Zilog - Z8019533FSC00TR Datasheet

Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	Z8S180
Number of Cores/Bus Width	1 Core, 8-Bit
Speed	33MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	100-QFP
Supplier Device Package	100-QFP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8019533fsc00tr

PIN DESCRIPTION

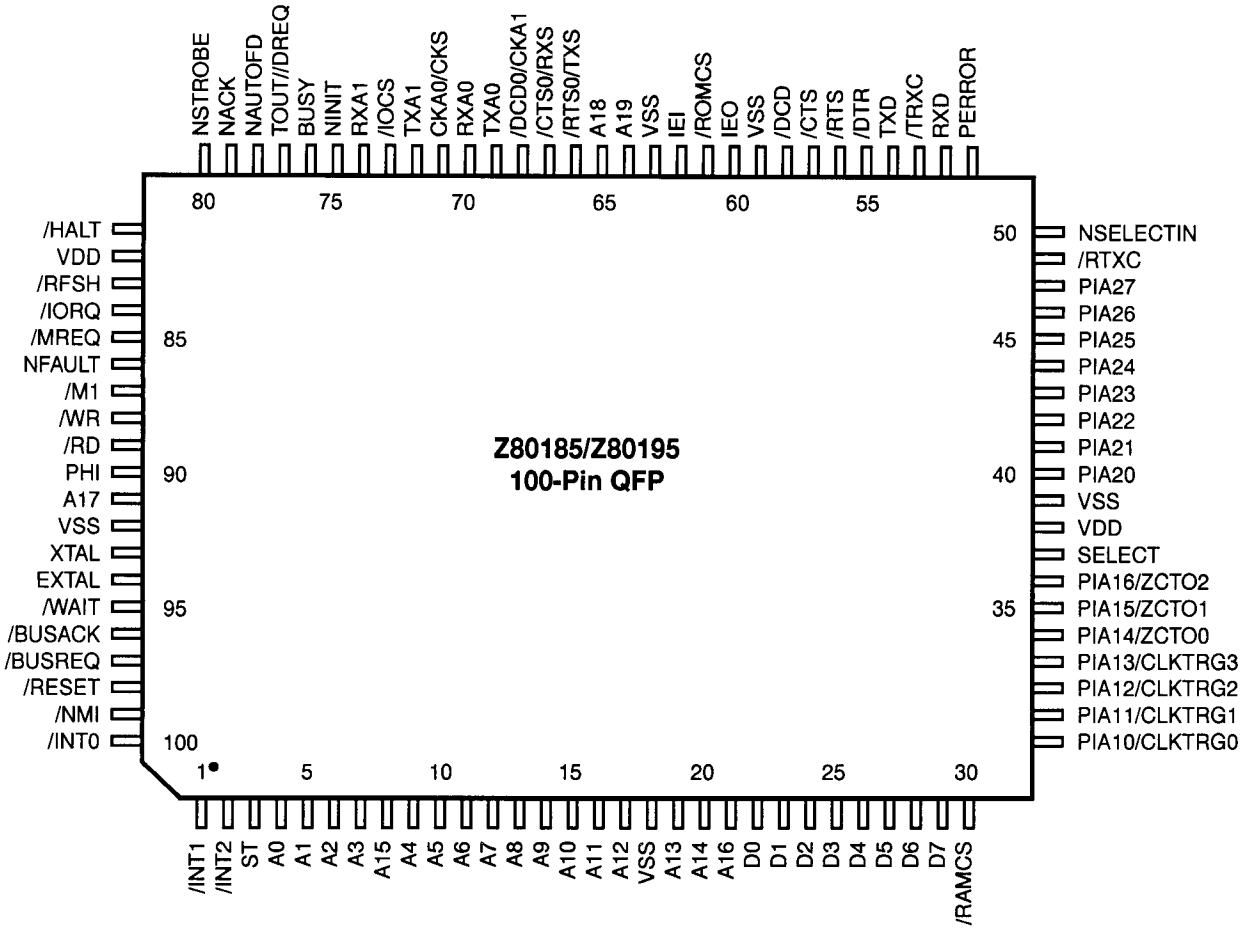


Figure 2. 100-Pin QFP Pin Assignments

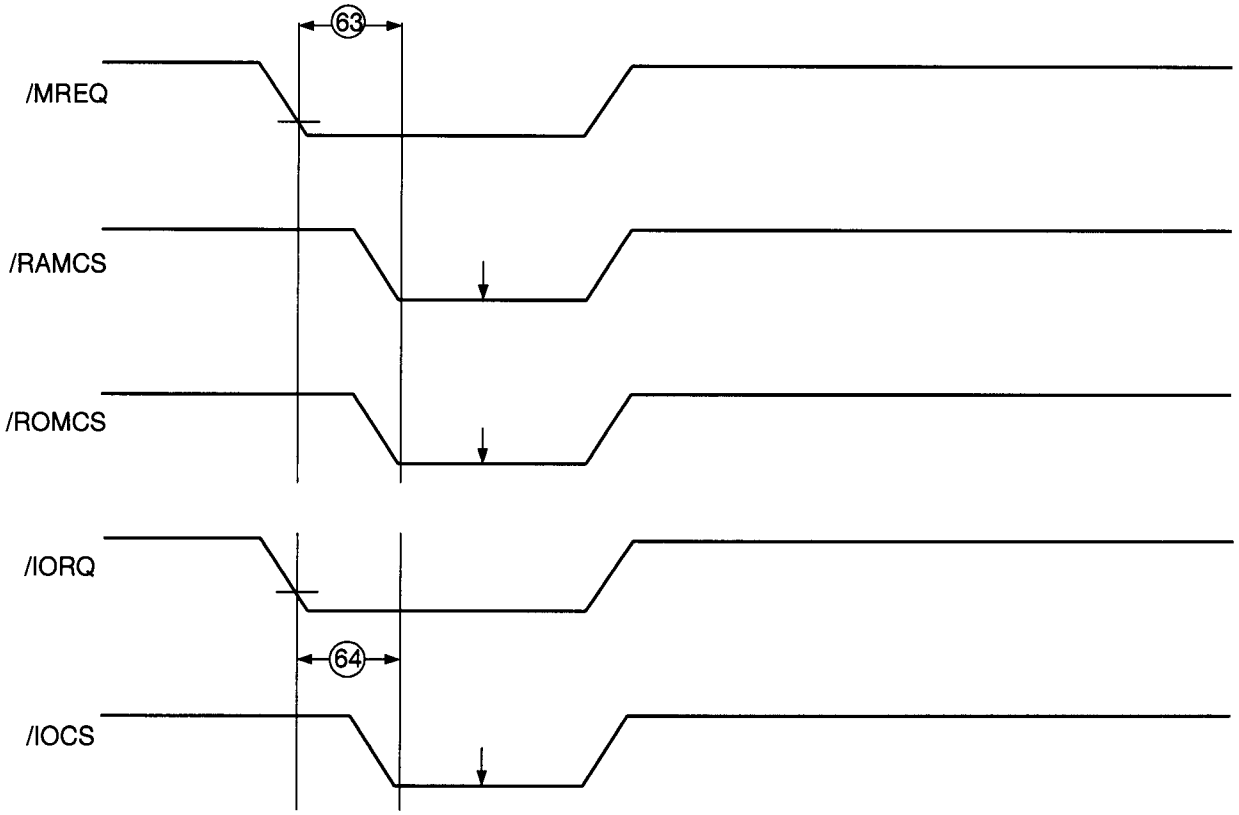


Figure 11. /ROMCS and /RAMCS Timing

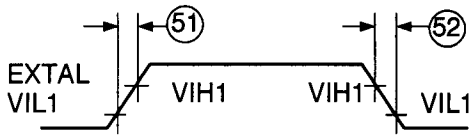


Figure 12. External Clock Rise Time and Fall Time

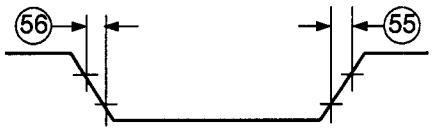


Figure 13. Input Rise and Fall Time
(Except EXTAL, /RESET)

EMSCC System Timing

No.	Symbol	Parameter	20 MHz Min	20 MHz Max	Notes
2	TdRxC(W)	/RxC to /Wait Inactive	13	18	(1,2)
4	TdRxC(INT)	/RxC to /INT Valid	15	22	(1,2)
6	TdTxC(W)	/TxC to /Wait Inactive	8	17	(1,3)
8	TdTxC(INT)	/TxC to /INT Valid	9	17	(1,3)
10	TdExT(INT)	/DCD or /CTS to /INT Valid	3	9	(1)

Notes:
1. Open-drain output, measured with open-drain test load.
2. /RxC is /RTxC or /TRxC, whichever is supplying the receive clock.
3. /TxC is /TRxC or /RTxC, whichever is supplying the transmit clock.
4. Units equal to TcPc

These AC parameter values are preliminary and subject to change without notice.

