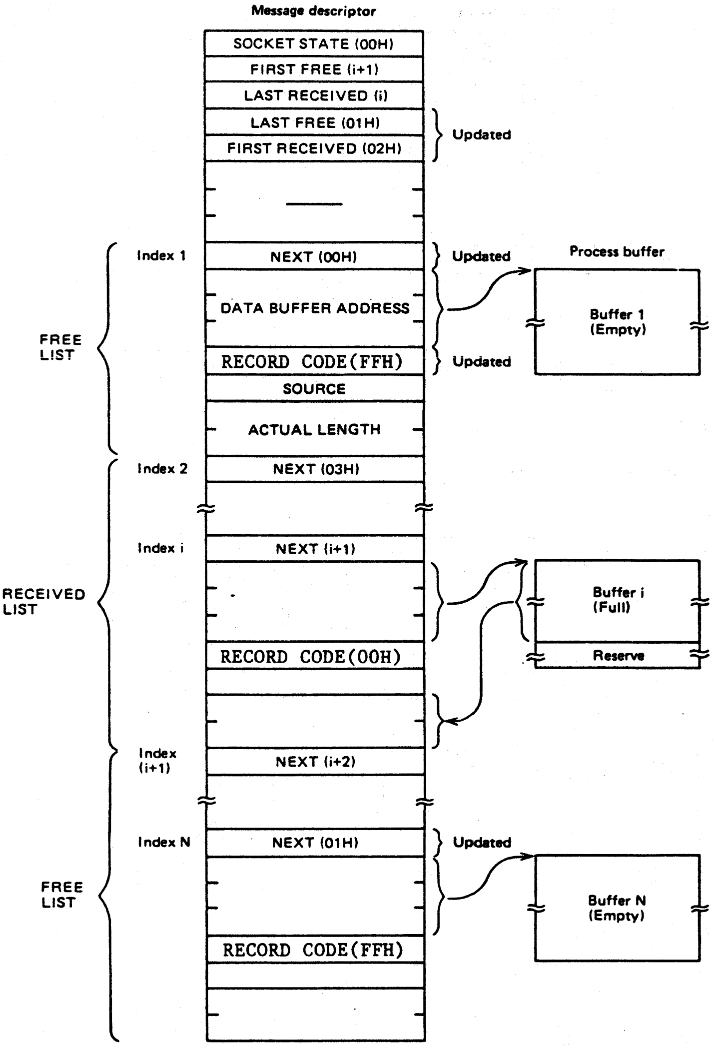
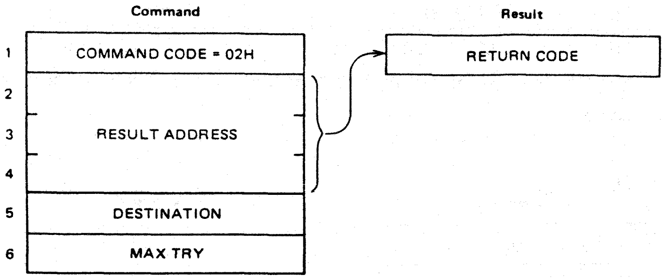


Fig. 4-4 Updated List on Receiving the i-th Packet (by the chip)



4-5 Updated list on Processing the First Message
(by the host system)

4.2.7 ECHO



The ECHO command transmits the ECHO packet to the other nodes on the network. With this command, the host system can learn whether the destination node of the ECHO packet exists on the network.

(1) Command

- DESTINATION
  This parameter specifies the node number of the destination.
- MAX TRY
  This parameter specifies the maximum allowable number of retransmission attempts (0 to 127). It is generally specified as a small value such as 01H or 02H.

(2) Result

- RETURN CODE

00H	: Indicates that the presence of the destination node on the network was acknowledged by transmitting the first ECHO packet (i.e., acknowledgment packet was received).
01H to 7FH	: Indicates the number of retransmission attempts made before the acknowledgment packet was received.
80H	: Indicates that the acknowledgment packet was not received after retransmission attempts had been made the number of times specified by MAX TRY.

86H : Indicates that the parameter DESTINATION was invalid (over 40H).

8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

4.2.8 LOOP BACK

	Command
1	COMMAND CODE = 1AH
2	DESTINATION DATA ADDRESS
3	
4	
5	CRC SIZE
6	SOURCE DATA ADDRESS
7	
8	
9	LENGTH
10	

No result

The LOOP BACK command transfers the data held by a buffer in memory via the uPD72105 to another buffer in the node. The host system may check the data for changes during transfer by comparing the contents of both the buffers.

(1) Command

- CRC SIZE

  This parameter specifies the number of bits in the CRC code. When its value is 02H, the CRC code consists of 16 bits and when it is 04H, 32 bits.
- SOURCE DATA ADDRESS

  This parameter specifies the first address of the buffer containing the data to be transferred.
- DATA LENGTH

  This parameter specifies the number of bytes of data to be transferred.

(2) Result

- RETURN CODE

  00H : Indicates that the loop back function completed normally.

  8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

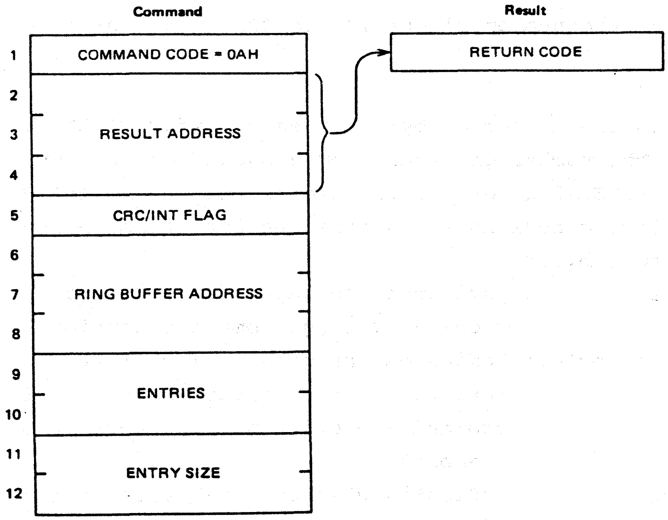
- LOOP BACK DATA

  These bytes are the data transferred by the loop back function.

- CRC

  These bytes are the CRC code generated with transferred data.

4.2.9 INIT MONIT



The INIT MONIT command starts network monitoring to receive all the packets on the network (including those transmitted to the other nodes), and then transfers a packet information and an additional information of each packet to the specified ring buffer. Note that the received packets also remain on the network, not influencing the normal receive operation.

(1) Command

- CRC/INT FLAG

  Bit D0 (CRC FLAG):

  When this bit is 1, the packet information for a packet that caused a CRC error is also transferred. When it is 0, however, the packet information is not transferred.

  Bit D1 (INT FLAG):

  When this bit is 1, an interrupt is generated each time a packet information and an additional information are transferred to the buffer at the last entry in the ring buffer.

- RING BUFFER ADDRESS

  Specifies the first address of the ring buffer in which the packet information and additional information are to be stored.

- ENTRIES

  Specifies the number of entries (buffers) in the ring buffer.

- ENTRY SIZE

  Specifies the number of bytes in each buffer of the ring buffer. The number of bytes from a packet defined by [(ENTRY SIZE)-5 bytes] is transferred to the buffer with the following additional information contained in the remaining 5 bytes:

1st byte	: Indicates the sequence of the packet received during command execution.
2nd to 4th bytes	: Indicates the count value of the internal counter after the start of command execution until reception of the packet.
5th byte	: Indicates whether a CRC error occurred. The value 00H indicates that an error has not occurred, whereas any other value indicates occurrence of an error.

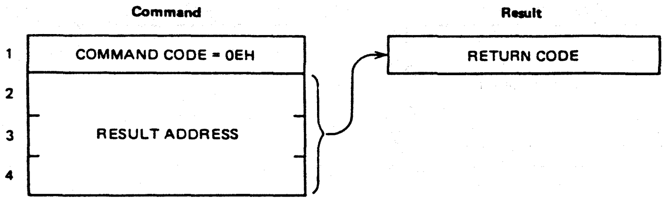
(2) Result

° RETURN CODE

00H : Indicates that the command has been accepted.

