



Z80S183

GENERAL-PURPOSE INTEGRATED
MICROPROCESSOR

PRELIMINARY PRODUCT SPECIFICATION

PS000500-ZMP0599



©1999 by ZiLOG, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval of ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses are conveyed, implicitly or otherwise, by this document under any intellectual property rights.

TABLE OF CONTENTS

| | |
|--------------------------------------------------------------|------------|
| ARCHITECTURAL OVERVIEW 1 | |
| BLOCK DIAGRAM AND OPERATIONAL OVERVIEW | 2 |
| PIN DESCRIPTIONS | 3 |
| OPERATIONAL DESCRIPTION | 10 |
| PROCESSOR DESCRIPTION | 10 |
| DEVICE AND VERSION ID REGISTERS | 13 |
| INTERRUPTS | 13 |
| MEMORY | 20 |
| INPUT/OUTPUT | 24 |
| CLOCK CIRCUITS | 27 |
| RESET CONDITIONS | 29 |
| POWER MANAGEMENT | 29 |
| REDUCED DRIVE/LOW NOISE FEATURES | 33 |
| I/O PORTS | 33 |
| DMA CHANNELS | 36 |
| WATCH-DOG TIMER | 43 |
| PROGRAMMABLE RELOAD TIMERS (PRTs) | 43 |
| REAL TIME CLOCK (RTC) | 46 |
| DIGITAL/ANALOG CONVERTER (DAC) | 48 |
| ANALOG/DIGITAL CONVERTER (ADC) | 49 |
| PROGRAMMABLE I/O SEQUENCER (PIOS) | 52 |
| ASYNC SERIAL COMMUNICATIONS INTERFACES (ASCIs) | 57 |
| CLOCKED SERIAL INPUT/OUTPUT MODULE (CSI/O) | 67 |
| I/O REGISTERS | 70 |
| REGISTERS SUMMARY | 70 |
| BASIC DEVICE REGISTERS | 72 |
| INTERRUPT REGISTERS | 80 |
| MMU REGISTERS | 82 |
| ROM/RAM CHIP SELECT AND WAIT REGISTERS | 83 |
| I/O PORT REGISTERS | 87 |
| DMA REGISTERS | 98 |
| WATCH-DOG TIMER REGISTERS | 110 |
| PROGRAMMABLE RELOAD TIMER (PRT) REGISTERS | 113 |
| REAL TIME CLOCK (RTC) REGISTERS | 118 |
| DIGITAL TO ANALOG CONVERTER (DAC) REGISTERS | 124 |
| ANALOG TO DIGITAL CONVERTER (ADC) REGISTERS | 126 |
| PROGRAMMABLE INPUT/OUTPUT SEQUENCER (PIOS) REGISTERS | 129 |
| ASYNC SERIAL COMMUNICATIONS INTERFACE (ASCI) REGISTERS | 131 |
| CLOCKED SERIAL I/O (CSI/O) REGISTERS | 140 |
| INSTRUCTION SET | 141 |
| CLASSES OF INSTRUCTIONS | 142 |
| PROCESSOR FLAGS | 144 |
| CONDITION CODES | 145 |
| NOTATION | 146 |
| ASSEMBLY LANGUAGE SYNTAX | 147 |
| INSTRUCTION SUMMARY | 148 |

| | |
|------------------------------------------------------|------------|
| OP CODE MAP | 158 |
| ELECTRICAL CHARACTERISTICS | 165 |
| ABSOLUTE MAXIMUM RATINGS | 165 |
| STANDARD TEST CONDITIONS | 165 |
| DC CHARACTERISTICS | 166 |
| AC CHARACTERISTICS | 168 |
| CAPACITANCE | 175 |
| TIMING DIAGRAMS | 175 |
| CHARACTERISTIC CURVES | 179 |
| SYSTEM DESIGN CONSIDERATIONS | 180 |
| ERRATA | 180 |
| APPLICATION NOTES/DEVELOPMENT TOOLS | 181 |
| ZiLOG DEBUG INTERFACE | 181 |
| PACKAGING | 182 |
| ORDERING INFORMATION 1 | 83 |
| PART NUMBER DESCRIPTION 1 | 83 |
| ROM CODE SUBMISSION | 183 |
| DISCLAIMER | 185 |
| PRECHARACTERIZATION PRODUCT | 185 |
| DOCUMENT INFORMATION | 185 |
| CHANGE LOG | 185 |

LIST OF FIGURES

| | | |
|------------|-----------------------------------------------------------|-----|
| FIGURE 1. | Z8S183 BLOCK DIAGRAM | 3 |
| FIGURE 2. | Z8S183 PIN DESCRIPTION | 4 |
| FIGURE 3. | IORD TIMING | 26 |
| FIGURE 4. | FUNDAMENTAL MODE CRYSTAL CIRCUIT < 20 MHz | 28 |
| FIGURE 5. | THIRD-OVERTONE CRYSTAL > 20 MHz | 28 |
| FIGURE 6. | LOW-FREQUENCY CRYSTAL CIRCUIT | 28 |
| FIGURE 7. | PROCESSOR/DMA OPERATION WITH LEVEL-SENSE REQUEST. | 38 |
| FIGURE 8. | PROCESSOR/DMA OPERATION WITH EDGE-SENSE REQUEST | 39 |
| FIGURE 9. | PIOS DIGITAL DATA OUTPUT ENTRY | 54 |
| FIGURE 10. | PIOS ANALOG DATA OUTPUT ENTRY | 54 |
| FIGURE 11. | PIOS ANALOG DATA INPUT ENTRY. | 54 |
| FIGURE 12. | PIOS INTERRUPT ENTRY | 55 |
| FIGURE 13. | PIOS END-OF-SEQUENCE ENTRY | 55 |
| FIGURE 14. | AN ASYNCHRONOUS CHARACTER. | 59 |
| FIGURE 15. | CSI/O OPERATION | 68 |
| FIGURE 16. | TEST CONDITION LOAD CIRCUIT. | 166 |
| FIGURE 17. | BASIC TIMING | 175 |
| FIGURE 18. | MEMORY READ TIMING (ONE WAIT STATE) | 176 |
| FIGURE 19. | MEMORY WRITE TIMING (NO WAIT STATES). | 176 |
| FIGURE 20. | I/O READ TIMING (AUTO WAIT STATE) | 177 |
| FIGURE 21. | I/O WRITE TIMING (AUTO + ONE WAIT STATE) | 177 |
| FIGURE 22. | BUS EXCHANGE TIMING | 178 |
| FIGURE 23. | INTERRUPT TIMING | 178 |
| FIGURE 24. | DMA TIMING | 179 |
| FIGURE 25. | CSI/O TIMING. | 179 |
| FIGURE 26. | CAPACITIVE LOAD CL VS. SWITCHING TIME | 179 |
| FIGURE 27. | ZDI CONNECTOR ON TARGET BOARD | 181 |
| FIGURE 28. | 100 PIN THIN QFP | 182 |
| FIGURE 29. | 100-PIN VQFP PACKAGE DIAGRAM | 18 |

LIST OF TABLES

| | | |
|-----------|---------------------------------------------------------------|----|
| TABLE 1. | PROCESSOR AND DEVICE PIN DESCRIPTIONS | 5 |
| TABLE 2. | UART AND CSI/O PIN DESCRIPTIONS | 8 |
| TABLE 3. | PORT AND PIOS PIN DESCRIPTIONS | 9 |
| TABLE 4. | ANALOG PIN DESCRIPTIONS | 9 |
| TABLE 5. | INTERRUPT OFFSETS AND PRIORITIES | 19 |
| TABLE 6. | LOW-POWER MODES | 31 |
| TABLE 7. | REDUCED DRIVE/LOW NOISE CONTROL BITS | 33 |
| TABLE 8. | SAR0B VALUE FOR A SOURCE IN I/O SPACE | 40 |
| TABLE 9. | DAR0B VALUE FOR A DESTINATION IN I/O SPACE | 40 |
| TABLE 10. | IAR1B VALUE | 41 |
| TABLE 11. | DMA0 SOURCE MODE. | 41 |
| TABLE 12. | DMA0 DESTINATION MODE | 41 |
| TABLE 13. | DMA1 OPERATING MODE. | 42 |
| TABLE 14. | OLD BRG DIVISION FACTORS | 58 |
| TABLE 15. | CSI/O CLOCK RATE SELECTION | 67 |
| TABLE 16. | FREE RUNNING COUNTER (0018H) | 72 |
| TABLE 17. | CLOCK CONTROL REGISTER (001EH) CLKCR | 73 |
| TABLE 18. | CPU CONTROL REGISTER (001FH) CCR | 73 |
| TABLE 19. | REFRESH CONTROL REGISTER (0036H) RCR | 74 |
| TABLE 20. | DEVICE ID LOW (003BH) | 74 |
| TABLE 21. | DEVICE ID HIGH (003CH) | 75 |
| TABLE 22. | REVISION ID 003DH) | 75 |
| TABLE 23. | REVISION ID (003DH) | 76 |
| TABLE 24. | OPERATING MODE CONTROL REGISTER (003EH) OMCR | 76 |
| TABLE 25. | I/O CONTROL REGISTER (003FH) IOCR | 77 |
| TABLE 26. | OUTPUT CONTROL REGISTER (007DH) OCR | 77 |
| TABLE 27. | POWER CONTROL REGISTER (007EH) PCR | 78 |
| TABLE 28. | SYSTEM CONFIGURATION REGISTER (007FH) SCR | 79 |
| TABLE 29. | INTERRUPT VECTOR LOW REGISTER (0033H) IL | 80 |
| TABLE 30. | INTERRUPT/TRAP CONTROL REGISTER (0034H) ITC | 80 |
| TABLE 31. | INT2-1 INTERRUPT EDGE REGISTER (0035H) IECR | 81 |
| TABLE 32. | COMMON BASE REGISTER (0038H) CBR | 82 |
| TABLE 33. | BANK BASE REGISTER (0039H) BBR | 82 |
| TABLE 34. | COMMON/BANK AREA REGISTER C(003AH) CBAR | 83 |
| TABLE 35. | WAIT STATE GENERATOR CONTROL REGISTER (006BH) WSGCR | 83 |
| TABLE 36. | ROM BOUND REGISTER (006CH) ROMBR | 84 |
| TABLE 37. | RAM LOWER BOUND REGISTER (006DH) RAMLBR | 85 |
| TABLE 38. | RAM UPPER BOUNDARY REGISTER (006EH) RAMUBR | 85 |
| TABLE 39. | RAM UPPER BOUND REGISTER (006EH) RAMUBR | 86 |

| | |
|-----------------------------------------------------------------------------|-----|
| TABLE 40. PORT A DATA REGISTER (0040H) DRA | 87 |
| TABLE 41. PORT A DATA DIRECTION REGISTER (0041H) DDRA | 88 |
| TABLE 42. PORT A ALTERNATE FUNCTION SELECT REGISTER (0042H) AFSA | 88 |
| TABLE 43. PORT A OUTPUT CONTROL REGISTER (0043H) OCRA | 89 |
| TABLE 44. PORT B DATA REGISTER (0044H) DRB | 89 |
| TABLE 45. PORT B DATA DIRECTION REGISTER (0045H) DDRB | 90 |
| TABLE 46. PORT B ALTERNATE FUNCTION SELECT REGISTER (0046H) AFSB | 91 |
| TABLE 47. PORT B OUTPUT CONTROL REGISTER (0047H) OCRB | 92 |
| TABLE 48. PORT C DATA REGISTER (0048H) DRC | 92 |
| TABLE 49. PORT C DATA DIRECTION REGISTER (0049H) DDRC | 93 |
| TABLE 50. PORT C ALTERNATE FUNCTION SELECT REGISTER (004AH) AFSC | 94 |
| TABLE 51. PORT C OUTPUT CONTROL REGISTER (004BH) OCRC | 94 |
| TABLE 52. PORT D DATA REGISTER (004CH) DRD | 95 |
| TABLE 53. PORT D DATA DIRECTION REGISTER (004DH) DDRD | 95 |
| TABLE 54. PORT D ALTERNATE FUNCTION SELECT REGISTER (004EH) AFSD | 96 |
| TABLE 55. PORT D OUTPUT CONTROL REGISTER (004FH) OCRD | 97 |
| TABLE 56. DMA0 SOURCE ADDRESS REGISTER LOW (0020H) SAR0L | 98 |
| TABLE 57. DMA0 SOURCE ADDRESS REGISTER LOW (0020H) SAR0L | 98 |
| TABLE 58. DMA0 SOURCE ADDRESS REGISTER HIGH (0021H) SAR0H | 99 |
| TABLE 59. DMA0 SOURCE ADDRESS REGISTER B (0022H) SAR0B | 99 |
| TABLE 60. DMA0 DESTINATION ADDRESS REGISTER LOW (0023H) DAR0L | 100 |
| TABLE 61. DMA0 DESTINATION ADDRESS REGISTER HIGH (0024H) DAR0H | 100 |
| TABLE 62. DMA0 DESTINATION ADDRESS REGISTER B (0025H) DAR0B | 101 |
| TABLE 63. DMA0 BYTE COUNT REGISTER LOW (0026H) BC0L | 101 |
| TABLE 64. DMA0 BYTE COUNT REGISTER HIGH (0027H) BC0H | 102 |
| TABLE 65. DMA1 MEMORY ADDRESS REGISTER LOW (0028H) MAR1L | 102 |
| TABLE 66. DMA1 MEMORY ADDRESS REGISTER HIGH (0029H) MAR1H | 102 |
| TABLE 67. DMA1 MEMORY ADDRESS REGISTER B (002AH) MAR1B | 103 |
| TABLE 68. DMA1 I/O ADDRESS REGISTER LOW (002BH) IAR1L | 103 |
| TABLE 69. DMA1 I/O ADDRESS REGISTER HIGH (002CH) IAR1H | 103 |
| TABLE 70. DMA1 I/O ADDRESS REGISTER B (002DH) IAR1B | 104 |
| TABLE 71. DMA1 BYTE COUNT REGISTER LOW (002EH) BCR1L | 105 |
| TABLE 72. DMA1 BYTE COUNT REGISTER HIGH (002FH) BCR1H | 106 |
| TABLE 73. DMA STATUS REGISTER (0030H) DSTAT | 106 |
| TABLE 74. DMA MODE REGISTER (0031H) DMODE | 107 |
| TABLE 75. DMA MODE REGISTER (0031H) DMODE | 108 |
| TABLE 76. DMA/WAIT CONTROL REGISTER (0032H) DCNTL | 109 |
| TABLE 77. WATCH-DOG TIMER MASTER REGISTER (0064H) WDTMR | 110 |

| | |
|--------------------------------------------------------------------|-----|
| TABLE 78. WATCH-DOG TIMER MASTER REGISTER (0064H) WDTMR | 111 |
| TABLE 79. WATCH-DOG TIMER COMMAND REGISTER (0065H) WDTCR | 112 |
| TABLE 80. PRT0 TIMER DATA REGISTER LOW (000CH) TMDROL | 113 |
| TABLE 81. PRT0 TIMER DATA REGISTER HIGH (000DH) TMDROH | 113 |
| TABLE 82. PRT0 RELOAD REGISTER LOW (000EH) RLDR0L | 113 |
| TABLE 83. PRT0 TIMER DATA REGISTER LOW (000CH) TMDROL | 114 |
| TABLE 84. PRT0 TIMER DATA REGISTER HIGH (000DH) TMDROH | 114 |
| TABLE 85. PRT0 RELOAD REGISTER LOW (000EH) RLDR0L | 114 |
| TABLE 86. PRT0 RELOAD REGISTER HIGH (000FH) RLDR0H | 115 |
| TABLE 87. TIMER CONTROL REGISTER (0010H) TCR | 115 |
| TABLE 88. PRT1 TIMER DATA REGISTER LOW (0014H) TMDR1L | 116 |
| TABLE 89. PRT1 TIMER DATA REGISTER HIGH (0015H) TMDR1H | 116 |
| TABLE 90. PRT1 RELOAD REGISTER LOW (0016H) RLDR1L | 116 |
| TABLE 91. PRT1 RELOAD REGISTER HIGH (0017H) RLDR1H | 117 |
| TABLE 92. RTC CONTROL/STATUS REGISTER (006FH) RTCCS | 118 |
| TABLE 93. RTC CONTROL/STATUS REGISTER (006FH) RTCCS | 119 |
| TABLE 94. RTC SECONDS REGISTER (0070H) RTCSEC | 120 |
| TABLE 95. RTC MINUTES REGISTER (0071H) RTCMIN | 120 |
| TABLE 96. RTC HOURS REGISTER 0072H) RTCHR | 120 |
| TABLE 97. RTC DAY OF THE WEEK REGISTER (0073H) RTCDAY | 121 |
| TABLE 98. RTC DATE REGISTER (0074H) RTCDAT | 121 |
| TABLE 99. RTC MONTH REGISTER (0075H) RTCMO | 122 |
| TABLE 100. RTC YEAR REGISTER (0076H) RTCYR | 122 |
| TABLE 101. RTC CENTURY REGISTER (0077H) RTCC | 123 |
| TABLE 102. RTC ALARM SECONDS REGISTER (0078H) ALARMS | 123 |
| TABLE 103. RTC ALARM MINUTES REGISTER (0079H) ALARMM | 123 |
| TABLE 104. RTC ALARM HOURS REGISTER (007AH) ALARMH | 124 |
| TABLE 105. DAC CONTROL REGISTER (0069H) DACCR | 124 |
| TABLE 106. DAC DATA REGISTER (006AH) DAC | 125 |
| TABLE 107. ADC CONTROL REGISTER 0 (0066H) ADCC0 | 126 |
| TABLE 108. ADC CONTROL REGISTER 1 (0067H) ADCC1 | 127 |
| TABLE 109. ADC RESULT REGISTER (0068H) ADC | 128 |
| TABLE 110. PIOS CONTROL REGISTER (0060H) PIOSCR | 129 |
| TABLE 111. PIOS ADDRESS/TYPE REGISTER (0061H) PIOSAT | 130 |
| TABLE 112. PIOS COUNTER LOW (0062H) PIOSCL | 130 |
| TABLE 113. PIOS COUNTER HIGH (0063H) PIOSCH | 130 |
| TABLE 114. ASCIO CONTROL REGISTER A (0000H) CNTL A0 | 131 |
| TABLE 115. ASCI1 CONTROL REGISTER A (0001H) CNTLA1 | 132 |
| TABLE 116. ASCIO CONTROL REGISTER B (0002H) CNTLB0 | 132 |

TABLE 117.ASCI1 CONTROL REGISTER B (0003H) CNTLB1133

TABLE 118.ASCIO STATUS REGISTER (0004H) STATO134

TABLE 119.ASCI1 STATUS REGISTER (0005H) STAT1135

TABLE 120.ASCIO TX DATA REGISTER (0006H) TDRO135

TABLE 121.ASCI1 TX DATA REGISTER (0007H) TDR1136

TABLE 122.ASCIO RX DATA REGISTER (0008H) RDRO136

TABLE 123.ASCI1 RX DATA REGISTER (0009H) RDR1136

TABLE 124.ASCIO EXTENSION CONTROL REGISTER (0012H) ASEXT0137

TABLE 125.ASCI1 EXTENSION CONTROL REGISTER (0013H) ASEXT1138

TABLE 126.ASCIO TIME CONSTANT LOW (001AH) ASTCOL138

TABLE 127.ASCIO TIME CONSTANT HIGH (001BH) ASTCOH139

TABLE 128.ASCI1 TIME CONSTANT LOW (001CH) ASTC1L139

TABLE 129.ASCIO TIME CONSTANT HIGH (001DH) ASTC1H139

TABLE 130. CSI/O CONTROL REGISTER (000AH) CNTR140

TABLE 131. CSI/O DATA REGISTER (000BH) TRDR141

TABLE 132.LOAD INSTRUCTION 142

TABLE 133.ARITHMETIC INSTRUCTIONS..... 142

TABLE 134.LOGICAL INSTRUCTIONS..... 142

TABLE 135.EXCHANGE INSTRUCTIONS 142

TABLE 136.PROGRAM CONTROL INSTRUCTIONS..... 143

TABLE 137.BIT MANIPULATION INSTRUCTIONS 143

TABLE 138.BLOCK TRANSFER INSTRUCTIONS 143

TABLE 139.ROTATE AND SHIFT INSTRUCTIONS 143

TABLE 140.INPUT/OUTPUT INSTRUCTIONS 144

TABLE 141.PROCESSOR CONTROL INSTRUCTIONS 144

TABLE 142.FLAG REGISTERFF 145

TABLE 143.FLAG SETTINGS DEFINITIONS..... 145

TABLE 144.CONDITION CODES 146

TABLE 145.SYMBOLS..... 146

TABLE 146.INSTRUCTION SUMMARY148

TABLE 147.OP CODE MAP (1ST OP CODE)158

TABLE 148.OP CODE MAP (2ND OP CODE AFTER OCBH)159

TABLE 149.OP CODE MAP (2ND OP CODE AFTER ODDH)160

TABLE 150.OP CODE MAP (2ND OP CODE AFTER OEDH)161

TABLE 151.OP CODE MAP (2ND OP CODE AFTER OFDH)162

TABLE 152.OP CODE MAP (4TH BYTE, AFTER ODDH, OCBH, AND D)163

TABLE 153.OP CODE MAP (4TH BYTE, AFTER OFDH, OCBH, AND D)164

TABLE 154.ABSOLUTE MAXIMUM RATINGS165

TABLE 155.DC CHARACTERISTICS, NORMAL TEMPERATURE RANGE166

TABLE 156.DC CHARACTERISTICS, EXTENDED TEMPERATURE RANGE167

TABLE 157.AC CHARACTERISTICS, NORMAL TEMPERATURE RANGE168
TABLE 158.AC CHARACTERISTICS, EXTENDED TEMPERATURE RANGE171

ARCHITECTURAL OVERVIEW

The Z80S183 is a general-purpose integrated microprocessor. It includes the Z8S180 processor, 32 bits of general purpose I/O, an Analog-to-Digital converter with eight multiplexed inputs, a Programmable I/O Sequencer, a Digital-to-Analog converter, a Watch-Dog Timer, two ASCI channels, two timers, a CSI/O channel, a Real Time Clock, 2KB of on-chip RAM, and 1KByte of on-chip ROM. It is packaged in a 100-pin VQFP.

The Z80S183 includes the following features:

- Code-compatible with Z80 & Z180
- On-chip wait state generator
- Two enhanced UART channels (ASCIs)
- Two 16-bit counters
- Three interrupt request inputs, two with optional edge-triggering
- Real time clock
- Two on-chip oscillators
- DC-to-33 MHz operating frequency @ 5.0V
- DC-to-20 MHz operating frequency @ 3.3V
- Clock divide by 2X or 1X
- Fully static CMOS design with low-power standby
- 2 KB of on-chip RAM
- 1 KB of on-chip ROM
- Eight 10-bit A/D channels
- One 10-bit D/A
- 32 bits of general-purpose I/O
- Low-power PLL oscillator
- Programmable Input/Output Sequencer (PIOS)
- Watch-Dog Timer (WDT)
- Clocked Serial I/O Interface (CSI/O)
- ZiLOG Debug Interface (ZDI)
- Power-down logic
- ASCI Tx complete output
- Economical 100-pin VQFP
- Interrupts on ports A and D

BLOCK DIAGRAM AND OPERATIONAL OVERVIEW

Figure 1 illustrates the block diagram for the Z8S183. In addition to a Z8S180-compatible processor, it includes the following modules:

32 Bits of General Purpose I/O. Four 8-bit ports are selectively multiplexed with on-chip peripheral functions (ASCIs, CSI/O, PRT, PIOS), and are individually programmable as inputs or outputs. Each I/O pin can source and sink 15mA.

Programmable I/O Sequencer. An engine that is independent of the processor, that can drive programmable waveforms onto 8 digital outputs, as well as initiating A/D and D/A conversions

Two ASCI Channels. Asynchronous serial channels with baud rate generators, modem control, and status.

Two 16-bit Timers. Down-counters with interrupt capability.

CSI/O. Clocked serial I/O can be used for serial memory or peripheral interface.

Watch-Dog Timer. This circuit helps detect code runaway and helps minimize the negative effects thereof. A range of time-out values is available. The $\overline{\text{RESET}}$ pin can be forced Low at the terminal count of the Watch-Dog Timer.

Eight Channel Analog-To-Digital Converter. A 10-bit converter with eight multiplexed inputs.

Digital to Analog Converter. 10-bit resolution.

2KB of On-Chip RAM. Used for stack and other read/write operations.

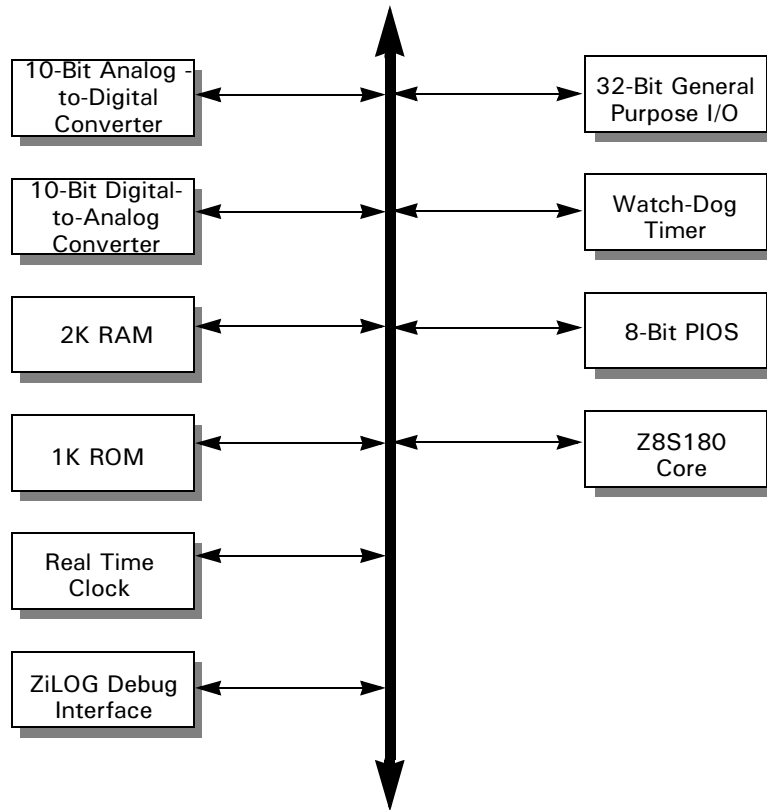


FIGURE 1. Z8S183 BLOCK DIAGRAM

PIN DESCRIPTIONS

Figure 2 illustrates the Z8S183 pinout. Table 2 describes the processor and device pins. Table 2 describes the Asynchronous Serial Communications Interface (ASCI) and Clocked Serial I/O (CSI/O) pins. Table 3 describes the Port and Programmable I/O Sequencer (PIOS) pins. Table 4 describes the analog pins.

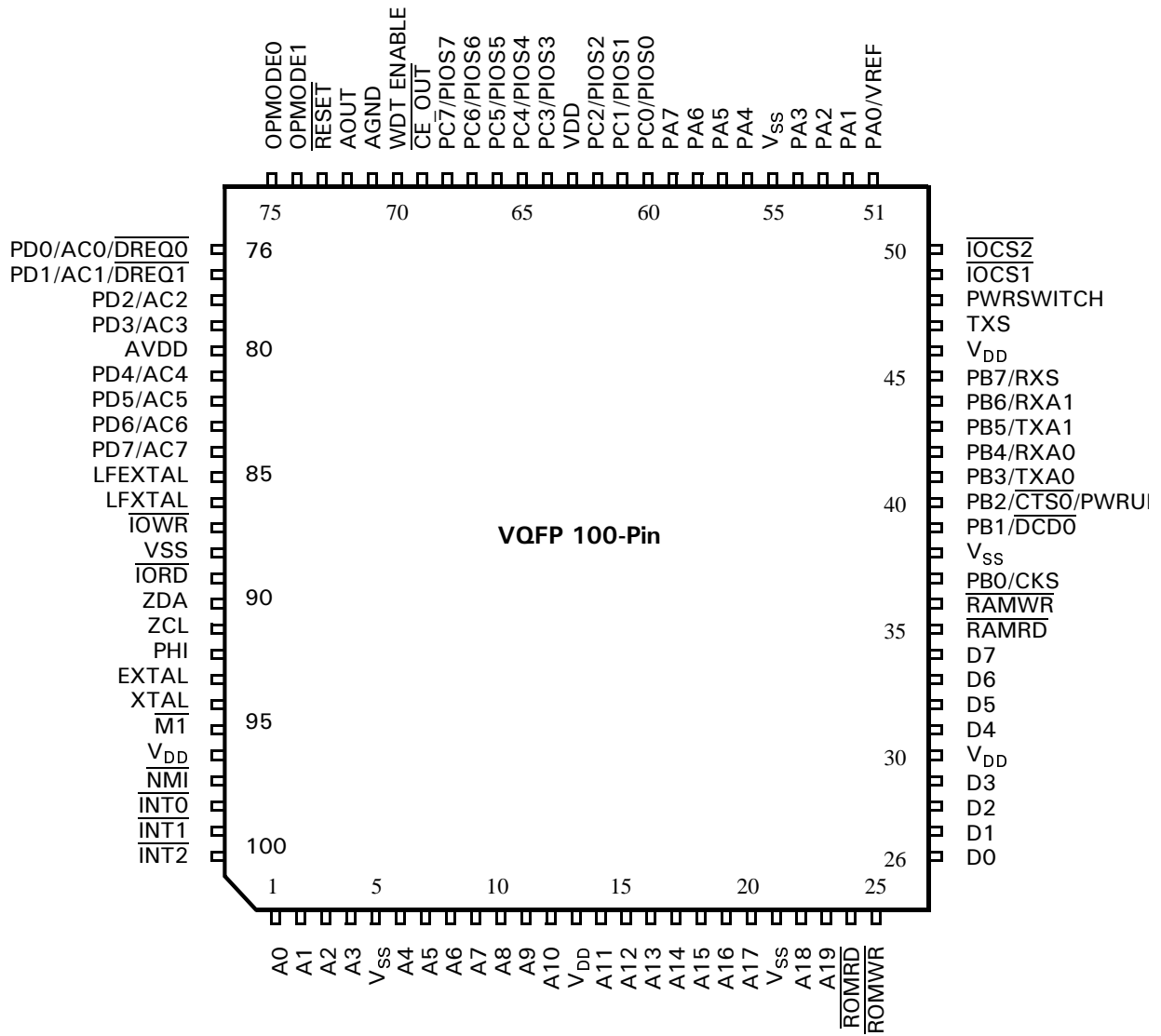


FIGURE 2. Z8S183 PIN DESCRIPTION

TABLE 1. PROCESSOR AND DEVICE PIN DESCRIPTIONS

| Symbol | Pin # | Function | Type | Description |
|-----------------------------|--------------------------------------|------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A19-0 | 1-4, 6-12, 14- 20, 22-23 | Address Bus | Bidirectional, 3-state | These lines select a location in memory or I/O space to be read or written. The Z8S183 does not drive these lines during Reset nor external bus acknowledge cycles. Drive is optional during LOW-POWER modes. |
| $\overline{\text{CE_OUT}}$ | 69 | Chip Enable Out | Open drain output | An open-drain output, with no internal pull-up. Can be used to enable an external RAM, as described in "Using the Power Control Register" on page 30. |
| D7-0 | 26- 29, 31-34 | Data Bus | Bidirectional, 3-state | These lines transfer information to and from I/O and memory devices. The Z80S183 drives these lines only during write cycles. |
| EXTAL | 93 | Oscillator or Clock In | Input | This pin can be connected to a crystal or to an external clock. If a crystal is used, this signal is not a logic level. |
| $\overline{\text{INT0}}$ | 98 | Interrupt Request 0 | Input, active Low | This signal can be driven Low in an open-drain fashion by external I/O devices. The processor responds to this request at the end of the current instruction cycle if it is enabled, and the $\overline{\text{NMI}}$ and $\overline{\text{BUSREQ}}$ signals are inactive. This pin can only be used in Z80/180 Mode 1, in which the processor acknowledges this request by interrupting to location 0038H. |
| $\overline{\text{INT1-2}}$ | 99, 100 | Interrupt Requests 1-2 | Inputs, active Low or edge-triggered | These signals are generated by external devices. The processor acknowledges a request on one of these lines at the end of the current instruction cycle, so long as the $\overline{\text{NMI}}$, $\overline{\text{BUSREQ}}$, and $\overline{\text{INT0}}$ signals are inactive. The processor acknowledges one of these requests with an internal cycle, in which a fixed vector corresponding to one of the pins is used to select an interrupt service routine. These pins may be programmed for active Low level, rising or falling edge interrupts. The state of the external $\overline{\text{INT1}}$ and $\overline{\text{INT2}}$ pins can be read in the Interrupt Edge Register. |
| $\overline{\text{IOCS1-2}}$ | 49, 50 | I/O Chip Selects 1-2 | Outputs, active Low | If bit 2 of the System Control register (SCR) is 1, IOCS1 goes Low for accesses to I/O addresses 0080-87H, and IOCS2 goes Low for accesses to 0088-8FH. |

TABLE 1. PROCESSOR AND DEVICE PIN DESCRIPTIONS (CONTINUED)

| Symbol | Pin # | Function | Type | Description | | | | | | | | | | | | | | | |
|---------------------------------|--------|-----------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------|----------------|---|---|--------|---|---|----------|---|---|---------------------|---|---|-----------------------------------------|
| $\overline{\text{IOR}}\text{D}$ | 89 | I/O Read | Output, active Low, 3-state | $\overline{\text{IOR}}\text{D}$ Low indicates that the Z80S183 is reading data from a location in I/O space. The addressed I/O device uses this signal to gate data onto the processor data bus. The Z80S183 does not drive this line during Reset, nor during bus acknowledge cycles. | | | | | | | | | | | | | | | |
| $\overline{\text{IOW}}\text{R}$ | 87 | I/O Write | Output, active Low, 3-state | $\overline{\text{IOW}}\text{R}$ indicates that bits D7–0 hold valid data to be stored at the addressed I/O location. The Z80S183 does not drive this line during Reset, nor during bus acknowledge cycles. | | | | | | | | | | | | | | | |
| LFExtal | 85 | Low-Frequency Crystal or Clock | Input | A low-frequency crystal or line frequency input can be connected to this pin, for use by the real time clock. A low-frequency crystal can be also used with the FLL for a system clock. | | | | | | | | | | | | | | | |
| LFXTAL | 86 | Low Frequency Crystal | Output | A low-frequency crystal can be connected to this pin, for use by the Real Time Clock or as the system clock. | | | | | | | | | | | | | | | |
| $\overline{\text{M}}\text{1}$ | 95 | Machine cycle 1 | Output, active Low | Together with $\overline{\text{ROM}}\text{RD}$ or $\overline{\text{RAM}}\text{RD}$, $\overline{\text{M}}\text{1}$ Low indicates that the current cycle is the fetch cycle of an instruction execution. | | | | | | | | | | | | | | | |
| $\overline{\text{NMI}}$ | 97 | Nonmaskable Interrupt | Input, falling-edge active | $\overline{\text{NMI}}$ has a higher priority than $\overline{\text{INT}}\text{0}$ and is always recognized at the end of an instruction, regardless of the state of the interrupt enable flip-flops. This signal forces processor execution to location 0066H. This input includes a Schmitt trigger to allow RC rise times. | | | | | | | | | | | | | | | |
| OPMOD2–1 | 74–75 | Operating Mode Select 2,1 | Inputs | These pins select the basic operating mode of the Z80S183. A rising edge on OPMOD1 makes the Z80S183 generate a Power On Reset (POR). <table border="0" style="margin-left: 20px;"> <tr> <td>OPMOD1</td> <td>OPMOD2</td> <td>Operating Mode</td> </tr> <tr> <td>L</td> <td>L</td> <td>Normal</td> </tr> <tr> <td>L</td> <td>H</td> <td>Reserved</td> </tr> <tr> <td>H</td> <td>L</td> <td>Internal RO enabled</td> </tr> <tr> <td>H</td> <td>H</td> <td>$\overline{\text{BUSREQ}}$ is generated</td> </tr> </table> | OPMOD1 | OPMOD2 | Operating Mode | L | L | Normal | L | H | Reserved | H | L | Internal RO enabled | H | H | $\overline{\text{BUSREQ}}$ is generated |
| OPMOD1 | OPMOD2 | Operating Mode | | | | | | | | | | | | | | | | | |
| L | L | Normal | | | | | | | | | | | | | | | | | |
| L | H | Reserved | | | | | | | | | | | | | | | | | |
| H | L | Internal RO enabled | | | | | | | | | | | | | | | | | |
| H | H | $\overline{\text{BUSREQ}}$ is generated | | | | | | | | | | | | | | | | | |
| PHI | 92 | System Clock | Output | This output is the Z80S183’s master clock, and is provided for use by external logic. The frequency of this clock may be equal to or half of the crystal or input clock frequency, depending on an internal register bit. | | | | | | | | | | | | | | | |

TABLE 1. PROCESSOR AND DEVICE PIN DESCRIPTIONS (CONTINUED)

| Symbol | Pin # | Function | Type | Description |
|---------------------------|-----------------------------|--------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PWRSWITCH | 48 | Power Switch | Output | This pin is a positive logic output controlled by bit 5 of the Power Control register. Bit 5 resets to 0, so this pin goes Low at any Reset, including one initiated by a rising edge on PWRUP or OPMOD1. See "Using the Power Control Register" on page 30. |
| PWRUP | 40 | Power Up | Input | If bit 6 is 1 in the Power Control Register, a rising edge on PB2/CTS0/PWRUP resets the part, as described in "Using the Power Control Register" on page 30. |
| $\overline{\text{RAMRD}}$ | 35 | RAM Read | Output, active Low, 3-state | $\overline{\text{RAMRD}}$ Low indicates that the processor reads data from a memory location in the address range of external RAM. The Z80S183 does not drive this line during Reset, nor during bus acknowledgment. |
| $\overline{\text{RAMWR}}$ | 36 | RAM write | Output, active Low, 3-state | $\overline{\text{RAMWR}}$ Low indicates D7–D0 hold data to be stored at a memory location in the address range of external RAM. The Z80S183 does not drive this line during Reset, nor during bus acknowledgement. |
| $\overline{\text{RESET}}$ | 73 | Master Reset | Input/Output, active Low | This signal initializes the Z80S183 and other devices in the system. This input must be Low for a minimum of six system clock cycles, and is held Low until the clock is stable. $\overline{\text{RESET}}$ can be programmed as an output, allowing Z80S183 to reset external devices. The Power On Reset and Watch-Dog Timer (if enabled) blocks perform a global reset by forcing $\overline{\text{RESET}}$ Low. This input includes a Schmitt trigger to allow RC rise times. |
| $\overline{\text{ROMRD}}$ | 24 | ROM Read | Output, active Low, 3-state | $\overline{\text{ROMRD}}$ Low indicates that the processor wants to read data from a memory location in the address range of external ROM/Flash. The Z80S183 does not drive this line during reset, nor during bus acknowledge cycles. |
| $\overline{\text{ROMWR}}$ | 25 | ROM Write | Output, active Low, 3-state | $\overline{\text{ROMWR}}$ Low indicates that D7–D0 hold a byte to be stored at the addressed memory location in the address range of external Flash memory. The Z80S183 does not drive this line during Reset, nor during bus acknowledge cycles. |
| V _{DD} | 13, 30, 46, 63, 96 | Power Supply | | These pins carry power to the device. They must be tied to the same voltage externally. |

TABLE 1. PROCESSOR AND DEVICE PIN DESCRIPTIONS (CONTINUED)

| Symbol | Pin # | Function | Type | Description |
|-----------------|-------------------|------------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| V _{SS} | 5, 21, 38, 55, 88 | Ground | | These pins are the ground references for the device. They must be tied to the same voltage externally. |
| WDT Enable | 70 | Watch-Dog Timer Enable | Input | If this pin is High, the Watch-Dog Timer cannot be disabled by software nor by Reset. If this pin is Low at power-up, these three events occur: <ul style="list-style-type: none"> – A Power On Reset disables the Watch-Dog Timer. – A WDT Reset does not change the status of the WDT. – Software can enable or disable it as needed. |
| XTAL | 57 | Crystal | Input/Output | Crystal oscillator connection. This pin is left open if an external clock is used instead of a crystal. This pin does not carry a logic level. (See “DC Characteristics” on page 165.) |
| ZCL | 91 | ZDI Clock | Input | The clock for the ZiLOG Debugging Interface. This input includes a Scmitt trigger. |
| ZDA | 90 | ZDI Data | Input/output, open drain | Data for the ZiLOG Debugging Interface. This input includes a Scmitt trigger. |

TABLE 2. UART AND CSI/O PIN DESCRIPTIONS

| Symbol | Pin # | Function | Type | Description |
|--------------------------|--------|-----------------------|-------------------|---------------------------------------------|
| $\overline{\text{CTS0}}$ | 40 | Clear To Send 0 | Input, active Low | Transmit control signal for ASCI channel 0. |
| $\overline{\text{DCD0}}$ | 39 | Data Carrier Detect 0 | Input, active Low | Receive control signal for ASCI channel 0. |
| RXA0–1 | 42, 44 | Receive Data 0,1 | Inputs | ASCI Receive data. |
| RXS | 45 | CSI/O Receive Data | Input | Receive data for the CSI/O channel. |
| TXA0–1 | 41, 43 | Transmit Data 0, 1 | Outputs | ASCI Transmit data. |
| TXS | 47 | CSI/O Transmit Data | Output | Transmit data from the CSI/O channel. |

TABLE 3. PORT AND PIOS PIN DESCRIPTIONS

| Symbol | Pin # | Function | Type | Description |
|--------|-----------------|----------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PA7-0 | 51-54, 56-59 | Port A | Input/outputs | These pins can be configured as inputs or outputs, with or without level-sensitive, active Low interrupt request capability, on a bit-by-bit basis. |
| PB7-0 | 37, 39-45 | Port B | Input/outputs | These pins can be configured as port inputs or outputs, or ASCII or CSI/O signals, on a bit-by-bit basis. |
| PC7-0 | 60-62, 64-68 | Port C | Input/outputs | These pins can be configured as port inputs or outputs, or Programmable Input/Output Sequencer outputs, on a bit-by-bit basis. Also, PC0 can be used as a 50 or 60 Hz time base for the real time Clock. |
| PD7-0 | 76-79, 81-84 | Port D | Input/outputs | These pins can be configured as inputs or outputs, with or without level-sensitive, active Low interrupt request capability, or as inputs to the A/D converter, on a bit-by-bit basis. PD0 and PD1 can also act as $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ respectively. |

TABLE 4. ANALOG PIN DESCRIPTIONS

| Symbol | Pin # | Function | Type | Description |
|--------|-----------------|---------------------------|--------|----------------------------------------------------------|
| AC7-0 | 76-79, 81-84 | Analog Channels | Inputs | Inputs to the Analog to Digital converter. |
| AGND | 71 | Analog Ground | | Reference ground for the analog circuitry. |
| AOUT | 72 | Analog Output | Output | The output of the Digital to Analog converter. |
| AVDD | 80 | Analog Power | | Power for the analog circuitry. |
| VREF | 51 | External Analog Reference | Input | External analog reference voltage, multiplexed with PA0. |

OPERATIONAL DESCRIPTION

This section describes, using text, tables, and figures, how the various parts of the Z80S183 operate. This description is presented from the processor outward to the peripherals. In the latter parts of this section, refer to the corresponding section of "I/O Registers" on page 70, which present the Z80S183's I/O registers. Cross-reference links are included in both sections to aid these references.