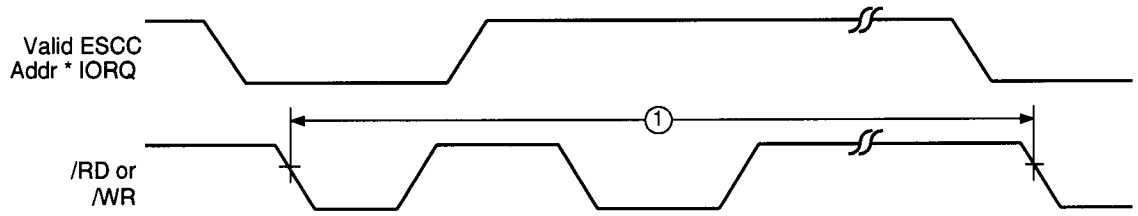


Figure 19. EMSCC External Bus Master Timing

External Bus Master Interface Timing (SCC Related Timing)

No	Symbol	Parameter	Z80185 / Z80195 (20 MHz) Min	Z80185 / Z80195 (20 MHz) Max	Z80185 / Z80195 (33 MHz) Min	Z80185 / Z80195 (33 MHz) Max	Unit	Notes
1	TrC	Valid Access Recovery Time	4TcC		4TcC		ns	(1)

Notes:
1. Applies only between transactions involving the EMSCC.

These AC parameter values are preliminary and subject to change without notice.

T_{CC} = EMSCC Clock Period Time

Z80185 EMSCC Signals

TXD. *Transmit Data* (output). This output transmits serial data at standard TTL levels.

RXD. *Receive Data* (input). This input receives serial data at standard TTL levels.

/TRXC. *Transmit/Receive Clock* (input or output). This pin functions under program control. /TRXC may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

/RTXC. *Receive/Transmit Clock* (input). This pin functions under program control. /RTXC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous mode.

/CTS. *Clear To Send* (input, active Low). If this pin is programmed as an "auto enable", a Low on it enables the EMSCC transmitter. If not programmed as an auto enable, it can be used as a general-purpose input. This pin is Schmitt-trigger buffered to accommodate slow rise-times. The EMSCC detects transitions on this input and can interrupt the processor on either logic level transition.

/DCD. *Data Carrier Detect* (input, active Low). This pin functions as an EMSCC receiver enable when programmed as an "auto enable"; otherwise it can be used as a general-purpose input pin. The pin is Schmitt-trigger buffered to accommodate slow rise-times. The EMSCC detects transitions on this pin and can interrupt the processor on either logic level transition.

EMSCC Signals

/RTS. *Request to Send* (output, active Low). When the Request to Send (RTS) bit in Write Register 5 is set, the/RTS signal goes Low. When the RTS bit is reset in the Asynchronous mode and auto enables is on, the signal goes High after the transmitter is empty. In Synchronous mode, or in Asynchronous mode with auto enables off, the /RTS pin strictly follows the state of the RTS bit. Thus the pin can be used as a general-purpose output. In a special "AppleTalk" mode on the Z80185, the pin is under hardware control.

/DTR. *Data Terminal Ready* (outputs, active Low). The"/DTR//REQ" functionality found in other SCC family members has been reconfigured internal to the EMSCC megacell. The /DTR output is routed to this pin, while the/REQ signal is routed to the DMA request multiplexing logic as described in a later section on the EMSCC. This pin follows the state of the DTR bit in WR5 of the EMSCC.

Note: The /W/REQ pin present on other SCC family members has its two possible functions reconfigured internal to the EMSCC, and both functions are handled internally to the Z80185. The Wait output of the EMSCC drives the/WAIT signal in a wire-ORed fashion with other internal and external peripherals. The /REQ component is routed to the DMA request multiplexing logic as described in a later section on the EMSCC.

Z80185 Parallel Ports

PIA16-14. *Port 1, Bits 6-4 or CTC ZC/TO2-0* (input/output). These lines can be configured as inputs or outputs, or as the "zero count/timeout" outputs of three of the four CTC channels, on a bit-by-bit basis.

PIA13-10. *Port 1, Bits 3-0 or CTC CLK/TRG3-0* (input/output). These lines can be configured as inputs or outputs, or as the "clock/trigger" inputs of the four CTC channels, on a bit-by-bit basis.

PIA27-20. *Port 2, Data, or Bidirectional* (input/output). These lines can be configured as inputs or outputs on a bit-by-bit basis when not used for Bidirectional Centronics operation. However, when used for Bidirectional Centronics operation, software and hardware controls the direction of all eight as a unit.

PIN DESCRIPTIONS (Continued)

Bidirectional Centronics Pins

nStrobe, nAutoFd, nSelectIn, nInit (input/outputs). These are inputs when using P27-20 for the Peripheral side of a Centronics controller, or outputs when using P27-20 for the Host side of such an interface. In certain P1284 modes, these pins assume other names as described in the section on the Centronics P1284 controller. When not using P27-20 for a Centronics controller, these pins can be used as general-purpose inputs or outputs.

Busy, nAck, PError, nFault, Select (input/outputs). These are outputs when using P27-20 for the Peripheral side of a Centronics P1284 controller, or inputs when using P27-20 for the Host side of such an interface. In certain P1284 modes, these pins have other names as described in the section on the Centronics P1284 controller. When not using P27-20 for a Centronics P1284 controller, these pins can be used as general-purpose outputs or inputs. These pins always function in the opposite direction of the preceding group.

System Control Signals

ST. *Status* (output, active High). This signal is used with the /M1 and /HALT output to indicate the nature of each CPU machine cycle.

/RESET. *Reset Signal* (input, active Low). /RESET signal is used for initializing the Z80185 and other devices in the system. It must be kept Low for at least three system clock cycles.

IEI. *Interrupt Enable Signal* (input, active High). IEI is used with IEO to form a priority daisy-chain when there are external interrupt-driven Z80-compatible peripherals.

IEO. *Interrupt Enable Output Signal* (output, active High). In an interrupt daisy-chain, IEO controls the interrupt of external peripherals. IEO is active when IEI is 1 and the CPU is not servicing an interrupt from the on-chip peripherals.

/IOCS. /IOCS decodes /IORQ, /M1, and as many address lines as are necessary to ensure it is activated for an I/O space access to any register in any block of eight registers that does not contain any on-chip registers. Also included in the decode is any programmed relocation of the "180 register set" in the ICR, and the "Decode High I/O" bit in the System Configuration Register. If the "180 registers" aren't relocated, and "Decode High I/O" is 0, /IOCS is active from address XX40 though XXD7, XXF8 through XXFF, and NN00 through NN3F, where NN are non-zero. If the "180 registers" are not relocated and "Decode High I/O" is 1,/IOCS is active from 0040 through 00D7, and 00F8 through FFFF. /IOCS is active when an external master is in control of the bus, as well as when the Z80185 processor has control.

/RAMCS. *RAM Chip Select* (output, active Low). This signal is driven Low for memory accesses at addresses that fall between the values programmed into the RAMLBR and RAMUBR registers. It is active when an external master has control of the bus, as well as when the Z80185 processor is in control.

/ROMCS. *ROM Chip Select* (output, active Low). This output is driven Low for memory accesses between the top of on-chip ROM (if on-chip ROM is enabled) and the value programmed into the ROMBR register. It is active when an external master has control of the bus, as well as when the Z80185 processor is in control.

XTAL. *Crystal* (input, active High). This pin functions as the Crystal oscillator connection and should be left open if an external clock is used instead of a crystal. The oscillator input is not a TTL level (reference DC Characteristics section).

EXTAL. *External Clock/Crystal* (input, active High). This pin functions as a Crystal oscillator connection. An external clock can be input to the Z80185 on this pin when a crystal is not used. This input is Schmitt-triggered.