When the uPD72105 accepts an INIT MONIT command, it becomes unable to perform ordinary communications, and can only accept the commands INIT MONIT, MONIT OFF, MONIT ON, and INIT. To release the network monitor mode, the INIT command must be executed.

4.2.10 MONIT OFF



The MONIT OFF command suspends DMA transfer to the ring buffer while the uPD72105 is monitoring the network.

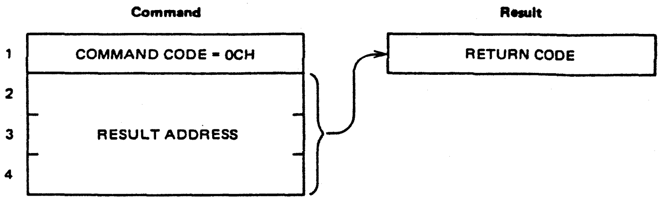
(1) Result

° RETURN CODE

00H : Indicates that the command has been accepted.

8AH : Indicates that the command was not accepted, because the chip has not been in the network monitor mode.

4.2.11 MONIT ON



The MONIT ON command resumes DMA transfer to the ring buffer suspended by the uPD72105 while it is monitoring the network.

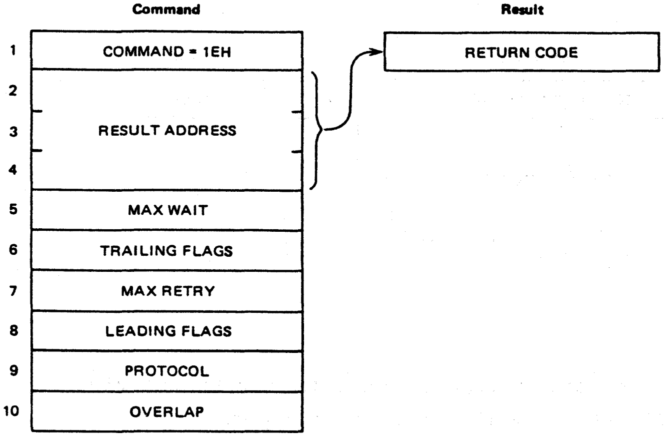
(1) Result

- RETURN CODE

  00H : Indicates that the command has been accepted.

  8AH : Indicates that the command was not accepted, because the chip has not been in the network monitor mode.

4.2.12 SET PARM



The SET PARM command modifies the internal parameters of the uPD72105.

(1) Command

° MAX WAIT

This parameter specifies the maximum random wait time before transmission.

The following values are acceptable, and the default value is 0FH.

MAX WAIT	Maximum wait time (us)
01H	5
03	15
07	35
0F	75
1F	155
3F	315
7F	635
FF	1275

° TRAILING FLAGS

The parameter specifies the number of bytes in the TRAILING FLAG of a packet. The default value is 08H.

° MAX RETRY

This parameter specifies the maximum allowable number of retransmission attempts. The default value is 0AH.

° LEADING FLAGS

This parameter specifies the number of bytes in the LEADING FLAG of a packet. The default value is 0AH.

° PROTOCOL

This parameter has the same functions as the parameter PROTOCOL of the INIT command.

° OVERLAP

This parameter specifies the number of bits that overlap timewise between the TRAILING FLAG of the message packet being received and LEADING FLAG of the acknowledge packet to be transmitted in response to the packet being received.

° TRYINT

This parameter specifies the minimum value of the retransmission intervals.

TRYINT	Minimum interval (us)
00H	Use prohibited
01	3
02	6
⋮	⋮
FE	762
FF	765

° HOST

The parameter HOST of this command has no meaning.
The HOST which was determined by INIT command can not be changed with this parameter.

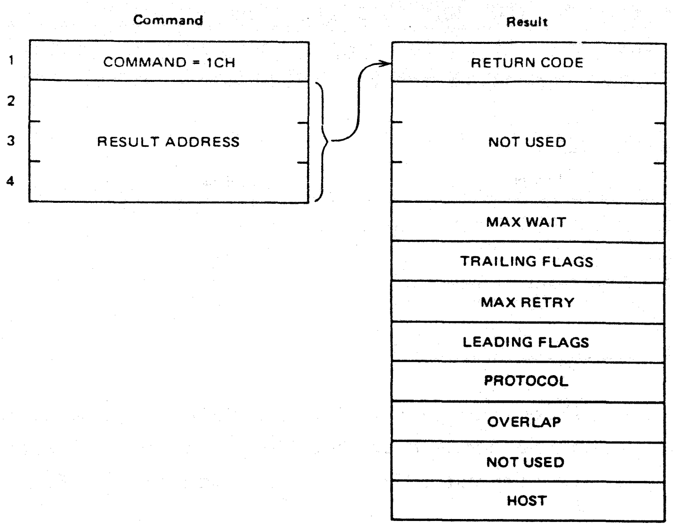
(2) Result

° RETURN CODE

00H : Indicates that the command has been accepted.

8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

4.2.13 GET PARM



The GET PARM command reads the internal parameters of the uPD72105. To partially modify the internal parameters, the parameters of the SET PARM command can be created by first executing the GET PARAM command and then modifying the result.

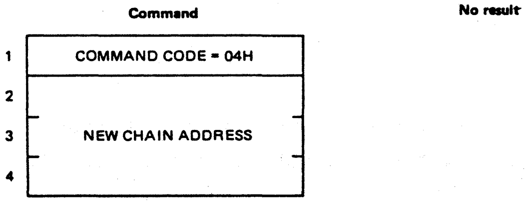
(1) Result

° RETURN CODE

00H : Indicates that the command has been accepted.

8AH : Indicates that the command was not accepted, because the chip'has been in the network monitor mode.

4.2.14 NEW CHAIN



The NEW CHAIN command executes the command stored in the address specified by NEW CHAIN ADDRESS. The uPD72105 does not generate an interrupt for this command, but does so after execution of the command in the specified address.

4.2.15 NEW DEFLT ADDR

	Command
1	COMMAND CODE = 06H
2	NEW DEFAULT COMMAND ADDRESS
3	
4	

No result

The NEW DEFLT ADDR command sets a default value of the command address from which the command is fetched when the address FIFO is empty. This command address is 0 immediately after resetting.
On receiving the command, the uPD72105 executes the command specified by NEW DEFAULT COMMAND ADDRESS. The uPD72105 does not generate an interrupt due to this command, but does after execution of the command in the specified address.

4.2.16 CLR STAT

	Command
1	COMMAND CODE=26H
2	RESULT ADDRESS
3	
4	

Result
RETURN CODE

The CLR STAT command clears the contents of event counters to initialize the counters. The counters manage the transmit/receive statistics.

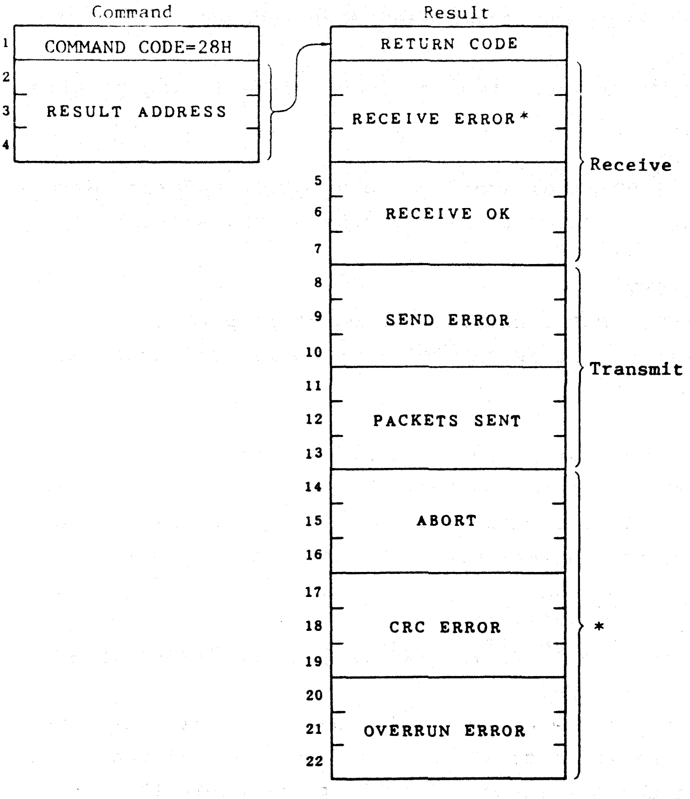
(1) Result

- RETURN CODE

  00H : Indicates that the command has been accepted.