PROCESSOR DESCRIPTION

The Z80S183 is an 8-bit microprocessor that performs certain 16-bit operations. In both data sizes, the processor includes an accumulator. Register A is the accumulator for 8-bit operations, and the HL register pair is the accumulator for 16-bit operations.

Processor Program Registers

In addition to register A, there are six more 8-bit registers named B, C, D, E, H, and L, which can also be operated on as 16-bit register pairs BC, DE, and HL. Flag register F completes the basic register bank.

Two of these basic register banks are included in all Z80 and Z180 processors. High-speed exchange between these banks can be used by a program internally, or one bank can be allocated to the mainline program and the other to interrupt service routines.

Finally, two Index registers IX and IY allow *base and displacement* addressing in memory. IX and IY are not included in the register banks on the Z80 and Z180; there is only one copy of each.

Memory Management Unit (MMU)

To the 16-bit, 64 KB memory addressing capability of the Z80, all Z180 processors add a Memory Management Unit (MMU) that expands the addressing capability to 20 bits (1 MB). With the MMU, the 64 KB logical addressing space can be divided into one to three areas of programmable size and location in the 1-MB physical memory space.

I/O Space

A separate I/O space includes on-chip and off-chip peripheral devices. On the Z80, I/O space included 8-bit addresses and 256 bytes. All Z180 processors feature an expanded I/O space with 16-bit addresses and 64 KB. The Z80S183 includes an extensive set of on-chip peripherals in I/O space, which can be augmented by external peripherals.

Processor Control Registers

In addition to the data-oriented registers described above, the Z80S183 processor includes several other control registers. Unlike the registers in I/O space that are described in Section 4, these control registers have no addresses, but are used implicitly in certain processor operations.

Program Counter (PC). This 16-bit register tracks program execution by the processor, which automatically increments PC while fetching instructions. The processor stores PC on the stack when it executes a CALL or RST instruction, or an interrupt or TRAP occurs. The processor loads PC with a new value when it executes a JUMP, CALL, RST, or RET instruction, and when an interrupt, Trap, or Reset occurs. PC resets to 0000.

Stack Pointer (SP). The processor decrements this 16-bit register by 2, and stores a 16-bit value in memory at this updated address, when it executes a PUSH, CALL, or RST instruction, and when an interrupt or Trap occurs. The processor fetches a 16-bit value from memory at the address in SP, and then increments SP by 2, when it executes a POP, RET, RETI, or RETN instruction. Software can store the value in SP in memory, load SP from memory or another register, or load it with a constant/immediate value. Further, software can add or subtract the value in SP to or from another register, and can increment or decrement SP. Finally, software can exchange the 16-bit value in memory, to which SP currently points, with the contents of a 16-bit register. SP resets to 0000B.

Flags (F). The processor includes two Flag registers each containing six bits, named Zero (Z), Carry (CF), Sign (S), Parity or Overflow (P/V), Half-Carry (HC) and Add/Subtract (N). Certain flags are automatically updated as part of executing certain instructions. Subsequent instructions can then use the flags, either as an operand (ADC, SBC, DAA), or to determine whether to perform a JUMP, CALL, or RET operation. The flags can be saved on the stack with a PUSH instruction, or restored from the stack with a POP instruction. The two sets of flag registers are paired with the two A accumulators; the current pair is toggled by the EX AF, AF' instruction.

Interrupt High Address (I). The contents of this register are used as the eight high-order address bits, when the processor fetches the address of an interrupt service routine from memory, for an interrupt from the INT1 or INT2 pin, or from an on-chip peripheral. The I register points at a table of interrupt service routine addresses, that starts at a 256-byte boundary in the 64 KB logical address space. The I register resets to 0, and can be read or written by the dedicated instructions LD A, I and LD I, A.

R Counter (R). On the Z8018x family processors this register contains a count of executed fetch cycles. R resets to 0, and can be read or written by the dedicated instructions LD A, R and LD R, A.

Observing Read Data from On-Chip Devices

Bit 7 of the Output Control Register (OCR, illustrated on page 77) determines whether the Z80S183 drives data from on-chip ROM, RAM, and I/O registers, onto the D7–D0 pins for debugging and monitoring purposes. When this bit is 0, as it is after a reset, the D7–D0 pins remain in high-impedance state during read cycles from on-chip devices, saving power.

If software sets this bit to 1 during device initialization, the Z80S183 drives read data from on-chip devices onto D7–D0, allowing it to be captured by debugging instruments such as logic analyzers.

Illegal Instruction Traps

Like most processors, the defined instruction set for the Z8018x family does not fully cover all possible sequences of binary values. The Op Code maps, in the section, “Op Code Map” on page 157, include numerous blank cells. These cells

represent Op Code sequences for which no operation is defined, and are commonly called *illegal* instructions.

When a Z80S183 or other Z8018x processor fetches one of these sequences, it performs a Trap sequence as follows:

1. The Trap bit is 1 in the Interrupt/Trap Control register.
2. The UFO bit in the Interrupt/Trap If the register is set to 0 if the processor detected the condition while fetching the second byte of the instruction. The UFO bit clears to 1 if in the Interrupt/Trap Control register. If the condition was detected while fetching the third byte, UFO is 1.
3. The processor decrements the Stack Pointer (SP) by 2 and stores the 16-bit logical address from PC, in memory at the new SP address. This address points to the last byte of the illegal Op Code sequence.
4. The processor then clears PC and resumes execution at logical address 0000.

Trap Handling. The code at logical address 0000B can optionally store the value of SP in memory, and then set SP to an area of memory dedicated to its private stack.

In all cases, the trap-handling routine stores as many registers among AF, BC, DE, HL, IX, and IY as it may use, by pushing them onto the stack. A general-purpose routine stores all of these registers, those in the alternate set, the value of I and the state of the Interrupt Enable flag.

Next, the Trap-handling code distinguishes among the four events that can bring execution to address 0000B:

- A Reset
- A Trap
- An RST 0 instruction
- A program error, such as a JUMP to a null pointer

The code detects a Trap by reading the Interrupt/Trap Control register (ITC) and checking bit 7 (Trap). If bit 7 is 1, a Trap has occurred, and the code handles it as follows:

1. Clears the Trap bit by writing a 0 to bit 7 of the ITC
2. Fetches the PC value stored on the stack
3. Examines bit 6 of the ITC (UFO).
4. Decrements the PC value by 1, if the UFO bit is 0; otherwise, decrements it by 2, so that it points to the start of the illegal instruction.

The next action of the trap handling routine depends on the application and its stage of development.

Extending the Instruction Set. Core software can use illegal instructions as extensions to the Z8018x instruction set. To accomplish this, the trap handler must fetch and examine each illegal instruction. If an illegal instruction is an extension,

the trap handler performs the extended operation that the instruction indicates. It then advances the stacked PC value over the instruction, restores the saved register values, and returns to the next instruction.

Error Message vs. Restart. Except for these extended instructions, the trap handling software can perform either of the following actions:

- Output an error message and wait for someone to examine the situation and restart the application
- Attempt to restart the application immediately

The former course is more common in the debugging/development stages of an application, while the latter may be more appropriate in the production/deployment stage. In the latter case, software may log the event for future readout, using an external storage medium or just in memory.

DEVICE AND VERSION ID REGISTERS

Three registers (described on page 74), allow software to determine if it is operating on a Z80S183, as well as the device version. These registers are a mandatory feature of the ZDI interface that is described in “ZiLOG Debug Interface” on page 180, and are also available in I/O space.

I/O addresses 003BH and 003CH read as 01 and 00 respectively, indicating that the Z80S183 was one of the first devices to incorporate a ZDI interface. I/O address 003DH reads as 00 for revision AB, 01 for revision BA, and will feature higher values on future revisions.

INTERRUPTS

ZiLOG Z80 and Z180 processors have a rich legacy of sophisticated interrupt capabilities. Because of the lack of an I/O Request signal on the Z80S183, its interrupt subsystem is substantially simpler and easier to describe than those of other 8018x devices.

The following topics that are significant for other 8018x processors, do not apply to the Z80S183:

- $\overline{\text{INT0}}$ Mode 0 and 2 interrupts,
- Interrupt acknowledge cycles,
- Interrupt daisy chains,
- Interrupt Pending and Interrupt Under Service bits, and
- RETI instructions.

Interrupt Resources in the Z80S183

IEF1 and IEF2. These bits are internal to the processor and can only be affected and manipulated by certain specific events:

- A Reset clears IEF1 and IEF2

- An EI instruction sets IEF1 and IEF2
- A DI instruction clears IEF1 and IEF2
- An NMI sequence copies IEF1 to IEF2, then clears IEF1
- A maskable interrupt clears IEF1 and IEF2
- An LD A, I or LD A, R instruction copies IEF2 to the P/V flag
- An RETN instruction copies IEF2 to IEF1

When IEF1 is 1, RESET and BUSREQ are both High, and no falling edge has occurred on NMI, the Z80S183 checks for maskable interrupt requests from external pins and on-chip peripherals, as it completes each instruction, or each instruction iteration for HALT, the block I/O instructions, block move instructions, and block scan instructions.

The I Register. The Z80S183 uses the contents of this register as A15–8 of the logical address for fetching interrupt service routine addresses from memory, in response to interrupt requests on INT1 and INT2, and from internal peripherals.

See “Interrupt Registers”, which starts on page 80, for other registers associated with interrupts.

The IL Register. The Z80S183 uses bits 7-5 of this register as A7–5 of the logical address for fetching interrupt service routine addresses from memory, in response to interrupt requests on INT1 and INT2, and from internal peripherals.

The Interrupt/Trap Control Register (ITC). 2-0 of the ITC are individual Enable bits for the INT2, INT1, and INT0 pins, respectively. They reset to 001B, so that requests on INT0 can be enabled by an EI instruction after Reset.

Interrupt Edge Register (IER). By reading this register, software can sense the current state of the INT2 and INT1 pins, and whether an edge has been detected on each. Other bits in the IER select whether each of these pins is low-level sensitive, or rising- and/or falling-edge sensitive.

Nonmaskable Interrupt (NMI)

The Z80S183 latches falling edges on the NMI pin. A falling edge clears the DME bit in the DMA Status register (DSTAT), disabling the on-chip DMA channels. Only a Low on RESET or on BUSREQ takes precedence over NMI. Unless RESET or BUSREQ is Low, the Z80S183 checks for a falling edge on NMI as it completes each instruction (each instruction iteration of HALT, the block I/O instructions, block move instructions, and block scan instructions), and performs an NMI sequence if a falling edge has occurred.

An NMI sequence includes 4 steps:

1. The processor copies the state of the IEF1 bit to IEF2.
2. It clears IEF1 to prevent maskable interrupts.
3. It decrements SP by 2, and stores the logical address in the PC in memory at the new address in SP. For most interrupts, this value is the address of the

instruction the processor would have executed next, had no interrupt occurred. If the processor was stopped by HALT or SLP, this value is the address of the next instruction. In the event of an incomplete block transfer, block scan, or block I/O instruction, this value is the address of the instruction.

4. The processor loads 0066H into PC, and resumes execution from that logical address.

NMI Handling. NMI routines fall into two categories, based on whether the external hardware that drives $\overline{\text{NMI}}$ is capable of producing another falling edge on the pin, before the NMI service routine has completed its execution and returned to the interrupted process. The case when the NMI is not capable of producing another falling edge is called *Single Edge Guaranteed*. The case when the NMI can produce another falling edge is called *Repeated Edge Possible*. Debug monitors, which may display the state of the interrupt process, fall into the Repeated Edge category.

Single Edge Guaranteed. An NMI routine in this category is similar to other interrupt service routines. This routine has the option of storing the contents of SP in memory and loading SP with the address of a memory area that is dedicated for the stack. In any case, this routine stores as many of the registers as it may use during its execution.

Repeated Edge Possible. An NMI routine in this category starts with a PUSH AF instruction, then LOAD A from a dedicated location in memory that indicates whether the interrupted process is the NMI routine. If this location indicates that the process is the NMI routine, it immediately performs a POP AF and then a RETN instruction, to return to its former execution.

If the *in NMI* location is cleared, software sets it to 1. Then, if the NMI routine performs either of the following:

- Places a DI instruction in a *Save The Registers* routine that it shares with other means of entry, or
- Displays the I register or the interrupt-enable state of the interrupted process, and allows a user/programmer to change these (in essence, a debug monitor)

it performs LD A, I and PUSH AF instructions. This stores the I register at the address in SP plus one, and the interrupt enabled state (IEF2) in the P/V flag and in bit 2 of the memory location pointed to by SP.

If the NMI routine uses a common *Save The Registers* subroutine that it shares with other entry points, the save subroutine can perform a DI instruction to prevent interruption by maskable interrupts.

The NMI routine has the option to store the SP value in a dedicated location in memory, and load SP with the address of a dedicated NMI stack area.

In any case, the NMI routine must PUSH as many other registers as it uses. A debug monitor typically performs PUSH operations on all registers in both banks, so that it can display them.

Reenabling The DMA Channels. In an NMI service routine in an application that uses the DMA channels, software next reads the DSTAT register and reenables any DMA operation that was in progress, as described in the in “NMI and DME” on page 42.

Exiting The NMI Routine. On completion of its processing, an NMI routine restores the saved registers. If the routine used its own stack area, it then restores the SP value of the interrupted process. If the routine set an in NMI memory location on the way in, it clears this location to 0.

NMI routines that did not save the I register and IEF2 state at the start, can conclude with POP AF and RETN instructions. RETN copies the state of IEF2 back into IEF1, restoring the interrupt enable state of the interrupted process.

NMI routines which saved I and IEF2 at the start, conclude with a POP AF for the saved I register and IEF2 bit. Then an LD I,A, followed by a JP V to a POP AF, EI, RET sequence. The JP is followed by LD I , A, POP AF, and RET instructions.

INT0 Mode 1

The Z80S183 can only handles interrupts requested on the INT0 pin in Mode 1. All Z80S183 applications that enable interrupts and do not tie INT0 High, must include an IM 1 instruction before the first EI instruction.

The Z80S183 performs an INT0 interrupt sequence at the end of an instruction (each instruction iteration for HALT, the block I/O instructions, block move instructions, and block scan instructions), if all of the following are true:

- INT0 is Low
- Bit 0 of the Interrupt/Trap Control register is 1 to enable INT0
- The IEF1 bit is set to 1 to enable interrupts in general general
- RESET and BUSREQ are both High, and a negative edge on NMI has not been detected

When all of these conditions occur simultaneously, the Z80S183 responds as follows:

1. It clears IEF1 and IEF2 to prevent further interrupts.
2. It decrements SP by 2, and stores the contents of PC in memory at the new address in SP. This value is typically the address of the instruction the processor would have executed next, if no interrupt had occurred. If the processor was stopped by HALT or SLP, this value is the address of the next instruction. In the event of an incomplete block transfer, block scan, or block I/O instruction, this value is the address of the instruction.
3. It loads 0038H into PC, and resumes execution from that logical address.

Interrupt Handling. Any Interrupt Service Routine (ISR) has the initial option of saving the contents of SP in memory, and loading SP with the address of a

memory area that is dedicated to its stack. Most interrupt service routines do not use this option.

An $\overline{\text{INT0}}$ ISR must save the contents of the registers it uses, using PUSH and/or EX AF, AF' and EXX instructions.

If the application includes a mechanism for allowing nested interrupts, the ISR can begin as specified by that mechanism, leading to an IE instruction that allows the ISR to be interrupted by other interrupts. Most applications do not allow for nested interrupts.

The ISR next reads status registers from each device that can request an interrupt on $\overline{\text{INT0}}$, to identify the cause of the interrupt. The ISR must process each interrupting device according to this status, and the device and application requirements.

Many ISRs read data from interrupting device(s), or write data to interrupting device(s). In addition, the ISRs can write registers in these devices, to modify its mode, status, or operation.

When interrupt processing is complete, if nested interrupts were allowed, the ISR ends as specified by the nesting mechanism. If nested interrupts were not allowed, the ISR restores the saved registers and concludes with EI and RET instructions.

Note: the Z80 and Z80180 instruction sets include an RETI instruction, that is used for servicing Z80 peripherals. Since the Z80S183 includes no such peripherals, nor does it allow them to be connected externally, there is no reason to ever conclude a Z80S183 ISR with an RETI. RET is both shorter and faster than RETI, and fills the same function.

$\overline{\text{INT1}}$ and $\overline{\text{INT2}}$

The Z80S183 performs an $\overline{\text{INT1}}$ or $\overline{\text{INT2}}$ interrupt sequence at the end of an instruction (each instruction iteration for HALT, the block I/O instructions, block move instructions, and block scan instructions), if all of the following are true:

- $\overline{\text{INT1}}$ and/or $\overline{\text{INT2}}$ meets the condition specified for it in the Interrupt Edge Control register (low level, rising edge, falling edge),
- Bit 2 or 1 of the Interrupt/Trap Control register (ITC) is 1 to enable this pin (if both pins are enabled and both pins meet the specified condition, $\overline{\text{INT1}}$ takes precedence over $\overline{\text{INT2}}$),
- IEF1 is 1, to enable interrupts in general,
- $\overline{\text{INT0}}$ is High or bit 0 of the ITC is 0,
- $\overline{\text{RESET}}$ and $\overline{\text{BUSREQ}}$ are High, and
- A negative edge on $\overline{\text{NMI}}$ has not been detected.

When all of these conditions occur simultaneously, the Z80S183 responds as follows:

1. It clears IEF1 and IEF2 to prevent further interrupts.
2. It decrements SP by 2, and stores the contents of PC in memory at the new address in SP. Typically, this value is the address of the instruction the

processor would have executed next, if no interrupt had occurred. If the processor was stopped by `HALT` or `SLP`, this value is the address of the next instruction. In the event of an incomplete block transfer, block scan, or block I/O instruction, this value is the address of the instruction.

3. Next, the processor forms a logical memory address using the contents of the I register as A15–8, bits 7–5 of the IL register as A7–5, and 0 as A4–0 for $\overline{\text{INT1}}$ or 2 in A4–0 for $\overline{\text{INT2}}$.
4. Finally, the processor fetches a 16-bit logical address from memory at that logical address, loads it into PC, and resumes instruction execution from there.

INT1–2 Handling. All of the considerations noted for $\overline{\text{INT0}}$ ISRs in “Interrupt Handling” on page 16, also apply to ISRs for $\overline{\text{INT1}}$ and $\overline{\text{INT2}}$. One additional step is required if the pin is edge-triggered: read the INT2-1 Interrupt Edge Register (IECR), and write the value (including 1 in bit 5 or 4) back to the IECR to clear the edge-detection logic.

On-Chip Interrupts

The Z80S183 performs an interrupt sequence for an on-chip device at the end of an instruction (each instruction iteration for `HALT`, the block I/O instructions, block move instructions, and block scan instructions), if all of the following are true:

- An interrupting condition has occurred in the device,
- That condition is interrupt-enabled in the device’s registers,
- IEF1 is 1, to enable interrupts in general,
- No higher-priority internal device is requesting an interrupt (see Table 5 below for the relative priorities of internal devices),
- Neither $\overline{\text{INT1}}$ nor $\overline{\text{INT2}}$ is interrupting,
- $\overline{\text{INT0}}$ is High or bit 0 of the ITC is 0,
- $\overline{\text{RESET}}$ and $\overline{\text{BUSREQ}}$ are both High, and
- A negative edge on $\overline{\text{NMI}}$ has not been detected.

When all of these conditions occur simultaneously, the Z80S183 responds as follows:

1. It clears IEF1 and IEF2 to prevent further interrupts.
2. It decrements SP by 2, and stores the contents of PC in memory at the new address in SP. Typically, this value is the address of the instruction the processor would have executed next, if no interrupt had occurred. If the processor was stopped by a `HALT` or `SLP` instruction, this value is the address of the next instruction. For an incomplete block transfer, block scan, or block I/O instruction, this value is the address of the instruction.

3. Next the processor forms a logical memory address using the contents of the I register as A15–8, bits 7–5 of the IL register as A7–5, and the value corresponding to the interrupting device as A4–0.
4. Finally, the processor fetches a 16-bit logical address from memory at that logical address, loads it into PC, and resumes instruction execution from there.

On-Chip Interrupt Handling. The only difference between handling an on-chip interrupt, and the considerations noted for $\overline{INT0}$ ISRs in “Interrupt Handling” on page 16, is that the ISR for an on-chip device never needs to differentiate among several devices connected to an INT pin, only among interrupt sources within the device.

TABLE 5. INTERRUPT OFFSETS AND PRIORITIES

| Device | Priority | A4–0 Offset |
|-----------------------------------|----------|-------------|
| INT1 pin | highest | 0 |
| INT2 pin | | 2 |
| PRT0 | | 4 |
| PRT1 | | 6 |
| DMA0 | | 8 |
| DMA1 | | 10 = 0AH |
| CSI/O | | 12 = 0CH |
| ASCI0 | | 14 = 0EH |
| ASCI1 | | 16 = 10H |
| Programmable I/O Sequencer (PIOS) | | 18 = 12H |
| Port A | | 20 = 14H |
| Port D | | 22 = 16H |
| A/D Converter | | 24 = 18H |
| Real Time Clock (RTC) | | 26 = 1AH |
| Reserved | | 28 = 1CH |
| ZiLOG Debug Interface (ZDI) | lowest | 30 = 1EH |

NOTE: Devices are ordered identically with respect to interrupt priority and offset value

MEMORY

Z8018x family processors include a 64 KB logical memory space in which software operates, and a 1 MB physical memory address space in which on-chip and external memory reside. The Memory Management Unit (MMU) translates 16-bit logical addresses to 20-bit physical addresses dynamically, as part of each memory access.

Memory Structure

On the Z80S183, memory is divided into four categories:

- 1 KB of on-chip ROM

- 2 KB of on-chip RAM
- External ROM or Flash memory using on-chip decoding
- External RAM using on-chip decoding

Table 28 on page 79 describes the System Configuration Register (SCR), which includes bits that enable or disable each of these four memory categories. Software designers must be cautious when programming this register, to *not* disable the memory in which the current code sequence resides.

On-chip ROM can be enabled or disabled at Reset time, by the state of the OPMODE1–0 pins. These pins control the initial state of the on-chip ROM Enable bit in the SCR. If on-chip ROM is enabled, it occupies physical addresses 00000–003FFH.

The last 256 bytes of on-chip RAM are always accessible to the Programmable I/O Sequencer (PIOS) module. If processor access to on-chip RAM is enabled, another bit in the SCR controls whether A19–16 are included in address decoding for on-chip RAM.

The on-chip address decoder for external ROM or Flash memory decodes from physical address 00000 through a programmable upper limit. When external ROM is enabled, memory accesses at addresses below the upper limit, drive the $\overline{\text{ROMRD}}$ or $\overline{\text{ROMWR}}$ pin Low. (Among the capabilities of the $\overline{\text{ROMWR}}$ pin are programming of Flash memories.)

The on-chip address decoder for external RAM decodes between programmable lower and upper limits. When external RAM is enabled, memory accesses at addresses between these two limits, drive the $\overline{\text{RAMRD}}$ or $\overline{\text{RAMWR}}$ pin Low.

Note: If software programs the Memory Chip Select Logic and System Configuration Register so that some addresses do not match either the ROM or RAM chip selection, accesses to these addresses do not appear on the external bus. Avoid this possibility by programming the active chip selects to cover the entire memory address space.

Addressing Modes

Instructions can specify a memory address in several ways. Z80S183 addressing modes include:

Relative Addressing. JR and DJNZ instructions include a signed 8-bit displacement that specifies a range of addresses -126 to $+129$ from the Op Code, to which program control can be transferred.

Direct Addressing. In this mode, instructions include a 16-bit logical address.

Register Indirect Addressing. In this mode, the address is taken from one of the register pairs BC, DE or HL.

Indexed Addressing. In this mode, instructions include an 8-bit signed displacement from the address in an index register IX or IY.

Other contexts in which memory is accessed include instruction fetching, interrupts, DMA operations, and cycles generated by external masters while BUSACK is Low.

Memory Management Unit (MMU)

The MMU translates the 16-bit addresses used by software, called *logical* addresses, into 20-bit *physical* addresses, as part of all memory accesses performed by the processor. The MMU has no effect on accesses performed by the DMA channels, which include 20-bit address registers. It also has no effect on addresses in I/O space, which always have A19–16 0.

The MMU resets to a state in which it has no effect on addresses in processor cycles, passing A15–0 through without change and keeping A19–16 0. If an application needs 64 KB of memory or less, it ignores the MMU.

Even when the MMU has been programmed to perform active address transactions, it passes A11 0 from the logical to the physical address. The MMU manages memory in 4 KB blocks.

The section titled “MMU Registers” on page 82, describes the registers associated with the MMU.

MMU Operation. The MMU compares bits 15–12 of each logical address to two 4-bit fields in its Common/Base Address Register (CBAR), in an unsigned manner.

If bits 15–12 of a logical address are less than the value in bits 3–0 of the CBAR, the MMU considers the address to be in Common Area 0. For these addresses, it passes bits 15–12 to the A15-12 pins unchanged, and sets pins A19–16 to 0.

If bits 15–12 of a logical address are greater than or equal to the value in bits 3–0 of the CBAR, but are less than the value in bits 7–4 of the CBAR, the MMU considers the address to be in the Bank Area. For such addresses, it adds the value in its 8-bit Bank Base Register (BBR) to bits 15–12 of the logical address, and outputs the 8-bit sum on pins A19–12.

If bits 15–12 of a logical address are greater than or equal to the value in bits 7–4 of the CBAR, the MMU considers the address to be in Common Area 1. For such addresses, it adds the value in its 8-bit Common Base Register (CBR) to bits 15–12 of the logical address, and outputs the 8-bit sum on A19–12.

Note: The value in bits 7–4 of the CBAR must never be less than the value in bits 3–0 of the CBAR.

MMU Configurations. In the general case, the MMU divides the 64 KB logical memory space into three parts, with Common Area 0 located at the start of the 1 MB physical address space, and the Bank Area and Common Area 1 relocatable to other parts of the physical address space. These three parts are under control of the Bank Base Register and Common Base Register, respectively.

Certain combinations of values in the CBAR result in the logical address space being divided into fewer active areas:

- If the CBAR contains 0, all logical addresses fall into Common Area 1, and are relocated to a contiguous 64 KB area starting at the address in the CBR times 4096.
- If CBAR3–0 are 0 but CBAR7–4 are non-zero, the Bank Area and Common Area 1 are active. Logical addresses less than $(CBAR7-4) * 4096$ are relocated by the Bank Base Register, while other addresses are related by the Common Base Register.
- If CBAR7–4 and CBAR3–0 are equal and not 0, Common Area 0 and Common Area 1 are active. Logical addresses less than $(CBAR3-0) * 4096$ are not relocated, and map to the start of physical memory. Other addresses are relocated by the Common Base Register.

The MMU After Reset. Because the CBAR resets to 11110000B, logical addresses 0000–EFFFH are in the Bank Area and F000–FFFFH are in Common Area 1 after Reset. But since the BBR and CBR both reset to 0, the MMU passes all logical addresses through without change, with A19–16 all 0.

On-Chip ROM

Bit 7 in the System Configuration Register (page 79) controls whether physical addresses 00000–003FFH access on-chip ROM or external memory. A 1 in this bit enables on-chip ROM. At reset, this bit is set to 1 if the OPMOD1 pin is High and the OPMOD2 pin is Low. Otherwise, this bit is cleared to 0.

On-Chip RAM

Bits 6–5 in the System Configuration Register control processor access to on-chip RAM. If bit 6 is 0, on-chip RAM is disabled. If bits 6–5 are 10, on-chip RAM does not decode bits 19–16 of physical addresses, and responds to all physical addresses with A15–11 all 1: xF800 through xFFFFH. If bits 6–5 are 11B, on-chip RAM responds to physical addresses with A19–11 all 1: addresses FF800–FFFFFH.

The Programmable I/O Sequencer (PIOS) can always read the last 256 bytes of on-chip RAM, regardless of bits 6–5 in the SCR.

External ROM/Flash Decoding

Bit 4 of the System Configuration Register enables or disables an on-chip address decoder for external ROM or Flash memory. When this bit is 1, memory accesses at physical addresses less than the upper limit programmed in the ROM Boundary Register (ROMBR, page 84), drive the $\overline{\text{ROMRD}}$ or $\overline{\text{ROMWR}}$ pin Low. (The $\overline{\text{ROMWR}}$ pin can be used to program Flash memories.)

When SCR bit 4 is 1, $\overline{\text{ROMRD}}$ or $\overline{\text{ROMWR}}$ goes Low for addresses with A19–12 less than or equal to the contents of ROMBR, that is, for addresses less than $(\text{ROMBR}+1) * 4096$.

External RAM Decoding

Bit 3 of the System Configuration Register (page 79) enables or disables an on-chip address decoder for external RAM. When this bit is 1, memory accesses at physical addresses between the lower limit programmed in the RAM Lower Bound Register (RAMLBR), and the upper limit programmed in the RAM Upper Bound Register (RAMUBR), drive the $\overline{\text{RAMRD}}$ or $\overline{\text{RAMWR}}$ pin Low. These registers are described on page 86.

When SCR bit 3 is 1, $\overline{\text{RAMRD}}$ or $\overline{\text{RAMWR}}$ goes Low for addresses with A19–12 greater than or equal to the contents of RAMLBR, and less than or equal to the contents of RAMUBR, that is, for addresses A in the range

$$(\text{RAMLBR}) * 4096 \leq A \leq (\text{RAMUBR}+1) * 4096$$

Wait State Generators

The Z80S183 includes two registers that control automatic insertion of wait states into memory and I/O accesses. The outputs of these two generators are logically OR (Low-active OR, positive-logic AND) with the external $\overline{\text{WAIT}}$ pin, such that for a given address, the number of wait states taken is the largest number among these two facilities and any external wait-state generator.

The DMA/Wait Control register (DCNTL) is shown on page 109, and is present on all 8018x family members. DCNTL is one of the DMA registers, but the wait states that it controls apply to processor cycles as well as to those generated by the DMA channels.

Bits 7–6 select the number of wait states for all memory accesses.

Bits 5–4 select the number of wait states for I/O accesses other than those to 180 registers. The Z80S183 interprets both fields as a binary number of wait states:

| | |
|----|----------------|
| 00 | No wait states |
| 01 | 1 wait state |
| 10 | 2 wait states |
| 11 | 3 wait states |

The Wait State Generator Control register (WSGCR) is described on page 83, and is unique to the Z80S183.

Bits 7–6 control the number of wait states for memory accesses in the $\overline{\text{ROMRD}}/\overline{\text{ROMWR}}$ address range.

Bits 5–4 control the number of wait states for memory accesses in the $\overline{\text{RAMRD}}/\overline{\text{RAMWR}}$ address range.

Bits 3–2 control the number of wait states for other memory accesses, but these cycles do not appear on the external bus.

The Z80S183 interprets these fields as follows:

| | |
|----|----------------|
| 00 | No wait states |
| 01 | 1 wait state |
| 10 | 2 wait states |
| 11 | 4 wait states |

No DRAM Refresh

ZiLOG’s Z80 and Z8018x families have traditionally included dynamic RAM refresh logic. This logic is identical on all Z8018x devices including the Z80S183, but the Z80S183 does not have a $\overline{\text{RFSH}}$ pin with which to signal refresh cycles, nor a Refresh Control Register.

INPUT/OUTPUT

The Z80S183 includes an I/O space that is distinct from memory space. I/O space is accessed by means of IN and OUT instructions rather than LD, PUSH, POP, and other instructions that access memory space. The MMU passes addresses in I/O space through without change; such addresses always have A19–16 all 0.

I/O Instructions

The original Z80 featured a 256-byte I/O space. The following instructions are specific to the Z80’s 256-byte I/O space, and should not be used on the Z80S183 except to access external I/O devices that do not decode A15–8:

```
OUT (port), A
IND
INDR
INI
INIR
OTDR
OTIR
OUTD
OUTI
```

The following instructions ensure that A15–8 are all 0, and can be used to access the Z80S183’s on-chip I/O registers, as well as external devices that decode A15–8 as all 0:

```

IN0    r, (port)
OUT0   (port), r
OTDM
OTDMR
OTIM
OTIMR
    
```

The following instructions drive A15–0 from the BC register pair, and can be used to access the full 64 K-byte I/O space:

```

IN     r, (C)
OUT   (C), r
    
```

The following instruction can access the entire 64 K-byte I/O space, by pre-loading the MS 8 bits of the address into A. (This step is unnecessary for external devices that do not decode A15-8.)

```

IN     A, (port)
    
```

Relocating the 80180 Registers

The section, “Registers Summary” on page 70, describes how the Z80S183’s I/O registers are divided into 80180-registers and Z80S183-specific registers. The latter registers are always located in the range 0040–007FH. After a reset, the 80180 Registers are located in the range 0000–003FH, but bits 7–6 of the I/O Control Register (page 77) allow software to relocate the 80180 Registers to higher addresses:

| IOCR 7–6 | 180 Register Addresses |
|----------|--------------------------|
| 00 | 0000–003FH |
| 01 | Reserved, do not program |
| 10 | 0080–00BFH |
| 11 | 00C0–00FFH |

Relocating the 180 registers is included to ease porting of Z80 applications to the Z8018x family.

Caution: Use this facility with caution because certain tools may assume that the 80180 Registers are located in the 0000–003FH range. Those tools need to be reconfigured (re-assembled, recompiled) to allow for relocated 180 Registers.

Write Enable/Lock for Critical Registers

Some registers are protected from inadvertent modification by software. Writing a 0BH to the Watch-Dog Timer Command Register (WDTCR, page 112) sets bit 0 of the Watch-Dog Timer Master register to 1 (WDTMR, page 111) and enables writing to these registers. Writing any other value to WDTCR clears bit 0 of the WDTMR to 0 and prevents writing to these registers.

The Write Enable state has no effect on reading the registers protected by this mechanism, which include.

- System Configuration register (SCR, page 79)
- Power Control register (PCR, page 78)
- Port A-D Data Direction registers (DDRA–D, pages 88–95)
- All Real Time Clock registers (pages 119–123)

I/O Chip Selects

If bit 2 in the System Control Register (SCR) is 1, the Z80S183 drives the $\overline{\text{IOCS1}}$ pin for accesses to I/O addresses 0080–87H, and drives the $\overline{\text{IOCS2}}$ pin Low for accesses to 0088–8FH. If SCR bit 2 is 0, $\overline{\text{IOCS1}}$ and $\overline{\text{IOCS2}}$ remain High at all times.

I/O Waits

Bits 5–4 of the DMA/Wait Control Register can be used to insert wait states into I/O cycles with the Z80S183-specific registers at addresses 0040–007FH, and into I/O cycles with external devices. This field is interpreted as a binary number of wait states:

| | |
|----|----------------|
| 00 | No wait states |
| 01 | 1 wait state |
| 10 | 2 wait states |
| 11 | 3 wait states |

$\overline{\text{IORD}}$ Timing

Bit 5 in the Operating Mode Control Register (OMCR, shown on page 76) controls the timing of the $\overline{\text{IORD}}$ signal when software reads from an external I/O device. If this bit is 1, as it is after a reset, the Z80S183 drives $\overline{\text{IORD}}$ Low from the falling edge of PHI in the T1 clock cycle. If this bit is 0, it drives $\overline{\text{IORD}}$ Low one-half clock cycle later, from the rising edge of PHI at the start of T2. Both cases are illustrated in Figure 3.

Note: On other Z8018x family members, bits 7 and 6 in the OMCR control how the $\overline{\text{MI}}$ signal affects Z80 peripheral devices. Since the Z80S183 does not have an $\overline{\text{IORQ}}$ pin, it cannot be used with Z80 peripherals, and OMCR bits 7–6 do not matter.

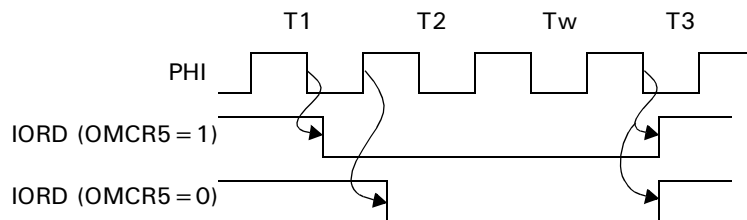


FIGURE 3. $\overline{\text{IORD}}$ TIMING

CLOCK CIRCUITS

The Z80S183 can be clocked in any of three ways:

- By an external TTL- or CMOS-level clock on the EXTAL pin,
- By a crystal connected to its XTAL and EXTAL pins, or
- By a low-frequency crystal (typically 32.768 KHz) connected to its LFXTAL and LFEXTAL pins.

An external clock signal must be free of overshoot or ringing, must make continuous, monotonic, and rapid transitions in both directions, and must meet the minimum High and Low times specified in “AC Characteristics” on page 167.

Clock Selection

Bits 1–0 of the System Configuration Register, which is shown on page 79, select the source of the main device clock (PHI) between the XTAL/EXTAL pins and the LFXTAL/LFEXTAL pins, and in the latter case, a multiplier for the clock:

| SCR 1–0 | PHI source |
|---------|---------------------------|
| 00 | XTAL/EXTAL |
| 01 | LFXTAL/LFEXTAL |
| 10 | LFXTAL/LFEXTAL times 1004 |
| 11 | LFXTAL/LFEXTAL times 502 |

Because these bits reset to 00, an application that requires LFXTAL and LFEXTAL must start up using XTAL and EXTAL. The circuit in Figure 6 includes a connection that satisfies this need.

Divide-by-2 vs. Direct Option

Regardless of the source of PHI, bit 7 of the CPU Control Register (CCR, described on page 73) controls whether the Z80S183 uses the signal selected by bits 1–0 of the SCR directly as PHI, or whether it divides the signal by 2 to obtain PHI.

If CCR bit 7 is 0, as it is after a reset, the part divides the selected signal by 2. This mode insulates the part against an asymmetric waveform on the selected signal. If CCR bit 7 is 1, the Z80S183 uses the selected signal directly. In this case, if an external clock is connected to EXTAL, the clock must meet the minimum High and Low times specified in “AC Characteristics” on page 167.

Circuits

When using a crystal connected to XTAL and EXTAL, locate the crystal as close as possible to the pins. This placement minimizes the trace lengths between the crystal, the pins, and the two capacitors shown in Figure 4, which illustrates the connection of a fundamental mode crystal up to and including 20 MHz. C1 and C2 are 20–30 pF, with 22 pF a typical value.

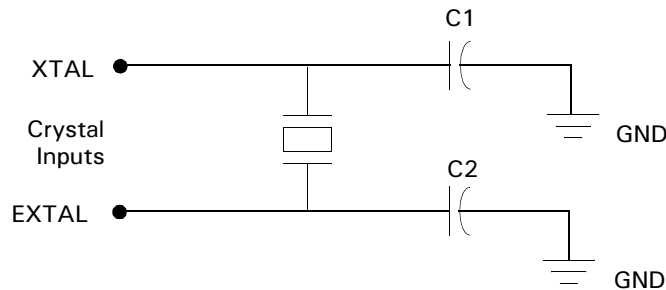


FIGURE 4. FUNDAMENTAL MODE CRYSTAL CIRCUIT \leq 20 MHz

For frequencies above 20 MHz, use a third-overtone crystal and include an LC *tank* circuit to filter the fundamental frequency, as shown in Figure 5. Again, it is essential to minimize trace lengths by locating all of the components as close as possible to the XTAL and EXTAL pins.

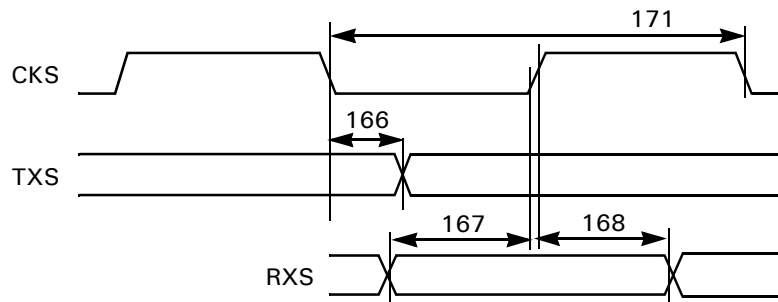


FIGURE 5. THIRD-OVERTONE CRYSTAL $>$ 20 MHz

A low-frequency crystal can be connected between the LFXTAL and LFEXTAL pins without any other components (see Figure 6). If the LF crystal is used as the clock source for the Real Time Clock, it must be exactly 32.768 KHz.

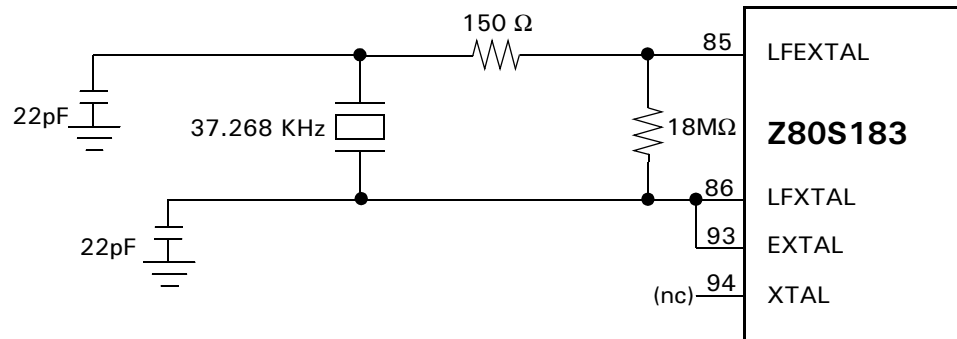


FIGURE 6. LOW-FREQUENCY CRYSTAL CIRCUIT

Crystal Specifications

The following specifications apply to fundamental mode crystals up to 20 MHz:

- Fundamental, parallel type (AT cut recommended)
- Load capacitance: $C_L = C_1 = C_2 = 20\text{--}30$ pF (22 pF typical)
- Equivalent Resistance $R_S \leq 60$ ohms
- $C_{IN} = C_{OUT} = 15\text{--}22$ pF

Reduced Oscillator Drive Option

Bit 6 in the Clock Control Register, described on page 73, controls the gain of the XTAL/EXTAL oscillator. When bit 6 is 0, as it is after a reset, a crystal connected to XTAL and EXTAL is driven strongly, to guarantee that oscillation always starts. This drive is suitable for traditional crystals packaged in HC-49-type packages, but may be too powerful for crystals packaged for miniaturized applications like PCMCIA.