PHI. *System Clock* (output, active High). This output is the processor's reference clock, and is provided for the use of external logic. The frequency of this output may be equal to, or one-half that of the crystal or input clock frequency, depending on an internal register bit.

Z80185 MPU FUNCTIONAL DESCRIPTION (Continued)

Baud Rate Generator

The Baud Rate Generator (BRG) has two modes. The first is the same as in the Z80180. The second is a 16-bit down counter that divides the processor clock by the value in a 16-bit time constant register, and is identical to the EMSCC BRG. This allows a common baud rate of up to 512 Kbps to be selected. The BRG can also be disabled in favor of an external clock on the CKA pin.

The Receiver and Transmitter will subsequently divide the output of the BRG (or the signal from the CKA pin) by 1, 16 or 64, under the control of the DR bit in the CNTLB register, and the X1 bit in the ASCI Extension Control Register. To compute baud rate, use the following formulas.

If ss2,1,0 = 111, baud rate = f_{CKA} / Clock mode

else if BRG mode baud rate = f_{PHI} / (2 * (TC+2) * Clock mode)

else baud rate = f_{PHI} / ((10 + 20*PS) * 2^ss * Clock mode)

Where:
BRG mode is bit 3 of the ASEXT register
PS is bit 5 of the CNTLB register

TC is the 16-bit value in the ASCI Time Constant registers The TC value for a given baud rate is:

TC = (f_{PHI} / (2 * baud rate * Clock mode)) - 2

Clock mode depends on bit 4 in ASEXT and bit 3 in CNTLB:

X1	DR		Clock Mode
0	0	=	16
0	1	=	64
1	0	=	1
1	1	=	Reserved, do not use.

2^ss depends on the three LS bits of the CNTLB register:

ss2	ss1	ss0	2^ss
0	0	0	= 1
0	0	1	= 2
0	1	0	= 4
0	1	1	= 8
1	0	0	= 16
1	0	1	= 32
1	1	0	= 64
1	1	1	= External Clock from CKA0 (see above).

The ASCIs require a 50 percent duty cycle when CKA is used as an input. Minimum High and Low times on CKA0 are typical of most CMOS devices.

RDRF is set, and if enabled an Rx Interrupt or DMA Request is generated, when the receiver transfers a character from the Rx Shift Register to the Rx FIFO. The FIFO merely provides margin against overruns. When there's more than one character in the FIFO, and software or a DMA channel reads a character, RDRF either remains set or is cleared and then immediately set again. For example, if a receive interrupt service routine doesn't read all the characters in the RxFIFO, RDRF and the interrupt request remain asserted.

The Rx DMA request is disabled when any of the error flags PE or FE or OVRN are set, so that software can identify with which character the problem is associated.

If Bit 7, RDRF Interrupt Inhibit, is set to 1 (see Figures 32 and 33), the ASCI does not request a Receive interrupt when its RDRF flag is 1. Set this bit when programming a DMA channel to handle the receive data from an ASCI. The other causes for an ASCI Receive interrupt (PE, FE, OVRN, and for ASCI0, DCD) continue to request Rx interrupt if the RIE bit is 1. (The Rx DMA request is inhibited if PE or FE or OVRN is set, so that software can tell where an error occurred.) When this bit is 0, as it is after a Reset, RDRF will cause an ASCI interrupt if RIE is 1.

Programmable Reload Timer (PRT)

This logic consists of two separate channels, each containing a 16-bit counter (timer) and count reload register. The time base for the counters is derived from the system clock (divided by 20) before reaching the counter. PRT channel 1 provides an optional output to allow for waveform generation.

The TOUT output of PRT1 is available on a multiplexed pin.

Clocked Serial I/O (CSIO)

The pins for this function are multiplexed with the RTS, CTS, and clock pins for ASCI0. **Note:** It is possible to use both ASCI0 and the CSIO at the same time. If bit 4 of the System Configuration Register is set to 1, the CKS clock signal will internally drive the clock for ASCI0 instead of the system clock.

Z8S180 POWER-DOWN MODES

The following is a detailed description of the enhancements to the Z8S180 from the standard Z80180 in the areas of STANDBY, IDLE, and STANDBY-QUICK RECOVERY modes.

Add-On Features

There are five different power-down modes. SLEEP and SYSTEM STOP are inherited from the Z80180. In SLEEP mode, the CPU is in a stopped state while the on-chipI/Os are still operating. In I/O STOP mode, the on-chip I/Os are in a stopped state while leaving the CPU running. In SYSTEM STOP mode, both the CPU and the on-chip I/Os are in the stopped state to reduce current consumption. The Z8S180 has added two additional power-down modes, STANDBY and IDLE, to reduce current consumption even further. The differences in these power-down modes are summarized in Table 2.

Table 2. Power Down Modes

Power-Down Modes	CPU Core	On-Chip I/O	OSC.	CLKOUT	Recovery Source	Recovery Time (Minimum)
SLEEP	Stop	Running	Running	Running	RESET, Interrupts	1.5 Clock
I/O STOP	Running	Stop	Running	Running	By Programming	-
SYSTEM STOP	Stop	Stop	Running	Running	RESET, Interrupts	1.5 Clock
IDLE†	Stop	Stop	Running	Stop	RESET, Interrupts, BUSREQ	8 +1.5 Clock
STANDBY†	Stop	Stop	Stop	Stop	RESET, Interrupts, BUSREQ	2^{17} +1.5 Clock (Normal Recovery) 2^{6} +1.5 Clock (Quick Recovery)

Note: † IDLE and STANDBY modes are only offered in the Z8S180. Note that the minimum recovery time can be achieved if INTERRUPT is used as the Recovery Source.

STANDBY Mode

The Z8S180 is designed to save power. Two low-power programmable power-down modes have been added: STANDBY mode and IDLE mode. The STANDBY/IDLE mode is selected by multiplexing D6 and D3 of the CPU Control Register (CCR, I/O Address = 1FH).

To enter STANDBY mode:

1. Set D6 and D3 to 1 and 0, respectively.
2. Set the I/O STOP bit (D5 of ICR, I/O Address = 3FH) to 1.
3. Execute the SLEEP instruction.

When the device is in STANDBY mode, it behaves similar to the SYSTEM STOP mode as it exists on the Z80180, except that the STANDBY mode stops the external oscillator, internal clocks and reduces power consumption to 50 µA (typical).

Since the clock oscillator has been stopped, a restart ofthe oscillator requires a period of time for stabilization. An 18-bit counter has been added in the Z8S180 to allow for oscillator stabilization. When the part receives an external IRQ or BUSREQ during STANDBY mode, the oscillator is restarted and the timer counts down 2^{17} counts before acknowledgment is sent to the interrupt source.

The recovery source needs to remain asserted for the duration of the 2^{17} count, otherwise standby will be resumed.

The following is a description of how the device exits STANDBY for different interrupts and modes of operation.

STANDBY Mode Exit with /RESET

The /RESET input needs to be asserted for a duration long enough for the crystal oscillator to stabilize, and then exit from the STANDBY mode. When /RESET is de-asserted, it goes through the normal reset timing to start instruction execution at address (logical and physical) 0000H.

The clocking is resumed within the Z8S180 and at the system clock output after /RESET is asserted when the crystal oscillator is restarted, but not yet stabilized.