  8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

4.2.17 GET STAT



The GET STAT command reads the contents of event counters to store them in a memory area following to the RETURN CODE.

(1) Result

- RETURN CODE

  00H : Indicates that the command has been accepted.

  8AH : Indicates that the command was not accepted, because the chip has been in the network monitor mode.

° RECEIVE ERROR

The RECEIVE ERROR indicates the count of errors which have occurred on receiving packets. The value is the total of ABORT, CRC ERROR and OVERRUN ERROR.

° RECEIVE OK

The RECEIVE OK indicates the count of recieving packets normally.

° SEND ERROR

The SEND ERROR indicates the count of Tx underrun errors which have occurred to send aborts on transmitting packets.

° PACKETS SENT

The PACKETS SENT indicates the count of packet transmissions. It is the total of normal and abnormal transmissions

° ABORT

The ABORT indicates the count of receiving aborts.

° CRC ERROR

The CRC ERROR indicates the count of CRC errors which have occurred on receiving packets. The polynominals of CRC are as follows.

16 bits: $X^{16}+X^{12}+X^{5}+1$

32 bits: $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}$
$+X^{2}+X+1$

The initial value of CRC is 1. The calculation is done for the packet information. The result of calculation is inverted the bit status, 1 or 0, to transmit the MSB first on transmitting. The result of calculation is compared with $^{LSB}F0B8^{MSB}$ (for 16-bit) or $^{LSB}DEBB20E3^{MSB}$ (for 32-bit) on receiving, and the receiving is regarded as CRC error.

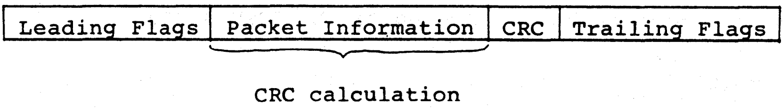


Fig. 4-6 Frame Format

° OVERRUN ERROR

The OVERRUN ERROR indicates the count of Rx overrun errors which have occurred on receiving packets.

5. CONTROL

The host system needs to have additional data in memory for some commands to issue them to the uPD72105. These data include: a message descriptor and buffer for the RCV LIST command, a buffer for the SETUP RCV command, and transmit data for the SEND command.

A control flowchart of the uPD72105 is shown in Fig. 5-1. Prior to issuing a command request to the uPD72105, the host system constructs a command table in memory and writes FFH as the RETURN CODE in the area of the table indicated by RESULT ADDRESS. Then it writes the first address of the memory on which the command table was created to the address FIFO of the uPD72105.

The command address writing procedure is shown in Fig. 5-2. In writing data to the address FIFO, the host system always checks whether the WNR(D1) bit of the status register is set to 1. Setting the command address is completed by writing 3-byte data.

If a command request is issued without setting the address FIFO, the uPD72105 fetches a command from the default address (see NEW DEFLT ADDR COMMAND) and executes it.

After setting the command address, the host system issues a command request (the CRQ signal is "H" or the CMDRQ(D0) bit in the control register is 1) and then waits for a command completion interrupt to be generated by the uPD72105. When the host system detects the command completion interrupt (the INT signal is "H" or the IRQ(D4) bit of the status register is 1), the host system resets this interrupt (CLRINT signal is "H" or CIRQ(D3) bit of the control register is 1).

In this procedure, the host system completes controlling the uPD72105 and then reads the RETURN CODE stored in RESULT ADDRESS.

However, when a command is to be executed while setting-up a receive socket, an interrupt of command execution completion and that of packet receive may simultaneously be generated. In this case, the host system fails to detect the packet receive

interrupt because the receive interrupt has been also reset after servicing the command completion interrupt. Therefore, the host system may service only the command completion interrupt, and fails to service the receive interrupt.
To avoid this, when a command is executed during the setting-up of the receice socket, the process can be controlled as follows: when the uPD72105 generates an interrupt, the RETURN CODE of the received command is first checked and if it is other than FFH or FEH, the receive completion interrupt is serviced. Subsequently, the service routine for the command completion interrupt is executed without waiting for the interrupt to occur.
By doing so, depending on the timing of an interrupt, the host system may start servicing the next interrupt immediately before it is generated. This results in no causes of interrupt remaining to be serviced even when an interrupt has been detected. In such cases, he host system must ignore the interrupt.