To reduce the gain of the oscillator, write a 1 to bit 6 of the Clock Control Register. This action reduces the drive to about 25% of normal mode, and reduces the maximum oscillator frequency from 33 to 20 MHz.

RESET CONDITIONS

The effects of Reset on each of the registers in I/O space is described in Tables 16–131 in the section describing “I/O Registers” on page 70. Among processor registers, the following registers and state bits are cleared to 0: PC, SP, I, IEF1, IEF2, R, and F. The following are not changed by Reset: A, B, C, D, E, H, L, IX, and IY.

The Z80S183 resets itself upon power-up. When power is applied, the device detects power rising. When the oscillator starts, the Power On Reset circuitry holds the Z80S183 in reset for 2^{16} clock cycles, driving $\overline{\text{RESET}}$ Low to provide a reset to external peripherals. This Power On Reset sequence also occurs in response to a rising edge on OPMOD1 pin. If bit 6 of the Power Control Register (PCR) is 1, in response to a rising edge on PB2/CTS0/PWRUP.

Another possible source of Reset is the Watch-Dog Timer (WDT). See the section “Watch-Dog Timer” on page 43, for more information on the WDT.

POWER MANAGEMENT

The Z80S183’s low-power modes are controlled by the Standby and Idle/Quick bits in the CPU Control Register (page 73), the IOSTOP bit in the I/O Control Register (page 77), and execution of SLP and HALT instructions.

The section titled “LOW-POWER Modes” on page 30 describes the LOW-POWER modes.

Using the Power Control Register

The Power Control Register (PCR, page 78) is unique to the Z80S183, and offers additional power control options to the board designer and programmer.

$\overline{\text{CE_OUT}}$. This pin can be connected to the $\overline{\text{CE}}$ or $\overline{\text{CS}}$ pin(s) of external RAM, and to an external pullup resistor. The Z80S183 drives this pin Low when bit 7 of the Power Control Register is 1. Software sets this bit before trying to access external RAM. When PCR bit 7 is 0, the Z80S183 does not drive $\overline{\text{CE_OUT}}$, and the external resistor pulls it High, which helps safeguard the RAM against modification. Software clears PCR bit 7 to 0 before entering a LOW-POWER mode.

The RAM and the external pullup can be powered from a supply that is active when the Z80S183 is not powered, for example, from standby power or from a battery.

PB2/CTS0/PWRUP. This pin can be connected to a rising-edge-active *wake up* signal from external logic. To use this feature, software sets bit 6 of the Power Control Register to 1 before entering a LOW-POWER mode. When PCR bit 6 is 1, a rising edge on PB2/CTS0/PWRUP causes a Reset that is identical to a Power On Reset. This reset brings the Z80S183 out of the LOW-POWER mode and into normal operation. When bit 6 of the PCR is 0, PB2/CTS0/PWRUP cannot cause a reset, and can be used for other purposes.

Note: A rising edge on the OPMOD1 pin also causes a Power On Reset. Assuming that OPMOD1 remains High, the device executes code from on-chip ROM after the Reset. After a Power On Reset caused by the PWRUP pin, execution starts in on-chip ROM or external memory, depending on the state of OPMOD1.

PWRSWTCH. This pin can be connected to a Low-active power switch that controls power to external devices (for example, a P-channel FET). It is a direct positive-logic output controlled by bit 5 of the Power Control Register, which resets to 0 making PWRSWTCH Low, which in turn applies power to the external devices. Before entering a LOW-POWER mode, software can write a 1 to bit 5 of the PCR, making PWRSWTCH High and removing power from the external devices.

$\overline{\text{Tristate A19-0}}$, $\overline{\text{RAMRD}}$, $\overline{\text{RAMWR}}$, $\overline{\text{ROMRD}}$, $\overline{\text{ROMWR}}$, $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$, $\overline{\text{IOCS1-2}}$, **TXS**. If bit 4 of the Power Control Register is 1, these pins are 3-stated when the Z80S183 is in a LOW-POWER mode. This action completes the PWRSWTCH power control mechanism, in that it keeps external memories and peripherals from drawing power from these signals. PCR bit 4 resets to 0. To use this facility, software sets PCR bits 5 and 4 to 1 before entering a LOW-POWER mode.

LOW-POWER Modes

The IOSTOP bit in the I/O Control Register (page 77) controls operation of the ASCIs, PRTs and CSI/O. When this bit is 0, these peripherals operate normally. When this bit is 1, they are disabled, reducing power use.

The Standby and Idle/Quick bits in the CPU Control Register (page 73) control the mode the Z80S183 enters when it executes an SLP instruction. If the application uses the XTAL/EXTAL oscillator and Standby is 1, an SLP instruction stops the oscillator. This mode uses less power than any other mode, but requires time to restart the oscillator in response to a reset, an interrupt request, or optionally a bus request.

When Standby is 1, the Idle/Quick bit controls the number of PHI clocks the device waits after reenabling the oscillator and before restarting operation (0 selects 2^{17} (128K) clocks, and 1 selects 64 clocks).

When Standby is 0, the oscillator runs for the duration of the SLP instruction, but clocking is blocked to most of the Z80S183. The Idle/Quick bit controls whether the oscillator output is driven onto the PHI pin. A 1 in Idle/Quick disables clocking on PHI.

When Standby is 1, the BREXT bit, bit 5 in the CPU Control Register (page 73), controls whether the Z80S183 restarts the oscillator in response to a Low on $\overline{\text{BUSREQ}}$, with a 1 enabling this response.

Table 6 below details the interaction of these various bits and states, including the conditions that make the Z80S183 leave each LOW-POWER mode and resume normal operation.

TABLE 6. LOW-POWER MODES

| Instruction | Standby | Idle/Quick | IOSTOP | Mode: Operation |
|------------------------|---------|------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| other than HALT or SLP | X | X | 0 | NORMAL: The processor fetches instructions and executes them, possibly sharing the bus with on-chip DMAs and external masters. On-chip peripherals operate under software control. |
| other than HALT or SLP | X | X | 1 | I/O STOP: The processor, MMU, DMAs, and external masters operate normally, but the ASCIs, PRTs, and CSI/O are disabled to reduce power consumption. All the ASCIs, PRTs, and CSI/O can do is generate an interrupt combinatorially. Software can switch the Z80S183 between NORMAL and I/O STOP mode as appropriate |
| HALT | X | X | 0 | HALT: The processor continually fetches the op code following the Halt, but does not execute it, possibly sharing the bus with on-chip DMAs and external masters. Halt mode reverts to Normal mode if: <ul style="list-style-type: none"> – $\overline{\text{RESET}}$ or $\overline{\text{NMI}}$ is Low, – $\overline{\text{INT0}}$ is Low and enabled, or – An enabled interrupt is requested from an ASCI, PRT, CSI/O, DMA or $\overline{\text{INT21}}$. |
| HALT | X | X | 1 | HALT and I/O STOP: Similar to HALT mode except that the ASCIs, PRTs, and CSI/O are disabled. |

TABLE 6. LOW-POWER MODES (CONTINUED)

| Instruction | Standby | Idle/Quick | I/OSTOP | Mode: Operation |
|-------------|---------|------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SLP | 0 | 0 | 0 | SLEEP: Clocking is blocked to the processor and DMAs, and refresh logic, so that bus activity is not generated active on the bus. The bus can be granted to external masters. I/O operates except for DMA. SLEEP mode reverts to Normal mode under the same conditions as for HALT mode, except that the DMAs cannot interrupt. |
| SLP | 0 | 0 | 1 | SYSTEM STOP: The oscillator continues running and generating PHI, but clocking is blocked to most of the chip. Bus granting can occur. SYSTEM STOP mode reverts to Normal mode if: <ul style="list-style-type: none"> - $\overline{\text{RESET}}$ or $\overline{\text{NMI}}$ is Low, - $\overline{\text{INT2-0}}$ is Low - An enabled interrupt is requested by an on-chip peripheral that can generate an interrupt combinatorially. |
| SLP | 0 | 1 | 1 | IDLE: The oscillator continues running but PHI is blocked, as is clocking to most of the chip. Bus granting can occur if the BREXT bit (CCR5) is 1. IDLE mode can revert to NORMAL mode under the same circumstances as from SYSTEM STOP mode. |
| SLP | 1 | 0 | 1 | STANDBY: The XTAL/EXTAL oscillator is stopped. This mode does not apply to applications that use LFXTAL/LFEXTAL. Bus granting can occur if the BREXT bit (CCR5) is 1, in which case the device reactivates the oscillator, waits for 2^{17} (128K) clocks, grants the bus, and deactivates the oscillator again after the bus request is negated. STANDBY mode can revert to NORMAL mode under the same circumstances as from System Stop mode. If one of these stimuli occur while the Z80S183 is waiting for 128K clocks before responding to an enabled bus request, or while the bus is granted, it re-enters Normal mode when the bus request is negated. Otherwise, the Z80S183 reactivates the oscillator and wait for 2^{17} (128K) clocks before commencing normal operation. |
| SLP | 1 | 1 | 1 | STANDBY WITH QUICK RECOVERY: Similar to STANDBY mode, except that the Z80S183 waits only 64 clocks after enabling the oscillator, before granting the bus or resuming normal operation. |

In SLEEP, SYSTEM STOP, IDLE, or either STANDBY mode, if the Z80S183 exits the mode because of $\overline{\text{NMI}}$ or an enabled interrupt with the IEF1 flag 1, the device resumes operation by performing the interrupt, with the return address as the instruction after the SLP instruction. If the device exits the LOW-POWER mode because of an individually-enabled interrupt request, but IEF1 is 0, the Z80S183 resumes by executing the instruction after the SLP.

REDUCED DRIVE/LOW NOISE FEATURES

Reduced drive is available on many of the outputs of the Z80S183. These options reduce power consumption for applications that do not need the full drive and slew rate capability provided in normal mode. Even more importantly for some applications, invoking these features results in less noise induced onto power and ground.

The following register bits govern the drive strength on various output pins. Each of these bits resets to 0, which selects normal/full drive strength for the outputs. To save power and reduce noise for outputs that do not need the maximum drive and slew rate, software sets the associated bit to 1 during device initialization. This action reduces the drive to about 25% of full strength.

TABLE 7. REDUCED DRIVE/LOW NOISE CONTROL BITS

| Register | Shown on | Bit | Output Pins |
|-------------------------------|----------|-----|---------------------------------------|
| CPU Control Register (CCR) | page 73 | 4 | PHI |
| | | 1 | \overline{IORD} , \overline{IOWR} |
| | | 0 | A19-0, D7-0 |
| Output Control Register (OCR) | page 77 | 3 | PA7-0 |
| | | 2 | PB7-0 |
| | | 1 | PC7-0 |
| | | 0 | PD7-0 |

I/O PORTS

The Z80S183 includes four 8-bit I/O ports called A through D. All four ports feature the same basic capabilities. The ports are controlled by three registers for each port: a Data Register, a Data Direction Register, and an Output Control Register.

The ports differ primarily in interrupt capability and in pin-multiplexing with other functions, controlled by an Alternate Function Select (AFS) Register for each port.

The next two sections describe the common characteristics shared by all the ports, then the unique capabilities controlled by the AFS register for each port

The section titled, "I/O Port Registers" on page 87, describes the registers associated with the I/O ports.

Data Registers

Writing to a Data Register affects the data that is driven onto pins that are designated as outputs in the Data Direction and Output Control Registers. Reading from the Data Register returns the states of the pins, for both inputs and outputs. The output latches cannot be read separately.

Data Direction Registers and Output Control Registers

These registers determine which pins in each port are inputs and which are outputs, and for outputs, select one of three output modes. The following table shows the four possible states for each pin:

| DDR | OCR | FUNCTION |
|-----|-----|-------------------------------------------------|
| 0 | 0 | Totem pole output (active pullup) |
| 0 | 1 | Open-drain output (no internal pullup) |
| 1 | 0 | Input |
| 1 | 1 | Open-drain output with internal pullup resistor |

The totem pole outputs actively drive both High and Low. The classic open-drain output only drives Low, and relies on an external pull-up resistor to ensure a High voltage when no open-drain driver is driving Low. (This external pull-up also avoids excessive current draw by the Z80S183's receiver and any other CMOS receivers that may be connected to the signal. The 11B state is similar but connects an internal pull-up resistor of about 15K Ohms, eliminating the need for an external pullup.

All four DDRs reset to all 1s and all four OCRs reset to all 0s, so Reset configures all port pins as inputs.

The Data Direction Registers can only be written if the Register Write Enable bit is 1 in the WDT registers.

Port A Alternate Function Select (AFSA)

This register controls interrupts from port A. AFSA resets to all zeroes, disabling all interrupts from port A. Setting any of these bits to 1 enables the corresponding pin to interrupt. When an interrupt is requested by the port and interrupts are enabled, the Z80S183 fetches the address of the Interrupt Service Routine (ISR) from memory at address (I : IL : 20). The ISR must read the Data Register to determine which pin(s) caused the interrupt. Pins selected for interrupts are level-sensitive and active Low. Regardless of how this register is programmed, software must set DDRA for each port pin. Setting a pin as an output, and enabling interrupt for it, allows software to force an interrupt.

Port B Alternate Function Select (AFSB)

This register controls alternate functions for the port B pins. AFSB resets to all 0s, so that all of the pins are assigned to port B. Setting a bit to 1 selects the following alternate function for the pin:

- PB0: CKS
- PB1: DCDO
- PB2: CTS0 or PWRUP
- PB3: TXA0

PB4: RXA0
PB5: TXA1
PB6: RXA1
PB7: RXS

Selecting the alternate function for a pin disables any control of the pin by the DDRB register, but the corresponding bit in OCRB must be 0.

Port B Weak Latch Disable Feature

Most of the inputs on the Z80S183 have *weak latch* circuits to reduce power consumption if an input is not driven by an external device. A weak latch can come up in either state at Power On, and thereafter is easily over-driven by an external driver, or for bidirectional pins like the Port pins, by an internal driver. Typically, the only effect of weak latches is to prevent an input voltage from floating in the threshold region, which makes the receiver circuit draw High current.

For the PB7–0 pins only, the weak latches can be disabled by setting bit 4 of the Output Control Register to 1, (described on page 77). When this bit 4 is 0, as it is after a Reset, weak latches are enabled on Port B.

Port C Alternate Function Select (AFSC)

AFSC resets to all 0s, so that all of the pins are assigned to port C. Setting a bit to 1 assigns that pin to the Programmable I/O Sequencer (PIOS). In this ALTERNATE FUNCTION mode, the DDRC and OCRC registers determine the output drive on the pin. This feature allows the PIOS to be used with any of the output modes of the normal port function. To use PC0 as a 50 or 60 Hz time base for the Real Time Clock, leave AFSC bit 0 at 0, DDRC bit 0 at 1, and OCRC bit 0 at 0.

Port D Alternate Function Select (AFSD)

This register controls interrupts from Port D. AFSD resets to all 0s, disabling all interrupts from port D. Setting any of these bits to 1 enables the corresponding pin to interrupt. When an interrupt is generated by the port and interrupts are enabled, the Z180 fetches the address of the Interrupt Service Routine (ISR) from memory at (I: IL: 22). The ISR must read the Data Register to determine which pin(s) caused the interrupt. Pins selected for interrupts are level-sensitive and active Low. Regardless of how this register is programmed, software must set DDRD for each port pin. Setting a pin as an output, and enabling interrupt for it, allows software to force an interrupt.

Note:

1. PD0 and PD1 also function as $\overline{\text{DREQ}}_0$ and $\overline{\text{DREQ}}_1$ respectively. The DMA section describes how to take DMA Requests from these pins, which programmed inputs in this case.

To use the A/D capability of Port D, leave the Alternate Function Select Register in PORT mode and set the DDRD and OCRD for input.

| Register | On Page | Bit | Output Pins |
|-------------------------------|---------|-----|-----------------------------------------------------|
| CPU Control Register (CRC) | 68 | 4 | PHI |
| | | 1 | $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$ |
| | | 0 | A19–0, D7–0 |
| Output Control Register (OCR) | 71 | 3 | PA7–0 |
| | | 2 | PB7–0 |
| | | 1 | PC7–0 |
| | | 0 | 7–0 |

DMA CHANNELS

The Z80S183 includes two DMA channels called DMA0 and DMA1. Both channels can transfer data between memory and a peripheral in I/O space. In addition, DMA0 can perform memory-to-memory block transfers, and transfers between memory and memory-mapped I/O devices.

Both DMA channels are of the *flowthrough* type, in which each byte transferred requires two bus cycles—the first to read the source and the second to write the destination. As a result, neither memory nor peripherals are subject to any special considerations for bus cycles controlled by the DMA channels, because such cycles are identical to processor bus cycles.

DMA transfer can occur as fast as 6 clocks/byte. At 33 MHz, this speed corresponds to 5.5 MBPS. Destination/output devices require Edge-Sensitive request mode, in which the maximum rate is 9 clocks/byte, or 3.67 MBPS at 33 MHz.

The section, “DMA Registers” on page 98, describes the registers associated with the DMA channels.

DMA Basics

Each channel has two 20-bit address registers and a 16-bit byte count. For DMA0 the address registers are called the Source and Destination Address Registers (SAR and DAR). DMA1’s address registers are called the Memory and I/O Address Registers (MAR and IAR).

Each address register is divided into three Z80S183 I/O registers, called L (Low), H (High), and B.

When a DMA channel is operating, it drives A7–0 from the L register and A15–8 from the H register. For memory addresses (always for MAR) the channel drives A19–16 from the least significant four bits of the B register. For I/O addresses (always for IAR) the channel uses the least significant three bits of the B register to select the source of the DMA Request signal that controls data transfer.

Each byte count register is divided into two Z80S183 I/O registers, called L (Low) and H (High).

After programming a DMA channel's address and byte count registers, software starts the channel by setting the Enable bit in the DSTAT register (DE0 or DE1). As/after the channel transfers each byte, it decrements the byte count register. When a channel has decremented the byte count to 0, it goes inactive by clearing its Enable bit to 0.

Software can select whether a channel increments or decrements a memory address as/after it transfers each byte. DMA0 also features an option to keep a memory address fixed. This fixed address option is intended for use with a memory-mapped I/O device that provides a DMA Request signal on the PD0/AC0/DREQ0 pin.

DMA Requests

An external peripheral, that needs a DMA channel to transfer data for it, must provide a DMA request signal to the PD0/AC0/DREQ0 pin for DMA0, and/or to the PD1/AC1/DREQ1 pin for DMA1. A DMA request can be connected directly to one of these pins if only one external peripheral is serviced by that DMA channel. If more than one external device is to use a DMA channel, external selection logic must be included in the application, to route the current device's request to the channel's DREQ pin.

The otherwise-unused bits 19--16 of the I/O address of a peripheral, select either the external $\overline{\text{DREQ}}$ pin or an internal peripheral as the source of each DMA channel's request. For a memory-mapped peripheral (a source or destination of a DMA0 memory-to-memory operation that is programmed to use a fixed address), the PD0/AC0/DREQ0 pin is always used as the Request signal.

The DMA request signal indicates when an input or source peripheral has a byte to be transferred to memory, or when an output or destination peripheral needs a byte from memory.

Edge- vs. Level-Sensitive Requests

DMA requests can be programmed to be low-level sensitive or falling-edge sensitive. For an output/destination peripheral, the timing requirements on the DMA Request signal dictate falling-edge mode. An input/source peripheral can use either an edge- or level-sensitive DMA Request.

Figure 7 illustrates the timing of a level-sensitive DMA Request. DMA operation is triggered when the channel samples the DMA request line Low. The channel samples the Request line again, at the rising PHI edge that begins the second-last clock cycle of the write cycle to the destination. If the Request line is Low at that time, as it is at the rightmost down-arrow below, the channel continues on to transfer another byte. If the Request is High, as at the leftmost down-arrow below, the DMA channel relinquishes use of the bus (to the processor, the other DMA channel, or an external master) after completing the write cycle.

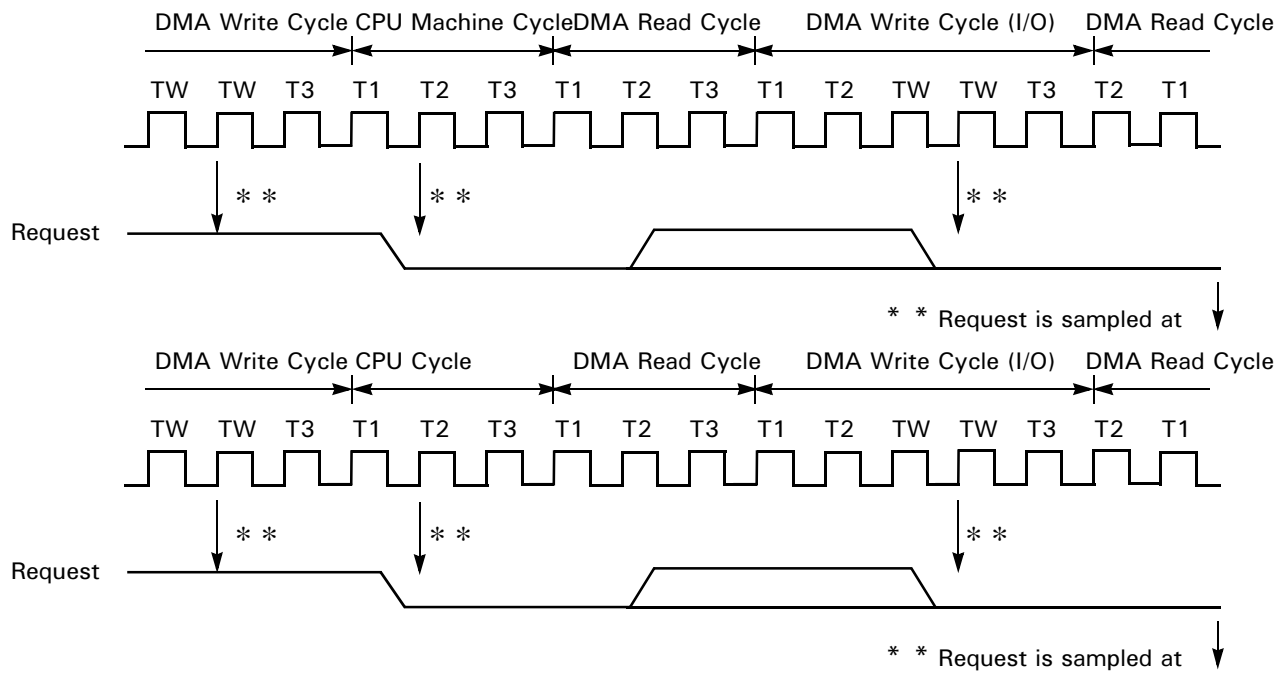


FIGURE 7. PROCESSOR/DMA OPERATION WITH LEVEL- SENSE REQUEST

Figure 8 illustrates the timing of an edge-sensitive request. At the first down-arrow, the DMA channel writes a byte to the destination. However, a new falling edge has not occurred by the second-to-last rising PHI edge of the cycle. The channel relinquishes the bus to the processor.

By the same sampling point in the subsequent processor cycle, a new falling edge has occurred on the Request line, so the DMA channel assumes control of the bus, and reads and then writes a byte.

At the same point in the DMA write cycle, a new falling edge has not yet occurred, so the channel returns bus control to the processor. The channel does not operate again until the Request line goes High and then Low again, some time after the right edge of Figure 8.

8

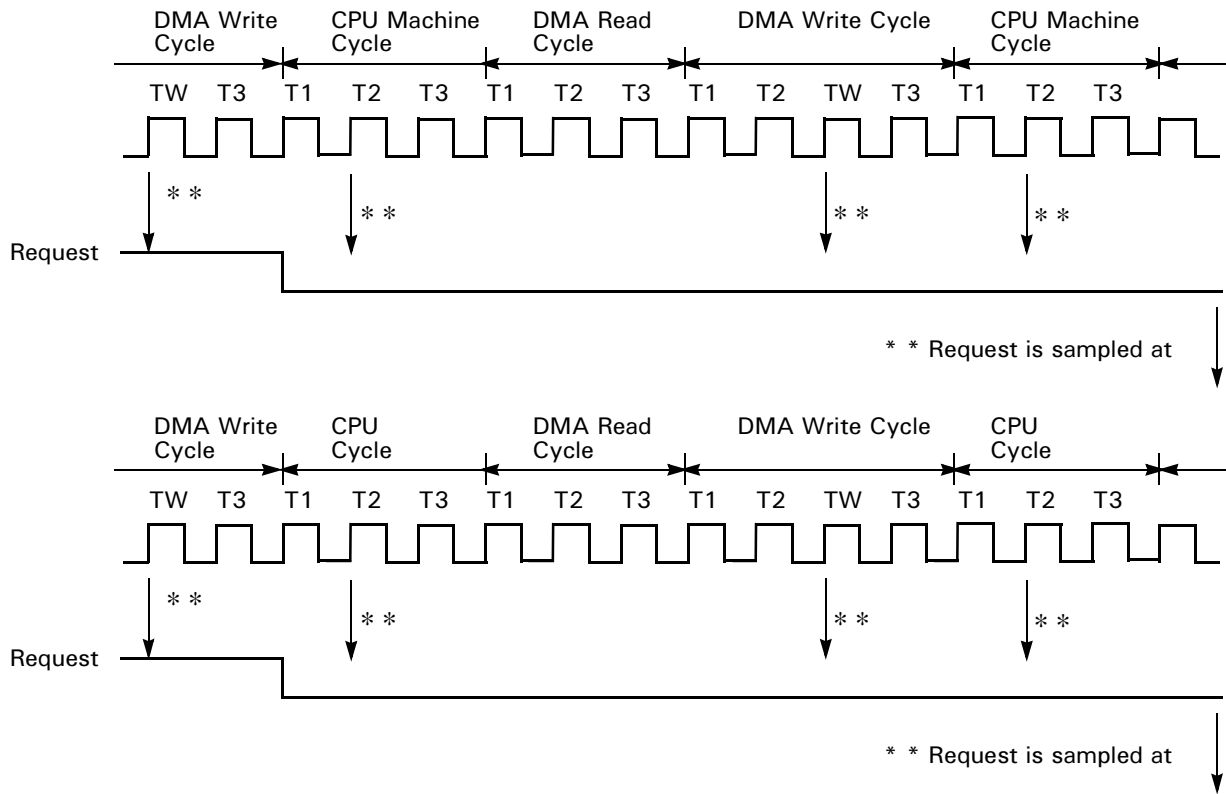


FIGURE 8. PROCESSOR/DMA OPERATION WITH EDGE-SENSE REQUEST

Memory-to-Memory Modes

In a DMA0 memory-to-memory operation, in which both the source and destination are programmed for address incrementing or decrementing, there is no peripheral to supply a request signal to control the transfer. In this case, software can select between two modes of operation by programming MMOD, bit 1 of the DMA mode register.

If MMOD is 0, the processor and DMA channel alternate bus cycles until the DMA completes the block transfer and decrements the byte count to 0. This sequence is called CYCLE STEAL mode.

If MMOD is 1, the DMA channel performs continuous cycles until the block transfer is complete. The processor can perform no other actions during this time. This sequence is called BURST mode.

DMA Interrupts

Software can enable interrupts from each DMA channel, which then requests an interrupt after decrementing its byte count to 0. When the processor acknowledges this interrupt, it fetches the address of the interrupt service routine from memory at (I : IL : 8) for DMA0, or (I : IL : 10) for DMA1.

If the interrupt service routine does not have another block of data for the DMA channel to transfer, it prevents further interrupts by clearing the interrupt enable bit (DSTAT bit 2 for DMA0, bit 3 for DMA1), before it reenables interrupts with an EI instruction. If the ISR programs the DMA channel for another transfer, interrupts can be reenabled with an EI instruction, after restarting the DMA by writing to DCNTL.

Setting Up a DMA Transfer

Write the Address Registers. For DMA0, this process includes registers SAR0L, SAR0H, SAR0B, DAR0L, DAR0H, and DAR0B are included. If the source is in I/O space, write SAR0B with a code to select the source of the DMA Request for DMA0, as described in Table 8:

TABLE 8. SAR0B VALUE FOR A SOURCE IN I/O SPACE

| SAR0B bits 2–0 | DMA Request Source |
|----------------|--------------------------|
| 000 | PD0/AC0/DREQ0 pin |
| 001 | ASCIO RDRF |
| 010 | ASC11 RDRF |
| 011 | Reserved, do not program |
| 1xx | Reserved, do not program |

If the DMA destination is in I/O space, write DAR0B with a code to select the source of the DMA Request for DMA0, as described in Table 9:

TABLE 9. DAR0B VALUE FOR A DESTINATION IN I/O SPACE

| DAR0B bits 2–0 | DMA Request Source |
|----------------|--------------------------|
| 000 | PD0/AC0/DREQ0 pin |
| 001 | ASCIO TDRE |
| 010 | ASC11 TDRE |
| 011 | Reserved, do not program |
| 1xx | Reserved, do not program |

For DMA1, software must write registers MAR1L, MAR1H, MAR1B, IAR1L, IAR1H, and IAR0B. Write IAR1B with a code to select (with the DIM1 bit in the DCNTL) the source of the DMA Request for DMA1, as described in Table 10:

TABLE 10. IAR1B VALUE

| IAR1B bits 2–0 | DIM1 | DMA Request Source |
|----------------|-------------|--------------------------|
| 000 | X | PD1/AC1/ DREQ1 pin |
| 001 | 0 (mem→I/O) | ASCIO TDRE |
| | 1 (I/O→mem) | ASCIO RDRF |
| 010 | 0 (mem→I/O) | ASC11 TDRE |
| | 1 (I/O→mem) | ASC11 RDRF |
| 011 | X | Reserved, do not program |
| 1xx | X | Reserved, do not program |

Write the Byte Count Registers. Write the less-significant byte to BCR0L or BCR1L, and the more-significant byte to BCR0H or BCR1H. An all-0 value causes the DMA to transfer 65,536 bytes.

For DMA0, Write the DMA Mode Register. Bits 3–2 select the operating mode for the source, as described in Table 11. Bits 5–4 select the operating mode for the destination, as described in Table 12. For memory-to-memory block transfers, bit 1 (MMOD) selects between Cycle Steal and Burst modes, as described in “Memory-to-Memory Modes” on page 39.

TABLE 11. DMA0 SOURCE MODE

| DMODE 3–2 | Mode |
|-----------|-----------------------------------------------------|
| 00 | Increment Memory Address |
| 01 | Decrement Memory Address |
| 10 | Fixed Memory Address (request on PD0/AC0/DREQ0 pin) |
| 11 | Fixed I/O Address |

TABLE 12. DMA0 DESTINATION MODE

| DMODE 5–4 | Mode |
|-----------|-----------------------------------------------------|
| 00 | Increment Memory Address |
| 01 | Decrement Memory Address |
| 10 | Fixed Memory Address (request on PD0/AC0/DREQ0 pin) |
| 11 | Fixed I/O Address |

Write the DCNTL Register. If both DMA channels can be used simultaneously, software reads DCNTL, modifies the bits noted below for the current DMA channel, and writes back the result. Otherwise, software can simply write DCNTL.

Bits 7–4 select the number of waits to insert for Memory and I/O accesses, as described in the section “Wait State Generators” on page 23. For DMA0, bit 2 selects between edge- and level-sensitivity on the DMA Request. For DMA1, bit 3 selects between edge- and level-sensitivity, and bits 1–0 select the operating mode as described in Table 13:

TABLE 13. DMA1 OPERATING MODE

| DCNTL 1–0 | Mode |
|-----------|----------------------------------------------|
| 00 | Increment Memory Address → Fixed I/O Address |
| 01 | Decrement Memory Address → Fixed I/O Address |
| 10 | Fixed I/O Address → Increment Memory Address |
| 11 | Fixed I/O Address → Decrement Memory Address |

Write the DSTAT Register to Enable the DMA Channel. If both DMA channels can be used simultaneously, software reads DSTAT, modifies the bits noted below for the current DMA channel, and writes back the result. Otherwise, software can simply write DSTAT.

For DMA0, write 110 to bits 6–4, and write a 1 to bit 2 if DMA0 interrupts when it has decremented its byte count to 0, or a 0 if not.

For DMA1, write a 1 to bit 7, 01 to bits 5–4, and a 1 to bit 3 if DMA1 interrupts when the when it has decremented its byte count to 0, or a 0 if not

NMI and DME

When software writes to DSTAT to enable either DMA channel, this action also sets the DMA Master Enable (DME) bit (bit 0 in DSTAT). A 1 in this bit enables operation by either or both DMA channels.

To guarantee that a Non Maskable Interrupt (NMI) is handled promptly, when it detects $\overline{\text{NMI}}$ Low the Z80S183 sets DME to 1 to suspend DMA operation.

The NMI service routine reads DSTAT immediately after saving the registers. For each DMA channel, if the DE bit (DSTAT7 or 6) is 1, and the associated device (if any) has not overrun or underrun, the service routine clears that channel’s DWE bit (DSTAT5 or 4) to 0. Then, if either DWE bit is 0, the routine writes the result back to DSTAT. This sequence sets DME again and reenables DMA operation.

DMA Channel Completion

While a DMA channel is operating, software can stop it by reading DSTAT, clearing bits 6 and 4 for DMA0, or 7 and 5 for DMA1, and writing the result back to DSTAT.

Otherwise, if software enabled the channel to interrupt, when the channel has decremented its byte count to 0, an interrupt is requested.

If software did not enable the DMA channel to interrupt, it can read the Enable bit in DSTAT (bit 6 for DMA0, bit 7 for DMA1) to determine whether/when the DMA channel finishes transferring the current block of data. In some applications, software can use status or an interrupt from the associated peripheral device to determine completion of the block transfer.

Handling DMA Interrupts

When the conditions noted in “On-Chip Interrupt Handling” on page 19 are met with respect to a DMA interrupt request, the processor fetches the interrupt service routine (ISR) address from memory at (I : IL : 8) for DMA0 or (I : IL : 10) for DMA1. The ISR performs the following functions, as a minimum:

- Reads the DSTAT to verify that the DE bit is 0 for the DMA channel corresponding to the service routine entry point. If a common ISR is used for both DMAs, the DSTAT value indicates which DMA channel(s) has (have) completed operation.
- Reprograms the channel’s registers and restarts it if the channel is to continue operation.
- Clears the DIE bit for the channel, in the DSTAT register, to prevent another interrupt for the same DMA completion.
- Concludes with EI and RET instructions, to return to the interrupted process.

WATCH-DOG TIMER

The Watch-Dog Timer (WDT) helps protect against unreliable software, power line faults that put the processor into unusual states, and other system-level problems.

When the WDT is enabled, software must periodically reload it, to prevent it from resetting the processor or the entire application. The time period, within which software must reload the WDT, is programmable among 2^{18} , 2^{22} , 2^{25} , or 2^{27} system clocks.

The registers in the WDT are detailed in “Watch-Dog Timer Registers” on page 110.

Several provisions of the WDT enhance its integrity against runaway execution. The WDT can only be reloaded by writing the specific value 4EH to the WDT Command register. The WDT can only be disabled by:

- Setting bit 1 of the WDT Master register to 1
- Writing the value 40H to the WDT Command register
- Clearing bit 1 to 0 in the Master register.

Core code at logical address 0000B typically identifies a Reset if the SP contains 0000B. At reset, two bits in the WDT Master register (WDTMR) help identify the cause of the reset. Bit 3 is 1 for a Power On Reset, while bit 2 is 1 for a WDT Reset. Reading The WDTMR clears these bits for subsequent reads.

PROGRAMMABLE RELOAD TIMERS (PRTs)

The Z80S183 includes two Programmable Reload Timers called PRT0 and PRT1. Each includes a 16-bit down-counter that can be read at any time, and a reload value that can be dynamically programmed. Each PRT can interrupt the processor when it counts down to 0 and reloads.

Note: On other 8018x family devices, PRT1 has waveform-generation capability on a pin called TOUT, but on the Z80S183 this function is superseded by the more capable Programmable I/O Sequencer (PIOS).

“Programmable Reload Timer (PRT) Registers” on page 113, details the PRT registers. Reset clears both Timer Downcount Enable bits (TDE1, TDE0) in the Timer Control register to 0, which prevents PRT operation.

Starting a PRT

To start a PRT:

1. Write the initial down-count value to the Timer Data Registers (TMDR0L and TMDR0H, or TMDR1L and TMDR1H),
2. Write the second (and possibly constant) down-count value to the Timer Reload Registers (RLDR0L and RLDR0H, or RLDR1L and RLDR1H).
3. Read the Timer Control Register (TCR),
4. Set the appropriate Timer Downcount Enable bit (TDE0 or TDE1) to 1.
5. Set or clear the corresponding Timer Interrupt Enable bit (TIE0 or TIE1) depending on whether an interrupt is desired when the count is decremented to 0, and
6. Write the result value back to the TCR.

The read-modify-write procedure of steps 3-6 ensures that starting one PRT does not affect the operation of the other. Applications that only use one PRT can simply write the desired value to the TCR instead.

Stopping a PRT

To stop a running PRT:

1. Read the TCR.
2. Clear the PRT's TDE bit, and
3. Write the result value back to the TCR.

A PRT may be stopped because its timing function is no longer needed, or before rewriting the RLDR value as described in the next section.

PRT Operation

While a PRT is running, it decrements the down-counter every 20 PHI clocks. When it counts down to 0, a PRT automatically performs the following actions:

1. Reloads the TMDR from the RLDR
 2. Sets the TIE bit in the TCR to 1
 3. Requests an interrupt if the TIE bit in the TCR is 1.
-

After reading a TIF bit as 1 in the TCR, software can clear it by reading either half of that PRT's TMDR. However, reading a TMDR, without first reading a TIF bit as 1 in the TCR, does not clear the TIF bit.

Software can read a down-counter, from TMDR0L and TMDR0H or TMDR1L and TMDR1H at any time. The PRTs ensure that the two 8-bit values read by INO instructions are consistent, provided that software reads TMDR0L or TMDR1L first. Reading one of these registers captures the more significant byte of the down-counter in a separate 8-bit latch, from which the value is provided when software subsequently reads TMDR0H or TMDR1H.

Writing an RLDR. Software can write a new reload value to RLDR0L and RLDR0H or RLDR1L and RLDR1H, while a PRT is running, but there is no hardware safeguard against the down-counter decrementing to 0 between the two 8-bit OUT0 instructions needed to write the new reload value. If this occurs, the value loaded into the down-counter may be incorrect.

If a new reload value is written in response to a PRT interrupt or to detecting a TIF bit 1 in the TCR, and the count values are large enough to prevent this problem, software can write the RLDR. Otherwise software must perform the following steps to load an RLDR:

1. Reads the Timer Control Register (TCR).
2. Clears the PRT's TDE bit.
3. Writes the result back to the TCR.
4. Writes the PRT's RLDR (L and H, in either order).
5. Writes the value from step 1 (with the TDE bit 1) back to the TCR.

Handling PRT Interrupts

When the conditions noted in "On-Chip Interrupt Handling" on page 19 are met with respect to an interrupt request from a PRT, the processor reads the address of the interrupt service routine from memory at address (I : IL : 4) for PRT0, or (I : IL : 6) for PRT1. The PRT ISR performs the following actions:

1. Save as many registers of the interrupted process as it may use (worst case), by means of PUSH, EX AF,AF', and/or EXX instructions.
2. Read the TCR, verifying that the TIF bit is set to 1.
3. Read the PRT's TMDRL register to clear the TIF bit, preventing another interrupt for the same zero-count.
4. If the RLDR value must be changed for the next down count sequence, the ISR proceeds as described in "Writing an RLDR", described previously.
5. Next, the ISR performs any necessary time-periodic functions in service to the overall application.
6. When these timer functions are completed, the ISR restores the saved registers, and returns to the interrupt process using EI and RET instructions.

If both PRTs are active and both are started and stopped, either at interrupt or mainline level, the mainline code that reads, modifies, and writes the TCR must protect against conflicts with an ISR for the other PRT. It surrounds the read-modify-write sequence (steps 1–5 in “Writing an RLDR” on page 45) with DI and EI instructions.

PRTs and Reset. Both TMDRs and both RLDRs reset to FFFFH, and the TCR resets to all 0s, which inhibits PRT operation until a TDE bit is set.

REAL TIME CLOCK (RTC)

The Real Time Clock module operates like a watch chip, maintaining readable registers ranging from seconds to centuries. An application that uses the RTC must provide it with one of several specific time bases. A 32.768 crystal can be connected to the LFXTAL and LFEXTAL pins, or a 50- or 60-Hz clock derived from AC power can be connected to the PC0 pin.

The RTC also includes an alarm function. Programmable registers containing alarm seconds, minutes, and hours are continually compared to the corresponding clock values. When all three registers match, the RTC can interrupt the processor. If a Snooze function is required, it must be implemented by software.

The section titled, “Real Time Clock (RTC) Registers” on page 118, describes the registers in the RTC. All of them are read/write. After software sets these registers, the RTC maintains the time provided there is Power On the V_{DD} pins. Reset has no effect on the RTC.

Note: The Register Write Enable bit must be set as described on page 25, before any of the RTC registers can be written.

Configuring the RTC

First, software must select the time base for the RTC: a 32.768 KHz crystal on the EXTAL and LFEXTAL pins, or a 50- or 60-Hz clock on the PC0 pin, in bits 4–3 of the RTC Command/Status register. The RTC divides the selected base clock by 32768, 50, or 60 to produce a 1 Hz clock that increments the Seconds register.

Setting the Time and Date.

The first step in setting the time and date is to determine them, from a user interface or a network or serial link. Next software must convert the time and date to the BCD format used in the RTC registers. The RTC does not include any hardware support for deriving the day of the week from the other values. Finally, software writes these values to the RTC Seconds, Minutes, Hours, Day of the Week, Date, Month, Year, and Century registers.

Note: the RTC registers increment in Binary Coded Decimal (BCD) format, and values written to these registers must be encoded in this format. In most of the RTC registers, bits 7-4 contains a tens digit and bits 3-0 contain a units digit. Neither field may contain any of the values 1010B through 1111B. Most of the RTC registers have further restrictions on the range of values that can be written to them, which are noted in “RTC Incrementing”, later.