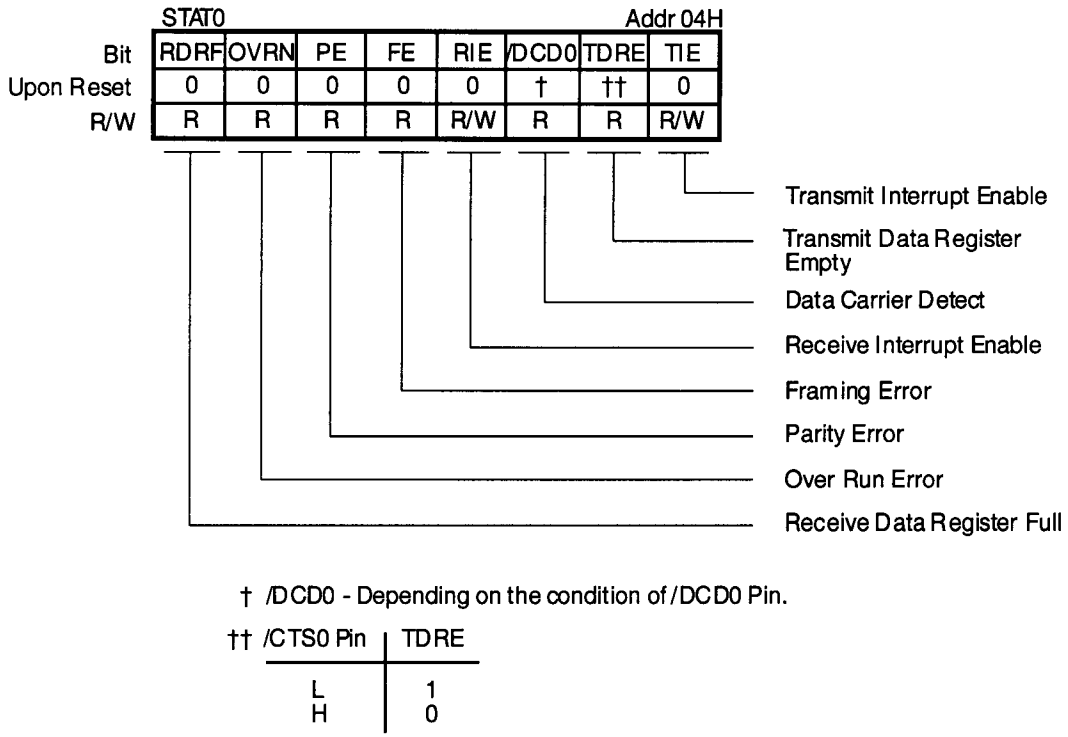


Figure 26. ASCI Status Register (Ch. 0)

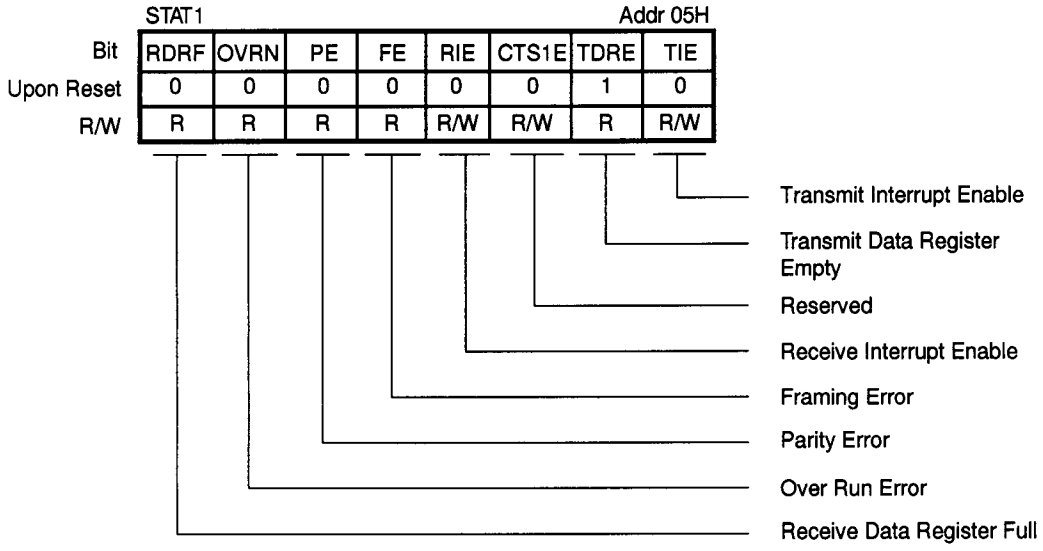


Figure 27. ASCI Status Register (Ch. 1)

TIMER DATA REGISTERS

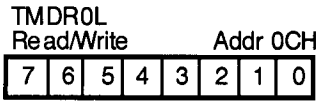
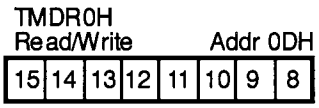


Figure 36. Timer 0 Data Register L



When Read, read Data Register L before reading Data Register H.

Figure 38. Timer 0 Data Register H

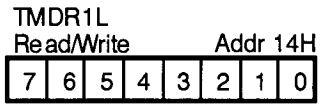
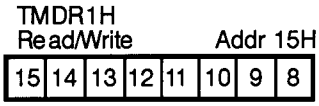


Figure 37. Timer 1 Data Register L



When Read, read Data Register L before reading Data Register H.

Figure 39. Timer 1 Data Register H

TIMER RELOAD REGISTERS

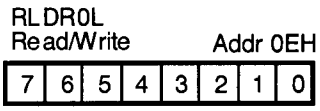


Figure 40. Timer 0 Reload Register L

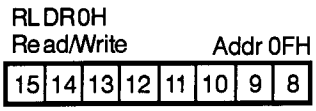


Figure 42. Timer 0 Reload Register H

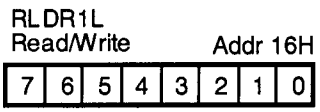


Figure 41. Timer 1 Reload Register L

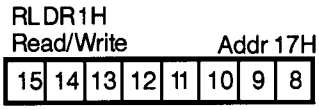
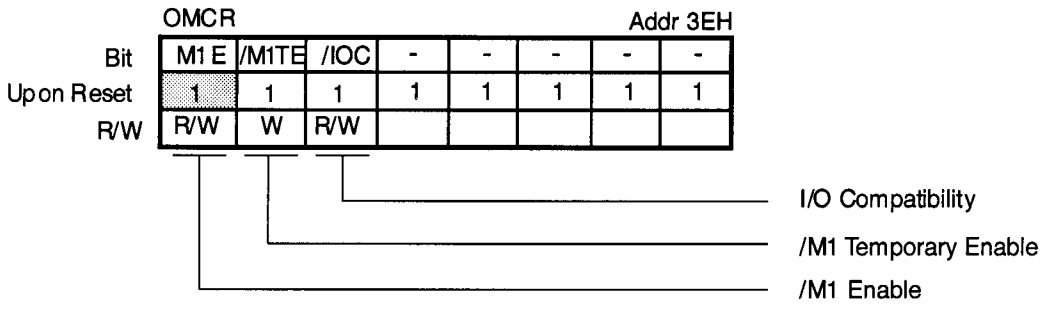


Figure 43. Timer 1 Reload Register H

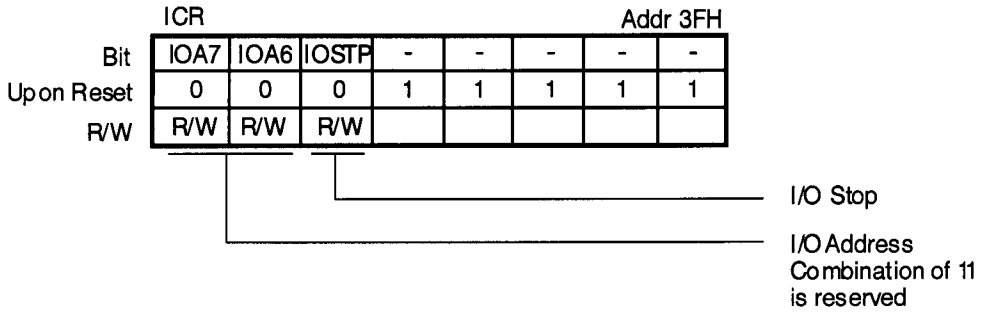
SYSTEM CONTROL REGISTERS



Bit	M1 E	/M1TE	/IOC	-	-	-	-	-
Upon Reset	1	1	1	1	1	1	1	1
R/W	R/W	W	R/W					

Notes:
1. This register should be programmed to 0x0xxxxb (x = don't care) as a part of Initialization.
2. If the M1E bit is set to 1, the process ordoes not fetch a RETI instruction.