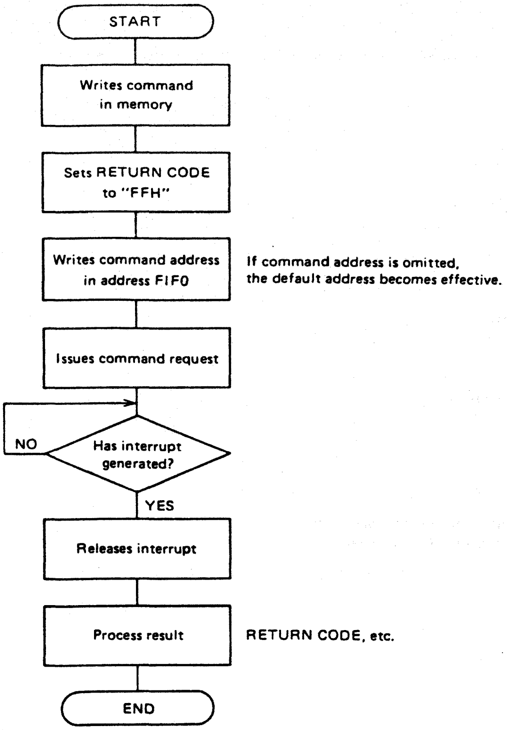


Fig. 5-1 CSMAC Control Flowchart

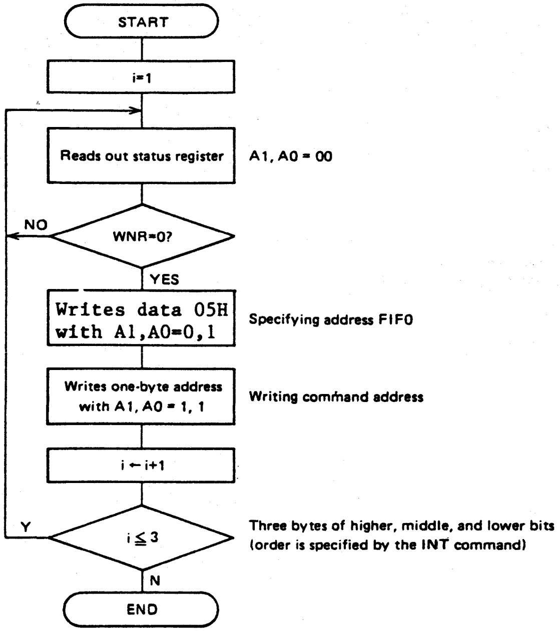


Fig. 5-2 Command Address Writing

6. SYSTEM CONFIGURATION EXAMPLES

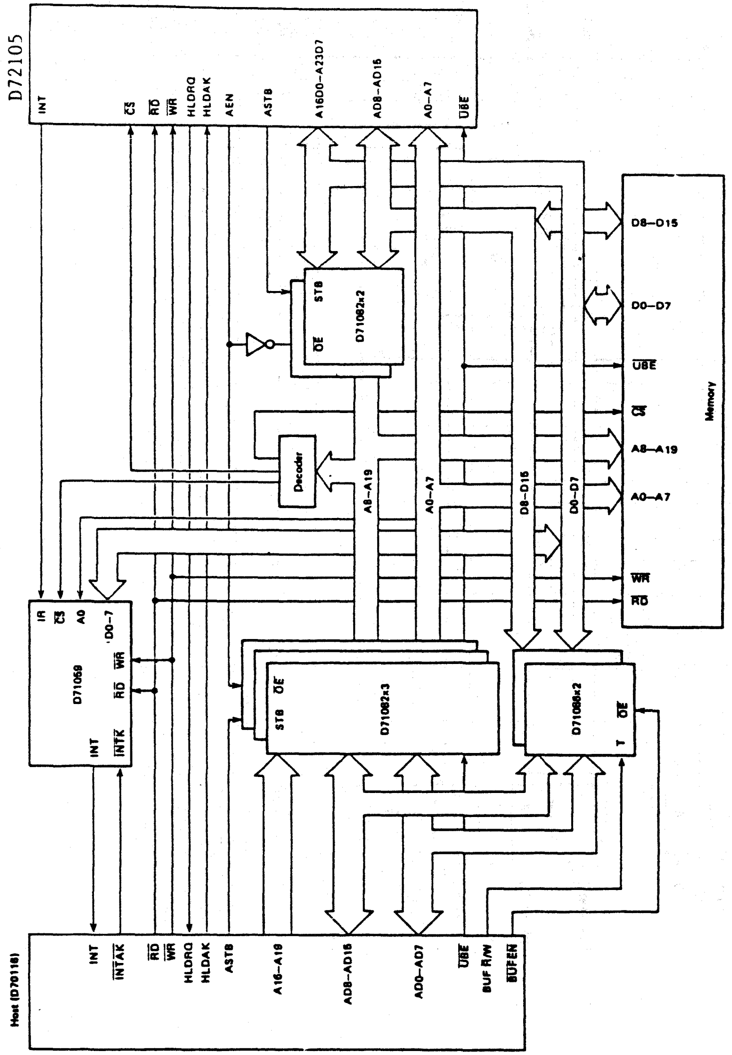


Fig. 6-1 System Configuration Example I (Memory Mapped I/O)

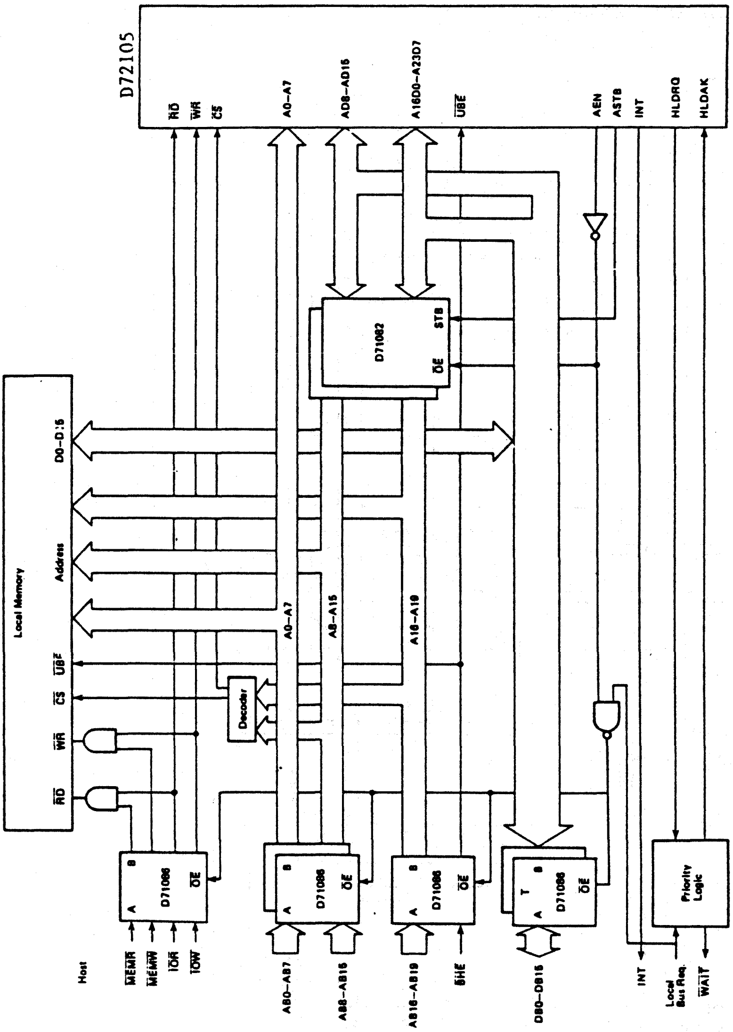


Fig. 6-2 System Configuration Example II

Fig. 6.3 Example for serial interface

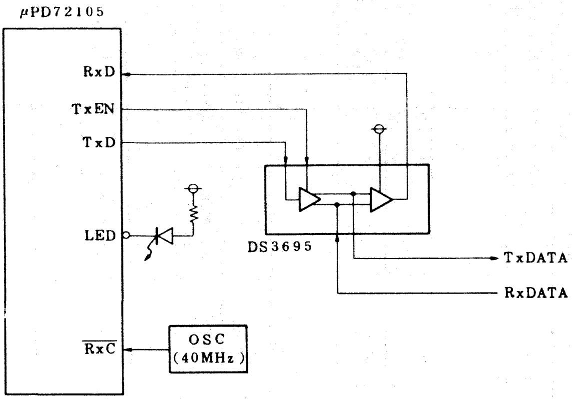
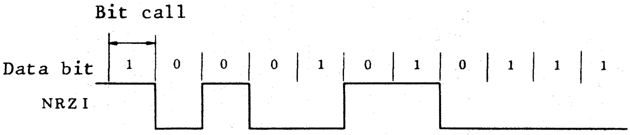


Fig. 6.4 NRZI Data Format



Packet Format

Message Packet

Packet Information

User Message

I	Leading Flags	Message Header	User Control ×(0-255)	User Data ×(0-2047)	CRC ×2	Trailing Flags
II	Leading Flags	Message Header	User Control ×(0-255)	User Data ×(0-2048)	CRC ×4	Trailing Flags

I	Destination Address	Source Address	Validation "A5"	Destination Socket	Retry	Parity	User Data Length	User Control Length
II	Destination Address	Source Address	Validation "5A"	Destination Socket	Retry	Parity	User Data Length	User Control Length

Echo Packet

I	Leading Flags	Destination Address	Source Address	Validation "A5"	Socket "00"	(no use) ×5	CRC ×2	Trailing Flags
II	Leading Flags	Destination Address	Source Address	Validation "5A"	Socket "00"	Retry	CRC ×4	Trailing Flags

Acknowledgement Packet

I	Leading Flags	Destination Address "50"	ACK/NAK Code	Validation "A5"	(no use)	CRC ×2	Trailing Flags	
II	Leading Flags	Destination Address	Source Address	Validation "5A"	Socket "01"	ACK/NAK Code	CRC ×4	Trailing Flags

Sync Packet

I	Leading Flags	Destination Address "60"	Source Address	Validation "A5"	(no use) ×6	Retry	CRC ×2	Trailing Flags
II	Leading Flags	Destination Address	Source Address	Validation "5A"	Socket "02"	Retry	CRC ×4	Trailing Flags

I = OMNINET I protocol, II = OMNINET II protocol

μPD72105

LOCAL AREA NETWORK CONTROLLER

TARGET SPECS

uPD72105 AC/DC Target spec.