Writing the RTC registers. While writing a new time and date into the RTC registers, there is a slight chance that a 1 Hz clock edge that causes a rollover will occur between writes, so that the registers contain the wrong value. To protect against this possibility:

1. Write all of the registers starting with the Seconds register.
2. Read the Seconds register.

If the value from step 2 is less than the value written to the Seconds register in step 1, a rollover occurred and the RTC registers may be wrong: Return to step 1.

Reading the RTC registers. When reading the RTC registers, there is a slight chance that a 1Hz clock edge that causes a rollover will fall between reads, so that the set of values read is wrong. To protect against this possibility:

3. Read all the registers, starting with the Seconds register,
4. Read the Seconds register again,

If the Seconds value from step 2 is less than the Seconds value from step 1, a roll-over occurred: return to step 1.

RTC Operation

RTC Incrementing. This section describes the functions of clocks and calendars.

1. The Seconds register rolls over from 59 to 0, at which time the Minutes registers is incremented.
2. The Minutes register rolls over from 59 to 0, at which time the Hours register is incremented.
3. The Hours register rolls over from 23 to 0, at which time the Day of the Week and Date registers are incremented. The RTC is a 24-hour clock.
4. The Day of the Week register rolls over from 7 to 1.
5. The contents of the Month, Year, and Century registers determine how the Date register rolls over. When the Date register rolls over, the Month register is incremented.
 - If the Month is 1, 3, 5, 7, 8, 10, or 12, the Date register rolls over from 31 to 1.
 - If the Month is 4, 6, 9, or 11, the Date register rolls over from 30 to 1.
 - If the Month is 2 and the Year and Century registers indicate a leap year, the Date register rolls over from 29 to 1. Otherwise it rolls over from 28 to 1.
6. The Month Register rolls over from 12 to 1, at which time the Year register is incremented.
7. The Year register rolls over from 99 to 0, at which time the Century register is incremented.

Setting and Polling the Alarm. To set an alarm, write the Alarm Seconds, Alarm Minutes, and Alarm Hours registers. The RTC continually compares these values with those in the Seconds, Minutes, and Hours registers, and sets the Alarm bit (bit 7) in the RTC Control/Status register (RTCCS) whenever all of these registers are equal. If the RTC Alarm interrupt is disabled, software can poll bit 7 in the RTCCS periodically to detect an Alarm.

Handling an Alarm Interrupt. If an interrupt is desired when the RTC sets the Alarm bit, sets bit 6 (IE) in the RTC Control/Status register after writing the three Alarm registers. Thereafter, when the Alarm bit (RTCCS7) is 1 and the conditions in “On-Chip Interrupt Handling” on page 19 have been met with respect to the RTC request, the Z80S183 responds by fetching the RTC interrupt service routine (ISR) address from memory at address (I : IL : 26). Most of this ISR is application-dependent, but as a minimum the ISR performs the following actions:

1. Save as many registers of the interrupted process as it may use, using PUSH, EX AF, AF', and/or EXX instructions.
2. Check that the Alarm bit (bit 7) of the RTC Control/Status (RTCCS) register is 1. If bit 7 is 0, the ISR may log this unknown interrupt, before restoring the saved register values and returning to the interrupted process using EI and RET instructions.

If a future alarm at a different time is desired, the ISR must reprogram the Alarm registers with that time. If no future Alarm is required, clear the IE bit in the value read in step 2. If an Alarm is required at the same time tomorrow, increment the Alarm time by 1 second, and write that value to the Alarm registers. This action prevents a continuing match from setting the Alarm bit immediately.

3. Clear bit 7 in the value read in step 2, and write it back to RTCCS.
4. Restore the saved registers after application-dependent processing is complete, and return to the interrupted process by means of EI and RET instructions.

If an Alarm at the same time tomorrow was needed in step 2, when the next RTC Alarm interrupt occurs (1 second later) repeat steps 1–4, decrementing the Alarm registers back to their original value.

DIGITAL/ANALOG CONVERTER (DAC)

The Digital-to-Analog Converter converts 10-bit digital values to analog voltages using a 10-stage resistor ladder. The DAC output is protected against latch-up and can drive output loads.

The AVDD and AGND pins provide the analog power for both the ADC and DAC modules. To achieve the specified accuracy, the voltages must be within the specified tolerances of V_{DD} and V_{SS} respectively. For maximum accuracy, isolate and filter AVDD and AGND from power supply noise.

“Digital to Analog Converter (DAC) Registers” on page 124, describes the I/O registers associated with the D/A Converter. These registers include a Data register containing the eight most significant bits of the digital value, and a

Control register (DACCR) which holds the two least significant bits plus configuration and enable bits.

Reference Voltage Selection

Bits 5–4 of the Control register select the reference voltage that is multiplied by the binary fraction represented by the digital value to be converted to obtain the voltage produced on the AOUT pin.

| DACCR5–4 | Reference Voltage |
|----------|-------------------|
| 00 | internal 2.6 V |
| 01 | internal 4.2 V |
| 1X | PA0/VREF pin |

Notes:

1. The DAC and A/D Converter share the PA0/VREF pin as an external analog reference voltage.
2. The reference voltage selected by bits 5–4 must be less than AV_{DD} and V_{DD} .

Example: If the 10-bit binary value in the Data and Control registers is 1000000000B, and bits 5–4 are 00B, the voltage on the AOUT pin is 2.1 volts.

Programming the DAC. To convert a 10-bit value to an analog voltage:

1. Write the eight most significant bits of the value to the DAC data register.
2. Write the two least significant bits of the value to bits 7–6 of the DAC Control register, along with the reference voltage selection in bits 5–4, and a 1 in bit 2 to enable analog voltage drive on the AOUT pin.

The voltage on AOUT settles to the programmed value within 1 microsecond after step 2.

Conversion Time Waffle. At the time this document was written, there is concern about the DAC conversion time in one case. If the selected reference voltage, times the fraction represented by the digital value, is within 0.5 volts of AV_{DD} and V_{DD} , settling may take longer than 1 microsecond. This issue will be characterized and clarified when silicon is available.

ANALOG/DIGITAL CONVERTER (ADC)

The Analog-to-Digital Converter is a 10-bit successive-approximation converter with 8 input pins. Conversion time is 64 PHI clocks, or about 1.92 microseconds at 33 MHz. Software determines the completion of a conversion using an interrupt or by polling.

The AV_{DD} and A_{GND} pins provide the analog power for both the ADC and DAC modules. To achieve the specified accuracy, the voltages on these pins must be

within the specified tolerances from V_{DD} and GND respectively. For maximum accuracy, isolate and filter AV_{DD} and A_{GND} from power supply noise.

Though the ADC can operate with small analog reference voltages, noise and offset are independent of the reference. Accuracy is maximized for reference voltages close to AV_{DD} and V_{DD} .

“Analog to Digital Converter (ADC) Registers” on page 126, describes the I/O registers associated with the A/D Converter. They include two Control registers, ADCC0 and ADCC1, and a Data register from which the software can read the most significant eight bits of the digital value when a conversion is complete.

Channel Selection

The ADC can divide the voltage on any of the pins in Port D by the reference voltage, to produce a 10-bit binary fraction. This fraction can be read from the Data register and ADCC1, when the conversion is complete. Bits 2–0 of ADCC1 select the port pin for each conversion:

| ADCC1 2–0 | Selected Channel |
|-----------|------------------|
| 000 | PD0/AC0/DREQ0 |
| 001 | PD1/AC1/DREQ1 |
| 010 | PD2/AC2 |
| 011 | PD3/AC3 |
| 100 | PD4/AC4 |
| 101 | PD5/AC5 |
| 110 | PD6/AC6 |
| 111 | PD7/AC7 |

Reference Voltage Selection

Bits 1–0 of ADCC0 select the reference voltage by which the voltage on the selected channel is divided, to produce the 10-bit binary fraction.

| ADCC0 1–0 | Reference Voltage |
|-----------|-------------------|
| 0X | PA0/VREF pin |
| 10 | internal 4.2 V |
| 11 | internal 2.6 V |

Notes:

1. The D/A Converter and ADC share the PA0 / VREF pin as an external analog reference voltage.
2. The reference voltage selected by bits 1–0 must be less than AV_{DD} and V_{DD} .

Programming the ADC

To convert the voltage on one of the Port D pins to a 10-bit binary fraction:

1. Write ADCC0 bits 1–0 to select the desired reference voltage, with a 1 in bit 2 to enable the A/D converter.
2. Write ADCC1 bits 2–0 to select the desired Port D pin, with a 1 in bit 3 to start the conversion. Include a 1 in bit 5 if an interrupt at the end of the conversion is desired. If so, refer to the next section, “Handling an ADC Interrupt”.
3. Read ADCC1 until bit 4 is 1, indicating polled operation. This operation indicates that the conversion is complete. Because the conversion always takes 64 clocks, 21 NOP instructions can be substituted for reading ADCC1, but NOPs require more memory.
4. Read the ADC Data register to obtain the eight most significant bits of the result. If all 10 bits of the result are needed, read ADCC1 (bits 7–6 are the two least significant bits).

Handling an ADC Interrupt

An interrupt takes at least 18 PHI clocks, plus the overhead of saving and restoring registers, reenabling interrupts, and returning to the interrupted process. Because a complete ADC conversion takes 64 clocks, many Z8S183 applications use polling rather than ADC interrupts, for A/D conversions initiated by the processor.

A/D interrupts are more useful for A/D conversions initiated by the PIOS module, described in “Programmable I/O Sequencer (PIOS)” on page 52.

Regardless of the initiator, when the conditions noted in “On-Chip Interrupt Handling” on page 19 are met with respect to an ADC interrupt request, the Z8S183 fetches the address of the ADC interrupt service routine (ISR) from memory at (I : IL : 24). This ISR is application-dependent, but at a minimum it must:

1. Save as many registers of the interrupted process as it may use, using of PUSH, EX AF, AF', and/or EXX instructions.
2. Read the Data register into H and ADCC1 into L if all 10 bits are required. If only the most significant eight bits of the result are required, the ISR reads ADCC1 into A to clear the conversion complete flag and prevent further interrupts for the same conversion, and then read the Data register into A.
3. Process the resulting value. This step is application-dependent.
4. Start another conversion, if desired, as in steps 1–2 of “Programming the ADC”.
5. Restore the registers using POP, EX AF, AF' and/or EXX instructions.
6. Return to the interrupted process using EI and RET instructions.

Sampling and Conversion

During the first 3 PHI clocks of the 64 required for a complete conversion, the ADC samples the voltage on the selected pin. To ensure the accuracy of the conversion, the voltage must remain constant during this time. During the last 61 clocks of the conversion, the pin voltage can change without affecting the result.

Restarting Conversion

An A/D conversion can be started at any time. If a conversion is in progress and a new start command is written to ADCC1, the conversion in progress is aborted and a new conversion is initiated. Software must not change bits 2–0 of ADCC1 while a conversion is in progress, unless the write that changes these bits also includes a 1 in bit 3 to start a new conversion.

Saving Power

When the ADC is not needed, a small amount of power can be saved by clearing bit 2 of ASCC0, shutting down the ADC.

PROGRAMMABLE I/O SEQUENCER (PIOS)

The Programmable Input/Output Sequencer can be used to create multiple complex digital waveforms and trigger A/D and D/A conversions without processor intervention. In addition the PIOS can generate processor interrupts at selected points during its operation.

The last 256 bytes of on-chip RAM can be read by the PIOS and can be read and written by the processor. Depending on the value of bit 5 in the System Control Register, the processor may locate these bytes at addresses FFF00–FFFFFH or xFFF00–xFFFFFH, that is, with or without decoding A19–16.

To the PIOS, this memory area is organized as 64 4-byte entries, each containing:

- A 14- or 16-bit delay value
- A 2-bit entry type
- An 8- or 10-bit data value
- A 6-bit address of the next entry

When the PIOS fetches an entry from RAM, it stops processing if the next-address field in the entry is zero. Otherwise it loads the entry into internal registers and begins counting down the delay value. When the PIOS has counted the delay down to zero, it performs the action specified by the rest of the entry. The PIOS then fetches a new entry using the Next Address value as A7–2 of the memory address.

The section, “Programmable Input/Output Sequencer (PIOS) Registers” on page 129, details the I/O registers in the PIOS, which include:

- A Control register that software can use to configure the PIOS

- An Address/Type register into which the PIOS fetches control information from each entry
- Timer Low and High registers, into which the PIOS fetches a 16-bit delay value from each entry, and then counts it down to zero.

Configuring the PIOS

Bits 1–0 of the Control register select the clock used to count down the delay value in each PIOS entry, among PHI, PHI/256, PHI/1024, or PHI/4096.

Software must not change these bits while the PIOS is running. To ensure an accurate delay in the first entry, software set bits 1–0 to the desired value before setting bit 7 to enable PIOS operation.

Bit 6 of the Control register enables or disables interrupts from the PIOS, if and when it encounters interrupt entries.

Other registers that affect the PIOS include:

- The AFSC (“Port C Alternate Function Select (AFSC)” on page 35) controls which Port C pins are controlled by the PIOS
- The DDRC and OCRC (“Data Direction Registers and Output Control Registers” on page 34) control Port C pin functions
- The System Configuration Register (SCR, page 79) controls whether the processor as well as the PIOS can access on-chip RAM, and if so, at what addresses.

Creating PIOS Entries in On-Chip RAM

Bit 6 of the System Control Register must be 1 to allow the processor to set up PIOS entries in on-chip RAM. Each entry includes 4 bytes and starts at an address that is a multiple of 4.

When the processor first enables the PIOS, the PIOS always processes the entry at `xFF00–3H` or `FFF00–3H` first.

If a new PIOS sequence is needed in the future, software can build the next sequence in on-chip RAM while the PIOS is still running its current sequence, provided that the total length of both sequences is 64 entries or less.

To accomplish this task, software builds the next sequence in those parts of the PIOS RAM that are not used by the current sequence. Next, software can store the first entry of the next sequence at location `xFF00–3H` or `FFF00–3H`. When the PIOS has completed its current sequence, software can start the next sequence as described in “Starting and Polling the PIOS” on page 55.

Digital Data Output Entries. Figure 9 illustrates the format of a PIOS Digital Data Output entry. The first two bytes contain a delay value, in the units selected by bits 1–0 of the PIOS Control register. As with all Z80/180 multi-byte binary values, the least significant byte is stored first, at the lower address. An all-0 delay value indicates 65,536 clocks. The third byte holds a value that the PIOS is to output to Port C after the specified delay. Bits 7–2 of the 4th and last byte hold a

non-zero address of the next entry, with 00 in bits 1–0 to identify this entry as a Digital Data entry.

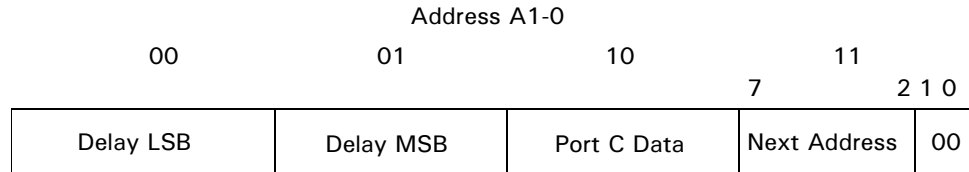


FIGURE 9. PIOS DIGITAL DATA OUTPUT ENTRY

Analog Data Output Entries. Figure 10 illustrates the format of a PIOS Analog Data Output entry. The first byte, and bits 5–0 of its second byte, contain a delay value in the units selected by bits 1–0 of the PIOS Control register. As with all Z80/180 multibyte binary values, the least significant byte is stored first, at the lower address. An all-zero delay value indicates 65,536 clocks. The third byte holds bits 9-2, and bits 7-6 of the 2nd byte contain bits 1-0, of a 10-bit value that the PIOS is to output to the D/A Converter after the specified delay. Bits 7-2 of the 4th and last byte hold a non-zero address of the next entry, with 10 in bits 1-0 identifying this entry as an Analog Data Output entry.

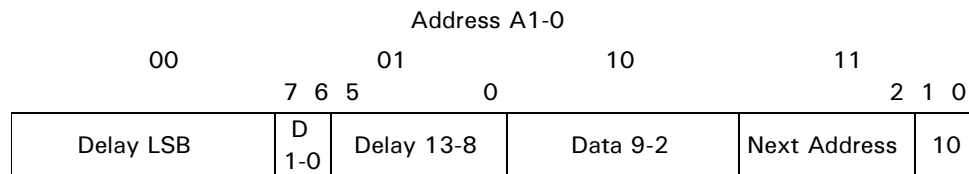


FIGURE 10. PIOS ANALOG DATA OUTPUT ENTRY

Analog Data Input Entries. Figure 11 describes the format of a PIOS Analog Data Input entry. The first two bytes contain a delay value, in the units selected by bits 1-0 of the PIOS Control register. As with all Z80/180 multibyte binary values, the least significant byte is stored first, at the lower address. An all-0 delay value indicates 65,536 clocks. After the specified delay, bits 2-0 of the third byte hold a channel (port D pin) number, the voltage on which is to be converted to a digital value. Bits 7-2 of the fourth byte contain the non-zero address of the next entry, and 01 in bits 1-0 identify this entry as an Analog Data Input entry.

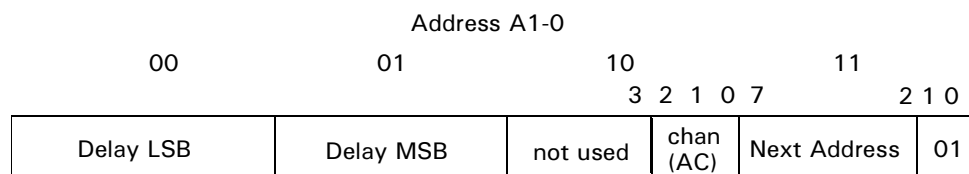


FIGURE 11. PIOS ANALOG DATA INPUT ENTRY

Interrupt Entries. Figure 12 illustrates the format of a PIOS Interrupt entry. The first two bytes contain a delay value, in the units selected by bits 1-0 of the PIOS Control register. As with all Z80/180 multibyte binary values, the least significant byte is stored first, at the lower address. An all-0 delay value indicates 65,536 clocks. The third byte of an Interrupt entry is not used. Bits 7-2 of the fourth byte hold the non-zero address of the next entry, with 11 in bits 1-0 identifying this entry as an Interrupt entry.

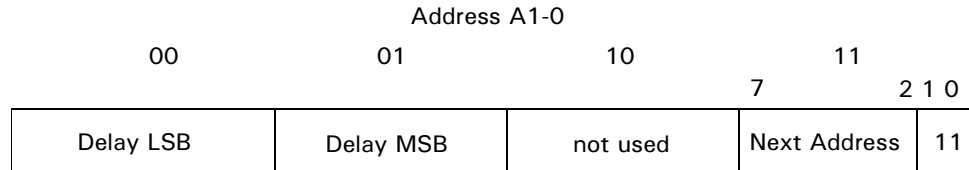


FIGURE 12. PIOS INTERRUPT ENTRY

End of Sequence Entry. Figure 13 illustrates the format of an entry that identifies the end of a PIOS sequence. This type of entry is 0 in bits 7-2 of its fourth byte. The other bits in the entry can be any value. When the PIOS fetches such an entry, it stops processing immediately, clearing bit 7 of its Control register.

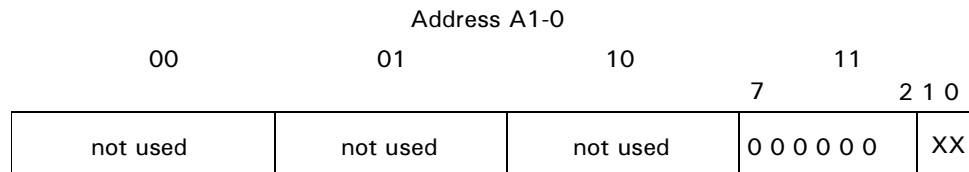


FIGURE 13. PIOS END-OF-SEQUENCE ENTRY

Starting and Polling the PIOS

After software has built a complete PIOS sequence in PIOS RAM, including the first entry in addresses xFF00-3H or FFF00-3H, and the PIOS has completed any previous sequence, software proceeds as follows:

1. If a change is required to the clock selection, write the new value to the Control register as a separate step. This action ensures accuracy of the delay associated with the first entry.
2. Write a 1 to bit 7 of the Control register, maintaining the prior clock selection in bits 1-0. If the sequence contains one or more Interrupt entries, enable PIOS interrupts by writing a 1 in bit 6 of this value.
3. The PIOS fetches all bits of each entry simultaneously. While the PIOS runs, the software can track its progress through the sequence by reading the PIOS Address/Type register.

4. If the Next Address field of an entry is non-zero, the PIOS counts down the delay value specified in the first two bytes, using the clock selected by bits 1-0 of the Control register.

Note: For a D/A entry, the PIOS automatically loads zeroes into the two most significant bits of the delay counter. D/A entries are limited to a 16,383 clock delay, or 65,536 for an all-zero value.

5. After the PIOS has counted the delay value down to zero, it performs the action specified by bits 1-0 of the last byte of the entry:
 - Writes an 8-bit value to Port C,
 - Writes a 10-bit value to the D/A converter,
 - Starts an A/D conversion on a particular channel, or
 - Requests an interrupt.
6. Software can stop PIOS operation at any time, by clearing bit 7 of the Control register. Otherwise, when the PIOS reaches an entry with 0 in its Next Address field, it clears bit 7 of the Control register and stops. Software polls the Control register to detect when this occurs, or the last active entry in the sequence can be an interrupt entry.

Handling PIOS Interrupts

If bit 6 of the PIOS Control register is 1 and the PIOS fetches an Interrupt entry, it requests an interrupt. When the conditions noted in “On-Chip Interrupts” on page 18 have been met for this request, the processor fetches the address of the PIOS interrupt service routine from memory at address (I : IL : 18). The ISR performs the following operations:

1. Save as many registers of the interrupted process as it may use (worst case), using PUSH, EX AF, AF', and/or EXX instructions.
2. Read the PIOS Status register and verify that bit 5 (Interrupt Pending), is 1. If not, the ISR can store logging information in memory if desired, before restoring the registers and using EI and RET instructions to return to the interrupted process.
3. Clear bit 5 if IP is 1, in the value read in step 2, and write the result back to the Control register. This action clears the IP bit and prevents further interrupts for the same PIOS event.
4. Read the Address/Type register, if there is more than one Interrupt entry in the PIOS sequence, read the Address/Type register to determine which entry caused this interrupt.
5. Reads ADC Control Register 1 and checks the conversion complete bit if this interrupt follows the start of an A/D conversion by the PIOS. Assuming that the delay in the interrupt entry has guaranteed that the conversion is complete, read the result value from the ADC Result register and, if desired, the two least significant bits from Control Register 1.

Note: Software can use either an A/D interrupt or a PIOS interrupt to signal the completion of an A/D conversion initiated by the PIOS.

6. Other processing is application-dependent. If the interrupt marks the end of a sequence, and another sequence must be run immediately, the ISR can start that sequence as described above.
7. Restore the saved registers, then use `EI` and `RET` instructions to reenable interrupts and return to the interrupted process.

ASYNC SERIAL COMMUNICATIONS INTERFACES (ASCIs)

The ASCIs are asynchronous full-duplex UARTs with the following features:

- 7- or 8-bit data
- Odd, even, or no parity
- One or two Tx Stop bits
- Checking for Parity, Framing, and Overrun errors
- Break Generation and Detection
- Choice of two Baud Rate Generators (BRGs)
- Rx and Tx interrupts
- DCD and CTS pins on ASCI0
- Operation with the on-chip DMA channels
- A MULTIPROCESSOR mode with an extra bit designating address vs. data characters

The registers associated with the ASCIs are described in section “Async Serial Communications Interface (ASCI) Registers” on page 131. Control registers A and B (CNTLA0, CNTLA1, CNTLB0, and CNTLB1) and the Extension Control registers (ASEXT0, ASEXT1) are typically written one time each, to configure an ASCI. The Status registers (STAT0 and STAT1) indicate the current state of each ASCI’s Receiver and Transmitter. Under control of this status, software can write bytes to be transmitted to the Transmit Data Registers (TDR0 and TDR1), and read received bytes from the Receiver Data Registers (RDR0 and RDR1).

Basic Clocking

Each ASCI uses the same clock for both transmitting and receiving. Because the Z80S183 does not include `CKA` pins for either channel, clocking for each ASCI must be derived from one of its two baud rate generators (BRG). Furthermore, bits 2-0 of each CNTLB register must not be left 111, as they are after a reset.

To use the older version of the BRG, which is compatible with the original ZiLOG Z80180:

- Clear bit 3 (BRG Mode) of the Extension Control register, to select the old BRG.
- Write bit 5 (PS) and bits 2-0 (SS) in the CNTLB register to select the value by which the old BRG divides the PHI clock to obtain the Basic Clock, as described in Table 14.

TABLE 14. OLD BRG DIVISION FACTORS

| PS (CNTLB5) | SS (CNTLB2-0) | Basic Clock = PHI Divided By |
|-------------|---------------|------------------------------|
| 0 | 000 | 10 |
| | 001 | 20 |
| | 010 | 40 |
| | 011 | 80 |
| | 100 | 160 |
| | 101 | 320 |
| | 110 | 640 |
| | 111 | No clock |
| 1 | 000 | 30 |
| | 001 | 60 |
| | 010 | 120 |
| | 011 | 240 |
| | 100 | 480 |
| | 101 | 960 |
| | 110 | 1920 |
| | 111 | No clock |

To use the newer BRG, which is compatible with the ZiLOG SCC family:

- Set bit 3 (BRG mode) of the Extension Control register, to select the new BRG.
- Write a 16-bit time constant to the Time Constant Low and High registers. This value is the factor by which the PHI is divided to produce the basic clock, divided by 2, minus 2. The new BRG calculates the basic clock as:

$$\text{basic clock} = \text{PHI} / (2 (\text{TC}+2))$$

Clock Mode. Since neither ASCI has a clock pin on the Z80S183, the 1X ISOCHRONOUS mode that can be used on other 8018x family members, cannot be used on the Z80S183, and bit 4 (X1 clock) in the Extension Control register must be 0.

If the DR bit (bit 3) in the CNTLB register is 0, the ASCI divides its basic clock by 16 to obtain the serial bit rate. If DR is 1, the ASCI divides the basic clock by 64.

Async Transmission

Figure 14 illustrates a single asynchronous character on the TXA or RXA pin. When a transmitter is disabled, or when it has completed sending all characters that the software or a DMA channel has provided, the transmitter maintained a High on the Tx/D pin. This state is also called 1 or Mark.

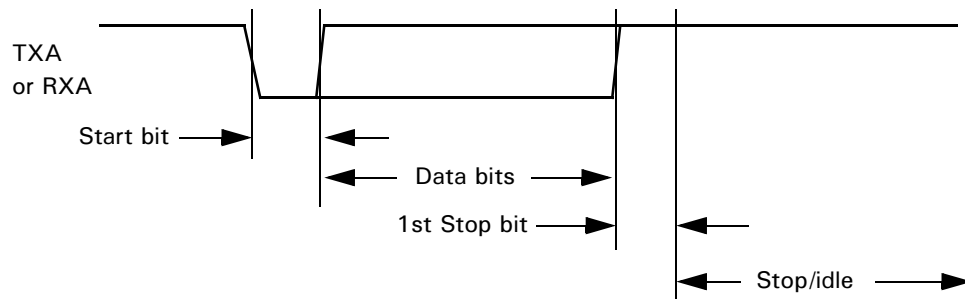


FIGURE 14. AN ASYNCHRONOUS CHARACTER

When software or a DMA channel provides a character to an idle ASCI transmitter:

1. It drives TXA Low on the next falling edge of the basic clock, to begin a start bit, and maintains TXA Low for 16 or 64 basic clocks depending on the value of DR.
2. It switches TXA to the state of the least significant bit (bit 0) of the character, and holds that for 16 or 64 clocks.
3. It repeats step 2 for each next-more-significant bit of the character, through the most significant bit (bit 6 or 7) of the character, and for a parity or MP bit if one of these is enabled.
4. It drives TXA High (1, Mark) again for a stop bit, and maintains it High for at least 16 or 64 clocks.
5. It continues this High of another 16 or 64 clocks if bit 0 of its CNTLA register is 1 to select two Tx stop bits.

The Tx character is now complete. If software or a DMA channel has provided another character to send, the transmitter begins another Start bit as described above. Otherwise, it maintains a High on TXA High until a character is provided.

Async Reception

The Receiver has a more complex task. When it is first enabled, or after a character has been received the receiver samples the RXA pin on the rising edge of each basic clock.

1. If RXA is High it remains in this state of waiting for a Start bit.
2. If the receiver samples RXA Low, it counts off half a bit time (8 basic clocks if DR is 0, or 32 basic clocks if DR is 1) then samples RXA again. If RXA is High, the receiver rejects the transient Low state on RXA as not representing a Start bit, and returns to step 1. The RXA pin is sampled again.
3. If RXA is still Low after half a bit time, it validates the Start bit. The receiver then counts off the number of basic clocks in a bit—16 if DR is 0, 64 if it is 1, — and samples RXA for the Least Significant data bit (bit 0). It continues this

routine for each progressively significant data bit (through bit 6 or 7), and for a Parity or MP bit if either is enabled.

Finally, the receiver counts off 16 or 64 more basic clocks and samples the first Stop bit. If the receiver's basic clock is close to the clock that the transmitter used to send the character, this occurs near the middle of the first Stop bit. If there is no noise or other error on the line, the receiver samples the Stop bit as High/1/Mark. However, if the receiver's and transmitter's clocks were sufficiently different, or if there is noise or an error on the line, the receiver samples the stop bit as Low/0/Space. This latter situation is called a *framing error*. When a framing error occurs, the receiver sets an error bit that accompanies the character through the receiver FIFO. The receiver then sets the FE bit in the STAT register when the character becomes the oldest one in the FIFO.

The received character is now complete, and the receiver returns to step 1 and samples the line for a new Start bit.

Combined Effect of BRG and Clock Mode

Combining the operation of the BRGs and the division by 16 or 64 performed by the transmitter and receiver, where:

- Serial rate is in bits/second
- PHI is the system clock frequency
- PS is the value written to bit 5 of CNTLB (0 or 1)
- ^ indicates exponentiation (2 to the power)
- SS is the binary value of bits 2-0 of CNTLB (0 thru 6)
- DR is bit 3 of CNTLB (0 or 1)
- TC is the 16-bit value in the ASTCL and ASTCH registers:

Old BRG: $serial\ rate = PHI / ((10 + 20*PS) * (2 ^ SS) * (16 + 48*DR))$

New BRG: $serial\ rate = PHI / (2 * (TC + 2) * (16 + 48*DR))$
 $TC = (PHI / (serial\ rate * (32 + 96*DR))) - 2$

Options

7 or 8 Data Bits. If bit 2 of CNTLA is 0, the transmitter sends, and the receiver accumulates, 7 data bits per character. If CNTLA2 is 1, the transmitter sends, and the receiver accumulates, 8 data bits per character.

Parity. If bit 1 of CNTLA is 1, the transmitter accumulates and sends a Parity bit after the data bits, and the receiver samples and checks these bits. If CNTLA1 is 1, a 1 in bit 4 of CNTLB selects odd parity and 0 in bit 4 selects even parity. Odd Parity means that a correct character includes an odd number of 1 bits, including the Parity bit. Even Parity means that a correct character includes an even number of 1 bits.

If the receiver samples the Parity bit in the incorrect state for a character, it sets an error bit that accompanies the character through the receiver FIFO. The receiver then sets the PE bit in the STAT register when the character becomes the oldest one in the FIFO.

Transmit Stop Bits. If bit 0 of CNTLA is 0, the transmitter sends a minimum of one Stop bit between characters. If CNTLA0 is 1, it sends at least two Stop bits between characters. Selecting two Stop bits has been known to work around timing mismatches between a transmitter and a receiver.

Note: The ASCI receivers on the Z80S183 check only one Stop bit, regardless of bit 0 of CNTLA. This is also true of ASCI receivers on other current 8018x family members, as well as other UARTs, but on the original Z80180, the ASCIs actually checked two Stop bits if CNTLA0 was 1.

Status

Break Conditions. Break conditions date back to the early days of async communications using Teletypewriters, which were functionally half-duplex in that text could only flow in one direction at a time.

If the operator of a receiving Teletype had a condition, or something to say, and wanted to interrupt the data from the other machine, the operator could press a Break key. Pressing this key drove the line to the 0/Space state for several character-times, which switched on a light at the other machine, and stopped its paper tape reader if it had been in use.

A Break is still defined as at least two character times of consecutive zeroes on the line. A receiver sees this as one or more all-0 characters with Framing errors.

Software can send a break by setting bit 0 of the Extension Control register. Before doing this, software can ensure that any characters previously written to the Transmit Data Register have been sent, by monitoring bit 0 of the Extension Control register. The duration of a transmitted Break is under software control—the Break is terminated by clearing the ASEXT bit to 0.

While receiving an all-zero character with a Framing error indicates a Break, bit 1 of the Extension Control register provides more specific break detection. When the receiver detects an all-zero character with a framing error, it sets a Break status bit that accompanies the character through the receiver FIFO. The receiver sets bit 1 in the ASEXT register to 1 when such a character becomes the oldest one in the FIFO.

After detecting a break, the receiver does not assemble any further characters until the RxA pin returns to High/1/Mark, signalling the end of the Break condition.

Rx Overrun. The ASCIs in the Z80S183 contain 4-character Rx FIFOs between the Rx Shift registers and Receive data registers. If a receiver receives a character and there are already 4 characters in the FIFO, an overrun status bit is set that accompanies the *preceding* character through the FIFO. The receiver then sets the OVRN bit (bit 6 in the STAT register) to 1 when that character becomes the oldest in the FIFO.

The receiver discards the character that triggers the overrun condition, *and subsequent characters*, until the last good character has come to the top of the FIFO so that OVRN is 1, and software writes a 0 to the EFR bit to clear OVRN.

Receive Status. There are four receive status bits in the STAT register and one in the Extension Control register. RDRF, bit 7 of STAT, is 1 whenever there is at least one received character in the Rx FIFO. RDRF is cleared:

- When the software or a DMA channel has read all characters out of the Rx FIFO
- By a reset
- In I/O STOP mode
- On ASCI0, when the $\overline{\text{DCD0}}$ pin is autoenabled and is High.

FE, PE, and OVRN in the STAT register are 1 when a character with a framing error or parity error, or the last character before an Rx Overrun, comes to the top of the receive FIFO. Similarly, bit 1 in the Extension Control register is 1 when a break character comes to the top of the FIFO.

Any of these bits remain 1 even if the character associated with the condition is read out of the receive FIFO. That is, these bits *latch* an error or exception condition. All four of these bits are cleared to 0 by reading CNTLA, clearing bit 3 (Error Flag Reset, or EFR) to 0, and writing the result back to CNTLA.

This action also allows the receiver to put subsequent characters into the receive FIFO after an Overrun, and if the receiver is handled by a DMA channel, allows the channel to service the receiver again.

Modem Control/Status. The only ASCII modem control or status signals on the Z80S183 are DCD0 and CTS0. The state of the DCD0 pin can be read as bit 2 of the STAT0 register, and that of CTS0 as bit 5 of CNTLB0. Both bits read as 1 if the pin is High/inactive.

Bit 6 in the Extension Control register controls whether DCD0 automatically enables and disables the receiver. Bit 5 controls whether CTS0 automatically enables and disables the transmitter.

In each case, if one of these ASEXT bits is 0, as it is after a reset, then a Low on the pin allows the module to operate. However, while a High disables the module. If one of these ASEXT bits is 1, software can read the state of the pin, but this state does not affect the hardware.

When ASEXT0 bit 6 is 0, a High on DCD0 forces the status bits RDRF, PE, FE, OVRN, and RxBreak to 0. If DCD0 goes Low thereafter, the next read of the STAT register still contains a 1 in bit 2 (DCD0). Subsequent reads of STAT indicate current status.

When ASEXT0 bit 5 is 0, a High on CTS0 clears the TDRE bit in STAT. This action prevents software or a DMA channel from putting further Tx data into TDR. As many as three characters (one from the Tx shift register, two from the 2-stage Tx FIFO) can appear on TDA0 after $\overline{\text{CTS0}}$ goes High.

DMA Operation

On the Z80S183, data can be transferred to or from either ASCI by either DMA channel. Setup for these operations is covered in the following ASCI and DMA topics:

“Starting a Transmitter”, later on this page,
“Starting a Receiver” on page 63,
“Handling ASCI Interrupts” on page 64, and
“Setting Up a DMA Transfer” on page 40.

A DMA channel used with an ASCI transmitter must be programmed to use the appropriate TDRE flag as its DMA request, and must be set up for edge-sensitive DMA request.

A DMA channel used with an ASCI receiver must be programmed to use the appropriate RDRF flag as its DMA request, and can be used in either edge- or level-sensitive mode. In this application, software sets both the RIE bit in the STAT register, and bit 7 of the Extension Control register, enabling ASCI receive interrupts in the event of errors, but not for every received character.

Note: The signal that an ASCI receiver provides as a Request to a DMA channel, is not simply RDRF but rather (RDRF and not PE and not FE and not OVRN and not RxBreak). DMA operation will be suspended if any of these errors or exceptional conditions occurs. This operation is performed so that software can determine the point in the receive data stream at which the error or condition occurred.

Programming Techniques

Starting a Transmitter. A Software can enable a transmitter at the same time as its associated receiver, or separately, as follows:

1. Write the CNTLB register to set the clocking and basic options.
2. Write the Extension Control register to select the BRG mode and, on ASCI0, the mode of the CTS0 pin.
3. Set bit 0 (TIE) in the STAT register to enable transmit interrupts, otherwise, clear it.
4. Set bit 5 to 1 of CNTLA to enable the transmitter.

The TDRE flag is set immediately. If transmit interrupts were enabled in step 3, an interrupt occurs immediately.

Note: If a DMA channel is to provide data to the transmitter, software can set it up whenever transmit data is available, including selecting TDRE as the DMA Request. Edge-sensitivity is required on the DMA request for transmit/output devices. If TDRE is set before the DMA channel is started, device software sets up the DMA to not include the first Tx character, and then write first character to the TDR, to initialize the ASCI-DMA handshake.

Starting a Receiver. A Software can enable a receiver at the same time as its associated transmitter, or separately, as follows:

1. Write the CNTLB register to set the clocking and basic options.

2. Write the Extension Control register to select the BRG mode and, on ASCI0, the mode of the $\overline{\text{DCD}}\overline{\text{U}}$ pin. If the receiver is to be handled by a DMA channel, set bit 7 of the Extension Control register to 1 to block the RDRF flag from requesting a receive interrupt.
3. Set bit 3 (RIE) in STAT to 1 if receive interrupts are desired, otherwise clear it.
4. Set up the DMA channel if the receiver is to be handled by a DMA channel). Select the channel, including selecting RDRF as the DMA Request signal.
5. Set the bit 6 in CNTLA to enable the receiver.

RDRF is 1 when there is data in the Rx FIFO. If RDRF interrupts are enabled, an interrupt occurs at that time.

Polled Transmission. After starting a transmitter without interrupts enabled, while there is data to send:

1. Read the STAT register (preferably more often than once per character-time) until bit 1 (TDRE) is 1.
2. Write the next character to be transmitted to the TDR. If there is more data to transmit, return to step 1.

Polled Reception. After starting a receiver without interrupt enabled:

1. Read the STAT register, at least once per character time, until it detects bit 7 (RDRF).
2. Read a received character from the RDR.
3. Process process the received character, then return to step 1.

Handling ASCI Interrupts. A 1 in bit 0 (TIE) of an ASCI's STAT register enables transmit interrupts, and a 1 in bit 3 (RIE) enables receive interrupts. If any of following are true:

- TIE and TDRE are both 1
- RIE is 1 and any of OVRN, PE, FE, or RxBreak (ASEXT bit 1) are 1
- RIE is 1, bit 7 of the Extension Control register is 0, and RDRF is 1
- RIE is 1, bit 6 in the Extension Control register is 0, and the $\overline{\text{DCD}}\overline{\text{U}}$ pin is High for ASCI0

then an ASCI requests an interrupt from the processor. When the conditions listed in "On-Chip Interrupt Handling" on page 19 are met with respect to this request, the processor fetches the address of the interrupt service routine (ISR) from (I : IL : 14) for ASCI0, or from (I : IL : 16) for ASCI1.

An ASCI ISR that handles both kinds of interrupts must:

1. Save as many registers as it may use, by means of PUSH, EX EX AF,AF', and/or EXX instructions.
2. Read the STAT register for this ASCI.

3. If TIE and TDRE in STAT are both 1:
 - a. If another Transmit character is available, write it to the TDR.
 - b. If not, clear the TIE bit until another Tx character is available.
4. If RIE in STAT is 1 and any of OVRN, PE, FE, or RxBreak (ASEXT bit 1) are 1:
 - a. If bit 7 of the Extension Control register is 1, indicating that a DMA channel is handling receive data, read the DSTAT register, clear the DE and DWE bits for that DMA channel, and write the result back to DSTAT.
 - b. Read the CNTLA register, clear bit 3 (EFR) to 0 and write the result back to CNTLA
 - c. Read the associated character from the RDR
 - d. For PE or FE conditions, applications can discard the character, replace it with a standard error character, or treat it like other characters.
 - e. For an OVRN condition (without any other error), most applications process this *last good* character normally, as in step 5. An application may then post an *overrun occurred here* notification in the received-data stream.
 - f. For Break, most applications discard the all-0 character. An application may post a *break occurred here* notification in the received-data stream. The receiver does not assemble more all-0 characters, but waits for RDA to go High before searching for a new Start bit.
5. If no errors were found in step 4, but RIE is 1, bit 7 of the Extension Control register is 0, and RDRF is 1, read the next received character from the RDR. Process it, of which the simplest case is storing it at the next memory location in a buffer.
6. If for ASCI0, RIE is 1, bit 6 in the Extension Control register is 0, and bit 2 of STAT is 1, carrier has been lost, and the ASCI receiver does nothing until the carrier returns. In this case, either clear RIE or set bit 6 in the Extension Control register, to prevent further interrupts because \overline{DCD} is High.
7. Optionally, at this point software can read STAT again, and return to step 3 if RIE and RDRF are both 1, or TIE and TDRE are both 1. This option saves on interrupt overhead.)
8. If the ISR disabled the receive DMA channel in step 4a, it restarts the channel.
9. Finally, the ISR must restore the saved registers, and return to the interrupted process by means of EI and RET instructions.