Figure 63. Operation Mode Control Register



Bit	IOA7	IOA6	IOSTP	-	-	-	-	-
Upon Reset	0	0	0	1	1	1	1	1
R/W	R/W	R/W	R/W					

Figure 64. I/O Control Register

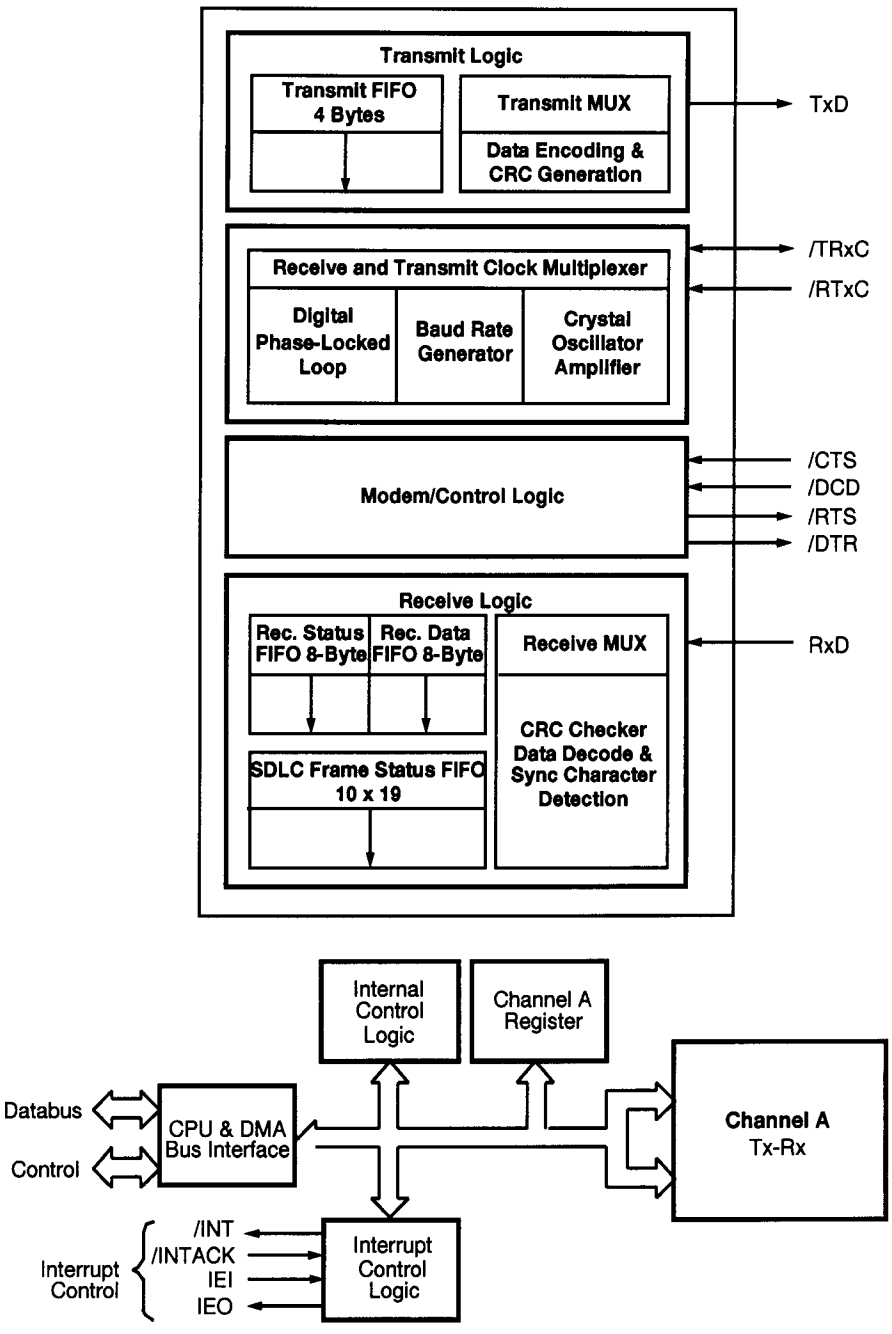


Figure 66. EMSCC Block Diagram

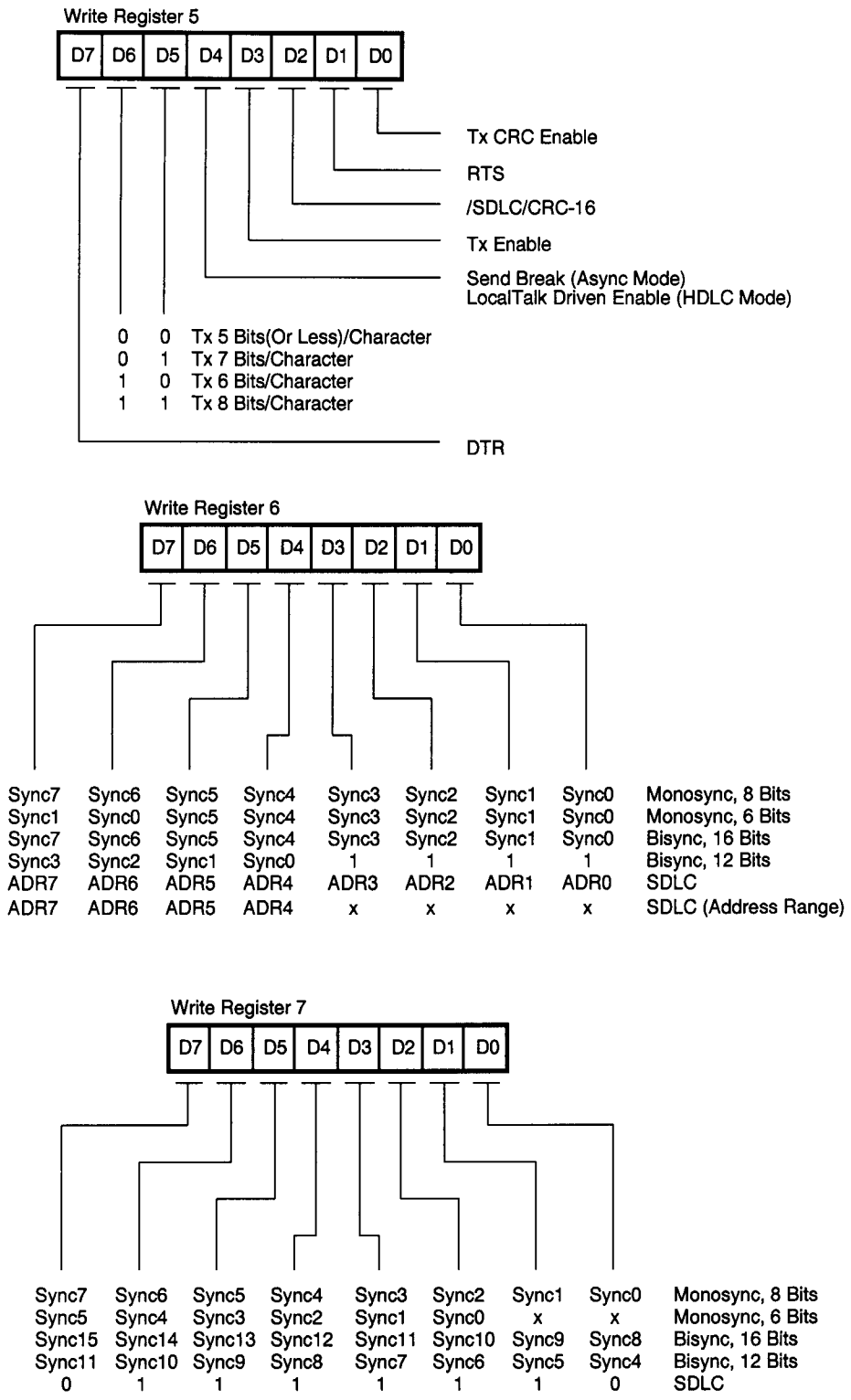


Figure 69. Write Register Bit Functions (Continued)