ABSOLUTE MAXIMUM RATINGS (Ta=25°C)

Parameter	Symbol	Test Conditions	Ratings	Units
Power Supply Voltage	VDD		-0.5 ~ +7.0	V
Input Voltage	VI		-0.5 ~ VDD+0.3	V
Output Voltage	VO		-0.5 ~ VDD+0.3	V
Operating Temperature	Topt		-40 ~ +85	°C
Storage Temperature	Tstg		-40 ~ +125	°C

DC CHARACTERISTICS (Ta=-40 +85°C, VDD=5V±10%)

Parameter	Symbol	Test Conditions	MIN	TYP	MAX	Units
Input Low Voltage	VILC	CLK Pin	-0.5		+0.8	V
	VILR	$\overline{\text{RXC}}$ when using on-chip DPLL	-0.5		+0.4	V
	VIL	Others	-0.5		+0.8	V
Input High Voltage	VIHC	CLK Pin, $\overline{\text{TEST}}$ Pin	+3.3		VDD+0.3	V
	VIHR	$\overline{\text{RXC}}$ when using on-chip DPLL	+2.4		VDD+0.3	V
	VIH	Others	+2.2		VDD+0.3	V
Output Voltage Low	VOL	IOL=2.5mA			+0.4	V
Output Voltage High	VOH	IOH=-400uA	0.7*VDD			V
Supply Current	IDD1	When operating *1		25	40	mA
Input Leakage Current	ILI	0V≦VIN≦VDD			± 10	uA
Output Leakage Current	ILO	0V≦VOUT≦VDD			± 10	uA

*1 system clock = 8MHz
DPLL clock = 40MHz

Fix B/W pin to "1" or "0".

uPD72105

AC CHARACTERISTIC (Ta=-40 +85°C, VDD=5V±10%) *2

BUS-MASTER MODE

Parameter	Symbol	Test Conditions	MIN	MAX	Units
CLK Cycle Time	TCYK		125	1000	nS
CLK Active Low	TKKL		50		nS
CLK Active High	TKKH		50		nS
CLK Rise Time	TKR	1.5-3.0V		10	nS
CLK Fall Time	TKF	3.0-1.5V		10	nS
HLDRQ↑ Delay Time from CLK↓	TDHQH			100	nS
HLDRQ↓ Delay Time from CLK↑	TDHQL			100	nS
HLDAK Setup Time to CLK↑	TSHA		35		nS
AEN↑ Delay Time from CLK↓	TDAEH			100	nS
AEN↓ Delay Time from CLK↑	TDAEL			100	nS
ASTB↑ Delay Time from CLK↑	TDSTH			70	nS
ASTB Pulse Width	TSTSTH		TKKH-15		nS
ASTB↓ Delay Time from CLK↓	TDSTL			100	nS
ADR/$\overline{\text{UBE}}$/$\overline{\text{RD}}$/$\overline{\text{WR}}$ Delay Time from CLK↑	TDA			100	nS
ADR/$\overline{\text{UBE}}$/$\overline{\text{RD}}$/$\overline{\text{WR}}$ Float Delay from CLK↑	TFA			70	nS
ADR Setup Time to ASTB↓	TSAST		TKKH-35		nS
ADR Hold Time from ASTB↓	THSTA		TKKL-20		nS
$\overline{\text{RD}}$↓ Delay Time from ADR Float	TDAR		TKKH-30		nS
$\overline{\text{RD}}$↓ Delay Time from CLK↓	TDRL			70	nS
$\overline{\text{RD}}$ Pulse Width	TRRL2		1.5TCYK -50		nS
$\overline{\text{RD}}$↑ Delay Time from CLK↑	TDRH			70	nS
DATA Setup Time to $\overline{\text{RD}}$↑	TSDR		70		nS
DATA Hold Time from $\overline{\text{RD}}$↑	THRD		0		nS
$\overline{\text{WR}}$↓ Delay Time from CLK↓	TDWL			70	nS
$\overline{\text{WR}}$ Pulse Width	TWWL2		1.5TCYK -50		nS
$\overline{\text{WR}}$↑ Delay Time from CLK↑	TDWH			70	nS
READY Setup Time to CLK↑	TSRY		35		nS
READY Hold Time from CLK↑	THRY		20		nS

uPD72105

Parameter	Symbol	Test Conditions	MIN	MAX	Units
CLRINT Pulse Width	TCLCLH		100		nS
INT↑ Delay Time from CLK↑	TDIH			100	nS
INT↓ Delay Time from CLRINT↑	TDIL			100	nS
$\overline{LED}$↓ Delay Time from CLK↑	TDLL			100	nS
$\overline{LED}$↑ Delay Time from CLK↑	TDLH			100	nS
CRQ Pulse Width	TCRCRH		100		nS

BUS-SLAVE MODE

Parameter	Symbol	Test Conditions	MIN	MAX	Units
$\overline{WR}$ Pulse Width	TWWL		100		nS
$\overline{CS}$ Hold Time from $\overline{WR}$↑	THWCS		0		nS
ADR/$\overline{UBE}$/$\overline{CS}$ Setup Time to $\overline{WR}$↓	TSAW		0		nS
ADR/$\overline{UBE}$ Hold Time from $\overline{WR}$↑	THWA		0		nS
DATA Setup Time to $\overline{WR}$↑	TSDW		100		nS
DATA Hold Time from $\overline{WR}$↑	THWD		20		nS
$\overline{RD}$ Pulse Width	TRRL		150		nS
ADR/$\overline{CS}$ Setup Time to $\overline{RD}$↓	TSAR		35		nS
ADR/$\overline{CS}$ Hold Time from $\overline{RD}$↑	THRA		0		nS
DATA Delay Time from $\overline{RD}$↓	TDRD			120	nS
DATA Float Delay from $\overline{RD}$↑	TFRD		10	100	nS
$\overline{RESET}$ Pulse Width	TRSTL		7TCYK		nS
VDD Setup Time to $\overline{RESET}$↑	TSVDD		1000		nS
1st $\overline{WR}$/$\overline{RD}$ from $\overline{RESET}$↑	TSYWR		2TCYK		nS
Recovery Time from $\overline{WR}$/$\overline{RD}$	TRVWR		200		nS
High Setup Time to HLDAK↑	TSWR		-20		nS
High Hold Time from AEN↓	THWR		100		nS

* 2 Input levels for AC test are
2.4V (as "1") and
0.4V (as "0").
Test points are
2.2V (as "1") and
0.8V (as "0").

uPD72105

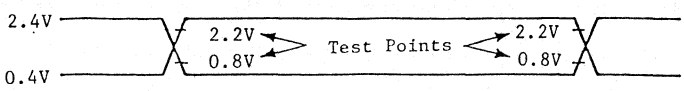
Serial Part

Parameter	Symbol	Test Conditions	MIN	MAX	Units
$\overline{\text{TXC}}$/$\overline{\text{RXC}}$ Cycle Time	TCYS		250	2000	nS
$\overline{\text{TXC}}$/$\overline{\text{RXC}}$ Active Low	TSSL		110		nS
$\overline{\text{TXC}}$/$\overline{\text{RXC}}$ Active High	TSSH		110		nS
$\overline{\text{TXC}}$/$\overline{\text{RXC}}$ Rise Time	TSR			20	nS
$\overline{\text{TXC}}$/$\overline{\text{RXC}}$ Fall Time	TSF			12	nS
TXD Delay Time from $\overline{\text{TXC}}$↓	TDTXD			100	nS
TXEN↑ Delay Time from $\overline{\text{TXC}}$↓	TDTEH			100	nS
TXEN↓ Delay Time from $\overline{\text{TXC}}$↓	TDTEL			100	nS
RXD Setup Time to $\overline{\text{RXC}}$↑	TSRXD		50		nS
RXD Hold Time from $\overline{\text{RXC}}$↑	THRXD		70		nS
$\overline{\text{RXC}}$ Cycle Time *3	TCYH	When using on-chip DPLL	25	200	nS
$\overline{\text{RXC}}$ Active Low	THHL	When using on-chip DPLL	5		nS
$\overline{\text{RXC}}$ Active High	THHH	When using on-chip DPLL	5		nS
$\overline{\text{RXC}}$ Rise Time	THR	When using on-chip DPLL		5	nS
$\overline{\text{RXC}}$ Fall Time	THF	When using on-chip DPLL		5	nS
TX, RX DATA Cycle Time	TCYD	When using on-chip DPLL	250	2000	nS
RXD Setup Timing	TRX	When using on-chip DPLL		+0.2TCYD -0.2TCYD	nS
$\overline{\text{TXC}}$ Active Low	TTTL	When using on-chip DPLL	0.5TCYD-25		nS
$\overline{\text{TXC}}$ Active High	TTTH	When using on-chip DPLL	0.5TCYD-25		nS
TXD Change Delay from $\overline{\text{TXC}}$↓	TDTX1	When using on-chip DPLL		50	nS
TXD Change Delay from $\overline{\text{TXC}}$↑	TDTX2	When using on-chip DPLL	0.5TCYD-50		nS

* 3 DPLLed clock cycle time should be from 250 ns to 2000ns.