Multiprocessor Mode

In this mode, the transmitter sends, and the receiver expects, an extra bit between the data and stop bits, that differentiates address from data characters. Other manufacturers' devices have a similar mode called NINE-BIT mode. To enable this mode for both the transmitter and receiver, set bit 6 of the CNTLB register as part of initializing the ASCI.

Note: MULTIPROCESSOR mode cannot be used with parity generation and checking.

In MULTIPROCESSOR mode, data is grouped into frames or messages, each preceded by an address character, which differs from data characters in that its extra bit is 1.

To send a frame:

1. Wait, if necessary, for the TDRE bit (bit 1 of the STAT register) to be 1, indicating that a new Tx character can be written to the TDR. The next two steps can be performed in an ASCII Interrupt Service Routine.
2. Read the CNTLB register, set bits 7-6 to 11, and write the result back to CNTLB. This action sets up the transmitter to send the first character of the frame with the extra bit 1.
3. Write the address value for the intended destination device to the TDR.
4. For each data character in the frame, again wait, if necessary, for the TDRE bit (bit 1 of the STAT register) to be 1. The following two steps can be performed in an ASCII ISR.
5. Read CNTLB, clear bit 7, and write the result back. These actions cause the transmitter to send the next character with the extra bit 0.
6. Write the next data character to the TDR. If there are more characters in the frame, return to Step 4.

If bit 6 of CNTLB is 1, to receive a frame in MULTIPROCESSOR mode:

1. Read CNTLA, set bit 7, and write the result back to CNTLA. This process conditions the receiver to ignore data characters, that contain a 0 in the extra bit.
2. Wait for the RDRF bit in the STAT register to be 1. The following steps can be done in an ASCII ISR.
3. Read CNTLA and verify that bit 3 is 1, namely that the next available character is an address character. If not, read the RDR, discard the data character, and return to step 2.
4. Read the RDR and check whether the value obtained indicates a frame destined for this processor (there may be one or more unique addresses for this node, plus a *broadcast* and/or *group* address) If not, discard the character and return to step 2. The hardware ignores the data characters in the frame.
5. Optionally, store the address character in memory. (The following steps may vary depending on the address.)
6. Read CNTLA, clear bit 7, and write the result back to CNTLA. This process conditions the receiver to assemble the following data characters.
7. For each data character in the frame, wait for the RDRF bit in the STAT register to be 1. The following steps can be processed in an ASCII interrupt service routine.

8. Read the data character from the RDR and store it in memory. (If frames are not a fixed length, software determines the frame length from its content.) If the frame is not complete, return to step 7.
9. Check the frame for checking or validation information, check it. Most protocols specify that a receiving node ignores a frame that fails checking/validation.
10. Process the frame based on its content. This routine is application-dependent.

CLOCKED SERIAL INPUT/OUTPUT MODULE (CSI/O)

The CSI/O allows synchronous communication with serial memories, peripherals, and other processors that include compatible interfaces. The CSI/O is a half-duplex interface that can send or receive 8-bit bytes, but not both simultaneously.

The CSI/O includes separate receive and transmit data pins, RXS and TXS, plus a clock pin, CKS, that can be either an input or an output. In either direction, CKS switches only during active data characters on the RXS and TXS pins.

The bit rate for the CSI/O can be up to $\text{PHI}/20$ for an internally-generated clock, and even faster with an externally-generated clock.

The CSI/O Control (CNTR) and Data (TRDR) registers are described in section “Clocked Serial Input/Output Module (CSI/O)” on page 67.

Clock Selection

After Reset, bits 2-0 of the CNTR are 111B, which conditions the CKS pin to be an input. If this Z80S183 is to provide the clock to the other station(s), write bits 2-0 with one of the values 000–110B depicted in Table 15. This value determines the factor by which the CSI/O divides PHI to produce the clock it drives onto CKS.

TABLE 15. CSI/O CLOCK RATE SELECTION

| CNTR 2-0 | CKS Bit Rate |
|----------|----------------------------|
| 000 | PHI/20 |
| 001 | PHI/40 |
| 010 | PHI/80 |
| 011 | PHI/160 |
| 100 | PHI/320 |
| 101 | PHI/640 |
| 110 | PHI/1280 |
| 111 | Input from external source |

Note: Wait at least one bit time after the transmitter clears the TE bit, or the receiver clears the RE bit, before changing the baud rate.

Operation

The clock signal on CKS and the data signals on TXS and RXS feature the same basic relationship whether this Z80S183 is driving or receiving the clock on CKS, and whether it is sending on TXS or receiving on RXS.

Figure 15 illustrates this relationship, along with the state of the flags in the CNTR.

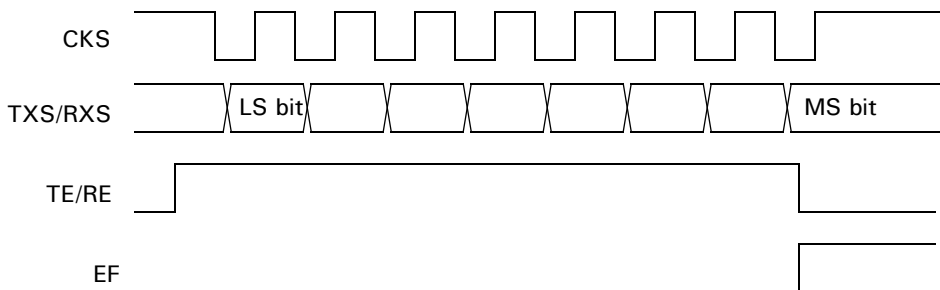


FIGURE 15. CSI/O OPERATION

CKS is High between bytes. The CSI/O operates as follows in the four possible cases of clock and data sourcing:

Output Clock, Output Data. After software writes a byte to be transmitted to the TRDR and sets the TE bit in the CNTR, the CSI/O drives CKS Low, and shortly thereafter drives bit 0 of the byte onto TXS.

Thereafter, it toggles CKS at the selected clock rate, driving each next-more-significant bit from each falling edge on TXS, until it has driven bit 7 onto TXS. Then the CSI/O clears the TE bit and sets the EF bit in the CNTR. The CSI/O drives CKS High to complete the operation.

Input Clock, Output Data. Software must write the byte to be transmitted to the TRDR, and then set the TE bit in the CNTR, before the external clock source drives the CKS pin Low for the first bit of the byte.

At each falling edge of CKS, CSI/O drives a bit onto TXS, beginning with bit 0. After driving bit 7 onto TXS, the CSI/O clears the TE bit and sets the EF bit in the CNTR.

Output Clock, Input Data. After software sets the RE bit in the TRDR, the CSI/O drives CKS Low, and thereafter toggles CKS at the selected clock rate, for a total of eight falling and eight rising edges. At each rising edge on CKS, the CSI/O samples one bit of the byte into the TRDR, starting with bit 0. After sampling bit 7, the CSI/O clears the RE bit and sets the EF bit in the CNTR.

Input Clock, Input Data. Software must set the RE bit in the CNTR, before the external clock source drives CKS Low for the first bit of the byte. At each rising edge on CKS, the CSI/O samples one bit of the byte into the TRDR, starting with bit 0. After sampling bit 7, the CSI/O clears the RE bit and sets the EF bit in the CNTR.

Transmitting a Byte. Both the TE and RE bits in the CNTR must be 0 before another byte can be sent. At that point, software must:

1. Write the byte to be transmitted into the TRDR.
2. Write a 1 to the TE bit (bit 4) in the CNTR, with the selected clock control value in bits 2-0. If no interrupt is necessary when the byte has been sent, write a 0 in bit 6 (EIE) of this register. Otherwise, write a 1 for the interrupt service routine.

Receiving a Byte. Both the TE and RE bits in the CNTR must be 0 before software can condition the CSI/O to receive another byte. At that point software must:

1. Write a 1 to the RE bit (bit 5) in the CNTR, with the selected clock control value in bits 2-0. For polled operation, write a 0 in bit 6 (EIE) of this value. Otherwise, write a 1 and clear the transmit flag for the Interrupt Service Routine.
2. Read the CNTR (for polled operation) periodically or in a tight loop, until RE is 0 and EF is 1. For interrupt-driven operation, the following step is performed in the interrupt service routine, as described in the next topic.
3. Read the TRDR to acquire the byte and clear the EF flag. If the sending station is to send more data, return to step 1.

Cancelling Transmission or Reception. Software can cancel a byte transmission or reception in progress, by writing 0s to the TE and RE bits. Avoid cancellation when the Z80S183 is sourcing CKS, because this action may hang the remote station's hardware in mid-byte. When operating with an external clock, software may cancel byte transmission or reception after a time-out period expires, indicating that the remote station has nothing more to send, or cannot accept further data.

Handling CSI/O Interrupts

If software sets bit 6 (EIE) in the CNTR when it starts a transmit or receive operation, the CSI/O requests an interrupt when it completes the operation and sets the EF bit. When the conditions listed in "On-Chip Interrupt Handling" on page 19 are met with respect to this request, the processor responds by fetching the address of the CSI/O interrupt service routine (ISR) from memory at address (I : IL : 12). The ISR then:

1. Saves as many registers as it may use, using PUSH, EX AF, AF', or EXX instructions.
2. Reads the CNTR. If the EF bit is 0, the ISR may log this unknown interrupt, before restoring the registers and returning to the interrupted process using EI and RET instructions.

Note: If the Transmit flag is cleared, read the TRDR.

3. Process the byte.
 - a. If further reception is necessary, write a 1 to RE as in step 1 of "Receiving a Byte", previously.

- b. If data is to be sent, write the first byte to the TRDR (this action clears the EF flag). Then set TE in the CNTR, as in steps 1–2 of “Transmitting a Byte”, previously. Set the Transmit flag for the next interrupt.
4. If the Transmit flag is set:
 - a. Write the next byte to the TRDR (this clears EF).
 - b. Then set TE in the CNTR, as in steps 1-2 of “Transmitting a Byte”, previously.
5. If there is data to be received, perform a dummy read of the TRDR to clear the EF flag, then write a 1 to RE as in step 1 of “Receiving a Byte”, previously. Clear the Transmit flag.
 - a. If no further data is to be sent or received, perform a dummy read of the TRDR to clear the EF flag.
6. Restore the saved registers and return to the interrupted process using EI and RET instructions.

I/O REGISTERS

“Processor Description” describes the processor registers and the Z80S183’s programming model. This section describes the registers in I/O space, that control the operation of the overall device and its on-chip peripherals.

REGISTERS SUMMARY

I/O registers on the Z80S183 are divided into two classes, the 80180 registers and other on-chip registers.

The 80180 registers:

- Are located at addresses between 0000 and 003FH
- Can be relocated to 0080–00BFH or 00C0–00FFH
- Require three cycles per I/O instruction

Other on-chip registers:

- Are located at addresses between 0040 and 007FH, and
- Require four cycles per I/O instruction.

All the on-chip registers on the Z80S183 decode 16-bit I/O addresses and require A15-8 to be all 0. These registers must be accessed using IN0 and OUT0 instructions.

The following table includes all on-chip registers in both classes. I/O addresses not described are not used.

| Register Name | Addr (hex) | Register Name | Addr (hex) |
|------------------------------------|------------|---------------------------------|------------|
| ASCIO Control Register A | 00 | ASC11 Control Register A | 01 |
| ASCIO Control Register B | 02 | ASC11 Control Register B | 03 |
| ASCIO Status Register | 04 | ASC11 Status Register | 05 |
| ASCIO Tx Data Register | 06 | ASC11 Tx Data Register | 07 |
| ASCIO Rx Data Register | 08 | ASC11 Rx Data Register | 09 |
| CSI/O Control Register | 0A | CSI/O Data Register | 0B |
| PRT0 Timer Data Register Low | 0C | PRT0 Timer Data Register High | 0D |
| PRT0 Timer Reload Register Low | 0E | PRT0 Timer Reload Register High | 0F |
| Timer Control Register | 10 | | |
| ASCIO Extension Control Reg | 12 | ASC11 Extension Control Reg | 13 |
| PRT1 Timer Data Register Low | 14 | PRT1 Timer Data Register High | 15 |
| PRT1 Timer Reload Register Low | 16 | PRT1 Timer Reload Register High | 17 |
| Free Running Counter | 18 | | |
| ASCIO Time Constant Low | 1A | ASCIO Time Constant High | 1B |
| ASC11 Time Constant Low | 1C | ASC11 Time Constant High | 1D |
| Clock Control Register | 1E | CPU Control Register | 1F |
| DMA0 Source Addr Register L | 20 | DMA0 Source Addr Register H | 21 |
| DMA0 Source Addr Register B | 22 | DMA0 Dest Addr Register L | 23 |
| DMA0 Dest Addr Register H | 24 | DMA0 Dest Addr Register B | 25 |
| DMA0 Byte Count Register L | 26 | DMA0 Byte Count Register H | 27 |
| DMA1 Memory Addr Register L | 28 | DMA1 Memory Addr Register H | 29 |
| DMA1 Memory Addr Register B | 2A | DMA1 I/O Addr Register L | 2B |
| DMA1 I/O Addr Register H | 2C | DMA1 I/O Addr Register B | 2D |
| DMA1 Byte Count Register L | 2E | DMA1 Byte Count Register H | 2F |
| DMA Status Register | 30 | DMA Mode Register | 31 |
| DMA/Wait Control Register | 32 | Interrupt Vector Low Register | 33 |
| Interrupt/Trap Control Register | 34 | Interrupt Edge Register | 35 |
| Refresh Control Register | 36 | | |
| MMU Common Base Register | 38 | MMU Bank Base Register | 39 |
| MMU Common/Bank Area Register | 3A | Device ID Low | 3B |
| Device ID High | 3C | Revision ID | 3D |
| Operating Mode Control Register | 3E | I/O Control Register | 3F |
| Port A Data Register | 40 | Port A Data Direction Register | 41 |
| Port A Alternate Function Register | 42 | Port A Output Control Register | 43 |
| Port B Data Register | 44 | Port B Data Direction Register | 45 |
| Port B Alternate Function Register | 46 | Port B Output Control Register | 47 |
| Port C Data Register | 48 | Port C Data Direction Register | 49 |
| Port C Alternate Function Register | 4A | Port C Output Control Register | 4B |
| Port D Data Register | 4C | Port D Data Direction Register | 4D |
| Port D Alternate Function Register | 4E | Port D Output Control Register | 4F |
| PIOS Control Register | 60 | PIOS Address/Type Register | 61 |
| PIOS Counter Low | 62 | PIOS Counter High | 63 |
| WDT Master Register | 64 | WDT Command Register | 65 |

| | | | |
|--------------------------|----|--------------------------|----|
| ADC Control Register 0 | 66 | ADC Control Register 1 | 67 |
| ADC Result Register | 68 | DAC Control Register | 69 |
| DAC Data Register | 6A | WSG Control Register | 6B |
| ROM Boundary Register | 6C | RAM Lower Bound Register | 6D |
| RAM Upper Bound Register | 6E | RTC Control/Status | 6F |
| RTC Seconds | 70 | RTC Minutes | 71 |
| RTC Hours | 72 | RTC Day of the Week | 73 |
| RTC Date | 74 | RTC Month | 75 |
| RTC Year | 76 | RTC Century | 77 |
| RTC Alarm Seconds | 78 | RTC Alarm Minutes | 79 |
| RTC Alarm Hours | 7A | Output Control Register | 7D |
| Power Control Register | 7E | System Control Register | 7F |

BASIC DEVICE REGISTERS

In these register tables, the abbreviation Resvd denotes Reserved bits, which must not be programmed.

TABLE 16. FREE RUNNING COUNTER (0018H)

| | | | | | | | | |
|---------------------------------------------------|-------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Count | | | | | | | |
| R/W | R | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|--------------------------------------------------------------------------------------|
| 7-0 | Count | R | | This value is counted down by 1 every 10 PHI clocks, including during I/O Stop mode. |

TABLE 17. CLOCK CONTROL REGISTER (001EH) CLKCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|-------|----------------|-------|---|---|---|---|---|
| Bit/Field | Resvd | Low Noise XTAL | Resvd | | | | | |
| R/W | ? | R/W | ? | | | | | |
| Reset | X | 0 | X | X | X | X | X | X |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

Bit

| Position | Bit/Field | R/W | Value | Description |
|----------|----------------|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6 | Low Noise XTAL | R/W | 1 | If this bit is 1, the oscillator operates in a Low Noise mode, in which the gain is reduced and the output drive is reduced to about 30% of normal operation. This mode can be used for PCMCIA applications, in which the crystal would otherwise be driven with too much power. This mode limits the crystal frequency to 20 MHz at $V_{DD} = 4.5V$ and 10 MHz at $V_{DD} = 3.0V$. |

TABLE 18. CPU CONTROL REGISTER (001FH) CCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|---------|----------|-------|-------|------------|-------|-------|-------|
| Bit/Field | X1 XTAL | Stand-by | BREXT | LNPHI | Idle/Quick | Resvd | LNCTL | LNA/D |
| R/W | R/W | R/W | R/W | R/W | R/W | ? | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | X | 0 | 0 |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

Bit

| Position | Bit/Field | R/W | Value | Description |
|----------|---------------------|-----|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | X1 XTAL | R/W | 0 1 | The crystal frequency is divided by 2. The crystal frequency is used directly. |
| 6,3 | Standby, Idle/Quick | R/W | 00 01 10 11 | SLP instruction enters Sleep or System Stop mode IOSTOP + SLP enters Idle mode IOSTOP + SLP enters Standby mode IOSTOP + SLP enters Quick Recovery Standby mode |
| 5 | BREXT | R/W | 1 | Z80S183 honors Bus Requests in Standby mode |
| 4 | LNPHI | R/W | 1 | PHI Low noise mode: 25% of normal drive |
| 1 | LNCTL | R/W | 1 | \overline{IORD} and \overline{IOWR} Low noise mode: 25% of normal drive |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------|
| 0 | LNA/D | R/W | 1 | A19-0/D7-0 Low noise mode: 25% of normal drive |

TABLE 19. REFRESH CONTROL REGISTER (0036H) RCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|------|------|-------|---|---|---|-------|---|
| Bit/Field | REFE | REFW | Resvd | | | | Cycle | |
| R/W | R/W | R/W | ? | | | | R/W | |
| Reset | 1 | 1 | X | X | X | X | 0 | 0 |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|--------------------------------------|
| 7 | REFE | R/W | 1 | The Z80S183 generates Refresh cycles |
| 6 | REFW | R/W | 0 | 2-clock Refresh cycles |
| | | | 1 | 3-clock Refresh cycles |
| 1-0 | Cycle | R/W | 00 | Refresh cycle every 10 PHI clocks |
| | | | 01 | Refresh cycle every 20 PHI clocks |
| | | | 10 | Refresh cycle every 40 PHI clocks |
| | | | 11 | Refresh cycle every 80 PHI clocks |

Note: Because the Z80S183 has no RFSH pin, always program this register with all 0s.

TABLE 20. DEVICE ID LOW (003BH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|----------------------|---|---|---|---|---|---|---|
| Bit/Field | LS byte of Device ID | | | | | | | |
| R/W | R | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------|-----|-------|-----------------------------------------------------------------------------|
| 7-0 | Device ID LS byte | R/W | 01H | LSB of the Device ID value used to identify a Z80S183 via the ZDI interface |

TABLE 21. DEVICE ID HIGH (003CH)

| | | | | | | | | |
|---------------------------------------------------|----------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS byte of Device ID | | | | | | | |
| R/W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------|-----|-------|-----------------------------------------------------------------------------|
| 7-0 | Device ID MS byte | R/W | 00 | MSB of the Device ID value used to identify a Z80S183 via the ZDI interface |

TABLE 22. REVISION ID 003DH)

| | | | | | | | | |
|---------------------------------------------------|-------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Revision ID | | | | | | | |
| R/W | R | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------|-----|-------|----------------------------------------|
| 7-0 | Revision ID | R | 00 | Identifies revision AB of the Z80S183. |
| | | | 01 | Identifies revision BA. |

TABLE 23. REVISION ID (003DH)

| | | | | | | | | |
|-----------|-------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Revision ID | | | | | | | |
| R/W | R | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------|-----|-------|-----------------------------------------|
| 7-0 | Revision ID | R | 00 | Identifies the revision of the Z80S183. |

TABLE 24. OPERATING MODE CONTROL REGISTER (003EH) OMCR

| | | | | | | | | |
|-----------|-----|------|-----|----------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | M1E | M1TE | IOC | Reserved | | | | |
| R/W | R/W | W | R/W | ? | | | | |
| Reset | 1 | 1 | 1 | X | X | X | X | X |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 7 | M1E | R/W | 0 | $\overline{M1}$ is driven Low during refetch of RETI instructions and $\overline{INT0}$ acknowledge cycles (Z80 peripheral compatible). |
| | | | 1 | $\overline{M1}$ is driven Low in all opcode fetches, $\overline{INT0}$ acknowledge cycles, and 1st cycle of NMI acknowledge. |
| 6 | M1TE | W | 0 | After a 0 is written to this bit, the next Op Code fetch asserts $\overline{M1}$ Low. (Automatically returns to 1 state.) |
| | | | 1 | M1E governs operation of $\overline{M1}$ |
| 5 | IOC | R/W | 0 | \overline{IORD} is driven Low from rising edge at start of T2 cycle (Z80 compatible) |
| | | | 1 | \overline{IORD} is driven Low from falling edge in T1 cycle (64180 compatible) |

TABLE 25. I/O CONTROL REGISTER (003FH) IOCR

| | | | | | | | | |
|---------------------------------------------------|-----|---|-------|----------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | IOA | | IOSTP | Reserved | | | | |
| R/W | R/W | | R/W | ? | | | | |
| Reset | 0 | 0 | 0 | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|---------------------------------------------------------------------------|
| 7-6 | IOA | R/W | 00 | 180 registers are located at 0000-003FH Do not program this value |
| | | | 01 | 180 registers are located at 0080-00BFH |
| | | | 10 | 180 registers are located at 00C0-00FFH |
| | | | 11 | Do not program this value |
| 5 | IOSTP | R/W | 0 | Normal operation |
| | | | 1 | Clocking is blocked to the ASCIs, PRTs, and CSI/O, disabling these units. |

TABLE 26. OUTPUT CONTROL REGISTER (007DH) OCR

| | | | | | | | | |
|----------------------------------------------------------------------|------|----------|---|---------------------------|------------------|------------------|------------------|------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | ROME | Reserved | | Port B Weak Latch Disable | Port A Low Noise | Port B Low Noise | Port C Low Noise | Port D Low Noise |
| R/W | R/W | ? | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | X | X | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate ? = Not applicable | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------------------|-----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | ROME Emulator mode | R/W | 0 | Read data from internal devices does not appear on D7-0. This feature saves power. |
| | | | 1 | Read data from internal devices is driven onto D7-0, for monitoring by an external logic analyzer or other instrumentation. |
| 5 | Port B Weak Latch Disable | R/W | 0 | Weak latches on Port B are enabled, as they are on other pins with input capability. Weak latches save power by preventing floating voltage levels. |
| | | | 1 | Weak latches on Port B are disabled, to allow quasi-analog use of these pins. |
| 3-0 | Port A-D Low Noise | R/W | 1 | Drive on the corresponding Port is reduced to 25% of its current capability. This setting saves power and reduces system noise, while slightly increasing switching times. |

TABLE 27. POWER CONTROL REGISTER (007EH) PCR

| | | | | | | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|---------------------|----------|---------------|-------|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | CEOUT | Reset on ↑ of PWRUP | PWR SWCH | LPM Addr Ctrl | Resvd | | | |
| R/W | R/W | R/W | R/W | R/W | ? | | | |
| Reset | 0 | 0 | 0 | 0 | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate ? = Not Applicable This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------------|-----|-------|-------------------------------------------------------------------------------------------|
| 7 | CEOUT | R/W | 0 | $\overline{\text{CEOUT}}$ pin is not driven |
| | | | 1 | $\overline{\text{CEOUT}}$ pin is driven Low |
| 6 | Reset on ↑ of PWRUP | R/W | 0 | PWRUP pin cannot Reset |
| | | | 1 | Rising edge on PWRUP Resets the Z80S183 |
| 5 | PWR SWCH | R/W | | Direct positive-logic output |
| 4 | LPM Address Control | R/W | 0 | Low power modes do not affect A19-0, RAMRD, RAMWR, ROMRD, ROMWR, IORD, IOWR, IOCS1-2, TXS |
| | | | 1 | These pins are 3-stated in any LOW-POWER mode |

TABLE 28. SYSTEM CONFIGURATION REGISTER (007FH) SCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|-------------------|-------------------|-----------------|----------------------------------|----------------------------------|---------------------------------|--------------|---|
| Bit/Field | Onchip ROM Enable | Onchip RAM Enable | RAM High Enable | $\overline{\text{ROMCS}}$ Enable | $\overline{\text{RAMCS}}$ Enable | $\overline{\text{IOCS}}$ Enable | Clock Select | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | OPMOD1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate
This register can only be written if the Register Write Enable bit (WDTMR 0) is 1.

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|----------------------------------|-----|----------------------|---------------------------------------------------------------------------------------------------------------------|
| 7 | On-chip ROM Enable | R/W | 0 1 | Addresses 00000–003FFH are off-chip Addresses 00000–003FFH are in on-chip ROM. |
| 6 | On-chip RAM Enable | R/W | 0 1 | On-chip RAM is not accessible to the processor On-chip RAM is accessible to the processor |
| 5 | RAM High Enable | R/W | 0 1 | On-chip RAM is at xF800–xFFFFH; addresses A19-16 are not decoded for on-chip RAM On-chip RAM is at FF800–FFFFFFH |
| 4 | $\overline{\text{ROMCS}}$ Enable | R/W | 1 | $\overline{\text{ROMCS}}$ pin is enabled/active |
| 3 | $\overline{\text{RAMCS}}$ Enable | R/W | 1 | $\overline{\text{RAMCS}}$ pin is enabled/active |
| 2 | $\overline{\text{IOCS}}$ Enable | R/W | 1 | The $\overline{\text{IOCS1}}$ and $\overline{\text{IOCS2}}$ pins are enabled/active |
| 1-0 | Clock Select | R/W | 00 01 10 11 | PHI taken from EXTAL PHI taken from LFEXTAL PHI = LFEXTAL times 1004 PHI = LFEXTAL times 502 |

INTERRUPT REGISTERS

See the Interrupts section, which starts on page 13, for more about these registers.

TABLE 29. INTERRUPT VECTOR LOW REGISTER (0033H) IL

| | | | | | | | | |
|-----------|-------|---|---|-------|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | IL7-5 | | | Resvd | | | | |
| R/W | R/W | | | ? | | | | |
| Reset | 0 | 0 | 0 | X | X | X | X | X |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-5 | IL7-5 | R/W | | The processor uses these bits as A7-5 (along with the contents of the I register as A15-8) when fetching an interrupt service routine address for $\overline{\text{INT1-2}}$, ASCIO-1, PRT0-1, DMA0-1, or the CSI/O. |

TABLE 30. INTERRUPT/TRAP CONTROL REGISTER (0034H) ITC

| | | | | | | | | |
|-----------|------|-----|------|-------|---|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Trap | UFO | IEF1 | Resvd | | INT2 en | INT1 en | INT0 en |
| R/W | RWOC | R | R | ? | | R/W | R/W | R/W |
| Reset | 0 | X | 0 | X | | 0 | 0 | 1 |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|------|--------|-----------------------------------------------------------------------------------------|
| 7 | Trap | RWOC | 1 | Illegal Instruction Trap. Writing a 0 to this bit clears it; writing a 1 has no effect. |
| 6 | UFO | R | 0 1 | Inst started at stacked PC-1 Inst started at stacked PC-2 |
| 5 | IEF1 | R | | Current state of interrupt enable |
| 2 | INT2 en | R/W | | $\overline{\text{INT2}}$ interrupt enable |
| 1 | INT1 en | R/W | | $\overline{\text{INT1}}$ interrupt enable |
| 0 | INT0 en | R/W | | $\overline{\text{INT0}}$ interrupt enable |

TABLE 31. INT2-1 INTERRUPT EDGE REGISTER (0035H) IECR

| | | | | | | | | |
|-----------|------|------|-------------------------------|-------------------------------|-----------------------------------|---|-----------------------------------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | INT2 | INT1 | $\overline{\text{INT2}}$ Edge | $\overline{\text{INT1}}$ Edge | $\overline{\text{INT2}}$ Mode Sel | | $\overline{\text{INT1}}$ Mode Sel | |
| R/W | R | R | RW1C | RW1C | RW | | RW | |
| Reset | X | X | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------------|------|----------------------|------------------------------------------------------------------------------------------------|
| 7 | INT2 | R | | $\overline{\text{INT2}}$ pin state, 0 is Low |
| 6 | INT1 | R | | $\overline{\text{INT1}}$ pin state, 0 is Low |
| 5 | $\overline{\text{INT2}}$ Edge | RW1C | 0 1 | R: edge not detected, W: no effect R: edge detected, W: clear this bit |
| 4 | $\overline{\text{INT1}}$ Edge | RW1C | 0 1 | R: edge not detected, W: no effect R: edge detected, W: clear this bit |
| 3-2 | $\overline{\text{INT2}}$ Mode | RW | 00 01 10 11 | Low-Level Interrupt Rising Edge Interrupt Falling Edge Interrupt Both Edges Interrupt |
| 1-0 | $\overline{\text{INT1}}$ Mode | RW | 00 01 10 11 | Low-Level Interrupt Rising Edge Interrupt Falling Edge Interrupt Both Edges Interrupt |

MMU REGISTERS

See “Memory Management Unit (MMU)” on page 10, for more about these registers.

TABLE 32. COMMON BASE REGISTER (0038H) CBR

| | | | | | | | | |
|---------------------------------------------------|-----------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Base of Common Area 1 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | Common 1 Area Base | R/W | | If the comparison of bits 15-12 of a logical address indicates that the address is in Common Area 1, this value (shifted left 12 bits, times 4096) is added to the logical address to form the physical address. |

TABLE 33. BANK BASE REGISTER (0039H) BBR

| | | | | | | | | |
|---------------------------------------------------|-------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Base of Bank Area | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|----------------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | Bank Area Base | R/W | | If the comparison of bits 15–12 of a logical address indicates that the address is in the Bank Area, this value (shifted left 12 bits, times 4096) is added to the logical address to form the physical address. |

TABLE 34. COMMON/BANK AREA REGISTER C(003AH) CBAR

| | | | | | | | | |
|---------------------------------------------------|------------------------|---|---|---|------------------------|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Bank/Common 1 Boundary | | | | Common 0/Bank Boundary | | | |
| R/W | R/W | | | | R/W | | | |
| Reset | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------------------|-----|-------|--------------------------------------------------------------------------------------------------------------|
| 7-4 | Bank/Common 1 Boundary | R/W | | If bits 15–12 of a logical address are greater than or equal to this value, the address is in Common Area 1. |
| 3-0 | Common 0/Bank Boundary | R/W | | If bits 15–12 of a logical address are less than this value, the address is in Common Area 0. |

Note: If bits 3-0 of this reg \leq bits 15-12 of a logical address $<$ bits 7-4 of this reg, the address is in the Bank Area. Do not program this register so that bits 3-0 $>$ bits 7-4. All comparisons are unsigned.

ROM/RAM CHIP SELECT AND WAIT REGISTERS

See “Memory”, which starts on page 19, for more detail about these registers.

TABLE 35. WAIT STATE GENERATOR CONTROL REGISTER (006BH) WSGCR

| | | | | | | | | |
|---------------------------------------------------|-----------|---|-----------|---|-------------|---|----------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | ROM Waits | | RAM Waits | | Other Waits | | Reserved | |
| R/W | R/W | | R/W | | R/W | | ? | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|---------------------------------------------------------------------------------------------------|
| 7-6 | ROM Waits | R/W | | This field determines how many wait states are inserted for memory addresses that activate ROMCS: |
| | | | 00 | No Wait states |
| | | | 01 | 1 Wait state |
| | | | 10 | 2 Wait states |
| | | | 11 | 4 Wait states |
| 5-4 | RAM Waits | R/W | | This field determines how many wait states are inserted for memory addresses that activate RAMCS: |
| | | | 00 | No Wait states |
| | | | 01 | 1 Wait state |
| | | | 10 | 2 Wait states |
| | | | 11 | 4 Wait states |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3-2 | Other Waits | R/W | | This field determines how many wait states are inserted for memory addresses that do not activate either $\overline{\text{ROMCS}}$ nor $\overline{\text{RAMCS}}$. (These cycles do not activate any control signals and thus are not visible.) |
| | | | 00 | No Wait states |
| | | | 01 | 1 Wait state |
| | | | 10 | 2 Wait states |
| | | | 11 | 4 Wait states |

TABLE 36. ROM BOUND REGISTER (006CH) ROMBR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|-----------------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---|---|---|
| Bit/Field | ROM Upper Bound | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |
| Bit Position | Bit/Field | R/W | Value | Description | | | | |
| 7-0 | ROM Upper Bound | R/W | | If bit 4 of the System Configuration Register is 1, as it is after a reset, memory accesses at addresses with A19-12 less than or equal to this value, activate $\overline{\text{ROMRD}}$ or $\overline{\text{ROMWR}}$ | | | | |

TABLE 37. RAM LOWER BOUND REGISTER (006DH) RAMLBR

| | | | | | | | | |
|---------------------------------------------------|-----------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | RAM lower Bound | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | RAM lower Bound | R/W | | If bit 3 of the System Configuration Register is 1, memory accesses at addresses with A19–12 greater than or equal to this value, less than or equal to the value in RAMUBR, and greater than the value in ROMBR, activate $\overline{\text{RAMRD}}$ or $\overline{\text{RAMWR}}$. |

TABLE 38. RAM UPPER BOUNDARY REGISTER (006EH) RAMUBR

| | | | | | | | | |
|---------------------------------------------------|-----------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | RAM Upper Bound | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | RAM lower Bound | R/W | | If bit 3 of the System Configuration Register is 1, memory accesses at addresses with A19–12 greater than or equal to this value, less than or equal to the value in RAMUBR, and greater than the value in ROMBR, activate $\overline{\text{RAMRD}}$ or $\overline{\text{RAMWR}}$. |

TABLE 39. RAM UPPER BOUND REGISTER (006EH) RAMUBR

| | | | | | | | | |
|---------------------------------------------------|-----------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | RAM Upper Bound | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | RAM Upper Bound | R/W | | If bit 3 of the System Configuration Register is 1, as it is after a reset, memory accesses at addresses with A19-12 less than or equal to this value, greater than or equal to the value in RAMLBR, and greater than the value in ROMBR, activate $\overline{\text{RAMRD}}$ or $\overline{\text{RAMWR}}$. |

I/O PORT REGISTERS

See “I/O Ports” on page 33, for more about these registers.

TABLE 40. PORT A DATA REGISTER (0040H) DRA

| | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | X | X | X | X | X | X | X | X |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PA7-0 | R/W | | Writing to this register sets the data that is driven onto those pins among PA7-0, that are designated as outputs in the Data Direction and Output Control Registers. Reading from the Data Register returns the state of pins PA7-0, for both inputs and outputs. The output latches cannot be read back separately. |

TABLE 41. PORT A DATA DIRECTION REGISTER (0041H) DDRA

| | | | | | | | | |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PA7 dir | PA6 dir | PA5 dir | PA4 dir | PA3 dir | PA2 dir | PA1 dir | PA0 dir |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: R = Read W = Write X = Indeterminate
This register can only be written if the Register Write Enable bit (WDTMR 0) is 1.

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PA7-0 dir | R/W | | Together with Output Control Register A, these bits determine which pins among PA7-0 are inputs and which are outputs, and for outputs, one of three output modes. The 4 possible settings for each pin are: <ul style="list-style-type: none"> 0 with 0 in the corresponding OCR bit, selects totem pole output. 0 with 1 in the corresponding OCR bit, selects open drain output. 1 with 0 in the corresponding OCR bit, selects input. 1 with 1 in the corresponding OCR bit, selects open drain output with an internal pullup resistor. |

TABLE 42. PORT A ALTERNATE FUNCTION SELECT REGISTER (0042H) AFSA

| | | | | | | | | |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PA7 int | PA6 int | PA5 int | PA4 int | PA3 int | PA2 int | PA1 int | PA0 int |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-------------------------------------------------------------------------------------------|
| 7-0 | PA7-0 int | R/W | | A 1 in one of these bits makes the corresponding pin a Low-active interrupt request line. |

TABLE 43. PORT A OUTPUT CONTROL REGISTER (0043H) OCRA

| | | | | | | | | |
|---------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PA7 oc | PA6 oc | PA5 oc | PA4 oc | PA3 oc | PA2 oc | PA1 oc | PA0 oc |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PA7-0 oc | R/W | | Together with Data Direction Register A, these bits determine which pins among PA7-0 are inputs and which are outputs, and for outputs, one of three output modes. The four configurations for each pin are described in Table 41. |

TABLE 44. PORT B DATA REGISTER (0044H) DRB

| | | | | | | | | |
|---------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PB7-0 | R/W | | Writing to this register sets the data that is driven onto those pins among PB7-0, that are designated as outputs in the Data Direction and Output Control Registers. Reading from the Data Register returns the state of pins PB7-0, for both inputs and outputs. The output latches cannot be read separately. |

TABLE 45. PORT B DATA DIRECTION REGISTER (0045H) DDRB

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PB7 dir | PB6 dir | PB5 dir | PB4 dir | PB3 dir | PB2 dir | PB1 dir | PB0 dir |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PB7-0 dir | R/W | | Together with Output Control Register B, these bits determine which pins among PB7-0 are inputs and which are outputs, and for outputs, one of three output modes. The four possible settings for each pin are: |
| | | | 0 | with 0 in the corresponding OCR bit, selects totem pole output. |
| | | | 0 | with 1 in the corresponding OCR bit, selects open drain output. |
| | | | 1 | with 0 in the corresponding OCR bit, selects input. |
| | | | 1 | with 1 in the corresponding OCR bit, selects open drain output with an internal pullup resistor. |

TABLE 46. PORT B ALTERNATE FUNCTION SELECT REGISTER (0046H) AFSB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit/Field | PB7 alt | PB6 alt | PB5 alt | PB4 alt | PB3 alt | PB2 alt | PB1 alt | PB0 alt |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------|
| 7 | PB7 alt | R/W | 0 | The PB7/RXS pin is PB7 |
| | | | 1 | The PB7/RXS pin is RXS |
| 6 | PB6 alt | R/W | 0 | The PB6/RXA1 pin is PB6 |
| | | | 1 | The PB6/RXA1 pin is RXA1 |
| 5 | PB5 alt | R/W | 0 | The PB5/TXA1 pin is PB5 |
| | | | 1 | The PB5/TXA1 pin is TXA1 |
| 4 | PB4 alt | R/W | 0 | The PB4/RXA0 pin is PB4 |
| | | | 1 | The PB4/RXA0 pin is RXA0 |
| 3 | PB3 alt | R/W | 0 | The PB3/TXA0 pin is PB3 |
| | | | 1 | The PB3/TXA0 pin is TXA0 |
| 2 | PB2 alt | R/W | 0 | The PB2/CTS0/PWRUP pin is PB2 |
| | | | 1 | The PB2/CTS0/PWRUP pin is CTS0 or PWRUP |
| 1 | PB1 alt | R/W | 0 | The PB1/DCD0 pin is PB1 |
| | | | 1 | The PB1/DCD0 pin is DCD0 |
| 0 | PB0 alt | R/W | 0 | The PB0/CKS pin is PB0 |
| | | | 1 | The PB0/CKS pin is CKS |

Note: A 1 in any of these bits disables control of the pin by the DDRB register, but the corresponding bit in OCRB is 0 in this case.

TABLE 47. PORT B OUTPUT CONTROL REGISTER (0047H) OCB

| | | | | | | | | |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PB7 oc | PB6 oc | PB5 oc | PB4 oc | PB3 oc | PB2 oc | PB1 oc | PB0 oc |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PB7-0 oc | R/W | | Together with Data Direction Register B, these bits determine which pins among PB7-0 are inputs and which are outputs, and for outputs, one of three output modes. The four configurations for each pin are described in Table 45. |

TABLE 48. PORT C DATA REGISTER (0048H) DRC

| | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | X | X | X | X | X | X | X | X |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PC7-0 | R/W | | Writing to this register sets the data that is driven onto those pins among PC7-0, that are designated as outputs in the Data Direction and Output Control Registers. Reading from the Data Register returns the state of pins PC7-0, for both inputs and outputs. The output latches cannot be read separately. |

TABLE 49. PORT C DATA DIRECTION REGISTER (0049H) DDRC

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PC7 dir | PC6 dir | PC5 dir | PC4 dir | PC3 dir | PC2 dir | PC1 dir | PC0 dir |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PC7-0 dir | R/W | | Together with Output Control Register C, these bits determine which pins among PC7-0 are inputs and which are outputs, and for outputs, one of three output modes. The four possible settings for each pin are: |
| | | | 0 | with 0 in the corresponding OCR bit, selects totem pole output. |
| | | | 0 | with 1 in the corresponding OCR bit, selects open drain output. |
| | | | 1 | with 0 in the corresponding OCR bit, selects input. |
| | | | 1 | with 1 in the corresponding OCR bit, selects open drain output with an internal pullup resistor. |

TABLE 50. PORT C ALTERNATE FUNCTION SELECT REGISTER R(004AH) AFSC

| | | | | | | | | |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PC7 alt | PC6 alt | PC5 alt | PC4 alt | PC3 alt | PC2 alt | PC1 alt | PC0 alt |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PC7-0 alt | R/W | | Writing a 1 to any of these bits assigns the corresponding pin to the Programmable I/O Sequencer (PIOS). In this mode the DDRC and OCRC registers still determine the output drive of the port, so that the PIOS can use any of the port output modes. To use PC0 as a 50 or 60 Hz time base for the Real Time Clock, leave AFSC bit 0 zero, DDRC bit 0 one, and OCRC bit 0 zero. |

TABLE 51. PORT C OUTPUT CONTROL REGISTER (004BH) OCRC

| | | | | | | | | |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PC7 oc | PC6 oc | PC5 oc | PC4 oc | PC3 oc | PC2 oc | PC1 oc | PC0 oc |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PC7-0 oc | R/W | | Together with Data Direction Register C, these bits determine which pins among PC7-0 are inputs and which are outputs, and for outputs, one of three output modes. The four configurations for each pin are described in Table 49. |

TABLE 52. PORT D DATA REGISTER (004CH) DRD

| | | | | | | | | |
|---------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PD7-0 | R/W | | Writing to this register sets the data that is driven onto pins among PD7-0, that are designated as outputs in the Data Direction and Output Control Registers. Reading from the Data Register returns the states of pins PD7-0, for both inputs and outputs. There is no way to read back the output latches separately. |

TABLE 53. PORT D DATA DIRECTION REGISTER (004DH) DDRD

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PD7 dir | PD6 dir | PD5 dir | PD4 dir | PD3 dir | PD2 dir | PD1 dir | PD0 dir |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PD7-0 dir | R/W | | Together with Output Control Register D, these bits determine which pins between PD7-0 are inputs and which are outputs, and for outputs, one of three output modes. The four possible settings for each pin are: 0 With 0 in the corresponding OCR bit, selects totem pole output. 0 With 1 in the corresponding OCR bit, selects open drain output. 1 With 0 in the corresponding OCR bit, selects input. 1 With 1 in the corresponding OCR bit, selects open-drain output with an internal pullup resistor. |

TABLE 54. PORT D ALTERNATE FUNCTION SELECT REGISTER (004EH) AFSD

| | | | | | | | | |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PD7 int | PD6 int | PD5 int | PD4 int | PD3 int | PD2 int | PD1 int | PD0 int |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-------------------------------------------------------------------------------------------|
| 7-0 | PD7-0 int | R/W | | A 1 in one of these bits makes the corresponding pin a Low-active interrupt request line. |

TABLE 55. PORT D OUTPUT CONTROL REGISTER (004FH) OCRD

| | | | | | | | | |
|---------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | PD7 oc | PD6 oc | PD5 oc | PD4 oc | PD3 oc | PD2 oc | PD1 oc | PD0 oc |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | PD7-0 oc | R/W | | Together with Data Direction Register D, these bits determine which pins among PD7-0 are inputs and which are outputs, and for outputs, one of three output modes. The four configurations for each pin are described in Table 52. |

DMA REGISTERS

See section “DMA Channels” on page 36, for more about these registers.