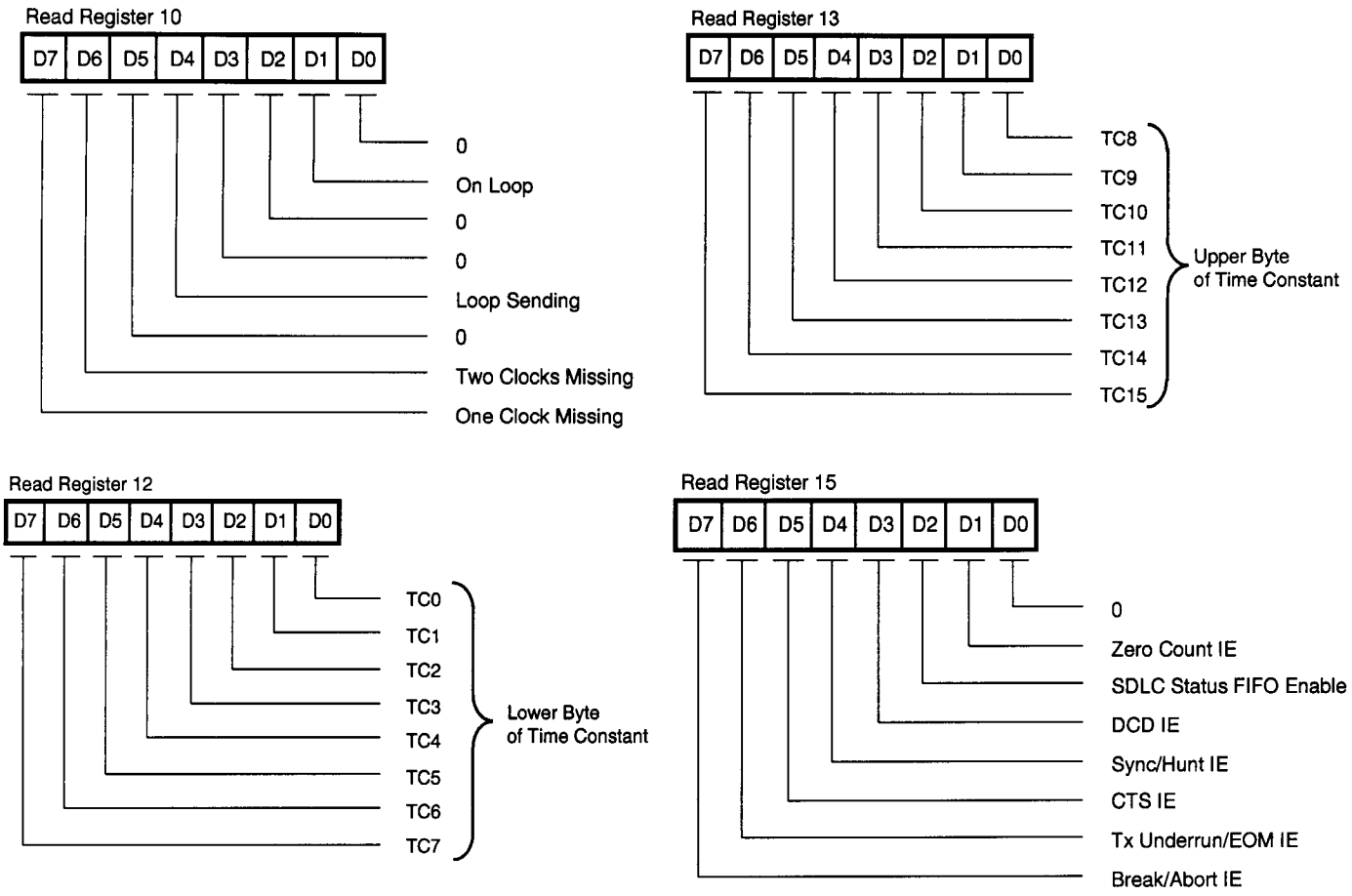


Figure 73. Read Register Bit Functions (Continued)

Z80185 BIDIRECTIONAL CENTRONICS P1284 CONTROLLER (Continued)

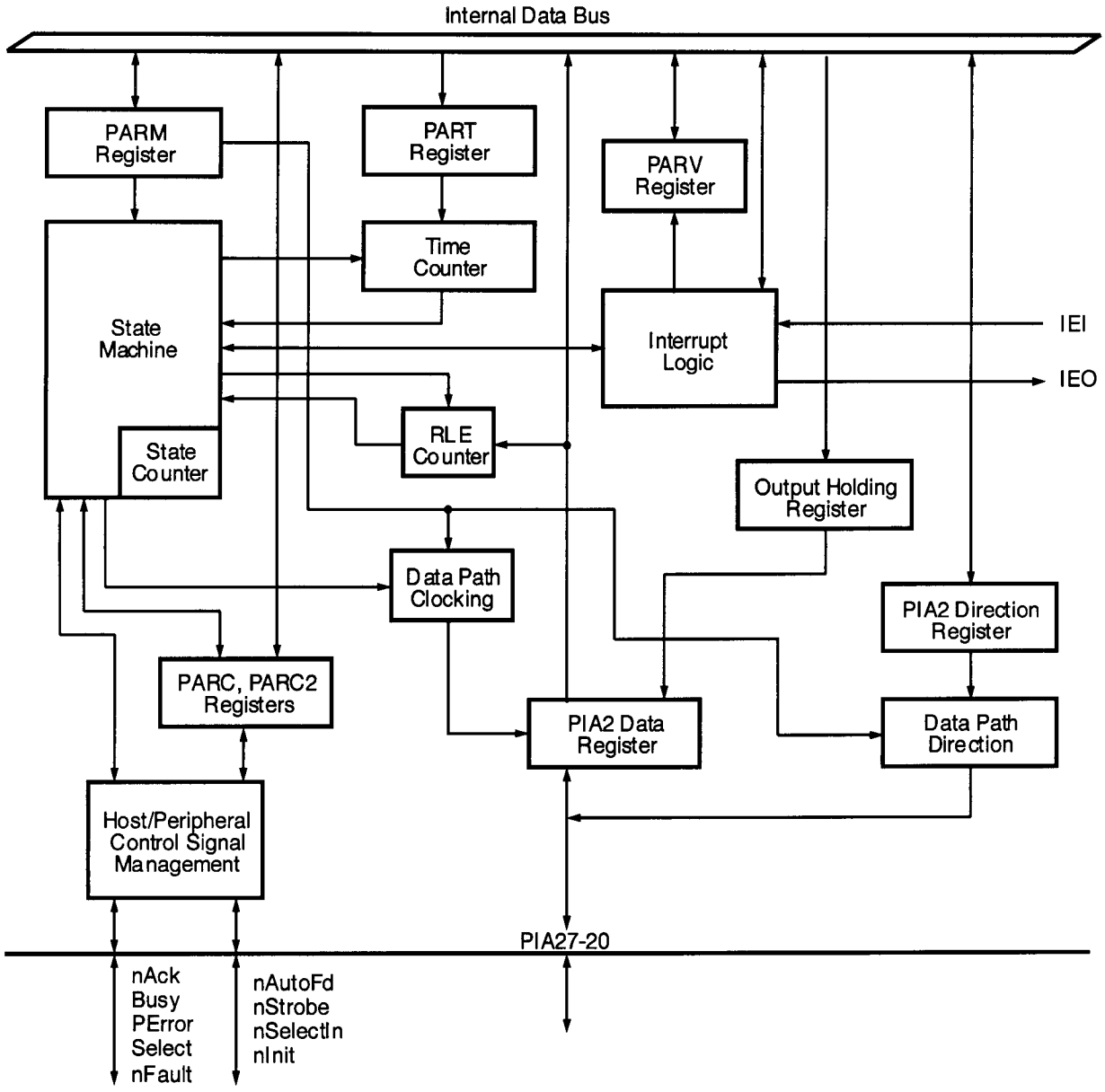


Figure 81. Bidirectional Centronics P1284 Controller Functional Block Description

Z80185 BIDIRECTIONAL CENTRONICS P1284 CONTROLLER (Continued)

Peripheral Nibble Mode

1. Software shouldn't set this mode until there is reverse data available to send. In other words, it should implement the P1284 "reverse idle mode" via software in Peripheral Compatibility/Negotiation mode. After software has driven nDataAvail (nFault), AckDataReq (PError), and Xflag (Select) all Low to signify that data is available, then driven PtrClk (nAck) High after 500 ns, and if requested programmed a DMA channel to provide data to send, when it sees HostBusy (nAutoFd) Low to request data, software should set this mode.

Setting this mode sets DREQ and Idle, but these settings do not request an interrupt. The PIA27-20 pins remain configured for data input but are not used. Instead, four of the five control outputs are driven with the LS and MS four bits of the Input/Output Register, as shown in Table 2, while PtrClk (nAck) serves as a handshake/clock output. On entering this mode the hardware begins routing bits 3-0 of the IOR to these lines.

2. If software, or a DMA channel, writes a byte to the Output Holding Register when Idle is set, the controller immediately transfers the byte to the IOR and clears Idle, and negates DREQ only momentarily to request another byte from software or the DMA channel.