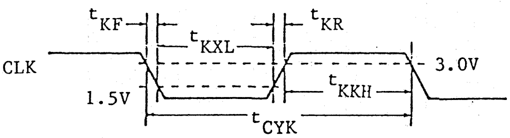
This is a target spec. and can be changed during development.

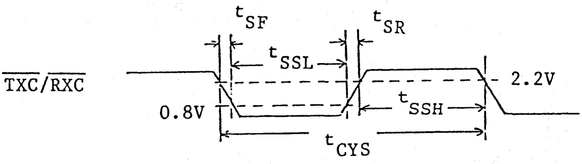
AC Timing Test Points

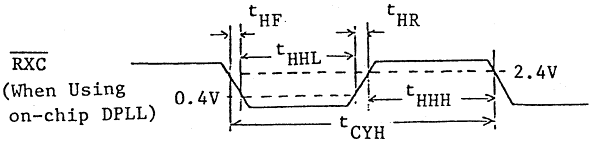


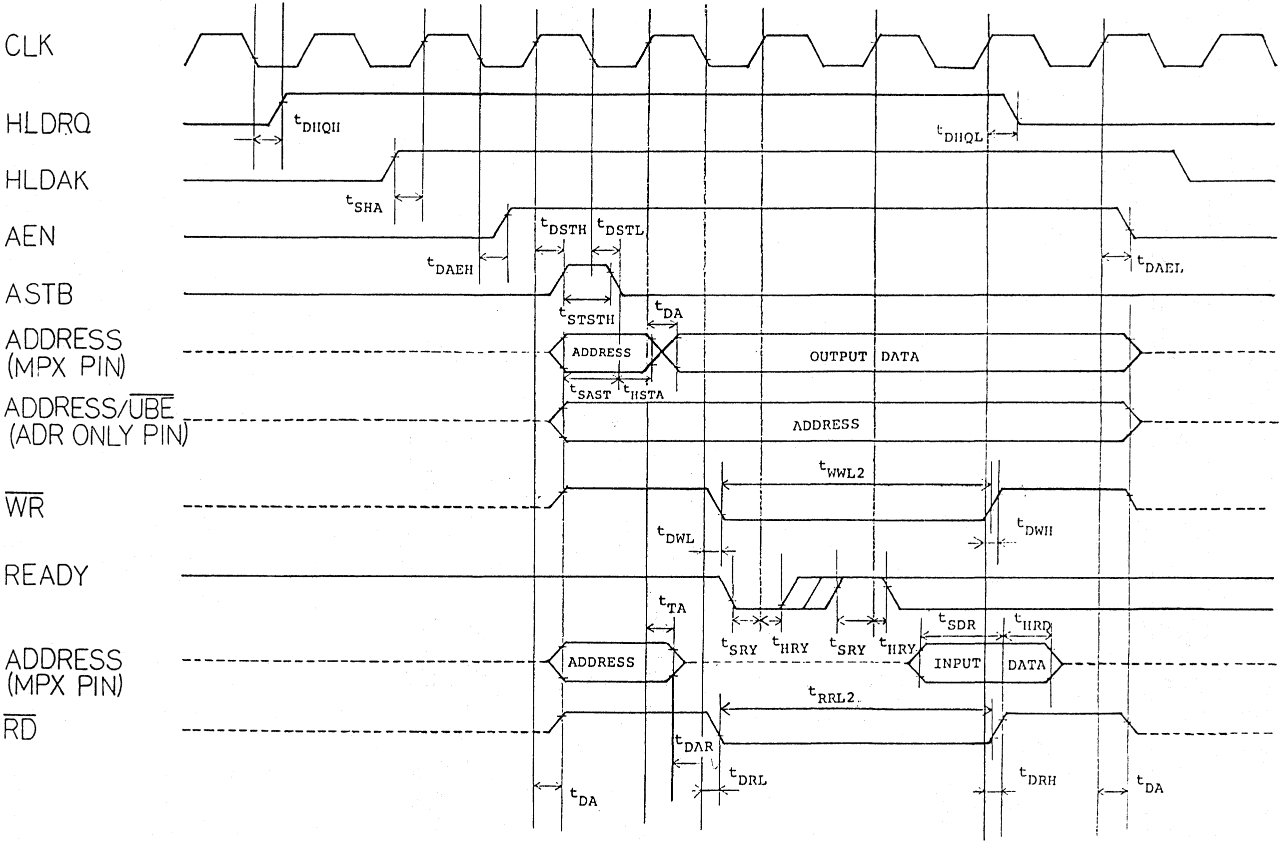
Timing Waveform

CLK Waveform

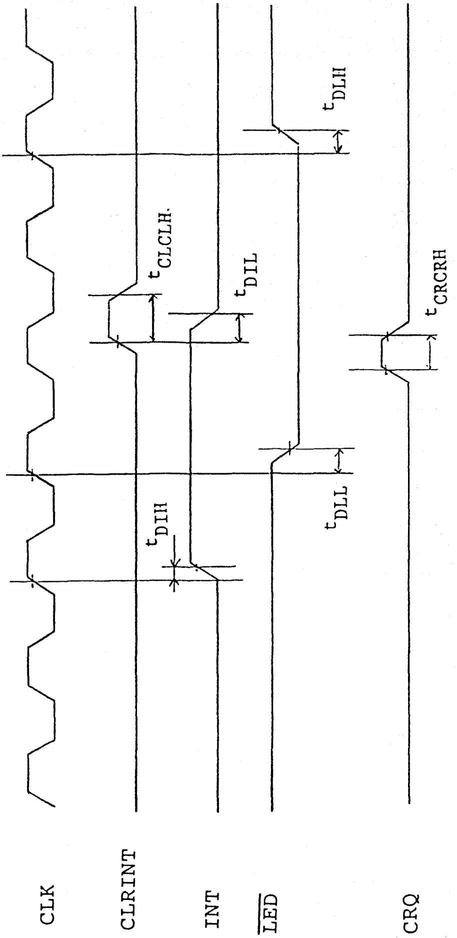




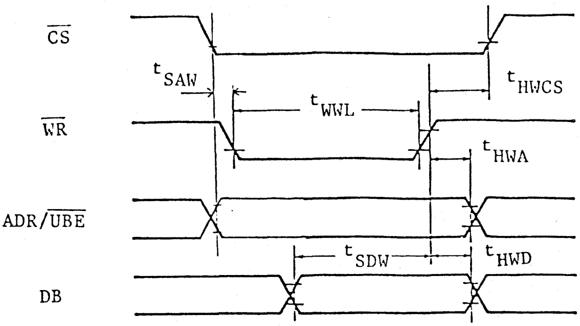


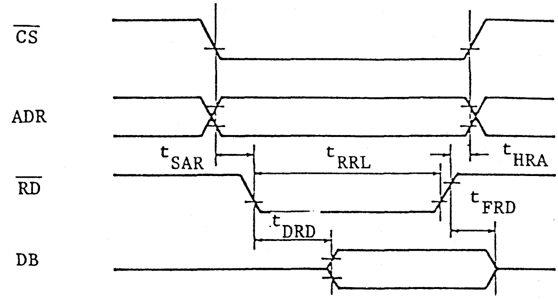


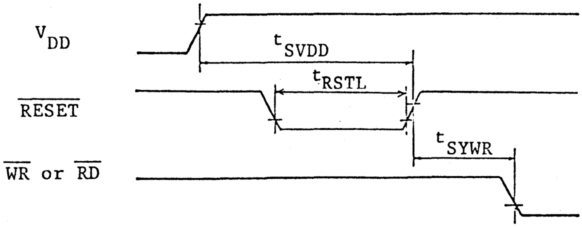
BUS-MASTER MODE

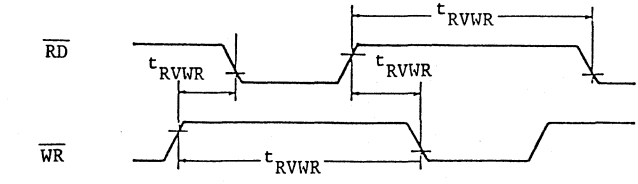

CLK
CLRINT
INT
$\overline{\text{LED}}$
CRQ
t_{DLH}
t_{CLCLH}
t_{DIL}
t_{CRCRH}
t_{DLL}
t_{DIH}