TABLE 56. DMA0 SOURCE ADDRESS REGISTER LOW (0020H) SAR0L

| | | | | | | | | |
|-----------|--------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of DMA0 Source Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------------------|-----|-------|----------------------------------------------|
| 7-0 | DMA0 Source Address LS byte | R/W | | LSB of the Source Address for DMA channel 0. |

TABLE 57. DMA0 SOURCE ADDRESS REGISTER LOW (0020H) SAR0L

| | | | | | | | | |
|-----------|--------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of DMA0 Source Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------------------|-----|-------|----------------------------------------------|
| 7-0 | DMA0 Source Address LS byte | R/W | | LSB of the Source Address for DMA channel 0. |

TABLE 58. DMA0 SOURCE ADDRESS REGISTER HIGH (0021H) SAR0H

| | | | | | | | | |
|---------------------------------------------------|------------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Middle byte of DMA0 Source Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------------------------|-----|-------|----------------------------------------------------|
| 7-0 | DMA0 Source Address middle byte | R/W | | Bits 15-8 of the Source Address for DMA channel 0. |

TABLE 59. DMA0 SOURCE ADDRESS REGISTER B (0022H) SAR0B

| | | | | | | | | |
|---------------------------------------------------|----------|---|---|---|-----------------------------|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Reserved | | | | MS part of DMA0 Source Addr | | | |
| R/W | | | | | R/W | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------------|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3-0 | MS part of DMA0 Source Address | R/W | | <p>If the Source Mode field in the DMODE register is 11, indicating an I/O source, these bits select which source device handshake line controls data transfer, as follows:</p> <p>X000 $\overline{\text{DREQ0}}$ pin X001 ASCIO RDRF X010 ASCI1 RDRF other Reserved, do not program</p> <p>Otherwise, these bits contain A19-16 of the DMA0 Source address.</p> |

TABLE 60. DMA0 DESTINATION ADDRESS REGISTER LOW (0023H) DAR0L

| | | | | | | | | |
|---------------------------------------------------|-------------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of DMA0 Destination Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|----------------------------------|-----|-------|---------------------------------------------------|
| 7-0 | DMA0 Destination Address LS byte | R/W | | LSB of the Destination Address for DMA channel 0. |

TABLE 61. DMA0 DESTINATION ADDRESS REGISTER HIGH (0024H) DAR0H

| | | | | | | | | |
|---------------------------------------------------|-----------------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Middle byte of DMA0 Destination Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------------------|-----|-------|---------------------------------------------------------|
| 7-0 | DMA0 Destination Address middle byte | R/W | | Bits 15-8 of the Destination Address for DMA channel 0. |

TABLE 62. DMA0 DESTINATION ADDRESS REGISTER B (0025H) DAR0B

| | | | | | | | | |
|---------------------------------------------------|----------|---|---|---|-------------------------------------|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Reserved | | | | MS part of DMA0 Destination Address | | | |
| R/W | | | | | R/W | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------------------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3-0 | MS part of DMA0 Destination Address | R/W | | <p>If the Dest Mode field in the DMODE register is 11, indicating an I/O destination, these bits select which destination device handshake line controls data transfer, as follows:</p> <p style="text-align: center;">$\overline{DREQ0}$ pin</p> <p>X000 ASCIO TDRE X001 ASCI1 TDRE X010 Reserved, do not program other</p> <p>Otherwise these bits contain A19-16 of the DMA 0 Destination address.</p> |

TABLE 63. DMA0 BYTE COUNT REGISTER LOW (0026H) BCR0L

| | | | | | | | | |
|---------------------------------------------------|----------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of DMA0 Byte Count | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|----------------------------------------------|
| 7-0 | DMA0 Byte Count LS byte | R/W | | LS byte of the Byte Count for DMA channel 0. |

TABLE 64. DMA0 BYTE COUNT REGISTER HIGH (0027H) BC0H

| | | | | | | | | |
|---------------------------------------------------|----------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS byte of DMA0 Byte Count | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|----------------------------------------------|
| 7-0 | DMA0 Byte Count MS byte | R/W | | MS byte of the Byte Count for DMA channel 0. |

TABLE 65. DMA1 MEMORY ADDRESS REGISTER LOW (0028H) MAR1L

| | | | | | | | | |
|---------------------------------------------------|--------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of DMA1 Memory Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------------------|-----|-------|--------------------------------------------------|
| 7-0 | DMA1 Memory Address LS byte | R/W | | LS byte of the Memory Address for DMA channel 1. |

TABLE 66. DMA1 MEMORY ADDRESS REGISTER HIGH (0029H) MAR1H

| | | | | | | | | |
|---------------------------------------------------|------------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Middle byte of DMA1 Memory Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------------------------|-----|-------|----------------------------------------------------|
| 7-0 | DMA1 Memory Address middle byte | R/W | | Bits 15-8 of the Memory Address for DMA channel 1. |

TABLE 67. DMA1 MEMORY ADDRESS REGISTER B (002AH) MAR1B

| | | | | | | | | |
|---------------------------------------------------|----------|---|---|---|---------------------------|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Reserved | | | | DMA1 Memory Address 19-16 | | | |
| R/W | | | | | R/W | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------------------|-----|-------|----------------------------------------|
| 3-0 | DMA1 Memory Address 19-16 | R/W | | Bits 19-16 of the DMA1 Memory address. |

TABLE 68. DMA1 I/O ADDRESS REGISTER LOW (002BH) IAR1L

| | | | | | | | | |
|---------------------------------------------------|-----------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of DMA1 I/O Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|-------|-----------------------------------------------|
| 7-0 | DMA1 I/O Address LS byte | R/W | | LS byte of the I/O Address for DMA channel 1. |

TABLE 69. DMA1 I/O ADDRESS REGISTER HIGH (002CH) IAR1H

| | | | | | | | | |
|---------------------------------------------------|-----------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS byte of DMA1 I/O Address | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|-------|-------------------------------------------------|
| 7-0 | DMA1 I/O Address MS byte | R/W | | Bits 15-8 of the I/O Address for DMA channel 1. |

TABLE 70. DMA1 I/O ADDRESS REGISTER B (002DH) IAR1B

| | | | | | | | | |
|---------------------------------------------------|------|------|----------|---|---|---------------------------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | AltE | AltC | Reserved | | | DMA1 I/O Handshake Select | | |
| R/W | R/W | R/W | ? | | | R/W | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | AltE | R/W | 1 | Set this bit only when both DMA channels are programmed for the same I/O source or I/O destination. In this case, a channel end condition (byte count = 0) on channel 0 sets bit 6 (AltC), which then enables channel 1's request and blocks channel 0's. Similarly, a channel end condition on channel 1 clears bit 6 (AltC), which then enables channel 0's request and blocks channel 1's. To use this feature with external requests, the request from the device must be routed or connected to both the $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ pins. |
| 6 | AltC | R/W | | If bit 7 (AltE) is 0, this bit has no effect. When bit 7 (AltE) is 1 and this bit is 0, the Request signal selected by bits 2-0 is not presented to channel 1, but channel 0's Request operates normally. When AltE is 1 and this bit is 1, the Request selected by SAR18-16 or DAR18-16 is not presented to channel 0, but channel 1's request operates normally. This bit can be written to select which channel operates first, but perform this operation only when both channels are stopped (both DE1 and DE0 are 0). |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------------------|-----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2-0 | DMA1 I/O Handshake Select | R/W | | <p>If bit DIM1 in the DCNTL register is 1, indicating an I/O source, these bits select which source handshake signal controls the transfer.</p> <p>$\overline{\text{DREQ1}}$ pin 000 ASCIO RDRF 001 ASCI1 RDRF 010 Reserved, do not program other</p> <p>If DIM1 is 0, indicating an I/O destination, these bits select which destination handshake signal controls the transfer, as follows:</p> <p>$\overline{\text{DREQ1}}$ pin ASCIO TDRE 000 ASCI1 TDRE 001 Reserved, do not program 010 other</p> |

TABLE 71. DMA1 BYTE COUNT REGISTER LOW (002EH) BCR1L

| | | | | | | | | |
|---------------------------------------------------|----------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of DMA1 Byte Count | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|----------------------------------------------|
| 7-0 | DMA1 Byte Count LS byte | R/W | | LS byte of the Byte Count for DMA channel 1. |

TABLE 72. DMA1 BYTE COUNT REGISTER HIGH (002FH) BCR1H

| | | | | | | | | |
|---------------------------------------------------|----------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS byte of DMA1 Byte Count | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|----------------------------------------------|
| 7-0 | DMA1 Byte Count MS byte | R/W | | MS byte of the Byte Count for DMA channel 1. |

TABLE 73. DMA STATUS REGISTER (0030H) DSTAT

| | | | | | | | | |
|---------------------------------------------------|-----|-----|------|------|------|------|----------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | DE1 | DE0 | DWE1 | DWE0 | DIE1 | DIE0 | Reserved | DME |
| R/W | R/W | R/W | W | W | R/W | R/W | ? | R/W |
| Reset | 0 | 0 | X | X | 0 | 0 | X | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | DE1 | R/W | | DMA channel 1 enable. This bit can only be written when DWE1 is 0 in a write operation. DMA channel 1 clears this bit when it counts its byte count down to 0. |
| 6 | DE0 | R/W | | DMA channel 0 enable. This bit can only be written when DWE0 is 0 in a write operation. DMA channel 0 clears this bit when it counts its byte count down to 0. |
| 5-4 | DWE1-0 | W | 0 | Writing a 0 to one of these bits makes the Z80S183 capture the value of the corresponding DE bit. |
| | | | 1 | Writing a 1 to one of these bits does not affect the state of the corresponding DMA channel. These bits always read as 11. |
| 3-2 | DIE1-0 | R/W | | Interrupt enable for DMA channels 1-0. If one of these bits is 1, that DMA channel interrupts when it decrements its byte count to 0. |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | DME | R | 0 | Operations of both DMA channels are disabled. Reset and a nonmaskable interrupt both clear this bit. |
| | | | 1 | Operation of a DMA channel that has its DE bit 1 is enabled. This bit is set when a 1 is written to a DE bit, and a 0 is written to the corresponding DWE bit in the same write operation. |

TABLE 74. DMA MODE REGISTER (0031H) DMODE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|----------|---|----------------|---|------------------|---|------|-------|
| Bit/Field | Reserved | | DMA0 Dest Mode | | DMA0 Source Mode | | MMOD | Resvd |
| R/W | ? | | R/W | | R/W | | R/W | ? |
| Reset | X | X | 0 | 0 | 0 | 0 | 0 | X |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|----------------|-----|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5-4 | DMA0 Dest Mode | R/W | 00 | This field controls operation of the destination side of DMA channel 0: Memory write, address increment Memory write, address decrement Memory (or memory mapped I/O) write, fixed address I/O write, fixed address |
| | | | 01 | |
| | | | 10 | |
| | | | 11 | |

TABLE 75. DMA MODE REGISTER (0031H) DMODE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|------------------|-----|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|---|------|----------|
| Bit/Field | Reserved | | DMA0 Dest Mode | | DMA0 Source Mode | | MMOD | Reserved |
| R/W | | | R/W | | R/W | | R/W | |
| Reset | X | X | 0 | 0 | 0 | 0 | 0 | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |
| Bit Position | Bit/Field | R/W | Value | Description Bit | | | | |
| 5-4 | DMA0 Dest Mode | R/W | 00 01 10 11 | This field controls operation of the destination side of DMA channel 0: Memory write, address increment Memory write, address decrement Memory (or memory mapped I/O) write, fixed address I/O write, fixed address | | | | |
| 3-2 | DMA0 Source Mode | R/W | 00 01 10 11 | This field controls operation of the source side of DMA channel 0: Memory read, address increment Memory read, address decrement Memory (or memory mapped I/O) read, fixed address I/O read, fixed address | | | | |
| 1 | MMOD | R/W | 0 1 | When the Source and Dest Mode fields above are both 0x, indicating memory to memory operation, no device request (that is, $\overline{DREQ0}$) controls data transfer on DMA channel 0. In this case, this bit selects between two modes: Cycle Steal mode: the DMA(s) and processor alternate bus cycles. Burst mode: DMA0 uses the bus continuously to complete the block transfer. | | | | |

TABLE 76. DMA/WAIT CONTROL REGISTER (0032H) DCNTL

| | | | | | | | | |
|---------------------------------------------------|--------------|---|-----------|---|-------------------|---|-----------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Memory Waits | | I/O Waits | | Request Sense 1-0 | | DMA1 Mode | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------|-----|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-6 | Memory Waits | R/W | 00 01 10 11 | This field controls how many wait states are injected into DMA and processor memory cycles: None 1 2 3 |
| 5-4 | I/O Waits | R/W | 00 01 10 11 | This field controls how many wait states are injected into DMA and processor I/O cycles: None 1 2 3 |
| 3-2 | Request Sense 1-0 | R/W | 0 1 | Each of these bits controls how the corresponding DMA channel samples its Request signal (except when DMA 0's Source and Dest Mode fields are both 0x) 0 Level sense: the DMA samples its Request again during the 2nd cycle for each byte 1 Edge sense: another falling edge is needed on the Request line before the DMA channel transfers another byte See section , which starts on page 37, for timing of both cases. |
| 1-0 | DMA1 Mode | R/W | 00 01 10 11 | This field controls the direction of both transfer and address stepping on DMA channel 1: 00 Incrementing Memory addrs to I/O 01 Decrementing Memory addrs to I/O 10 I/O to incrementing Memory addrs 11 I/O to decrementing Memory addrs |

WATCH-DOG TIMER REGISTERS

See “Watch-Dog Timer” on page 43, for more about these registers.

TABLE 77. WATCH-DOG TIMER MASTER REGISTER (0064H) WDTMR

| | | | | | | | | |
|-----------|--------|---------------|---|-------------|----------------|-----------|---------------------|------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Enable | Period Select | | Drive Reset | Power On Reset | WDT Reset | State Change Enable | Reg Write Enable |
| R/W | R | R/W | | R/W | RC | RC | R/W | R |
| Reset | 1 | 1 | 1 | POR = 1 | POR = 1 | WDT | 1 | 1 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|----------------------|-----|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | WDT Enabled | R | | If this read-only bit is 1 the Watch-Dog Timer is enabled. |
| 6-5 | Period Select | R/W | 00 01 10 11 | The field selects how long software can leave the Watch-Dog Timer unattended, before it resets the part: 2 ¹⁸ (262,144) PHI clocks 2 ²² (4,194,304) PHI clocks 2 ²⁵ (33,554,432) PHI clocks 2 ²⁷ (134,217,728) PHI clocks |
| 4 | Drive Reset | R/W | | If this bit is 1, as it is after a Power On Reset, expiration of the WDT drives the Reset pin Low, to reset external devices. WDT expiration resets the Z80S183 regardless of this bit. |
| 3 | Power On Reset | RC | | A Power On Reset sets this bit. (This includes a POR sequence caused by a rising edge on the OPMOD1 or PWRUP pin.) Reading this register clears this bit. |
| 2 | WDT Reset | RC | | WDT expiration sets this bit. Reading this register clears it. |
| 1 | State Change Enabled | R/W | | When this bit is 1, writing 40H to the WDT Command register disables the WDT, and writing 80H to it enables the WDT. Such changes only become effective after this bit is cleared. |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | Register Write Enable | R | | When this read-only bit is 1, as it is after a reset, the System Configuration Register, port Data Direction Registers, the Power Control Register, and all of the Real Time Clock registers can be written. When this bit is 0, these registers are read-only. |

TABLE 78. WATCH-DOG TIMER MASTER REGISTER (0064H) WDTMR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|--------|---------------|---|-------------|----------------|-----------|---------------------|------------------|
| Bit/Field | Enable | Period Select | | Drive Reset | Power On Reset | WDT Reset | State Change Enable | Reg Write Enable |
| R/W | R | R/W | | R/W | RC | RC | R/W | R |
| Reset | 1 | 1 | 1 | POR = 1 | POR = 1 | WDT | 1 | 1 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description | | | | | | | | |
|--------------|-----------------------------------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------------------------|----|---------------------------------|----|----------------------------------|----|-----------------------------------|
| 7 | WDT Enabled | R | | If this read-only bit is 1 the Watch-Dog Timer is enabled. | | | | | | | | |
| 6-5 | Period Select | R/W | | This field selects how long software can leave the Watch-Dog Timer unattended, before it resets the part: <table border="0" style="margin-left: 20px;"> <tr> <td>00</td> <td>2^{18} (262,144) PHI clocks</td> </tr> <tr> <td>01</td> <td>2^{22} (4,194,304) PHI clocks</td> </tr> <tr> <td>10</td> <td>2^{25} (33,554,432) PHI clocks</td> </tr> <tr> <td>11</td> <td>2^{27} (134,217,728) PHI clocks</td> </tr> </table> | 00 | 2^{18} (262,144) PHI clocks | 01 | 2^{22} (4,194,304) PHI clocks | 10 | 2^{25} (33,554,432) PHI clocks | 11 | 2^{27} (134,217,728) PHI clocks |
| 00 | 2^{18} (262,144) PHI clocks | | | | | | | | | | | |
| 01 | 2^{22} (4,194,304) PHI clocks | | | | | | | | | | | |
| 10 | 2^{25} (33,554,432) PHI clocks | | | | | | | | | | | |
| 11 | 2^{27} (134,217,728) PHI clocks | | | | | | | | | | | |
| 4 | Drive Reset | R/W | | If this bit is 1, as it is after a Power On Reset, expiration of the WDT drives the $\overline{\text{RESET}}$ pin Low, to reset external devices. WDT expiration resets the Z80S183 regardless of this bit. | | | | | | | | |
| 3 | Power On Reset | RC | | A Power On Reset sets this bit. (This includes a POR sequence caused by a rising edge on the OPMOD1 or PWRUP pin.) Reading this register clears this bit. | | | | | | | | |
| 2 | WDT Reset | RC | | WDT expiration sets this bit. Reading this register clears it. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | State Change Enabled | R/W | | When this bit is 1, writing 40H to the WDT Command register disables the WDT, and writing 0B0H to it enables the WDT. |
| 0 | Register Write Enable | R | | When this read-only bit is 1, as it is after a reset, the System Configuration Register, port Data Direction Registers, the Power Control Register, and all of the Real Time Clock registers can be written. When this bit is 0, these registers are read-only. |

TABLE 79. WATCH-DOG TIMER COMMAND REGISTER (0065H) WDTCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|-------------|-----|-----|-----|-----|-----|-----|-----|
| Bit/Field | WDT Command | | | | | | | |
| R/W | W | | | | | | | |
| Reset | n/a | n/a | n/a | n/a | n/a | n/a | n/a | n/a |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | WDT Command | W | | Software can write the following values to this register, to affect the status of the WDT and Z80S183: 0BH Set Register Write Enable (WDTMR bit 0) 40H Disables WDT if WDTMR bit 1 is 1 4EH Reloads/Restarts WDT 80H Enables WDT if WDTMR bit 1 is 1 Writing any value other than 0BH to this register clears the Register Write Enable bit. |

PROGRAMMABLE RELOAD TIMER (PRT) REGISTERS

See section “Programmable Reload Timers (PRTs)” on page 43, for more about these registers.

TABLE 80. PRT0 TIMER DATA REGISTER LOW (000CH) TMDROL

| | | | | | | | | |
|-----------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of PRT0 Counter | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|-----------------------------------------|
| 7-0 | LS byte of PRT0 Counter | R/W | | The LS 8 bits of the PRT0 down-counter. |

TABLE 81. PRT0 TIMER DATA REGISTER HIGH (000DH) TMDROH

| | | | | | | | | |
|-----------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS Byte of PRT0 Counter | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|-----------------------------------------|
| 7-0 | MS byte of PRT0 Counter | R/W | | The MS 8 bits of the PRT0 down-counter. |

TABLE 82. PRT0 RELOAD REGISTER LOW (000EH) RLDR0L

| | | | | | | | | |
|-----------|------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of PRT0 Reload Value | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------------------------|-----|-------|----------------------------------------------------------------------------------------------------|
| 7-0 | LS byte of PRT0 Reload Value | R/W | | The LS 8 bits of the value that is loaded into the PRT0 down-counter, when it is decremented to 0. |

TABLE 83. PRT0 TIMER DATA REGISTER LOW (000CH) TMDROL

| | | | | | | | | |
|---------------------------------------------------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of PRT0 Counter | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|-----------------------------------------|
| 7-0 | LS byte of PRT0 Counter | R/W | | The LS 8 bits of the PRT0 down-counter. |

TABLE 84. PRT0 TIMER DATA REGISTER HIGH (000DH) TMDROH

| | | | | | | | | |
|---------------------------------------------------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS Byte of PRT0 Counter | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|-----------------------------------------|
| 7-0 | MS byte of PRT0 Counter | R/W | | The MS 8 bits of the PRT0 down-counter. |

TABLE 85. PRT0 RELOAD REGISTER LOW (000EH) RLDR0L

| | | | | | | | | |
|---------------------------------------------------|------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of PRT0 Reload Value | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------------------------|-----|-------|----------------------------------------------------------------------------------------------------|
| 7-0 | LS byte of PRT0 Reload Value | R/W | | The LS 8 bits of the value that is loaded into the PRT0 down-counter, when it is decremented to 0. |

TABLE 86. PRT0 RELOAD REGISTER HIGH (000FH) RLDR0H

| | | | | | | | | |
|---------------------------------------------------|------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS Byte of PRT0 Reload Value | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------------------------|-----|-------|--------------------------------------------------------------------------------------------------|
| 7-0 | MS byte of PRT0 Reload Value | R/W | | The MS 8 bits of the value that is loaded into PRT0's down-counter, when it is decremented to 0. |

TABLE 87. TIMER CONTROL REGISTER (0010H) TCR

| | | | | | | | | |
|----------------------------------------------------------------------|------|------|------|------|-------|---|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | TIF1 | TIF0 | TIE1 | TIE0 | Resvd | | TDE1 | TDE0 |
| R/W | R | R | R/W | R/W | ? | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | X | X | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate ? = Not Applicable | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-6 | TIF1,0 | R | | One of these status bits is set when a PRT decrements its down-counter to 0. It is cleared when software has read this register and either byte of the TMDR. |
| 5-4 | TIE1,0 | R/W | | If one of these bits is 1, the corresponding_PRT requests an interrupt when its down-counter has counted down to 0 and it has set the TIF bit. |
| 1-0 | TDE1,0 | R/W | 0 1 | The corresponding PRT is stopped. The corresponding PRT is running. |

TABLE 88. PRT1 TIMER DATA REGISTER LOW (0014H) TMDR1L

| | | | | | | | | |
|---------------------------------------------------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of PRT1 Counter | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|---------------------------------------|
| 7-0 | LS byte of PRT1 Counter | R/W | | The LS 8 bits of PRT1's down-counter. |

TABLE 89. PRT1 TIMER DATA REGISTER HIGH (0015H) TMDR1H

| | | | | | | | | |
|---------------------------------------------------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS Byte of PRT1 Counter | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------------|-----|-------|---------------------------------------|
| 7-0 | MS byte of PRT1 Counter | R/W | | The MS 8 bits of PRT1's down-counter. |

TABLE 90. PRT1 RELOAD REGISTER LOW (0016H) RLDR1L

| | | | | | | | | |
|---------------------------------------------------|------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of PRT1 Reload Value | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------------------------|-----|-------|--------------------------------------------------------------------------------------------------|
| 7-0 | LS byte of PRT1 Reload Value | R/W | | The LS 8 bits of the value that is loaded into PRT1's down-counter, when it is decremented to 0. |

TABLE 91. PRT1 RELOAD REGISTER HIGH (0017H) RLDR1H

| | | | | | | | | |
|---------------------------------------------------|------------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS Byte of PRT1 Reload Value | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------------------------|-----|-------|--------------------------------------------------------------------------------------------------|
| 7-0 | MS byte of PRT1 Reload Value | R/W | | The MS 8 bits of the value that is loaded into PRT1's down-counter, when it is decremented to 0. |

REAL TIME CLOCK (RTC) REGISTERS

See the section titled “Real Time Clock (RTC)” on page 46, for more about these registers.

TABLE 92. RTC CONTROL/STATUS REGISTER (006FH) RTCCS

| | | | | | | | | |
|-----------|-------|-----|-------|--------------|---|-------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Alarm | IE | Resvd | Clock Select | | Resvd | | |
| R/W | R/W | R/W | ? | R/W | | ? | | |
| Reset | 0 | 0 | X | 0 | 0 | X | X | X |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable

This register can only be written if the Register Write Enable bit (WDTMR 0) is 1.

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | Alarm | R/W | | The RTC sets this bit when the hours, minutes, and seconds registers equal the programmed alarm value. Software can set this bit to 0, but either: <ul style="list-style-type: none"> – wait a second before doing so or – change the Alarm value before doing so, to prevent the continuing match from resetting Alarm. |
| 6 | IE | R/W | | If this bit is 1, the RTC requests an interrupt when the Alarm bit is 1. If the service routine for this interrupt clears Alarm by method a) above, it clears this bit during the <i>match second</i> to avoid further interrupts. |
| 4-3 | Clock Select | R/W | 0x | The RTC takes its clock from the LFXTAL and LFEXTAL pins, which must be connected to a 32.768 KHz crystal. |
| | | | 10 | The RTC takes its clock from the PC0 pin, which must be programmed as an input and connected to a 60 Hz line frequency. |
| | | | 11 | The RTC takes its clock from the PC0 pin, which must be programmed as an input and connected to a 50 Hz line frequency. |

TABLE 93. RTC CONTROL/STATUS REGISTER (006FH) RTCCS

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|-------|-----|-------|--------------|---|-------|---|---|
| Bit/Field | Alarm | IE | Resvd | Clock Select | | Resvd | | |
| R/W | R/W | R/W | ? | R/W | | ? | | |
| Reset | 0 | 0 | X | 0 | 0 | X | X | X |

Note: R = Read W = Write X = Indeterminate ? = Not Applicable
This register can only be written if the Register Write Enable bit (WDTMR 0) is 1.

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | Alarm | R/W | | The RTC sets this bit when the hours, minutes, and seconds registers equal the programmed alarm value. Software can write a 0 to this bit to 0, but should change the Alarm value before doing so, to prevent the continuing match from setting Alarm again. |
| 6 | IE | R/W | | If this bit is 1, the RTC requests an interrupt when the Alarm bit is 1. |
| 4-3 | Clock Select | R/W | 0x | The RTC takes its clock from the LFXTAL and LFEXTAL pins, which must be connected to a 32.768 KHz crystal. |
| | | | 10 | The RTC takes its clock from the PC0 pin, which must be programmed as an input and connected to a 60 Hz line frequency. |
| | | | 11 | The RTC takes its clock from the PC0 pin, which must be programmed as an input and connected to a 50 Hz line frequency. |

TABLE 94. RTC SECONDS REGISTER (0070H) RTCSEC

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---|---------|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | Seconds | | | | | | |
| R/W | R | R/W | | | | | | |
| Reset | 0 | NC | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-------------------------------------------------------------------|
| 6-0 | Seconds | R/W | | Incremented by the RTC once per second. BCD 59H is followed by 0. |

TABLE 95. RTC MINUTES REGISTER (0071H) RTCMIN

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---|---------|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | Minutes | | | | | | |
| R/W | R | R/W | | | | | | |
| Reset | 0 | NC | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------|
| 6-0 | Minutes | R/W | | Incremented by the RTC when the Seconds register increments from 59H to 0. This BCD register also increments from 59H to 0. |

TABLE 96. RTC HOURS REGISTER (0072H) RTCHR

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|----|----|-------|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | 0 | Hours | | | | | |
| R/W | R | R | R/W | | | | | |
| Reset | NC | NC | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------|
| 5-0 | Hours | R/W | | Incremented by the RTC when the Minutes register increments from 59H to 0. This BCD register increments from 23H to 0. |

TABLE 97. RTC DAY OF THE WEEK REGISTER (0073H) RTCDAY

| | | | | | | | | |
|-----------|---|---|---|---|---|-----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | 0 | 0 | 0 | 0 | Day | | |
| R/W | R | R | R | R | R | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | NC | NC | NC |

Note: R = Read W = Write X = Indeterminate
This register can only be written if the Register Write Enable bit (WDTMR 0) is 1.

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|----------------------------------------------------------------------------------------------------------------|
| 2-0 | Day | R/W | | Incremented by the RTC when the Hours register increments from 23H to 0. This register increments from 7 to 1. |

TABLE 98. RTC DATE REGISTER (0074H) RTCDAT

| | | | | | | | | |
|-----------|---|---|------|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | 0 | Date | | | | | |
| R/W | R | R | R/W | | | | | |
| Reset | 0 | 0 | NC | NC | NC | NC | NC | NC |

Note: R = Read W = Write X = Indeterminate
This register can only be written if the Register Write Enable bit (WDTMR 0) is 1.

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5-0 | Date | R/W | | Incremented by the RTC when the Hours register increments from 23H to 0. This BCD register increments from 28H, 30H, or 31H to 1, depending on the month and (for February) the year and century. |

TABLE 99. RTC MONTH REGISTER (0075H) RTCMO

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---|---|---|-------|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | 0 | 0 | Month | | | | |
| R/W | R | R | R | R/W | | | | |
| Reset | 0 | 0 | 0 | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------|
| 4-0 | Month | R/W | | Incremented by the RTC when the Date register increments to 1. This BCD register increments from 12H to 1. |

TABLE 100. RTC YEAR REGISTER (0076H) RTCYR

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|------|----|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Year | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | NC | NC | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|----------------------------------------------------------------------------------------------------------------------|
| 7-0 | Year | R/W | | Incremented by the RTC when the Month register increments from 12H to 1. This BCD register increments from 99H to 0. |

TABLE 101. RTC CENTURY REGISTER (0077H) RTCC

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---|---|---------|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | 0 | Century | | | | | |
| R/W | R | R | R/W | | | | | |
| Reset | 0 | 0 | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|--------------------------------------------------------------------------------------------|
| 5-0 | Century | R/W | | Incremented by the RTC in a BCD fashion, when the Years register increments from 99H to 0. |

TABLE 102. RTC ALARM SECONDS REGISTER (0078H) ALARMS

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---|---------------|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | Alarm Seconds | | | | | | |
| R/W | R | R/W | | | | | | |
| Reset | 0 | NC | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------|-----|-------|------------------------------------------------|
| 6-0 | Alarm Seconds | R/W | | The seconds component of the Alarm time (BCD). |

TABLE 103. RTC ALARM MINUTES REGISTER (0079H) ALARMM

| | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------|---|---------|----|----|----|----|----|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | 0 | Minutes | | | | | | |
| R/W | R | R/W | | | | | | |
| Reset | 0 | NC | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|----------------------------------------------------------------------------------------------------------------|
| 6-0 | Year | R/W | | Incremented by the RTC when the Month register increments from 12 to 1. This register increments from 99 to 0. |

TABLE 104. RTC ALARM HOURS REGISTER (007AH) ALARMH

| | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|

TABLE 104. RTC ALARM HOURS REGISTER (007AH) ALARMH

| | | | | | | | | |
|-----------------------------------------------------------------------------------------------------------------------------------------|---|---|-------------|----|----|----|----|----|
| Bit/Field | 0 | 0 | Alarm Hours | | | | | |
| R/W | R | R | R/W | | | | | |
| Reset | 0 | 0 | NC | NC | NC | NC | NC | NC |
| Note: R = Read W = Write X = Indeterminate This register can only be written if the Register Write Enable bit (WDTMR 0) is 1. | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------|-----|-------|----------------------------------------------|
| 4-0 | Alarm Hours | R/W | | The hours component of the Alarm time (BCD). |

DIGITAL TO ANALOG CONVERTER (DAC) REGISTERS

See “Digital/Analog Converter (DAC)” on page 48, for more about these registers.

TABLE 105. DAC CONTROL REGISTER (0069H) DACCR

| | | | | | | | | |
|----------------------------------------------------------------------|----------|---|------------|---|-------|---------------|-------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Data 1-0 | | Ref Select | | Resvd | Output Enable | Resvd | |
| R/W | R/W | | R/W | | ? | R/W | ? | |
| Reset | X | X | X | X | X | 0 | X | X |
| Note: R = Read W = Write X = Indeterminate ? = Not Applicable | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|----------------|-------------------------------------------------------------------------------------------------|
| 7-6 | Data 1-0 | R/W | | This field contains the 2 LS bytes of the 10-bit digital value that the DAC converts to analog. |
| 5-4 | Reference Voltage Select | R/W | 00 01 1x | Internal reference 2.6V Internal reference 4.2V DAC reference is PA0 pin |
| 2 | Output Enable | R/W | | A 1 in this bit enables the DAC to drive its analog result voltage onto the AOUT pin. |

TABLE 106. DAC DATA REGISTER (006AH) DAC

| | | | | | | | | |
|---------------------------------------------------|----------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Data 9-2 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | Data 9-2 | R | | Writing to this register sets bits 9-2 of the digital value the DAC is to convert. With bits 7-6 of the DACCR, this value is a 10-bit binary fraction of the selected reference voltage. For example, 1000000000B means that the DAC outputs half of the reference voltage. The DAC continuously converts this value, so rewriting this register produces the corresponding voltage on AOUT within the specified conversion time. |

ANALOG TO DIGITAL CONVERTER (ADC) REGISTERS

See section “Analog/Digital Converter (ADC)” on page 49, for more about these registers.

TABLE 107. ADC CONTROL REGISTER 0 (0066H) ADCC0

| | | | | | | | | |
|----------------------------------------------------------------------|-------|---|---|---|---|--------|------------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Resvd | | | | | Enable | Ref Select | |
| R/W | ? | | | | | R/W | R/W | |
| Reset | X | X | X | X | X | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate ? = Not Applicable | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|----------------|------------------------------------------------------------------------------------------------------------------------|
| 2 | Enable | R/W | | This bit enables the ADC and must be set prior to starting an ADC conversion. When this bit is 0, the ADC powers down. |
| 1-0 | Reference Voltage Select | R/W | 0x 10 11 | ADC Reference voltage is PA0 pin Internal Reference 4.2 V Internal Reference 2.6 V |

TABLE 108. ADC CONTROL REGISTER 1 (0067H) ADCC1

| | | | | | | | | |
|---------------------------------------------------|------------|---|-----|----|-------|----------------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Result 1-0 | | IE | CC | Start | Channel Select | | |
| R/W | R | | R/W | R | W | R/W | | |
| Reset | X | X | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-6 | Result 1-0 | R | | When a conversion is completed, these bits hold the LS bits of the digital value |
| 5 | IE | R/W | | If this bit is 1 when the ADC completes a conversion, an interrupt is requested. |
| 4 | Conversion Complete (CC) | R | | The ADC sets this bit when it completes a conversion. This bit is cleared when software writes a 1 to the Start bit, to begin a new conversion. |
| 3 | Start | W | | Writing a 1 to this bit causes the ADC to start a new conversion. If the ADC had a conversion in progress, it is aborted. |
| 2-0 | Channel Select | R/W | | These bits select which pin is sampled for a conversion. |
| | | | 000 | Pin PD0 |
| | | | 001 | Pin PD1 |
| | | | 010 | Pin PD2 |
| | | | 011 | Pin PD3 |
| | | | 100 | Pin PD4 |
| | | | 101 | Pin PD5 |
| | | | 110 | Pin PD6 |
| | | | 111 | Pin PD7 |

TABLE 109. ADC RESULT REGISTER (0068H) ADC

| | | | | | | | | |
|---------------------------------------------------|------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Result 9-2 | | | | | | | |
| R/W | R | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------|-----|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | Result 9-2 | R | | After the ADC has completed a conversion, this register contains bits 9-2 of the result digital value. With bits 7-6 of the ADCC1, this value is a 10-bit binary fraction of the selected reference voltage. For example, 1000000000B means that the voltage on the selected pin was half of the reference voltage. |

PROGRAMMABLE INPUT/OUTPUT SEQUENCER (PIOS) REGISTERS

See section “Programmable I/O Sequencer (PIOS)” on page 52, for more about these registers.

TABLE 110. PIOS CONTROL REGISTER (0060H) PIOSCR

| | | | | | | | | |
|----------------------------------------------------------------------|--------|-----|-----|-------|---|---|--------------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Enable | IE | IP | Resvd | | | Clock Select | |
| R/W | R/W | R/W | R/W | ? | | | R/W | |
| Reset | 0 | 0 | 0 | X | X | X | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate ? = Not Applicable | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | Enable | R/W | | A 1 in this bit enables the PIOS. |
| 6 | IE | R/W | | A 1 in this bit allows the PIOS to request an interrupt when it encounters an entry in its RAM, directing it to do so. |
| 5 | IP | R/W | | The PIOS sets this bit when it encounters an “interrupt” entry in its RAM. Interrupt service routines write a 0 to this bit to clear the PIOS interrupt request. |
| 1-0 | Clock Select | R/W | | This field determines the PIOS clock frequency |
| | | | 00 | PHI |
| | | | 01 | PHI/256 |
| | | | 10 | PHI/1024 |
| | | | 11 | PHI/4096 |

TABLE 111. PIOS ADDRESS/TYPE REGISTER (0061H) PIOSAT

| | | | | | | | | |
|---------------------------------------------------|--------------|---|---|---|---|---|------------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Next Address | | | | | | Entry Type | |
| R/W | R | | | | | | R | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|---------------------------------------------------------------------------------------------------------------|
| 7-2 | Next Address | R | | Software can read the Next Address value from the RAM entry most recently fetched by the PIOS, in this field. |
| 1-0 | Entry Type | R | | Software can read the Entry Type value from the RAM entry most recently fetched by the PIOS, in this field. |

TABLE 112. PIOS COUNTER LOW (0062H) PIOSCL

| | | | | | | | | |
|---------------------------------------------------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS byte of PIOS Counter | | | | | | | |
| R/W | R | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------|-----|-------|----------------------------------------------------------------------------------------|
| 7-0 | Counter LS byte | R | | Software can read the LS 8 bytes of the current PIOS Counter value from this register. |

TABLE 113. PIOS COUNTER HIGH (0063H) PIOSCH

| | | | | | | | | |
|---------------------------------------------------|-------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MS byte of PIOS Counter | | | | | | | |
| R/W | R | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------|-----|-------|------------------------------------------------------------------------------------------------------------------------|
| 7-0 | Counter MS byte | R | | Software can read the MS 8 bytes of the PIOS Counter value, at the time that PIOSCL was last read, from this register. |

ASYNC SERIAL COMMUNICATIONS INTERFACE (ASCI) REGISTERS

See section “Clocked Serial Input/Output Module (CSI/O)” on page 67, for more detail about these registers.

TABLE 114. ASCIO CONTROL REGISTER A (0000H) CNTLA0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|-----|-----|-----|----------|--------------|------|------|------|
| Bit/Field | MPE | RE | TE | Reserved | MPBR/ EFR | MOD2 | MOD1 | MOD0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 1 | X | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|------------------------------------|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | Multi-processor Mode Enable | R/W | | If this bit and the MP bit in CNTLB are both 1, only received characters having a 1 in an additional bit between the last data bit and the stop bit are placed in the Rx FIFO. If either the MP bit or this bit is 0, all received characters are placed in the Rx FIFO. |
| 6 | Receive Enable | R/W | | A 1 in this bit enables the receiver. Writing a 0 stops reception. |
| 5 | Transmit Enable | R/W | | A 1 in this bit enables the transmitter. Writing a 0 stops transmission. |
| 4 | Reserved | R/W | | On other 180 devices this bit controlled the $\overline{RTS0}$ output. |
| 3 | MP Bit Rcv/ Error Flag Reset | R/W | | Reading this bit returns the value of the multiprocessor (MP) bit. Read this register before reading the RDR. Writing a 0 to this bit clears the OVRN, FE, PE, and Break Detect bits. Writing a 1 has no effect. |
| 2 | MOD2 | R/W | 0 1 | 7 bit data (Tx and Rx) 8 bit data |
| 1 | MOD1 | R/W | 0 1 | No parity (Tx and Rx) Parity generated (Tx), checked (Rx) |
| 0 | MOD0 | R/W | 0 1 | 1 Stop bit transmitted 2 Stop bits transmitted |

TABLE 115. ASCI1 CONTROL REGISTER A (0001H) CNTLA1

| | | | | | | | | |
|----------------------------------------------------------------------|-----|-----|-----|-------|--------------|------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MPE | RE | TE | Resvd | MPBR/ EFR | MOD2 | MOD1 | MOD0 |
| R/W | R/W | R/W | R/W | ? | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | X | 1 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate ? = Not Applicable | | | | | | | | |

All bits in this register are as described in Table 114.