3. After data has been valid on the four control outputs for 500 ns (as controlled by the PART register), the controller drives the PtrClk (nAck) line Low. Then it waits for the host to drive the HostBusy (nAutoFd) line back to High, after which it drives PtrClk (nAck) back to High, switches the four control lines to bits 7-4 of the IOR, and begins waiting for the host to drive HostBusy (nAutoFd) back to Low. When bits 7-4 have been valid for 500 ns and the host has driven HostBusy (nAutoFd) Low, the controller drives PtrClk (nAck) Low again and begins waiting for the host to drive HostBusy (nAutoFd) High. When HostBusy (nAutoFd) has been driven High, the controller returns the four control outputs to the state set by software in PARC. At this point, if software or a DMA channel has not yet written another byte to the Output Holding Register (thus clearing DREQ), the controller sets Idle and waits for software to do so. If/when software or a DMA channel has written a new byte to the OHR, the controller transfers the byte to the IOR, sets DREQ, and clears Idle if it had been set. Then, when the control outputs have been valid for 500 ns, the controller drives PtrClk (nAck) to High. It then waits for the host to drive HostBusy (nAutoFd) back to Low, at which time it switches the four control lines back to bits 3-0 of the IOR and returns to the event sequence at the start of this paragraph.

If there is no more data to send, when the controller sets Idle, software should modify PARC to make nDataAvail (nFault) and AckDataReq (PError) High, and then change the mode to Peripheral Compatible/Negotiation. Then (after 500 ns) software should set PtrClk (nAck) back to High in PARC and enter Reverse Idle state.

Status interrupts in Peripheral Nibble mode include rising and falling edges on P1284Active (nSelectIn) and nInit. The controller sets the IllOp (Illegal Operation) bit if P1284Active (nSelectIn) goes Low in this mode, before it drives nAck High for the status states on the four control lines, or after the host drives HostBusy Low thereafter, in which case software should immediately enter Peripheral Compatibility/Negotiation mode. If P1284Active goes Low, but IllOp stays 0, indicating that the Host negated P1284Active in a legitimate manner, software should enter Peripheral Inactive mode for the duration of the "return to Compatibility mode", and then enter Peripheral Compatibility/Negotiation mode.

Host Byte Mode

1. When in Host Negotiation mode the software has presented the value hex 01 or 05 on PIA27-20, it has been acknowledged by the peripheral, and the peripheral has driven nDataAvail (nFault) and AckDataReq (PError) to Low to indicate data availability and then driven PtrClk (nAck) back to High, software should set this mode. This sets PIA27-20 as inputs regardless of the contents of register E2, and clears the Idle flag. The controller then waits 500 ns (as controlled by the PART register) before proceeding.

2. For each byte, the controller drives HostBusy (nAutoFd) Low to indicate readiness for a byte from the peripheral. Then it waits for PtrClk (nAck) to go Low, at which time it captures the state of PIA27-20 into the Input/Output Register; sets the DREQ bit to request software, or the DMA channel to take the byte, and drives HostBusy (nAutoFd) High and HostClk (nStrobe) Low. When software, or the DMA channel, has taken the byte (thus clearing DREQ) and the peripheral has driven PtrClk (nAck) back High, and at least 500 ns after driving HostClk (nStrobe) Low, the controller drives HostClk (nStrobe) back to High, and samples nDataAvail (nFault). If it is still Low, the controller returns to the event sequence at the start of this paragraph, otherwise it sets the Idle flag.

In response to Idle, software should enter Host Negotiation mode. Thereafter, it can set HostBusy (nAutoFd) Low, to enter Reverse Idle state, or enter Host Compatible mode (reference IEEE P1284 specification), or conduct a new negotiation.

If software is programmed not to accept all the data that a peripheral has available in this mode, it should first disable the DMA channel, if one is in use, and then wait for DREQ to be 1 and nAck to be 1. Then it should reprogram the controller back to Host Negotiation mode, read the last byte from the IOR, drive HostClk (nStrobe) back to High, and then drive P1284Active (nSelectIn) Low to instruct the peripheral to leave Byte mode.

There are no status interrupts in Host Byte mode.

Peripheral Byte Mode

1. Software should not set this mode until there is reverse data available to send — that is, it should implement the P1284 "reverse idle mode" via software in Peripheral Compatibility/Negotiation mode. The exact sequencing among PtrClk (nAck), nDataAvail (nFault), and AckDataReq (PError) differs according to whether this mode is entered directly from Negotiation or from reverse idle phase, and is controlled by software. But in either case, before software sets this mode, it should set nDataAvail (nFault) and AckDataReq (PError) to Low, then after 500 ns, set PtrClk (nAck) to High. When it detects that the host has driven HostBusy (nAutoFd) Low to request data, software should set this mode, which sets the DREQ and Idle flags.

2. In this mode, as long as P1284Active (nSelectIn) remains High, the controller drives PIA27-20 as outputs, regardless of the contents of register E2. When software, or a DMA channel, writes the first byte to the Output Holding Register, the controller immediately transfers the byte to the Input/Output Register, clears Idle but negates DREQ only momentarily, to request another byte from software, or the DMA channel.

3. After each byte is transferred to the IOR, the controller waits 500 ns data setup time (as controlled by the PART register) before driving PtrClk (nAck) Low, and thereafter waits for the host to drive HostBusy (nAutoFd) High. When this occurs, if software, or the DMA channel, has not written more data to the Output Holding Register, that is, if DREQ is still set, the controller sets the Idle flag and waits for software or the DMA channel to do so. If software, or the DMA channel, then writes data to the Output Holding Register, the controller clears DREQ and Idle. When there is data in the OHR and DREQ is 0, this guarantees that it is appropriate to keep nDataAvail (nFault), and AckDataReq (PError) Low to indicate that more data is available, and the controller drives PtrClk (nAck) back to High. The controller then waits for a rising edge on HostClk (nStrobe), and then for the host to drive HostBusy (nAutoFd) Low, at which time it transfers the byte from the OHR to the Output Register, sets DREQ, and then it returns to the event sequence at the start of this paragraph.

While this mode is in effect, software should monitor the interface for two conditions:

Case 1: Idle set and no more data to send, or

Case 2: P1284Active (nSelectIn) Low.

In Case #1, the software should write zero to register E3 to keep PIA27-20 outputs momentarily, and then set the mode back to Peripheral Compatibility, so that the interface is fully under software control, set nDataAvail (nFault) and AckDataReq (PError) High to signify no more data, wait 500 ns, and set PtrClk (nAck) back to High. When HostBusy goes back to Low, the software should set PIA27-20 back to inputs.

In Case #2, if a falling edge on P1284Active happens any time other than between a rising edge on HostClk (nStrobe), and the next falling edge on HostBusy (nAutoFd), the controller sets the IllOp bit to notify software that an immediate Abort is in order, in which case software should immediately enter Peripheral Compatibility/Negotiation Mode. If P1284Active goes Low, but IllOp is not set, meaning that the Host negated P1284Active in a "legal" manner, software should enter Peripheral Inactive Mode for the duration of the "return to Compatibility Mode", and then enter Peripheral Compatibility/Negotiation Mode.

Status interrupts in Peripheral Byte Mode include rising and falling edges on P1284Active (nSelectIn) and nInit.

Z80185 CTC, AND MISCELLANEOUS REGISTERS (Continued)

Bit 7. *Decode High I/O.* If this bit is 0, as it is after a Reset, A15-8 are not decoded for the registers for which A7-6 are 11, that is, the registers for the EMSCC, CTCs, I/O Ports, Bidirectional Centronics Controller. If this bit is 1, A15-8 must all be zero to access these registers, as for the other registers in the Z80185. When set to 0, this bit is compatible with the Z80181 and Z80182, and allows shorter, and more basic I/O instructions to be used to access these registers. Alternately, when set to 1, this bit allows more extensive off-chip I/O.

Bit 6. *Daisy-Chain Configuration Bit 2.* This bit is described with bits 1-0 below.

Bit 5. *Disable /ROMCS.* When this bit is 1, /ROMCS is forced to High, regardless of the status of the address decode logic. This bit Resets to 0 so that /ROMCS is enabled.