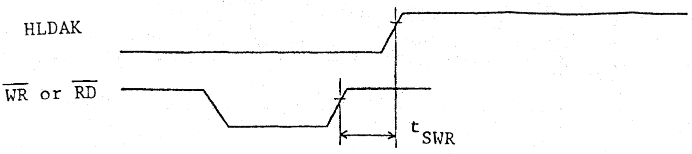
BUS SLAVE MODE

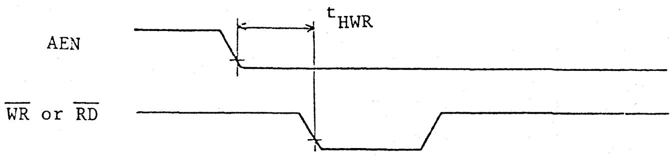


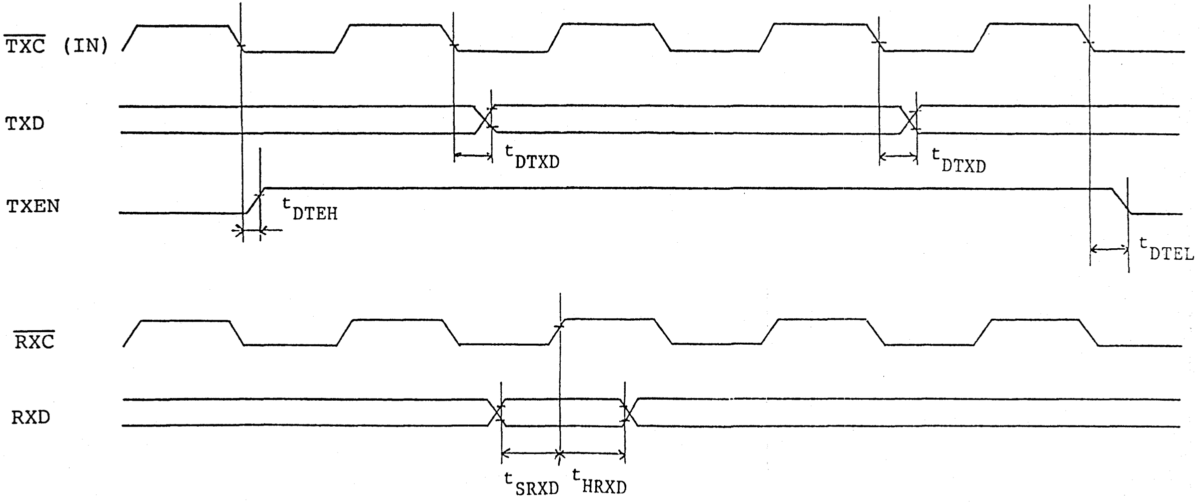




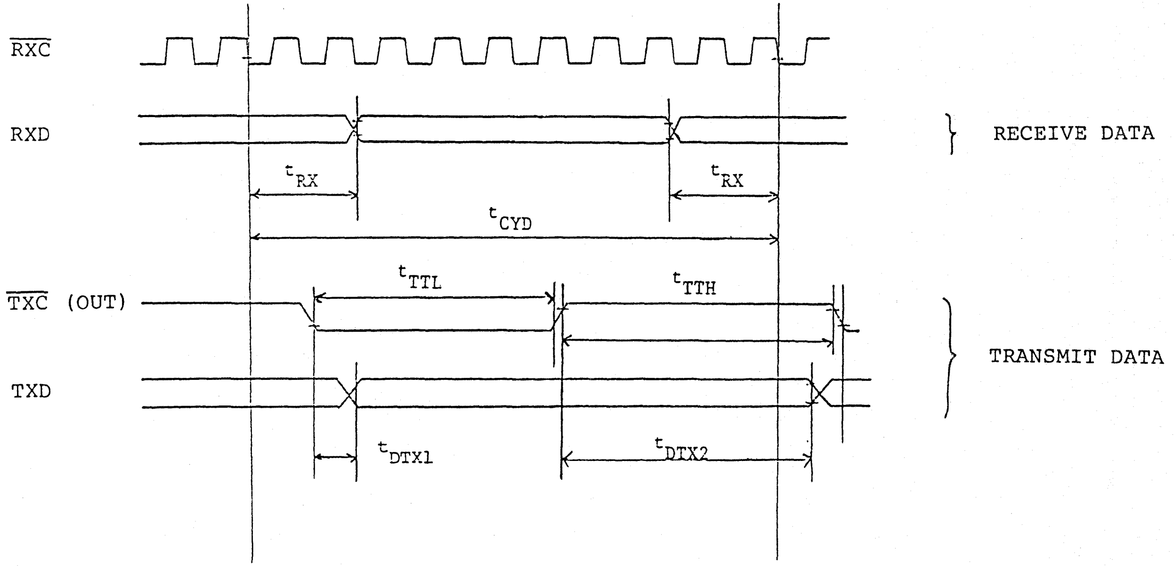



HLDAK
$\overline{WR}$ or $\overline{RD}$
t_{SWR}


t_{HWR}
AEN
$\overline{WR}$ or $\overline{RD}$



When not using on-chip DPLL



When using on-chip DPLL

EUROPEAN DISTRIBUTORS

AUSTRIA
A & D
ABRAHAMCZIK & DEMEL
GES. MBH. & CO KG
EICHENSTRASSE 58-64/1
1120 WIEN
TEL.: (222) 857661
TLX.: 134273

BELGIUM
CN ROOD
DE JAMBLINNE DE MEUXPLEIN 37
1040 BRUSSEL
TEL.: (02) 7352135
TLX.: 22846

MALCHUS ELECTRONICS PVBA
PLANTIN EN MORETUSLEI 172
2000 ANTWERPEN
TEL.: (032) 353256
TLX.: 33637

DENMARK
MER-EL A/S
VED KLAEDEBO 18
2970 HOERSHOLM
TEL.: (2) 571000
TLX.: 37360

FINLAND
OY FERRADO A/B
P.O. BOX 54
VALIMONTIE 1
00380 HELSINKI 38
TEL.: (0) 550002
TLX.: 122214

FRANCE
ASAP
MONSIEUR LEGRIS
42, RUE HENRI MATISSE
59930 LA CHAPELLE D'ARMENTIERES
TEL.: 20 35 11 10

ASAP
RUE DE TROIS PEUPLES
78190 MONTIGNY LE BRETONNEUX
TEL.: (1) 30 43 82 33
TLX.: 69 88 87

CCI
5, RUE MARCELIN BERTHELOT
BP 92
92164 ANTONY
TEL.: (1) 46 66 21 82
TLX.: 203881

CCI
5, RUE BATAILLE
69008 LYON
TEL.: 78 74 44 56

CEDIS (TOURS)
1, RUE DU DANEMARK
37100 TOURS
TEL.: 47 41 76 46

CELTI
Z. I. DE COURTABŒUF
9, AVENUE DU QUEBEC
91940 LES ULIS
TEL.: (1) 64 46 09 09

DIM INTER
65 - 67, RUE DES CITES
93300 AUBERVILLIERS
TEL.: (1) 48 34 93 70
TLX.: 230524

DIM INTER (COLMAR)
27, RUE KLEBER
68000 COLMAR
TEL.: 89 41 15 43

DIM INTER (VILLEURBANNE)
101, RUE DEDIEU
69100 VILLEURBANNE
TEL.: 78 68 32 29

EALING
BATIMENT AUVIDULIS
AVENUE D'OCEANIE
Z.A. D'ORSAY COURTABŒUF
BP 90
91943 LES ULIS CEDEX
TEL.: (1) 69 28 01 31

GEDIS
352, AVENUE G. CLEMENCEAU
92000 NANTERRE
TEL.: (1) 42 04 04 04

GEDIS (AIX)
MERCURE C
Z.I. D'AIX EN PROVENCE
13763 LES MILLES CEDEX
TEL.: 42 60 01 77

GEDIS (ALPES)
21, RUE DES GLAISONS
38400 ST. MARTIN D'HERES
TEL.: 75 51 23 32

SERTRONIQUE (LILLE)
20, RUE CABANIS
BP 35
59007 LILLE CEDEX
TEL.: 20 47 70 70

SERTRONIQUE (MANS)
60, RUE SAGEBIEN
CEDEX 43
72040 LE MANS
TEL.: 43 84 24 60
TLX.: 720019

TEKELEC
RUE CARLE VERNET
CITE DES BRUYERES
92310 SEVRES
TEL.: (1) 45 34 75 35

GERMANY
BIT-ELECTRONIC AG
DINGOLFINGER STRASSE 6
8000 MÜNCHEN 80
TEL.: (0 89) 41 80 07-0
TLX.: 5 212 931

GLEICHMANN + CO ELECTRONICS
GMBH
WORMSER STRASSE 34
6710 FRANKENTHAL
TEL.: (0 62 33) 2 42 77
TLX.: 4 65 270

GLYN GMBH
SCHÖNE AUSSICHT 30
6272 NIEDERNHAUSEN
TEL.: (0 61 27) 80 77
TLX.: 4 186 911

H3W ELEKTRONIK VERTRIEB GMBH
STAHLGRUBERRING 12
8000 MÜNCHEN 82
TEL.: (0 89) 42 92 71
TLX.: 5 214 514

MICROSCAN GMBH
ÜBERSEERING 31
2000 HAMBURG 60
TEL.: (0 40) 6 32 00 30
TLX.: 2 13 288

REIN ELEKTRONIK GMBH
LÖTSCHERWEG 66
4054 NETTETAL 1
TEL.: (0 21 53) 73 31 11
TLX.: 8 54 251

SYSTEM ELEKTRONIK VERTRIEB GMBH
HEESFELD 4
3300 BRAUNSCHWEIG
TEL.: (05 31) 31 40 95
TLX.: 9 52 351

ULTRATRONIK GMBH
MÜNCHENER STRASSE 6
8031 SEEFELD
TEL.: (0 81 52) 70 90
TLX.: 5 26 459

UNIELECTRONIC VERTRIEBS GMBH
LISE-MEITNER-STRASSE 8
6072 DREIEICH 1 B. FRANKFURT
TEL.: (0 61 03) 3 51 75
TLX.: 4 112 13

ITALY
ADELSY S.R.L.
VIA DEL FONDITORE, 5
LOCALITA ROVERI
40127 BOLOGNA
TEL.: (051) 532119

CLAITRON S.P.A.