TABLE 116. ASCI0 CONTROL REGISTER B (0002H) CNTLB0

| | | | | | | | | |
|---------------------------------------------------|------|-----|---------------------------------|-----|-----|--------------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | MPBT | MP | $\overline{\text{CTS}}$ / PS | PEO | DR | Speed Select | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | X | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------------------------|-----|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | Multi-Processor Bit Tx | R/W | | If the MP bit (bit 6) is 1, this bit defines the value to send in the MP bit with the next character written to the Transmit Data Register. |
| 6 | Multiprocessor Mode | R/W | | If this bit is 1, the ASCI sends, and expects to receive, an extra bit after the last data bit. The extra bit is 1 in address characters that begin frames and 0 in following data characters. |
| 5 | $\overline{\text{CTS}}$ /PS | R/W | | Reading this pin returns the state of the CTS pin (0 is Low, 1 is High). For writing, this bit is PS. If bits 2–0 in this register are not 111 and the BRG Mode bit in the Extension Control register is 0, PS determines whether the PHI clock is prescaled by 10 (for 0) or 30 (for 1) as the first stage in ASCI clocking. |
| 4 | Parity Even/Odd | R/W | | If MOD1 (CNTLA bit 1) is 1, this bit selects whether parity is generated and checked as even (for 0) or odd (for 1). |
| 3 | DR | R/W | 0 1 | The ASCI divides its basic clock by 16 to obtain its bit rate. The ASCI divides its basic clock by 64 to obtain its bit rate. |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2-0 | Speed Select | R/W | 111 (other) 000 001 010 011 100 101 110 | Do not program this value. If the BRGO Mode bit is 1, the output of the new BRG is the basic clock of the ASCI. If BRGO Mode is 0, these bits determine what the output of the Prescaler is divided by, to obtain basic clock for the ASCI: used as is Divide by 2 Divide by 4 Divide by 8 Divide by 16 Divide by 32 Divide by 64 The ASCI divides the basic clock by 16 or 64 to obtain the bit rate. |

TABLE 117. ASCI1 CONTROL REGISTER B (0003H) CNTLB1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|------|-----|-----|-----|-----|--------------|---|---|
| Bit/Field | MPBT | MP | PS | PEO | DR | Speed Select | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | X | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

All bits in this register are as described in Table 116, except that bit 5 has no function in write operations on the Z80S183 ASCI1.

TABLE 118. ASCI0 STATUS REGISTER (0004H) STAT0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|------|------|----|----|-----|------|------|-----|
| Bit/Field | RDRF | OVRN | PE | FE | RIE | DCD0 | TDRE | TIE |
| R/W | R | R | R | R | R/W | R | R | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | pin | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | RDRF | R | | The Z80S183 sets this bit when a character is received. Reading the last received byte from the Rx FIFO clears this bit. See Note below. |
| 6 | OVRN | R | | This bit is set when the last character received before an Overrun condition comes to the top of the Rx FIFO. It is cleared when a 0 is written to the EFR bit in CNTL0. See Note below. |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 | PE | R | | This bit is set if parity is enabled, and a character with a Parity Error comes to the top of the Rx FIFO. It is cleared when a 0 is written to the EFR bit in CNTLO. See Note below. |
| 4 | FE | R | | This bit is set if a character with a Framing Error (one in which the Stop bit was sampled as 0) comes to the top of the Rx FIFO. It is cleared when a 0 is written to the EFR bit in CNTLO. See Note below. |
| 3 | RIE | R/W | | If this bit is 1, the ASCI requests an interrupt when any of the flags OVRN, PE, FE, or Break Detect is set, or if bit 7 of the Extension Control register is 0 and RDRF is 1, or if bit 6 of the Extension Control register is 0 and DCD0 is Low. |
| 2 | DCD0 | R | | This bit is 1 whenever the $\overline{DCD0}$ pin is High. When $\overline{DCD0}$ goes Low, the next read of this register returns a 1. The next read returns a 0 if $\overline{DCD0}$ is still Low. |
| 1 | TDRE | R | | This bit is cleared when software writes a character to the TDR. It is set when the character leaves the TDR for transmission, by Reset, and in I/O Stop mode. It is cleared if bit 5 of the Extension Control register is 0 and $\overline{CTS0}$ is High. |
| 0 | TIE | R/W | | If this bit is 1, the ASCI requests an interrupt when TDRE is 1. |

Note: The RDRF, OVRN, PE, and FE bits are cleared by Reset, during I/O Stop mode, and for ASCI0, if the DCD Disable bit in the Extension Control register is 0 and $\overline{DCD0}$ is High.

TABLE 119. ASCI1 STATUS REGISTER (0005H) STAT1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|------|------|----|----|-----|----------|------|-----|
| Bit/Field | RDRF | OVRN | PE | FE | RIE | Reserved | TDRE | TIE |
| R/W | R | R | R | R | R/W | R | R | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | pin | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

This register is as described in Table 118, except that bit 2 has no function for Z80S183 ASCII.

TABLE 120. ASCIO Tx DATA REGISTER (0006H) TDR0

| | | | | | | | | |
|---------------------------------------------------|-----------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Character to Tx | | | | | | | |
| R/W | W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|------------------------------------------------------------------------------------------------------|
| 7-0 | Tx character | R/W | | Software can write a character to be transmitted to this register, when the TDRE flag in STAT0 is 1. |

TABLE 121. ASCI1 TX DATA REGISTER (0007H) TDR1

| | | | | | | | | |
|---------------------------------------------------|-----------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Character to Tx | | | | | | | |
| R/W | W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|------------------------------------------------------------------------------------------------------|
| 7-0 | Tx character | R/W | | Software can write a character to be transmitted to this register, when the TDRE flag in STAT1 is 1. |

TABLE 122. ASCIO RX DATA REGISTER (0008H) RDR0

| | | | | | | | | |
|---------------------------------------------------|--------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Received Character | | | | | | | |
| R/W | R | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|----------------------------------------------------------------------------------------------|
| 7-0 | Rx character | R/W | | When the RDRF flag in STAT0 is 1, software can read a received character from this register. |

TABLE 123. ASCI1 RX DATA REGISTER (0009H) RDR1

| | | | | | | | | |
|---------------------------------------------------|--------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | Received Character | | | | | | | |
| R/W | R | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------|-----|-------|----------------------------------------------------------------------------------------------|
| 7-0 | Rx character | R/W | | When the RDRF flag in STAT1 is 1, software can read a received character from this register. |

TABLE 124. ASCI0 EXTENSION CONTROL REGISTER (0012H) ASEXTO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|------|-------------------------------------|-------------------------------------|-------------|-------------|----------|-------------|-----------------------|
| Bit/Field | RDID | $\overline{\text{DCD0}}$ Disable | $\overline{\text{CTS0}}$ Disable | X1 Clock | BRG Mode | Start IE | Rx Break | Tx Break/ TxEnd |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: R = Read W = Write X = Indeterminate

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|----------------------------------|-----|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | Rx Data Interrupt Disable | R/W | | If RIE (STAT bit 3) and this bit are both 1, the ASCI requests receive interrupts only when OVRN, PE, or FE is set. If RIE is 1 and this bit is 0, the ASCI also requests interrupts when RDRF is set (for each received character). If RIE is 0 this bit has no effect. |
| 6 | $\overline{\text{DCD0}}$ Disable | R/W | 0 1 | $\overline{\text{DCD0}}$ auto-enables the receiver $\overline{\text{DCD0}}$ has no effect on the receiver |
| 5 | $\overline{\text{CTS0}}$ Disable | R/W | 0 1 | $\overline{\text{CTS0}}$ auto-enables the transmitter $\overline{\text{CTS0}}$ has no effect on the transmitter |
| 4 | X1 Clock | R/W | 0 | Always 0 on the Z80S183. |
| 3 | BRG Mode | R/W | 0 1 | The SS bits in Control Register B determine the factor by which the Prescaler output is divided, to obtain the ASCI's basic clock. 1 The ASCI's basic clock comes from the new BRG. |
| 2 | Start IE | R/W | | If this bit and RIE are both 1, the ASCI requests an interrupt when the start of a start bit is detected, for auto-bauding. Writing a 0 to this bit clears the interrupt request. |
| 1 | RxBreak | R | | This bit is 1 if the receiver has detected a Break condition, that is, if all bits in a character, including the Stop bit, are 0. The all-0 character is placed in the Rx FIFO if there is room, but the receiver does not assemble any more characters until the RXA pin has returned High. |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|---------------|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | TxBreak/TxEnd | R/W | | Writing a 1 to this bit makes the transmitter drive TXA Low to send a Break condition, until software writes a 0 to this bit. This bit reads as 0 while a character is transmitted, but becomes 1 when the number of Stop bits selected by MOD0 in CNTLA have been sent. |

TABLE 125. ASCI1 EXTENSION CONTROL REGISTER (0013H) ASEXT1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------------------------------------------------------------|------|-------|---|----------|----------|----------|----------|----------------|
| Bit/Field | RDID | Resvd | | X1 Clock | BRG Mode | Start IE | Rx Break | Tx Break/TxEnd |
| R/W | R/W | ? | | R/W | R/W | R/W | R | R/W |
| Reset | 0 | X | X | 0 | 0 | 0 | 0 | 0 |
| Note: R = Read W = Write X = Indeterminate ? = Not Applicable | | | | | | | | |

This register is as described in the previous table, except that bits 6–5 have no function for Z80S183 ASCI1.

TABLE 126. ASCIO TIME CONSTANT LOW (001AH) ASTCOL

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|--------------------------|---|---|---|---|---|---|---|
| Bit/Field | LS Byte of Time Constant | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|-------|--------------------------------------------------------------------------------|
| 7–0 | LS byte of Time Constant | R/W | | The Least Significant 8 bits of the ASCIO Baud Rate Generator’s Time Constant. |

TABLE 127. ASCI0 TIME CONSTANT HIGH (001BH) ASTC0H

| | | | | | | | | |
|---------------------------------------------------|--------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of Time Constant | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|-------|-------------------------------------------------------------------------------|
| 7-0 | MS byte of Time Constant | R/W | | The Most Significant 8 bits of the ASCI0 Baud Rate Generator's Time Constant. |

TABLE 128. ASCI1 TIME CONSTANT LOW (001CH) ASTC1L

| | | | | | | | | |
|---------------------------------------------------|--------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of Time Constant | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|-------|--------------------------------------------------------------------------------|
| 7-0 | LS byte of Time Constant | R/W | | The Least Significant 8 bits of the ASCI1 Baud Rate Generator's Time Constant. |

TABLE 129. ASCI0 TIME CONSTANT HIGH (001DH) ASTC1H

| | | | | | | | | |
|---------------------------------------------------|--------------------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | LS Byte of Time Constant | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|--------------------------|-----|-------|-------------------------------------------------------------------------------|
| 7-0 | MS byte of Time Constant | R/W | | The Most Significant 8 bits of the ASCI1 Baud Rate Generator's Time Constant. |

CLOCKED SERIAL I/O (CSI/O) REGISTERS

See section , which starts on page 67, for more about these registers.

TABLE 130. CSI/O CONTROL REGISTER (000AH) CNTR

| | | | | | | | | |
|-----------|----|-----|-----|-----|-------|-------------------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit/Field | EF | EIE | RE | TE | Resvd | Speed Select (SS) | | |
| R/W | R | R/W | R/W | R/W | ? | R/W | | |
| Reset | 0 | 0 | 0 | 0 | X | 1 | 1 | 1 |

Note: R = Read W = Write X = Indeterminate? = Not Applicable

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|----------------------------|-----|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | End Flag (EF) | R | | The CSI/O sets this bit to 1 when it finishes sending or receiving a byte. It clears this bit when software reads or writes the TRDR, on Reset, and during I/O Stop mode. |
| 6 | End Interrupt Enable (EIE) | R/W | | If this bit is 1, the CSI/O requests an interrupt when it completes sending or receiving a byte and sets EF. |
| 5 | Receive Enable (RE) | R/W | | Write a 1 to this bit to start a CSI/O receive operation. If the SS bits are 111, the CSI/O waits for 8 clock pulses on CKS. Otherwise, it outputs 8 clock pulses on CKS. In either case, it clocks data on RXS into the TRDR at each rising edge on CKS. After capturing the 8th bit, it clears this bit and sets EF. |
| 4 | Transmit Enable (TE) | R/W | | Write a 1 to this bit to start CSI/O transmission. If the SS bits are 111, the CSI/O waits for 8 clock pulses on CKS. Otherwise, it outputs 8 clock pulses on CKS. In either case, it clocks data from the TRDR onto TXS at each falling edge on CKS. After sending 8 bits, the CSI/O clears this bit and sets EF. |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-------------------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2-0 | Speed Select (SS) | R/W | | If these bits are 111, as they are after a Reset, the CSI/O takes external clocking from the CKS pin. Otherwise, it drives a clock onto CKS, that it derives from PHI as follows: |
| | | | 000 | PHI/20 |
| | | | 001 | PHI/40 |
| | | | 010 | PHI/80 |
| | | | 011 | PHI/160 |
| | | | 100 | PHI/320 |
| | | | 101 | PHI/640 |
| | | | 110 | PHI/1280 |

TABLE 131. CSI/O DATA REGISTER (000BH) TRDR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------------------------------------------|---------------------------------------|---|---|---|---|---|---|---|
| Bit/Field | Byte to Send (W) or Received Byte (R) | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | X | X | X | X | X | X | X | X |
| Note: R = Read W = Write X = Indeterminate | | | | | | | | |

| Bit Position | Bit/Field | R/W | Value | Description |
|--------------|-----------|-----|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7-0 | | R/W | | Software writes a byte to this register, before setting the TE bit, allowing the CSI/O to send it. Software reads a received byte from this register, after the CSI/O sets the EF bit in response to software setting the RE bit. |

INSTRUCTION SET

The Z80S183 includes the 8S180 processor, which is descended from the ZiLOG Z80. The 8-bit data bus and 20-bit address space fit well into a wide variety of mid-range embedded processing applications. This processor provides significantly more computing power than a microcontroller, at a fraction of the system cost of a larger microprocessor.

For details of these instructions see the Z80S183 User Manual, or the Z8S180 or Z80185 User Manuals until the Z80S183 UM is available.

CLASSES OF INSTRUCTIONS
TABLE 132. LOAD INSTRUCTION

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|--------------------|
| LD | dst,src | Load |
| POP | dst | Pop |
| PUSH | src | Push |

TABLE 133. ARITHMETIC INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|---------------------------------------|
| ADC | dst,src | Add with Carry |
| ADD | dst,src | Add |
| CP | A,src | Compare |
| CPD(R) | | Block Scan, decrementing (and Repeat) |
| CPI(R) | | Block Scan, incrementing (and Repeat) |
| DAA | | Decimal Adjust Accumulator |
| DEC | dst | Decrement |
| INC | dst | Increment |
| MLT | rr | Multiply |
| NEG | | Negate Accumulator |
| SBC | dst,src | Subtract with Carry |
| SUB | A,src | Subtract |

TABLE 134. LOGICAL INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|------------------------|
| AND | A,src | Logical AND |
| CPL | | Complement accumulator |
| OR | A,src | Logical OR |
| TST | A,src | Test accumulator |
| XOR | A,src | Logical Exclusive OR |

TABLE 135. EXCHANGE INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|------------------------------------|
| EX | AF,AF' | Exchange Accumulator and Flags |
| EX | DE,HL | Exchange DE and HL |
| EX | (SP),rr | Exchange register and top of stack |
| EXX | | Exchange register banks |

TABLE 136. PROGRAM CONTROL INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|----------|----------|-----------------------------------|
| CALL | cc,dst | Conditional Call |
| CALL | dst | Call |
| DJNZ | dst | Decrement and Jump if Non-Zero |
| JP | cc,dst | Conditional Jump |
| JP | dst | Jump |
| JR | cc',dst | Conditional Jump Relative |
| JR | dst | Jump Relative |
| RET | cc | Conditional Return |
| RET | | Return |
| RETI | | Return from Interrupt |
| RETN | | Return from Nonmaskable interrupt |
| RST | dst | Restart |

TABLE 137. BIT MANIPULATION INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|----------|----------|-------------|
| BIT | n,src | Bit test |
| RES | n,dst | Reset bit |
| SET | n,dst | Set bit |

TABLE 138. BLOCK TRANSFER INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|----------|----------|---------------------------------------|
| LDD(R) | | Block Move, decrementing (and Repeat) |
| LDI(R) | | Block Move, incrementing (and Repeat) |

TABLE 139. ROTATE AND SHIFT INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|----------|----------|-----------------------------------|
| RL | dst | Rotate Left |
| RLA | | Rotate Left Accumulator |
| RLC | dst | Rotate Left Circular |
| RLCA | | Rotate Left Circular Accumulator |
| RLD | | Rotate Left Decimal |
| RR | dst | Rotate Right |
| RRA | | Rotate Right Accumulator |
| RRC | dst | Rotate Right Circular |
| RRCA | | Rotate Right Circular Accumulator |
| RRD | | Rotate Right Decimal |

TABLE 139. ROTATE AND SHIFT INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|------------------------|
| SLA | dst | Shift Left |
| SRA | dst | Shift Right Arithmetic |
| SRL | dst | Shift Right Logical |

TABLE 140. INPUT/OUTPUT INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|-------------------------------------------------|
| IN | A, (n) | Input to A from port n |
| IN | r, (C) | Input to register from port in BC |
| INO | r, (n) | Input to r from port n in page 0 |
| IND(R) | | Block Input, decrementing (and Repeat) |
| INI(R) | | Block Input, incrementing (and Repeat) |
| OTDM(R) | | Block Output, page 0, decrementing (and Repeat) |
| OTIM(R) | | Block Output, page 0, incrementing (and Repeat) |
| OUT | (n), A | Output from A to port n |
| OUT | (C), r | Output from register to port in BC |
| OUTO | (n), r | Output from register to port n in page 0 |
| OUTD (OTDR) | | Block Output, decrementing (and Repeat) |
| OUTI (OTIR) | | Block Output, incrementing (and Repeat) |
| TSTIO | n | Test port (0,C) under mask |

TABLE 141. PROCESSOR CONTROL INSTRUCTIONS

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|-----------------------|
| CCF | | Complement Carry Flag |
| DI | | Disable Interrupts |
| EI | | Enable Interrupts |
| HALT | | Halt |
| IM | 0/1/2 | Interrupt Mode |
| NOP | | No Operation |
| SCF | | Set Carry Flag |
| SLP | | Sleep |

PROCESSOR FLAGS

Table 142 describes the Flag register. Bits in this register are set and cleared by certain instructions as described in the Z80S183 User Manual. Some of the Flags are tested by conditional JR, JP, CALL, and RET instructions, and some are used

by subsequent instructions such as ADC, SBC, and DAA. Accumulator A can also perform PUSH and POP instructions on the Flags.

TABLE 142. FLAG REGISTERFF

| | | | | | | | | |
|--------------------------------|---|---|---|----|---|-----|---|----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | S | Z | x | HC | x | P/V | N | CF |
| Reset | 0 | 0 | x | 0 | x | 0 | 0 | 0 |
| Note: X = Indeterminate | | | | | | | | |

| Bit/ Field | Bit Position | Description |
|---------------|-----------------|--------------------------|
| S | 7 | Sign Flag |
| Z | 6 | Zero Flag |
| | 5 | Reserved, do not program |
| HC | 4 | Half-carry Flag |
| | 3 | Reserved, do not program |
| P/V | 2 | Parity or Overflow Flag |
| N | 1 | Add/Subtract Flag |
| CF | 0 | Carry Flag |

CONDITION CODES

Table 143 describes the codes used in the *Flags Affected* columns of the Instruction Summary Table, Table 146, to indicate how each flag is affected by each type of instruction.

TABLE 143. FLAG SETTINGS DEFINITIONS

| Symbol | Definition |
|--------|---------------------------------------------------------|
| 0 | Cleared to 10 |
| 1 | Set to 1 |
| * | Set or cleared according to the result of the operation |
| - | Unaffected |
| X | Undefined |
| V | Set if Overflow or Underflow |
| P | Set if Parity or result is Even |
| NZ | Set if the count in B or BC is non-zero |

Table 144 describes the condition codes that can be used in conditional JP, CALL, and RET instructions in assembly language. A subset of these codes can also be used in JR instructions, which are shorter and faster than JPs.

TABLE 144. CONDITION CODES

| Mnemonic | Definition | Flag Settings | Valid in JR? |
|----------|---------------|---------------|--------------|
| C | Carry | CF = 1 | Y |
| NC | No Carry | CF = 0 | Y |
| Z | 0 | Z = 1 | Y |
| NZ | Non-Zero | Z = 0 | Y |
| M | Minus | S = 1 | N |
| P | Positive or 0 | S = 0 | N |
| PE | Parity Even | P/V = 1 | N |
| PO | Parity Odd | P/V = 0 | N |
| V | Overflow | P/V = 1 | N |
| NV | No Overflow | P/V = 0 | N |

NOTATION

Table 145 describes other notation used in the subsequent Instruction Summary table.

TABLE 145. SYMBOLS

| Symbol | Definition |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| (aa) | (mn), (IX ± d), (IY ± d), (BC), (DE), or (HL). |
| (BC), (DE), (HL) | The 8-bit contents of memory, at the address pointed to by a register pair |
| (IX ± d), (IY ± d) | The 8-bit content of memory at the address formed by adding the contents of the index register and the signed displacement d in the instruction. |
| (mn) | The 8-bit content of memory at the direct address mn |
| (SP) | The 16-bit contents of memory at the address pointed to by SP, and the next higher address. |
| ± d | Since d is signed, it would be more correct to just write + instead. But we write ± to emphasize that d is signed. |
| AF | A concatenated with F, with A as the more significant byte |
| b | A bit number 0–7 |
| cc | A condition code C, NC, Z, NZ, S, M, PE, PV, V, or NV |
| cc' | A condition code C, NC, Z, or NZ |
| d | An 8-bit signed displacement –128 to +127 |
| ee | A 16-bit register BC, DE, HL, SP, IX, or IY |
| IEF1,2 | The processor's two Interrupt Enable Flags. See the "Interrupt Registers" on page 80 section for more detail. |
| mn | A 16-bit immediate data value or direct address |
| n | A 8-bit immediate value or port number, 0–255 or 0–FFH |
| op1–op2 | A range of Op Code values, that includes some of the values between the Low and High values. See the Note. |
| PC | Program Counter |

TABLE 145. SYMBOLS

| Symbol | Definition |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| pp | A 16-bit register BC, DE, HL, SP, IX, IY, or AF |
| r, r' | An 8-bit register A, B, C, D, E, H, or L. |
| rr | A 16-bit register HL, IX, or IY. |
| s | An 8-bit register or memory location |
| SP | Stack Pointer |
| ss | A 16-bit register BC, DE, HL, or SP. |
| ss _H , ss _L | The more- and less-significant eight bits of a register pair |
| tt | A 16-bit register like ss, except that the value that designates HL in the ss encoding, here means <i>same as the destination register HL, IX, or IY.</i> |

Note: The symbol – between Op Codes (op1–op2), in the Op Codes column of the Instruction Summary table, indicates all the binary values between the lower and upper limits inclusive, that can be formed by incrementing the set of bits that differ between the lower and upper value.

Example: 00–C0 represents 00, 40, 80, and C0, while 40–BF represents all the values in that range.

ASSEMBLY LANGUAGE SYNTAX

For two-operand instructions, Z80 assembly language syntax puts the destination operand before the source operand.

Example: LD A, (1234) is a Load instruction, while LD (1234), A is a Store instruction.

Past Z80 assemblers allowed the destination operand to be omitted (implicit) if the Op Code mnemonic only allowed one destination operand, for example, AND L instead of AND A, L. Use of these short forms is discouraged because they are a source of possible error (the programmer thinks that the *implicit* destination is other than it really is). For the sake of legacy code, all known Z80 assemblers still accept the short form.

Caution: the assembly language uses C ambiguously, to designate one of the 8-bit registers as well as a condition code to test the Carry flag. This Product Specification uses CF to designate the Carry flag, and HC to designate the Half-Carry flag (as opposed to the 8-bit register H).

INSTRUCTION SUMMARY

The following table describes each type or class of instruction, using the notation described in the preceding sections. In cases where the Address Mode information can be both Destination (Dest) and Source code, this information spans both the Dest and Source columns (for example, the DEC instruction). The table is sorted by the assembly language mnemonics

TABLE 146. INSTRUCTION SUMMARY

| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|-----------------------------------------------------------------------------------------------------|--------------|------------|---------------------|----------------|---|----|-----|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| ADC A,s $A \leftarrow A + s + CF$ | | r | 88–8F | * | * | * | V | 0 | * |
| | | n | CE | | | | | | |
| | | (HL) | 8E | | | | | | |
| | | (IX/Y ± d) | DD/FD 8E | | | | | | |
| ADC HL,ss $HL \leftarrow HL + ss + CF$ | | | ED 4A–7A | * | * | * | V | 0 | * |
| | | | | | | | | | |
| ADD A,s $A \leftarrow A + s$ | | r | 80–87 | * | * | * | V | 0 | * |
| | | n | C6 | | | | | | |
| | | (HL) | 86 | | | | | | |
| | | (IX/Y ± d) | DD/FD 86 | | | | | | |
| ADD rr,tt $rr \leftarrow rr + tt$ | HL | | 09–39 | – | – | * | – | 0 | * |
| | IX/Y | | DD/FD 09–39 | | | | | | |
| AND A,s $A \leftarrow A \text{ and } s$ | | r | A0–A7 | * | * | 1 | P | 0 | 0 |
| | | n | E6 | | | | | | |
| | | (HL) | A6 | | | | | | |
| | | (IX/Y ± d) | DD/FD A6 | | | | | | |
| BIT b,m $Z \leftarrow \text{not (bit } b \text{ of } m)$ | | r | CB 40–7F | X | * | 1 | X | 0 | – |
| | | (HL) | CB 46–7E | | | | | | |
| | | (IX/Y ± d) | DD/FD CB d 46–7E | | | | | | |
| CALL cc,mn IF cc { $SP \leftarrow SP - 2$ $(SP) \leftarrow PC$ $PC \leftarrow mn$ } | | | C4–FC | – | – | – | – | – | – |
| | | | | | | | | | |
| CALL mn $SP \leftarrow SP - 2$ $(SP) \leftarrow PC$ $PC \leftarrow mn$ | | | CD | – | – | – | – | – | – |
| | | | | | | | | | |
| CCF $CF \leftarrow \text{not } CF$ | | | 3F | – | – | * | – | 0 | * |
| CP A,s $A - s$ | | r | B8–BF | * | * | * | V | 1 | * |
| | | n | FE | | | | | | |
| | | (HL) | BE | | | | | | |
| | | (IX/Y ± d) | DD/FD BE | | | | | | |
| CPD $A - (HL)$ $HL \leftarrow HL - 1$ $BC \leftarrow BC - 1$ | | | ED A9 | * | * | * | NZ | 1 | – |
| | | | | | | | | | |
| | | | | | | | | | |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|----------------------------------------------------------------------------------------------|--------------|--------|---------------------|----------------|---|----|-----|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| CPDR repeat {A ← (HL) HL ← HL - 1 BC ← BC - 1 } while (not Z and BC! = 0) | | | ED B9 | * | * | * | NZ | 1 | - |
| CPI A ← (HL) HL ← HL + 1 BC ← BC - 1 | | | ED A1 | * | * | * | NZ | 1 | - |
| CPIR repeat {A ← (HL) HL ← HL + 1 BC ← BC - 1 } while (not Z and BC! = 0) | | | ED B1 | * | * | * | NZ | 1 | - |
| CPL A ← not A | | | 2F | - | - | 1 | - | 1 | - |
| DAA A ← decimal adjust (A,F) | | | 27 | * | * | * | P | - | * |
| DEC ee ee ← ee - 1 | ss | | 0B-3B | - | - | - | - | - | - |
| | IX/Y | | DD/FD 2B | | | | | | |
| DEC m m ← m - 1 | r | | 05-3D | * | * | * | V | 1 | - |
| | (HL) | | 35 | | | | | | |
| | (IX/Y ± d) | | DD/FD 35 | | | | | | |
| DI IEF1,2 ← 0 | | | F3 | - | - | - | - | - | - |
| DJNZ d B ← B - 1 if B ≠ 0 {PC ← PC ± d} | | | 10 | - | - | - | - | - | - |
| EI IEF1,2 ← 1 | | | FB | - | - | - | - | - | - |
| EX AF,AF' AF ↔ AF' | | | 08 | * | * | * | * | * | * |
| EX (SP),rr (SP) ↔ rr | HL | | E3 | - | - | - | - | - | - |
| | IX/Y | | DD/FD E3 | | | | | | |
| EXX BC ↔ BC' DE ↔ DE' HL ↔ HL' | | | D9 | - | - | - | - | - | - |
| HALT | | | 76 | - | - | - | - | - | - |
| IM n | | | ED 40-58 | - | - | - | - | - | - |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|-------------------------------------------------------------------------------------------------------------------|--------------|--------|---------------------|----------------|---|----|-----|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| IN A,(n) $A \leftarrow (n)$ | | | DB | - | - | - | - | - | - |
| IN r,(C) $r \leftarrow (BC)$ | | | ED 40-78 | * | * | 0 | P | 0 | - |
| INO r,(n) $r \leftarrow (0,n)$ | | | ED 00-38 | * | * | 0 | P | 0 | - |
| INC ee $ee \leftarrow ee + 1$ | ss | | 03-33 | - | - | - | - | - | - |
| | IX/Y | | DD/FD 23 | | | | | | |
| INC m $m \leftarrow m + 1$ | r | | 04-3C | * | * | * | V | 0 | - |
| | (HL) | | 34 | | | | | | |
| | IX/Y | | DD/FD 34 | | | | | | |
| IND (HL) \leftarrow (C) $B \leftarrow B - 1$ $HL \leftarrow HL - 1$ | | | ED AA | X | * | X | X | 1 | - |
| INDR do {(HL) \leftarrow (C)} $B \leftarrow B - 1$ $HL \leftarrow HL - 1$ } while $B \neq 0$ | | | ED BA | X | 1 | X | X | 1 | - |
| INI (HL) \leftarrow (C) $B \leftarrow B - 1$ $HL \leftarrow HL + 1$ | | | ED A2 | X | * | X | X | 1 | - |
| INIR do {(HL) \leftarrow (C)} $B \leftarrow B - 1$ $HL \leftarrow HL + 1$ } while $B \neq 0$ | | | ED B2 | X | 1 | X | X | 1 | - |
| JP (rr) $PC \leftarrow rr$ | (HL) | | E9 | - | - | - | - | - | - |
| | (IX/Y) | | DD/FD E9 | | | | | | |
| JP cc,mn if cc { $PC \leftarrow mn$ } | | | C2-FA | - | - | - | - | - | - |
| JP mn $PC \leftarrow mn$ | | | C3 | - | - | - | - | - | - |
| JR cc',d if cc' { $PC \leftarrow PC \pm d$ } | | | 10-38 | - | - | - | - | - | - |
| JR d $PC \leftarrow PC \pm d$ | | | 18 | - | - | - | - | - | - |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|--------------------------------|--------------|------------|---------------------|----------------|---|----|------|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| LD (aa),A (aa) ← A | (BC) | | 02 | - | - | - | - | - | - |
| | (DE) | | 12 | | | | | | |
| | (HL) | | 77 | | | | | | |
| | (mn) | | 32 | | | | | | |
| | (IX/Y ± d) | | DD/FD 77 | | | | | | |
| LD (mn),ee (mn) ← ee | | HL | 22 | - | - | - | - | - | - |
| | | ss | ED 43-73 | | | | | | |
| | | IX/Y | DD/FD 22 | | | | | | |
| LD A,(aa) A ← (aa) | | (BC) | 0A | - | - | - | - | - | - |
| | | (DE) | 1A | | | | | | |
| | | (HL) | 7E | | | | | | |
| | | (mn) | 3A | | | | | | |
| | | (IX/Y ± d) | DD/FD 7E | | | | | | |
| LD A,I A ← I | | | ED 57 | * | * | 0 | IEF2 | 0 | - |
| LD A,R A ← R | | | ED 5F | * | * | 0 | IEF2 | 0 | - |
| LD ee,mn ee ← mn | ss | | 01-31 | - | - | - | - | - | - |
| | IX/Y | | DD/FD 21 | | | | | | |
| LD ee,(mn) ee ← (mn) | HL | | 2A | - | - | - | - | - | - |
| | ss | | ED 4B-7B | | | | | | |
| | IX/Y | | DD/FD 2A | | | | | | |
| LD I,A I ← A | | | ED 47 | - | - | - | - | - | - |
| LD m,n m ← n | r | | 06-3E | - | - | - | - | - | - |
| | (HL) | | 36 | | | | | | |
| | (IX/Y ± d) | | DD/FD 36 | | | | | | |
| LD m,r m ← r | r' | | 40-7F | - | - | - | - | - | - |
| | (HL) | | 70-77 | | | | | | |
| | (IX/Y ± d) | | DD/FD 70-77 | | | | | | |
| LD R,A R ← A | | | ED 4F | - | - | - | - | - | - |
| LD r,s r ← s | | r' | 40-7F | - | - | - | - | - | - |
| | | n | 06-3E | | | | | | |
| | | (HL) | 46-7E | | | | | | |
| | | (IX/Y ± d) | DD/FD 46-7E | | | | | | |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|------------------------------------------------------------------------------------------------|--------------|------------|---------------------|----------------|---|----|-----|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| LD SP,rr SP ← rr | | HL | F9 | - | - | - | - | - | - |
| | | IX/Y | DD/FD F9 | | | | | | |
| LDD (DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1 | | | ED A8 | - | - | 0 | NZ | 0 | - |
| LDDR do {(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1 } while BC != 0 | | | ED B8 | - | - | 0 | 0 | 0 | - |
| LDI (DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC - 1 | | | ED A0 | - | - | 0 | NZ | 0 | - |
| LDIR do {(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC - 1 } while BC != 0 | | | ED B0 | - | - | 0 | 0 | 0 | - |
| MLT ss ss ← ss _L * ss _H | | | ED 4C-7C | - | - | - | - | - | - |
| NEG A ← 0 - A | | | ED 44 | * | * | * | V | 1 | * |
| NOP | | | 00 | - | - | - | - | - | - |
| OR A,s A ← A OR s | | r | B0-B7 | * | * | 0 | P | 0 | 0 |
| | | n | F6 | | | | | | |
| | | (HL) | B6 | | | | | | |
| | | (IX/Y ± d) | DD/FD B6 | | | | | | |
| OTDM (O,C) ← (HL) B ← B - 1 C ← C - 1 HL ← HL - 1 | | | ED 8B | * | * | * | P | * | * |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|----------------------------------------------------------------------------------------------|--------------|--------|---------------------|----------------|---|----|-----|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| OTDMR do {(O,C) ← (HL)} B ← B - 1 C ← C - 1 HL ← HL - 1 } while B != 0 | | | ED 8B | 0 | 1 | 0 | 1 | * | 0 |
| OTDR do {(C) ← (HL)} B ← B - 1 HL ← HL - 1 } while B != 0 | | | ED BB | X | 1 | X | X | 1 | - |
| OTIM (O,C) ← (HL) B ← B - 1 C ← C + 1 HL ← HL + 1 | | | ED 83 | * | * | * | P | * | * |
| OTIMR do {(O,C) ← (HL)} B ← B - 1 C ← C + 1 HL ← HL + 1 } while B != 0 | | | ED 93 | 0 | 1 | 0 | 1 | * | 0 |
| OTIR do {(C) ← (HL)} B ← B - 1 HL ← HL + 1 } while B != 0 | | | ED B3 | X | 1 | X | X | 1 | - |
| OUT (C),r (BC) ← r | | | ED 41-79 | - | - | - | - | - | - |
| OUT (n),A (n) ← A | | | D3 | - | - | - | - | - | - |
| OUT0 (n),r (0,n) ← r | | | ED 01-79 | - | - | - | - | - | - |
| OUTD (C) ← (HL) B ← B - 1 HL ← HL - 1 | | | ED AB | X | * | X | X | 1 | - |
| OUTI (C) ← (HL) B ← B - 1 HL ← HL + 1 | | | ED AB | X | * | X | X | 1 | - |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

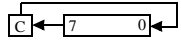
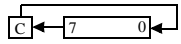
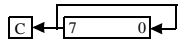
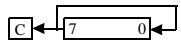
| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|---------------------------------------------------------------------------------------------------------------------------|--------------|------------|---------------------|----------------------------------|---|----|-----|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| POP pp pp ← (SP) SP ← SP + 2 | qq | | C1–F1 | (no change unless operand is AF) | | | | | |
| | IX/Y | | DD/FD E1 | | | | | | |
| PUSH pp SP ← SP–2 (SP) ← pp | | qq | C5–F5 | – | – | – | – | – | – |
| | | IX/Y | DD/FD E5 | | | | | | |
| RES b,m m ← m and not (2 ^b) | | r | CB 80–BF | – | – | – | – | – | – |
| | | (HL) | CB 86–BE | | | | | | |
| | | (IX/Y ± d) | DD/FD CB d 86–BE | | | | | | |
| RET PC ← (SP) SP ← SP + 2 | | | C9 | – | – | – | – | – | – |
| RET cc if cc {PC ← (SP) SP ← SP + 2} | | | C0–F8 | – | – | – | – | – | – |
| RETI PC ← (SP) SP ← SP + 2 + recognition by Z80 peripherals | | | ED 4D | – | – | – | – | – | – |
| RETN PC ← (SP) SP ← SP + 2 IEF1 ← IEF2 | | | ED 45 | – | – | – | – | – | – |
| RL m  (CF,m) ← rotL(CF,m) | | r | CB 10–17 | * | * | 0 | P | 0 | * |
| | | (HL) | CB 16 | | | | | | |
| | | (IX/Y ± d) | DD/FD CB d 16 | | | | | | |
| RLA  (CF,A) ← rotL(CF,A) | | | 17 | – | – | 0 | – | 0 | * |
| RLC m  (CF,m) ← rotL(m) | | r | CB 00–07 | * | * | 0 | P | 0 | * |
| | | (HL) | CB 06 | | | | | | |
| | | (IX/Y ± d) | DD/FD CB d 06 | | | | | | |
| RLCA  (CF,A) ← rotL(A) | | | 07 | – | – | 0 | – | 0 | * |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

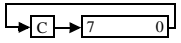
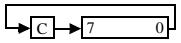
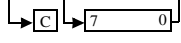
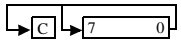
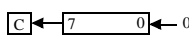
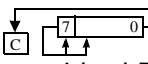
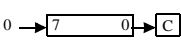
| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | | |
|-------------------------------------------------------------------------------------------------------------------------|----------------------------------|--------|---------------------|----------------|---|----|-----|---|----|---|
| | Dest | Source | | S | Z | HC | P/V | N | CF | |
| RLD tmp ← A[3:0] A[3:0] ← (HL)[7:4] (HL)[7:4] ← (HL)[3:0] (HL)[3:0] ← tmp | | | ED 6F | * | * | 0 | P | 0 | - | |
| RR m  (CF,m) ← rotR(CF,m) | r | | CB 18–1F | * | * | 0 | P | 0 | * | |
| | (HL) | | CB 1E | | | | | | | |
| | (IX/Y ± d) | | DD/FD CB d 1E | | | | | | | |
| RRA  (CF,A) ← rotR(CF,A) | r | | 1F | - | - | 0 | - | 0 | * | |
| | | | | | | | | | | |
| RRC m  (CF,m) ← rotR(m) | r | | CB 08–0F | * | * | 0 | P | 0 | * | |
| | (HL) | | CB 0E | | | | | | | |
| | (IX/Y ± d) | | DD/FD CB d 0E | | | | | | | |
| RRCA  (CF,A) ← rotR(A) | | | 0F | -* | - | 0 | - | 0 | * | |
| | | | | | | | | | | |
| RRD tmp ← (HL)[3:0] (HL)[3:0] ← (HL)[7:4] (HL)[7:4] ← A[3:0] A[3:0] ← tmp | | | ED 67 | * | * | 0 | P | 0 | - | |
| RST p SP ← SP - 2 (SP) ← PC PC ← 0,p note p=0,8,10,18,...38H | | | C7–FF | * | * | 0 | P | 0 | - | |
| | SBC A,s A ← A - s - CF | r | | 98–9F | * | * | * | V | 1 | * |
| | | n | | DE | | | | | | |
| | | (HL) | | 9E | | | | | | |
| | (IX/Y ± d) | | DD/FD 9E | | | | | | | |
| SBC HL,ss HL ← HL - ss - CF | r | | ED 42–72 | * | * | * | V | 1 | * | |
| SCF CF ← 1 | | | 37 | - | - | 0 | - | 0 | 1 | |
| SET b,m m ← m or (2^b) | r | | CB C0–FF | - | - | - | - | - | - | |
| | (HL) | | CB C6–FE | | | | | | | |
| | (IX/Y ± d) | | DD/FD CB d C6–FE | | | | | | | |

TABLE 146. INSTRUCTION SUMMARY (CONTINUED)

| Instruction and Operation | Address Mode | | Op Code(s) (Hex) | Flags Affected | | | | | |
|----------------------------------------------------------------------------------------------------------------------------|--------------|--------|---------------------|----------------|---|----|-----|---|----|
| | Dest | Source | | S | Z | HC | P/V | N | CF |
| SLA m  (CF,m) ← m + m | r | | CB 20–27 | * | * | 0 | P | 0 | * |
| | (HL) | | CB 26 | | | | | | |
| | (IX/Y ± d) | | DD/FD CB d 26 | | | | | | |
| SLP | | | ED 76 | – | – | – | – | – | – |
| SRA m  (m,CF) ← arith_shR(m) | r | | CB 28–2F | * | * | 0 | P | 0 | * |
| | (HL) | | CB 2E | | | | | | |
| | (IX/Y ± d) | | DD/FD CB d 2E | | | | | | |
| SRL m  (m,CF) ← logic_shR(m) | r | | CB 38–3F | 0 | * | 0 | P | 0 | * |
| | (HL) | | CB 3E | | | | | | |
| | (IX/Y ± d) | | DD/FD CB d 3E | | | | | | |
| SUB A,s A ← A – s | r | | 90–97 | * | * | * | V | 1 | * |
| | n | | D6 | | | | | | |
| | (HL) | | 96 | | | | | | |
| | (IX/Y ± d) | | DD/FD 96 | | | | | | |
| TST A,s A AND s | r | | ED 04–3C | * | * | 1 | P | 0 | 0 |
| | n | | ED 64 | | | | | | |
| | (HL) | | ED 34 | | | | | | |
| TSTIO n (0,C) AND n | | | ED 34 | * | * | 1 | P | 0 | 0 |
| XOR A,s A ← A XOR s | r | | A8–AF | * | * | 0 | P | 0 | 0 |
| | n | | EE | | | | | | |
| | (HL) | | AE | | | | | | |
| | (IX/Y ± d) | | DD/FD AE | | | | | | |