Bit 4. When this bit is 0, the /RTS0/TXS, /CTS0/RXS, and CKA0/CKS pins have the /RTS0, /CTS0 and CKA0 functions, respectively. When this bit is 1, the pins have the TXS, RXS, and CKS functions, and the CSIO facility can be used. When this bit is 1, if ASCI0 is used, the "CTS auto-enable" function must not be enabled. The multiplexing of CKA0 is important only with respect to output — the same external clock could be used for both ASCI0 and the CSIO.

Bit 3. When this bit is 0, the PCLK clock of the EMSCC is the same as the processor's PHI clock. When this bit is 1, this clock is PHI/2. Set this bit if the PHI clock is too fast for the EMSCC.

Bit 2. *ROM Emulator Mode Enable.* When this bit is 1, read data from on-chip sources is driven onto the D7-D0 pins, as shown in Table 6. This bit resets to 0.

Bits 1-0. These bits, plus bit 6, determine the routing of the on-chip interrupt daisy-chain, and thus the relative interrupt priority of the on-chip interrupt sources on the daisy-chain as shown in Table 5.

Table 5. Interrupt Daisy-Chain Routing

b6	b1	b0	Daisy Chain Configuration
0	0	0	IEI pin -> EMSCC -> CTC -> Bidirectional Centronics Controller -> IEO pin
0	0	1	IEI pin -> EMSCC -> Bidirectional Centronics Controller -> CTC -> IEO pin
0	1	X	IEI pin -> Bidirectional Centronics Controller -> EMSCC -> CTC -> IEO pin
1	0	0	IEI pin -> CTC -> EMSCC -> Bidirectional Centronics Controller -> IEO pin
1	0	1	IEI pin -> CTC -> Bidirectional Centronics Controller -> EMSCC -> IEO pin
1	1	X	IEI pin -> Bidirectional Centronics Controller -> CTC -> EMSCC -> IEO pin

Z80185 CTC, AND MISCELLANEOUS REGISTERS (Continued)

Interrupt Edge Register

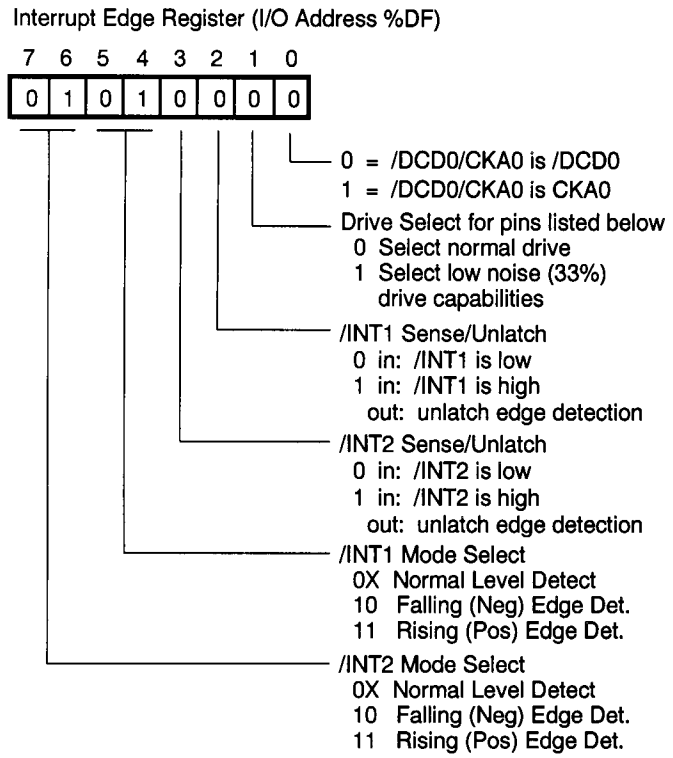


Figure 87. Interrupt Edge Register
(I/O Address %DF)

Bits 7-6. These bits control the interrupt capture logic for the /INT2 pin. When these bits are 0X, the /INT2 pin is level sensitive and Low active. When these bits are 10, negative edge detection is enabled. Any falling edge will latch an active Low on the internal /INT2 to the processor. This interrupt must be cleared by writing a 1 to bit 3 of this register. Programming these bits to 11 enables rising edge interrupts to be latched. The latch must be cleared in the same fashion as for a falling edge.

Bits 5-4. These bits control the interrupt capture logic for the external /INT1 pin. When these bits are 0X, the /INT1 pin is level sensitive and Low active. When these bits are 10, negative edge detection is enabled. Any falling edge will latch an active Low on the internal /INT1 to the processor. This interrupt must be cleared by writing a 1 to bit 2 of this register. Programming these bits to 11 enables rising edge interrupts to be latched. The latch must be cleared in the same fashion as for a falling edge.

Bit 3. Software can read this register to sense the state of the /INT2 pin. Writing a 1 to this bit clears the edge detection logic for /INT2.

Bit 2. Software can read this register to sense the state of the /INT1 pin. Writing a 1 to this bit clears the edge detection logic for /INT1.

Bit 1. This bit selects low noise or normal drive for the parallel ports, bidirectional Centronics controller pins, Chip Select pins, and EMSCC pins as follows:

PIA 10-13	/RTS	nFault
PIA 14-16/ZCT0 0-2	/DTR	nInit
PIA 27-20	TXD	nSelectIn
/ROMCS	/TRXC	nStrobe
/RAMCS	BUSY	PError
/IOCS	nAck	Select
IEO	nAutoFd	

A 1 in this bit selects the low noise option, which is a 33 percent reduction in drive capability. A 0 selects normal drive, and is the default after power-up. Additionally, refer to CPU Register (CCR) for a list of the pins that are programmable for low drive, via the CCR register.

Bit 0. If this bit is 1, the /DCD0/CKA1 pin has the CKA1 function. The pin is always connected to the DCD input of ASCI0, so if this pin is 1, and ASCI0 is used, it should not be programmed to use DCD as a receive auto-enable.

Individual Pin Selection Between PIA1 and CTCs

The assignment of the choice between PIA1 and CTC I/Os is controlled by the PIA1/CTC Pin Select Register (Figure 79).

Bit 7. Reserved, and should be programmed as 0.

Bits 6-4. When the PIA1 data direction register has set the corresponding pins as outputs, for each of these bits that is 0, the pin is driven with the state of the corresponding bit of the PIA 1 Data register, while for each of these bits that is 1, the associated pin is driven with the indicated CTC output. These bits Reset to 0.

Bits 3-1. These bits control whether the CLK/TRG inputs of CTCs 3-1 are taken from PIA3-1, respectively, or from the ZC/TO outputs of CTC2-0, respectively. These bits do not have any affect on the operating mode of the CTCs.

Bit 0. This bit is reserved and should be programmed as 0. CTC0's CLK/TRG0 input is always connected to the PIA10 pin.

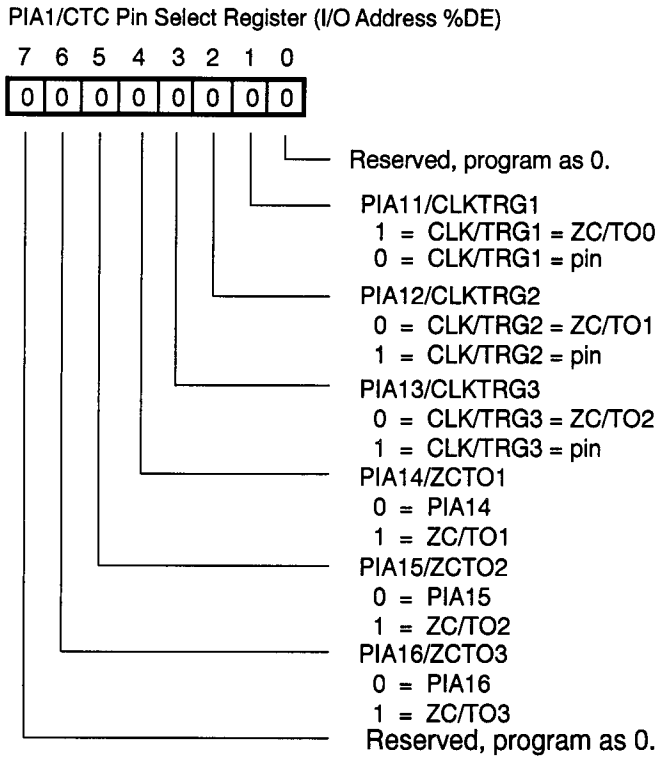


Figure 88. PIA1/CTC Pin Select Register
(I/O Address %DE)