VIA GALLARATE, 211
20151 MILANO
TEL.: (02) 3010091

MELCHIONI S.P.A.
VIA COLETTA, 37
20135 MILANO
TEL.: (02) 57941

PANTRONIC S.R.L.
VIA MATTIA BATTISTINI, 212/a
00167 ROMA
TEL.: (06) 6273909

NETHERLANDS
CN ROOD
CORT V.D. LINDENSTRAAT 11-13
2288 EV RIJSWIJK
TEL.: (070) 996360
TLX.: 31238

INNOCIRCUIT
MALCHUS ELECTRONICA
ADVIESGROEP
MALCHUS B.V.
FOKKERSTRAAT 511-513
3125 BD SCHIEDAM
TEL.: (010) 4 373777
TLX.: 21598

NORWAY
JAKOB HATTELAND ELECTRONIC A/S
P.B. 25
5578 NEDRE VATS
TEL.: (47) 63 111
TLX.: 428 50

PORTUGAL
AMPEREL S.A.
AV. FONTES PEREIRA DE MELO 47, 4D
1000 LISBOA
TEL.: (1) 53 26 98
TLX.: 18588

SPAIN
AMITRON S.A.
AVENIDA DE VALLADOLID 47 A
28008 MADRID
TEL.: (1) 247 93 13
TLX.: 45550

COMELTA S.A.
EMILIO MUNOZ 41, NAVE 1-1-2
MADRID 17
TEL.: (1) 7 54 30 01
TLX.: 42007

LOBER S.A.
MONTE ESQUINZA 28
MADRID 4
TEL.: (1) 442 11 00
TLX.: 49533

SWEDEN
NORDQVIST & BERG
BOX 1458
S-171 28 SOLNA
TEL.: + 46 87 64 67 10
TLX.: 10407 NORDSWE S

TH'S ELEKTRONIK
BOX 3027
16303 SPAANGA
TEL.: (0) 8362970
TLX.: 11145

SWITZERLAND
MEMOTEC AG
GASWERKSTRASSE 32
4901 LANGENTHAL
TEL.: (63) 28 11 22
TLX.: 9 82 550

TURKEY
BURC ELEKTRONIK
VE MAKINA
SANAYI VE TICARET A.S.
BANKATCHI-SOKAK 15/2
KÜCÜKESAT
ANKARA
TEL.: (0090) 41250300
TLX.: 43430

UNITED KINGDOM
ANZAC COMPONENTS LTD
BURNHAM LANE
SLOUGH SL1 6LN
ENGLAND
TEL.: (06286) 4701

DIALOGUE DISTRIBUTION LTD
WATCHMOOR ROAD
CAMBERLER
SURREY GU15 3AQ
ENGLAND
TEL.: (0276) 688001

FARNEHL ELECTRONIC
COMPONENTS LTD
CANAL ROAD
LEEDS LS12 2TU
ENGLAND
TEL.: (0532) 636311

IMPULSE ELECTRONICS LTD
HAMMOND HOUSE
CATERHAM
SURREY CR3 6XG
TEL.: (0883) 46433

STC MULTI COMPONENTS
EDINBURGH WAY
HARLOW
CM20 2DF
ENGLAND
TEL.: (0279) 442971

VSI ELECTRONICS LTD
ROYDOMBURY INDUSTRIAL PARK
HORSECROFT ROAD 9
HARLOW, 5
ESSEX CM19 5BYQM
TEL.: (0279) 29666

NEC OFFICES

NEC Electronics (Europe) GmbH, Oberrather Str. 4, 4000 Düsseldorf 30, W. Germany,
Tel. (02 11) 65 03 01, Telex 8 58 996-0

NEC Electronics (Germany) GmbH, Oberrather Str. 4, 4000 Düsseldorf 30,
Tel. (02 11) 65 03 02, Telex 8 58 996-0
- Königstr. 12, 3000 Hannover 1, Tel. (05 11) 31 60 91, Telex 9 230 109
- Arabellastr. 17, 8000 München 2, Tel. (0 89) 92 10 03-0, Telex 5 22 971
- Heilbronner Str. 314, 7000 Stuttgart 30, Tel. (07 11) 89 09 10, Telex 7 252 220

NEC Electronics (BNL) – Boschdijk 187a, NL-5612 HB Eindhoven, Tel. (0 40) 44 58 45,
Telex 51 923

NEC Electronics (Scandinavia) – Box 4039, S-18304 Täby, Tel. (08) 73 28 200,
Telex 13 839

NEC Electronics (France) S.A., 9, rue Paul Dautier, B. P. 187,
F-78142 Velizy Villacoublay Cedex, Tél. (1) 39 46 96 17, Télex 699 499

NEC Electronics (France) S.A., Representacion en Espana, Edificio «La Caixa»,
Paseo de la Castellana 51, E-28046 Madrid, Tél. (1) 41 94 150, Télex 41 316

NEC Electronics Italiana S.R.L., Via Fabio Filzi, 25A, I-20124 Milano, Tel. (02) 67 09 108,
Telex 315 355
- Rome Office, Via Monte Cervialto, 131, I-00139 Rome,
  Tel. (06) 8 11 12 91, Telex 623 323

NEC Electronics (UK) Ltd., Cygnus House, Sunrise Park Way, Milton Keynes, MK14 6NP,
Tel. (09 08) 69 11 33, Telex 777 565
- Dublin Office, 34/35 South William Street, Dublin 2, Ireland, Tel. (00 01) 71 02 00