OP CODE MAP

TABLE 147. OP CODE MAP (1ST OP CODE)

| | | LOWER NIBBLE (HEX) | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|-----------|------------|------------|------------|-----------|------------|-----------|-----------|-----------|------------|-------------|------------|-------------|------------|---------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| UPPER NIBBLE (HEX) | 0 | NOP | LD BC,nn | LD (BC),A | INC BC | INC B | DEC B | LD B,n | RLCA | EX AF,AF' | ADD HL,BC | LD A,(BC) | DEC BC | INC C | DEC C | LD C,n | RRCA |
| | 1 | DJNZ d | LD DE,nn | LD (DE),A | INC DE | INC D | DEC D | LD D,n | RLA | JR d | ADD HL,DE | LD A,(DE) | DEC DE | INC E | DEC E | LD E,n | RRA |
| | 2 | JR NZ,d | LD HL,nn | LD (nn),HL | INC HL | INC H | DEC H | LD H,n | DAA | JR d | ADD HL,HL | LD (HL),nn | DEC HL | INC L | DEC L | LD L,n | CPL |
| | 3 | JR NC,d | LD SP,nn | LD (nn),A | INC SP | INC (HL) | DEC (HL) | LD (HL),n | SCF | JR C,d | ADD HL,SP | LD A,(nn) | DEC SP | INC A | DEC A | LD A,n | CCF |
| | 4 | LD B,B | LD B,C | LD B,D | LD B,E | LD B,H | LD B,L | LD B,(HL) | LD C,A | LD C,B | LD C,C | LD C,D | LD C,E | LD C,H | LD C,L | LD C,(HL) | LD C,A |
| | 5 | LD D,B | LD D,C | LD D,D | LD D,E | LD D,H | LD D,L | LD D,(HL) | LD D,A | LD D,B | LD D,C | LD D,E | LD D,H | LD D,L | LD D,(HL) | LD D,A | LD D,A |
| | 6 | LD H,B | LD H,C | LD H,D | LD H,E | LD H,H | LD H,L | LD H,(HL) | LD H,A | LD L,B | LD L,C | LD L,D | LD L,E | LD L,H | LD L,L | LD L,(HL) | LD L,A |
| | 7 | LD (HL),B | LD (HL),C | LD (HL),D | LD (HL),E | LD (HL),H | LD (HL),L | HALT | LD (HL),A | LD A,B | LD A,C | LD A,D | LD A,E | LD A,H | LD A,L | LD A,(HL) | LD A,A |
| | 8 | ADD A,B | ADD A,C | ADD A,D | ADD A,E | ADD A,H | ADD A,L | ADD A,(HL) | ADC A,A | ADC A,B | ADC A,C | ADC A,D | ADC A,E | ADC A,H | ADC A,L | ADC A,(HL) | ADC A,A |
| | 9 | SUB A,B | SUB A,C | SUB A,D | SUB A,E | SUB A,H | SUB A,L | SUB A,(HL) | SBC A,A | SBC A,B | SBC A,C | SBC A,D | SBC A,E | SBC A,H | SBC A,L | SBC A,(HL) | SBC A,A |
| | A | AND A,B | AND A,C | AND A,D | AND A,E | AND A,H | AND A,L | AND A,(HL) | AND A,A | XOR A,B | XOR A,C | XOR A,D | XOR A,E | XOR A,H | XOR A,L | XOR A,(HL) | XOR A,A |
| | B | OR A,B | OR A,C | OR A,D | OR A,E | OR A,H | OR A,L | OR A,(HL) | OR A,A | CP A,B | CP A,C | CP A,D | CP A,E | CP A,H | CP A,L | CP A,(HL) | CP A,A |
| | C | RET NZ | POP BC | JP NZ,nn | JP nn | CALL NZ,nn | PUSH BC | ADD A,n | RST 0 | RET Z | RET | JP Z,nn | (Table 148) | CALL Z,nn | CALL nn | ADC A,n | RST 8 |
| | D | RET NZ | POP DE | JP NC,nn | OUT (n),A | CALL NC,nn | PUSH DE | SUB A,n | RST 10H | RET C | EXX | JP C,nn | IN A,(n) | CALL C,nn | (Table 149) | SBC A,n | RST 18H |
| | E | RET PO | POP HL | JP PO,nn | EX (SP),HL | CALL PO,nn | PUSH HL | AND A,n | RST 20 | RET PE | JP (HL) | JP PE,nn | EX DE,HL | CALL PE,nn | (Table 150) | XOR A,n | RST 28H |
| | F | RET P | POP AF | JP P,nn | DI | CALL P,nn | PUSH AF | OR A,n | RST 30H | RET M | LD SP,HL | JP M,nn | EI | CALL M,nn | (Table 151) | CP A,n | RST 38H |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

Notes:

n = 8-bit data
 nn = 16-bit addr or data
 d = signed 8-bit displacement

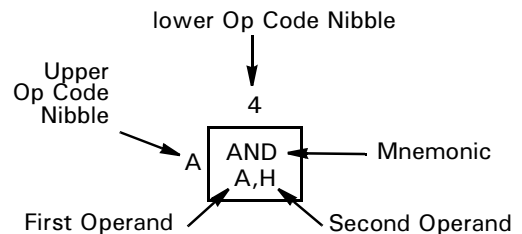


TABLE 148. OP CODE MAP (2ND OP CODE AFTER 0CBH)

| | | LOWER NIBBLE (HEX) | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|---------|---------|---------|---------|---------|------------|----------|---------|---------|---------|---------|---------|---------|------------|---------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| UPPER NIBBLE (HEX) | 0 | RLC B | RLC C | RLC D | RLC E | RLC H | RLC L | RLC (HL) | RLC RRCA | RRC B | RRC C | RRC D | RRC E | RRC H | RRC L | RRC (HL) | RRC A |
| | 1 | RL B | RL C | RL D | RL E | RL H | RL L | RL (HL) | RL A | RR B | RR C | RR D | RR E | RR H | RR L | RR (HL) | RR A |
| | 2 | SLA B | SLA C | SLA D | SLA E | SLA H | SLA L | SLA (HL) | SLA A | SRA B | SRA C | SRA D | SRA E | SRA H | SRA L | SRA (HL) | SRA A |
| | 3 | | | | | | | | | SRL B | SRL C | SRL D | SRL E | SRL H | SRL L | SRL (HL) | SRL A |
| | 4 | BIT 0,B | BIT 0,C | BIT 0,D | BIT 0,E | BIT 0,H | BIT 0,L | BIT 0,(HL) | BIT 0,A | BIT 1,B | BIT 1,C | BIT 1,D | BIT 1,E | BIT 1,H | BIT 1,L | BIT 1,(HL) | BIT 1,A |
| | 5 | BIT 2,B | BIT 2,C | BIT 2,D | BIT 2,E | BIT 2,H | BIT 2,L | BIT 2,(HL) | BIT 2,A | BIT 3,B | BIT 3,C | BIT 3,D | BIT 3,E | BIT 3,H | BIT 3,L | BIT 3,(HL) | BIT 3,A |
| | 6 | BIT 4,B | BIT 4,C | BIT 4,D | BIT 4,E | BIT 4,H | BIT 4,L | BIT 4,(HL) | BIT 4,A | BIT 5,B | BIT 5,C | BIT 5,D | BIT 5,E | BIT 5,H | BIT 5,L | BIT 5,(HL) | BIT 5,A |
| | 7 | BIT 6,B | BIT 6,C | BIT 6,D | BIT 6,E | BIT 6,H | BIT 6,L | BIT 6,(HL) | BIT 6,A | BIT 7,B | BIT 7,C | BIT 7,D | BIT 7,E | BIT 7,H | BIT 7,L | BIT 7,(HL) | BIT 7,A |
| | 8 | RES 0,B | RES 0,C | RES 0,D | RES 0,E | RES 0,H | RES 0,L | RES 0,(HL) | RES 0,A | RES 1,B | RES 1,C | RES 1,D | RES 1,E | RES 1,H | RES 1,L | RES 1,(HL) | RES 1,A |
| | 9 | RES 2,B | RES 2,C | RES 2,D | RES 2,E | RES 2,H | RES 2,L | RES 2,(HL) | RES 2,A | RES 3,B | RES 3,C | RES 3,D | RES 3,E | RES 3,H | RES 3,L | RES 3,(HL) | RES 3,A |
| | A | RES 4,B | RES 4,C | RES 4,D | RES 4,E | RES 4,H | RES 4,L | RES 4,(HL) | RES 4,A | RES 5,B | RES 5,C | RES 5,D | RES 5,E | RES 5,H | RES 5,L | RES 5,(HL) | RES 5,A |
| | B | RES 6,B | RES 6,C | RES 6,D | RES 6,E | RES 6,H | RES 6,L | RES 6,(HL) | RES 6,A | RES 7,B | RES 7,C | RES 7,D | RES 7,E | RES 7,H | RES 7,L | RES 7,(HL) | RES 7,A |
| | C | SET 0,B | SET 0,C | SET 0,D | SET 0,E | SET 0,H | SET 0,L | SET 0,(HL) | SET 0,A | SET 1,B | SET 1,C | SET 1,D | SET 1,E | SET 1,H | SET 1,L | SET 1,(HL) | SET 1,A |
| | D | SET 2,B | SET 2,C | SET 2,D | SET 2,E | SET 2,H | SET 2,L | SET 2,(HL) | SET 2,A | SET 3,B | SET 3,C | SET 3,D | SET 3,E | SET 3,H | SET 3,L | SET 3,(HL) | SET 3,A |
| | E | SET 4,B | SET 4,C | SET 4,D | SET 4,E | SET 4,H | SET 4,L | SET 4,(HL) | SET 4,A | SET 5,B | SET 5,C | SET 5,D | SET 5,E | SET 5,H | SET 5,L | SET 5,(HL) | SET 5,A |
| | F | SET 6,B | SET 6,C | SET 6,D | SET 6,E | SET 6,H | SET 6,L | SET 6,(HL) | SET 6,A | SET 7,B | SET 7,C | SET 7,D | SET 7,E | SET 7,H | SET 7,L | SET 7,(HL) | SET 7,A |

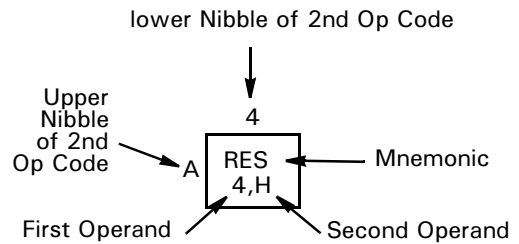


TABLE 149. OP CODE MAP (2ND OP CODE AFTER 0DDH)

| | | LOWER NIBBLE (HEX) | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|---|-----------|-----------|------------|-------------|---|---|--------------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| UPPER NIBBLE (HEX) | 0 | | | | | | | | | | ADD IX,BC | | | | | | |
| | 1 | | | | | | | | | | ADD IX,DE | | | | | | |
| | 2 | | LD IX,nn | LD (nn),IX | INC IX | | | | | | | ADD IX,IX | LD IX,(nn) | DEC IX | | | |
| | 3 | | | | | INC (IX±d) | DEC (IX±d) | LD (IX±d),n | | | | ADD IX,SP | | | | | |
| | 4 | | | | | | | LD B,(IX±d) | | | | | | | | | LD C,(IX±d) |
| | 5 | | | | | | | LD D,(IX±d) | | | | | | | | | LD E,(IX±d) |
| | 6 | | | | | | | LD H,(IX±d) | | | | | | | | | LD L,(IX±d) |
| | 7 | LD (IX±d),B | LD (IX±d),C | LD (IX±d),D | LD (IX±d),E | LD (IX±d),H | LD (IX±d),L | | LD (IX±d),A | | | | | | | | LD A,(IX±d) |
| | 8 | | | | | | | | ADD A,(IX±d) | | | | | | | | ADC A,(IX±d) |
| | 9 | | | | | | | | SUB A,(IX±d) | | | | | | | | SBC A,(IX±d) |
| | A | | | | | | | | AND A,(IX±d) | | | | | | | | XOR A,(IX±d) |
| | B | | | | | | | | OR A,(IX±d) | | | | | | | | CP A,(IX±d) |
| | C | | | | | | | | | | | | | (Table 152) | | | |
| | D | | | | | | | | | | | | | | | | |
| | E | | POP IX | | EX (SP),IX | | PUSH IX | | | | | JP (IX) | | | | | |
| | F | | | | | | | | | | | LD SP,IX | | | | | |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

Notes:
n = 8-bit data
nn = 16-bit addr or data
d = signed 8-bit displacement

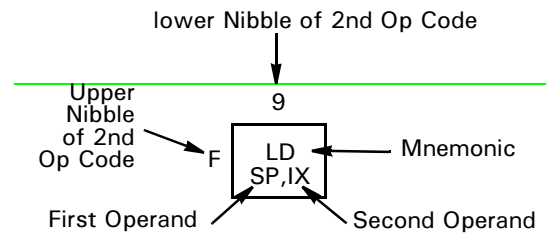


TABLE 150. OP CODE MAP (2ND OP CODE AFTER 0EDH)

| | | LOWER NIBBLE (HEX) | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|------------|-----------|------------|------------|------|------|--------|-----------|------------|-----------|------------|---------|------|------|--------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| UPPER NIBBLE (HEX) | 0 | INO B,(n) | OUTO (n),B | | | TST A,B | | | | INO C,(n) | OUTO (n),C | | | TST A,C | | | |
| | 1 | INO D,(n) | OUTO (n),D | | | TST A,D | | | | INO E,(n) | OUTO (n),E | | | TST A,E | | | |
| | 2 | INO H,(n) | OUTO (n),H | | | TST A,H | | | | INO L,(n) | OUTO (n),L | | | TST A,L | | | |
| | 3 | INO F,(n) | | | | TST A,(HL) | | | | INO A,(n) | OUTO (n),A | | | TST A,A | | | |
| | 4 | IN B,(C) | OUT (C),B | SBC HL,BC | LD (nn),BC | NEG | RETN | IM 0 | LD I,A | IN C,(C) | OUT (C),C | ADC HL,BC | LD BC,(nn) | MLT BC | RETI | | LD R,A |
| | 5 | IN D,(C) | OUT (C),D | SBC HL,DE | LD (nn),DE | | | IM 1 | LD A,I | IN E,(C) | OUT (C),E | ADC HL,DE | LD DE,(nn) | MLT DE | | IM 2 | LD A,R |
| | 6 | IN H,(C) | OUT (C),H | SBC HL,HL | LD (nn),HL | TST A,n | | | RRD | IN L,(C) | OUT (C),L | ADC HL,HL | LD HL,(nn) | MLT HL | | | RLD |
| | 7 | IN F,(C) | | SBC HL,SP | LD (nn),SP | TSTIO n | | SLP | | IN A,(C) | OUT (C),A | ADC HL,SP | LD SP,(nn) | MLT SP | | | |
| | 8 | | | | OTIM | | | | | | | | | OTDM | | | |
| | 9 | | | | OTIMR | | | | | | | | | OTDMR | | | |
| | A | LDI | CPI | INI | OUTI | | | | | LDD | CPD | IND | OUTD | | | | |
| | B | LDIR | CPIR | INIR | OTIR | | | | | LDDR | CPDR | INDR | OTDR | | | | |
| | C | | | | | | | | | | | | | | | | |
| | D | | | | | | | | | | | | | | | | |
| | E | | | | | | | | | | | | | | | | |
| | F | | | | | | | | | | | | | | | | |

Notes:
n = 8-bit data
nn = 16-bit addr or data
d = signed 8-bit displacement

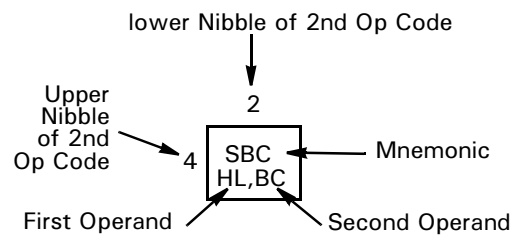


TABLE 151. OP CODE MAP (2ND OP CODE AFTER 0FDH)

| | | LOWER NIBBLE (HEX) | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|-------------|-------------|-------------|-------------|-------------|---------------|-------------|---|-----------|-----------|------------|-------------|---|---|---------------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| UPPER NIBBLE (HEX) | 0 | | | | | | | | | | ADD IY,BC | | | | | | |
| | 1 | | | | | | | | | | ADD IY,DE | | | | | | |
| | 2 | | LD IY,nn | LD (nn),IY | INC IY | | | | | | | ADD IY,IY | LD IY,(nn) | DEC IY | | | |
| | 3 | | | | | INC (IY±d) | DEC (IY±d) | LD (IY±d),n | | | | ADD IY,SP | | | | | |
| | 4 | | | | | | | LD B, (IY±d) | | | | | | | | | LD C, (IY±d) |
| | 5 | | | | | | | LD D, (IY±d) | | | | | | | | | LD E, (IY±d) |
| | 6 | | | | | | | LD H, (IY±d) | | | | | | | | | LD L, (IY±d) |
| | 7 | LD (IY±d),B | LD (IY±d),C | LD (IY±d),D | LD (IY±d),E | LD (IY±d),H | LD (IY±d),L | | LD (IY±d),A | | | | | | | | LD A, (IY±d) |
| | 8 | | | | | | | ADD A, (IY±d) | | | | | | | | | ADC A, (IY±d) |
| | 9 | | | | | | | SUB A, (IY±d) | | | | | | | | | SBC A, (IY±d) |
| | A | | | | | | | AND A, (IY±d) | | | | | | | | | XOR A, (IY±d) |
| | B | | | | | | | OR A, (IY±d) | | | | | | | | | CP A, (IY±d) |
| | C | | | | | | | | | | | | | (Table 153) | | | |
| | D | | | | | | | | | | | | | | | | |
| | E | | POP IY | | EX (SP),IY | | PUSH IY | | | | | JP (IY) | | | | | |
| | F | | | | | | | | | | | LD SP,IY | | | | | |

Notes:
n = 8-bit data
nn = 16-bit addr or data
d = signed 8-bit displacement

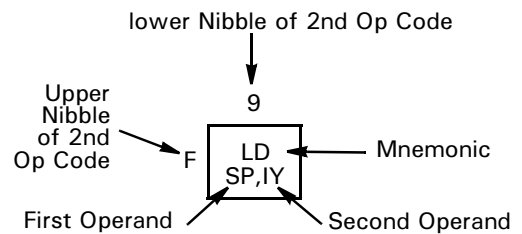


TABLE 152. OP CODE MAP (4TH BYTE, AFTER ODDH, OCBH, AND d)

| | | LOWER NIBBLE (HEX) | | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|---|---|---|---|---|--------------------|---|---|---|---|---|---|---|---|--------------------|--|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
| UPPER NIBBLE (HEX) | 0 | | | | | | | RLC (IX ± d) | | | | | | | | | RRC (IX ± d) | |
| | 1 | | | | | | | RL (IX ± d) | | | | | | | | | RR (IX ± d) | |
| | 2 | | | | | | | SLA (IX ± d) | | | | | | | | | SRA (IX ± d) | |
| | 3 | | | | | | | | | | | | | | | | SRL (IX ± d) | |
| | 4 | | | | | | | BIT 0, (IX ± d) | | | | | | | | | BIT 1, (IX ± d) | |
| | 5 | | | | | | | BIT 2, (IX ± d) | | | | | | | | | BIT 3, (IX ± d) | |
| | 6 | | | | | | | BIT 4, (IX ± d) | | | | | | | | | BIT 5, (IX ± d) | |
| | 7 | | | | | | | BIT 6, (IX ± d) | | | | | | | | | BIT 7, (IX ± d) | |
| | 8 | | | | | | | RES 0, (IX ± d) | | | | | | | | | RES 1, (IX ± d) | |
| | 9 | | | | | | | RES 2, (IX ± d) | | | | | | | | | RES 3, (IX ± d) | |
| | A | | | | | | | RES 4, (IX ± d) | | | | | | | | | RES 5, (IX ± d) | |
| | B | | | | | | | RES 6, (IX ± d) | | | | | | | | | RES 7, (IX ± d) | |
| | C | | | | | | | SET 0, (IX ± d) | | | | | | | | | SET 1, (IX ± d) | |
| | D | | | | | | | SET 2, (IX ± d) | | | | | | | | | SET 3, (IX ± d) | |
| | E | | | | | | | SET 4, (IX ± d) | | | | | | | | | SET 5, (IX ± d) | |
| | F | | | | | | | SET 6, (IX ± d) | | | | | | | | | SET 7, (IX ± d) | |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |

Notes:
d = signed 8-bit displacement

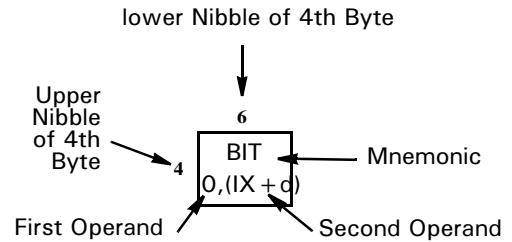
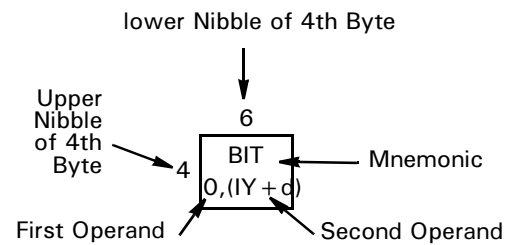


TABLE 153. OP CODE MAP (4TH BYTE, AFTER 0FDH, 0CBH, AND d)

| | | LOWER NIBBLE (HEX) | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|---|---|---|---|---|--------------------|---|---|---|---|---|---|---|--------------------|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| UPPER NIBBLE (HEX) | 0 | | | | | | | RLC (IY ± d) | | | | | | | | RRC (IY ± d) | |
| | 1 | | | | | | | RL (IY ± d) | | | | | | | | RR (IY ± d) | |
| | 2 | | | | | | | SLA (IY ± d) | | | | | | | | SRA (IY ± d) | |
| | 3 | | | | | | | | | | | | | | | SRL (IY ± d) | |
| | 4 | | | | | | | BIT 0, (IY ± d) | | | | | | | | BIT 1, (IY ± d) | |
| | 5 | | | | | | | BIT 2, (IY ± d) | | | | | | | | BIT 3, (IY ± d) | |
| | 6 | | | | | | | BIT 4, (IY ± d) | | | | | | | | BIT 5, (IY ± d) | |
| | 7 | | | | | | | BIT 6, (IY ± d) | | | | | | | | BIT 7, (IY ± d) | |
| | 8 | | | | | | | RES 0, (IY ± d) | | | | | | | | RES 1, (IY ± d) | |
| | 9 | | | | | | | RES 2, (IY ± d) | | | | | | | | RES 3, (IY ± d) | |
| | A | | | | | | | RES 4, (IY ± d) | | | | | | | | RES 5, (IY ± d) | |
| | B | | | | | | | RES 6, (IY ± d) | | | | | | | | RES 7, (IY ± d) | |
| | C | | | | | | | SET 0, (IY ± d) | | | | | | | | SET 1, (IY ± d) | |
| | D | | | | | | | SET 2, (IY ± d) | | | | | | | | SET 3, (IY ± d) | |
| | E | | | | | | | SET 4, (IY ± d) | | | | | | | | SET 5, (IY ± d) | |
| | F | | | | | | | SET 6, (IY ± d) | | | | | | | | SET 7, (IY ± d) | |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

Notes:
d = signed 8-bit displacement



ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This rating is a stress rating only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

TABLE 154. ABSOLUTE MAXIMUM RATINGS

| Parameter | Min | Max | Units | Notes |
|-----------------------------------------------------|------|-------|---------|-------|
| Ambient Temperature under Bias | -40 | + 105 | C | 1 |
| Storage Temperature | -65 | + 150 | C | |
| Voltage on any Pin with Respect to V_{SS} | -0.7 | + 12 | V | 2 |
| Voltage on V_{DD} Pin with Respect to V_{SS} | -0.3 | + 7 | V | |
| Total Power Dissipation | | TBD | mW | |
| Maximum Current out of V_{SS} | | TBD | mA | |
| Maximum Current into V_{DD} | | TBD | mA | |
| Maximum Current on Input and/or Inactive Output Pin | -TBD | + TBD | μ A | |
| Maximum Output Current | -TBD | TBD | mA | |

Notes:

1. Operating temperature is specified in DC Characteristics
2. This applies to all pins except where noted otherwise. Maximum current through a pin is specified below.

STANDARD TEST CONDITIONS

Unless otherwise noted, the DC and AC characteristics in this document are measured under standard test conditions that include the load circuit illustrated in Figure 16. This circuit closely mimics the loading presented by active devices such as memories and peripheral devices.

All voltages are referenced to the V_{SS} pins (ground, 0V). Positive current flows into the referenced pin.

All AC parameters assume a load capacitance of 100 pF. See “Characteristic Curves” on page 178 for the effect of lesser or greater total capacitance on the timing. AC timing measurements are referenced to the High and low voltage thresholds given in the DC specifications, as indicated in Figures 17 through 25.

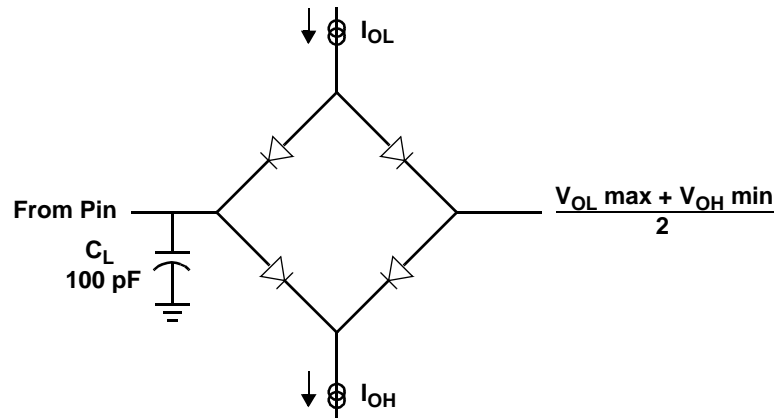


FIGURE 16. TEST CONDITION LOAD CIRCUIT

DC CHARACTERISTICS

Table 155 and Table 156 describe the DC Characteristics of the Z80S183, for temperature ranges $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$ and -40°C to $+85^\circ\text{C}$ respectively.

TABLE 155. DC CHARACTERISTICS, $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|-----------|----------------------------------------------------------------------------------------------------|-------------------------------------|----------------|-----|----------------|---------------|
| V_{IH1} | Input High Voltage (RESET, EXTAL, NMI) | | $V_{DD} - 0.6$ | | $V_{DD} + 0.3$ | V |
| V_{IH2} | Input High Voltage (Except $\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$, CKS) | | 2.0 | | $V_{DD} + 0.3$ | V |
| V_{IH3} | Input High Voltage (CKS) | | 2.4 | | $V_{DD} + 0.3$ | V |
| V_{IL1} | Input Low Voltage ($\overline{\text{RESET}}$, EXTAL, NMI) | | -0.3 | | 0.6 | V |
| V_{IL2} | Input Low Voltage (except $\overline{\text{RESET}}$, EXTAL, NMI) | | -0.3 | | 0.6 | V |
| V_{OH} | Output High Voltage | $I_{OH} = -200 \mu\text{A}$ | 2.4 | | | V |
| | | $I_{OH} = -20 \mu\text{A}$ | $V_{DD} - 1.2$ | | | |
| V_{OL} | Output Low Voltage | $I_{OL} = 2.2 \text{ mA}$ | | | 0.45 | V |
| I_{IL} | Input Leakage (All inputs except XTAL, EXTAL, LFX TAL, LFEXTAL) | $V_{IN} = 0.5$ to $V_{DD} - 0.5$ | | | 1.0 | μA |
| I_{OL} | Output Leakage | $V_{IN} = 0.5$ to $V_{DD} - 0.5$ | | | 1.0 | μA |
| V_{ICR} | Comparator Input Common Mode Voltage Range | | | | | |

TABLE 155. DC CHARACTERISTICS, $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$ (CONTINUED)

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|-----------|---------------------------------------|-----------------|-----|-----|-----|---------------|
| I_{DD} | Supply Current (Normal Operation) | 20 MHz (note 1) | | TBD | TBD | mA |
| | | 33 MHz (note 1) | | TBD | TBD | |
| I_{DD1} | Standby Current (System Stop Mode) | 20 MHz (note 1) | | TBD | TBD | mA |
| | | 33 MHz (note 1) | | TBD | TBD | |
| I_{DD2} | Standby Current (Standby Mode) | | | TBD | TBD | μA |

Notes:
 1. $V_{IH} > V_{DD}-1.0\text{ V}$, $V_{IL} < 0.8\text{ V}$, $V_{DD} = 5.0\text{ V}$, no outputs loaded.

TABLE 156. DC CHARACTERISTICS, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|-----------|----------------------------------------------------------------------------------------------------|-----------------------------------|--------------|-----|--------------|---------------|
| V_{IH1} | Input High Voltage ($\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$) | | $V_{DD}-0.6$ | | $V_{DD}+0.3$ | V |
| V_{IH2} | Input High Voltage (Except $\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$, CKS) | | 2.0 | | $V_{DD}+0.3$ | V |
| V_{IH3} | Input High Voltage (CKS) | | 2.4 | | $V_{DD}+0.3$ | V |
| V_{IL1} | Input Low Voltage ($\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$) | | -0.3 | | 0.6 | V |
| V_{IL2} | Input Low Voltage (except $\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$) | | -0.3 | | 0.6 | V |
| V_{OH} | Output High Voltage | $I_{OH} = -200\ \mu\text{A}$ | 2.4 | | | V |
| | | $I_{OH} = -20\ \mu\text{A}$ | $V_{DD}-1.2$ | | | |
| V_{OL} | Output Low Voltage | $I_{OL} = 2.2\ \text{mA}$ | | | 0.45 | V |
| I_{IL} | Input Leakage (All inputs except XTAL, EXTAL, LFXTAL, LFEXTAL) | $V_{IN} = 0.5$ to $V_{DD}-0.5$ | | | 1.0 | μA |
| I_{OL} | Output Leakage | $V_{IN} = 0.5$ to $V_{DD}-0.5$ | | | 1.0 | μA |
| V_{ICR} | Comparator Input Common Mode Voltage Range | | | | | |
| I_{DD} | Supply Current (Normal Operation) | 20 MHz (note 1) | | TBD | TBD | mA |
| | | 33 MHz (note 1) | | TBD | TBD | |
| I_{DD1} | Standby Current (System Stop Mode) | 20 MHz (note 1) | | TBD | TBD | mA |
| | | 33 MHz (note 1) | | TBD | TBD | |
| I_{DD2} | Standby Current (Standby Mode) | | | TBD | TBD | μA |

Note:
 1. $V_{IH} > V_{DD}-1.0\text{ V}$, $V_{IL} < 0.8\text{ V}$, $V_{DD} = 5.0\text{ V}$, no outputs loaded.

AC CHARACTERISTICS

Tables 157 and 158 give the AC Characteristics of the Z80S183, over temperature ranges $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$ and $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$ respectively, for the parameters described in Figures 17 through 25.

TABLE 157. AC CHARACTERISTICS, $T_A = 0^\circ\text{C}$ TO $+70^\circ\text{C}$, $C_L = 100$ pF

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|--------------------|--------------------|----------------------------------------------------------------------------------|----------------------|-----|----------------------|-------|------------------|
| | | | Min | Max | Min | Max | |
| 1 | f_{OSC} | Crystal Frequency | | 20 | | 33.33 | MHz |
| 2 | t_{EXCYC} | External Clock Cycle Time (EXTAL) | 50 | DC | 30 | DC | ns |
| 3 | t_{EXH} | External Clock High Time (EXTAL) | | 15 | | 10 | ns |
| 4 | t_{EXL} | External Clock Low Time (EXTAL) | | 15 | | 10 | ns |
| 5 | t_{EXr} | External Clock Rise Time (EXTAL) | | 10 | | 5 | ns |
| 6 | t_{EXf} | External Clock Fall Time (EXTAL) | | 10 | | 5 | ns |
| 7 | t_{CHW} | PHI High Time | 20 | | 12 | | |
| 8 | t_{CLW} | PHI Low Time | 20 | | 12 | | |
| 9 | t_{CYC} | PHI Cycle Time | 50 | DC | 30 | DC | ns |
| 10 | t_{RES} | $\overline{\text{RESET}}$ Setup to PHI Fall ² | 10 | | 8 | | ns |
| 11 | t_{REH} | $\overline{\text{RESET}}$ Hold from PHI Fall ² | 5 | | 5 | | ns |
| 12 | t_{Rr} | $\overline{\text{RESET}}$ Fall Time ¹ | | 50 | | 50 | ms |
| 13 | t_{RL} | $\overline{\text{RESET}}$ Low Time | 6 | | 6 | | t_{CYC} |
| 14 | t_{Rf} | $\overline{\text{RESET}}$ Rise Time ¹ | | 50 | | 50 | ms |
| 15 | t_{fr} | Input Fall Time (except EXTAL, $\overline{\text{RESET}}$) ¹ | | 50 | | 50 | ns |
| 16 | t_{fr} | Input Rise Time (except EXTAL, $\overline{\text{RESET}}$) ¹ | | 50 | | 50 | ns |
| 17 | t_{AV} | PHI Rise to Address Valid | | 15 | | 5 | ns |
| Memory Read | | | | | | | |
| 18 | t_{M1L} | PHI Rise to $\overline{\text{M1}}$ Fall | | 15 | | 15 | ns |
| 19 | t_{ASMR} | Address Valid to ($\overline{\text{ROMRD}}$ or $\overline{\text{RAMRD}}$) Fall | $t_{\text{CHW}}-5$ | | $t_{\text{CHW}}-10$ | | ns |
| 20 | t_{MRL} | PHI Fall to ($\overline{\text{ROMRD}}$ or $\overline{\text{RAMRD}}$) Fall | | 15 | | 15 | ns |
| 21 | t_{WTS} | $\overline{\text{WAIT}}$ Setup to PHI Fall ² | 15 | | 15 | | ns |
| 22 | t_{WTH} | $\overline{\text{WAIT}}$ Hold from PHI Fall ² | 5 | | 5 | | ns |
| 23 | t_{MRPL} | ($\overline{\text{ROMRD}}$ or $\overline{\text{RAMRD}}$) Width Low | $2t_{\text{CYC}}-15$ | | $2t_{\text{CYC}}-10$ | | ns |
| 24 | t_{RDS} | Read Data Setup to PHI Rise ¹ | 15 | | 15 | | ns |
| 25 | t_{M1H} | PHI Rise to $\overline{\text{M1}}$ Rise | | 15 | | 15 | ns |
| 26 | t_{RWH} | PHI Fall to ($\overline{\text{ROMRD}}$ or $\overline{\text{RAMRD}}$) Rise | | 15 | | 15 | ns |

TABLE 157. AC CHARACTERISTICS, $T_A = 0^\circ\text{C}$ TO $+70^\circ\text{C}$, $C_L = 100\text{ pF}$ (CONTINUED)

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|---------------------|-------------|------------------------------------------------------------------------------------|-----------------------------|-----|-----------------------------|-----|-------|
| | | | Min | Max | Min | Max | |
| 27 | t_{RDH} | Read Data Hold from (\overline{ROMRD} or \overline{RAMRD}) Rise ¹ | 0 | | 0 | | ns |
| 28 | t_{MRHAC} | (\overline{ROMRD} or \overline{RAMRD}) Rise to Address Change | t_{CLW-10} | | t_{CLW-10} | | ns |
| Memory Write | | | | | | | |
| 29 | t_{ASMW} | Address Valid to (\overline{ROMWR} or \overline{RAMWR}) Fall | t_{CYC-10} | | t_{CYC-10} | | |
| 30 | t_{WDV} | PHI Fall to Write Data Valid | | 20 | | 20 | ns |
| 31 | t_{MWDS} | Write Data Valid to (\overline{ROMWR} or \overline{RAMWR}) Fall | t_{CLW-15} | | t_{CLW-15} | | ns |
| 32 | t_{MRL} | PHI Rise to (\overline{ROMWR} or \overline{RAMWR}) Fall | | 15 | | 15 | ns |
| 33 | t_{MWPL} | (\overline{ROMWR} or \overline{RAMWR}) Width Low | t_{CYC+} t_{CHW-10} | | t_{CYC+} t_{CHW-10} | | ns |
| 34 | t_{WRH} | PHI Fall to (\overline{ROMWR} or \overline{RAMWR}) Rise | | 15 | | 15 | ns |
| 35 | t_{WDH} | (\overline{ROMWR} or \overline{RAMWR}) Rise to Write Data Change | t_{CLW-20} | | t_{CLW-12} | | ns |
| 36 | t_{WDZ} | PHI Rise to Write Data Float | | 10 | | 10 | ns |
| 37 | t_{MWHAC} | (\overline{ROMWR} or \overline{RAMWR}) Rise to Address Change | t_{CLW-20} | | t_{CLW-12} | | ns |
| I/O Read | | | | | | | |
| 38 | t_{ASIR} | Address Valid to \overline{IORD} Fall ($\overline{IOC} = 1$) | t_{CHW-10} | | t_{CHW-10} | | |
| 39 | | Address Valid to \overline{IORD} Fall ($\overline{IOC} = 0$) | t_{CYC-10} | | t_{CYC-10} | | |
| 40 | t_{IRL} | PHI Fall to \overline{IORD} Fall ($\overline{IOC} = 1$) | | 15 | | 15 | ns |
| 41 | | PHI Rise to \overline{IORD} Fall ($\overline{IOC} = 0$) | | 15 | | 15 | ns |
| 42 | t_{IRPL} | \overline{IORD} Width Low ($\overline{IOC} = 1$) | $3t_{CYC-15}$ | | $3t_{CYC-10}$ | | |
| 43 | | \overline{IORD} Width Low ($\overline{IOC} = 0$) | $2t_{CYC+}$ t_{CHW-15} | | $2t_{CYC+}$ t_{CHW-10} | | |
| 44 | t_{RWH} | PHI Fall to \overline{IORD} Rise | | 15 | | 15 | ns |
| 45 | t_{RDH} | Read Data Hold from \overline{IORD} Rise ¹ | 0 | | 0 | | ns |
| 46 | t_{IRHAC} | \overline{IORD} Rise to Address Change | t_{CLW-20} | | t_{CLW-12} | | ns |
| I/O Write | | | | | | | |
| 47 | t_{ASOW} | Address Valid to \overline{IOWR} Fall | t_{CYC-10} | | t_{CYC-10} | | |
| 48 | t_{OWDS} | Write Data Valid to \overline{IOWR} Fall | t_{CLW-10} | | t_{CLW-10} | | |
| 49 | t_{WRL} | PHI Rise to \overline{IOWR} Fall | | 15 | | 15 | ns |
| 50 | t_{OWPL} | \overline{IOWR} Width Low | $2t_{CYC+}$ t_{CHW-15} | | $2t_{CYC+}$ t_{CHW-10} | | ns |
| 51 | t_{WRH} | PHI Fall to \overline{IOWR} Rise | | 15 | | 15 | ns |

TABLE 157. AC CHARACTERISTICS, $T_A = 0^\circ\text{C}$ TO $+70^\circ\text{C}$, $C_L = 100\text{ pF}$ (CONTINUED)

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|----------------------------|-------------|--------------------------------------------------------------------|-------------------------|-----|-------------------------|-----|-----------|
| | | | Min | Max | Min | Max | |
| 52 | t_{WDH} | \overline{IOWR} Rise to Write Data Change | $t_{CLW}-20$ | | $t_{CLW}-12$ | | ns |
| 53 | t_{OWHAC} | \overline{IOWR} Rise to Address Change | $t_{CLW}-20$ | | $t_{CLW}-12$ | | ns |
| Bus Exchange Timing | | | | | | | |
| 54 | t_{BRS} | \overline{BUSREQ} Setup to PHI Fall ² | 10 | | 10 | | ns |
| 55 | t_{BRH} | \overline{BUSREQ} Hold after PHI Fall ² | 10 | | 10 | | ns |
| 56 | t_{RLAL} | \overline{BUSREQ} Low to \overline{BUSACK} Low | t_{CYC} $+t_{CLW}$ | | t_{CYC} $+t_{CLW}$ | | |
| 57 | t_{BAL} | PHI Rise to \overline{BUSACK} Fall | | 15 | | 15 | ns |
| 58 | t_{BZ} | PHI Rise to Bus Float | | 10 | | 10 | ns |
| 59 | t_{RHAH} | \overline{BUSREQ} High to \overline{BUSACK} High | 1 | 2 | 1 | 2 | t_{CYC} |
| 60 | t_{BAH} | PHI Fall to \overline{BUSACK} Rise | | 15 | | 15 | ns |
| 61 | t_{BV} | PHI Rise to Bus Valid | | 30 | | 20 | ns |
| Interrupt Timing | | | | | | | |
| 62 | t_{INTS} | $\overline{INT0}$ Setup to PHI Fall ² | 15 | | 15 | | ns |
| 63 | t_{INTH} | $\overline{INT0}$ Hold after PHI Fall ² | 5 | | 5 | | ns |
| 64 | t_{INTS} | $\overline{INT1-2}$ Setup to PHI Fall (level sense) ² | 15 | | 15 | | ns |
| 65 | t_{INTH} | $\overline{INT1-2}$ Hold after PHI Fall (level sense) ² | 5 | | 5 | | ns |
| 66 | t_{INTL} | $\overline{INT1-2}$ Pulse Width (edge sense) ¹ | 15 | | 15 | | ns |
| 67 | t_{NMIL} | \overline{NMI} Width low ¹ | 35 | | 25 | | ns |
| DMA Timing | | | | | | | |
| 68 | t_{DROS} | $\overline{DREQ0-1}$ Setup to PHI Rise (level sense) ² | 15 | | 15 | | ns |
| 69 | t_{DROH} | $\overline{DREQ0-1}$ Hold from PHI Rise (level sense) ² | 5 | | 5 | | ns |
| 70 | t_{DRQL} | $\overline{DREQ0-1}$ Low Width (edge sense) ¹ | t_{CYC} | | t_{CYC} | | |
| 71 | t_{DROH} | $\overline{DREQ0-1}$ High Width (edge sense) ¹ | t_{CYC} | | t_{CYC} | | |

TABLE 157. AC CHARACTERISTICS, $T_A = 0^\circ\text{C}$ TO $+70^\circ\text{C}$, $C_L = 100\text{ pF}$ (CONTINUED)

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|---------------------|-------------------|--------------------------------------|--------|-----|--------|-----|-------|
| | | | Min | Max | Min | Max | |
| CSI/O Timing | | | | | | | |
| 72 | t_{STDI} | CKS Low to TXS Valid | | 10 | | 10 | ns |
| 73 | t_{SRSI} | RXS Valid to CKS high ¹ | 15 | | 15 | | ns |
| 74 | t_{SRHI} | CKS High to RXS Invalid ¹ | 5 | | 5 | | ns |

Notes:

1. These timing requirements must be met to assure correct device operation.
2. These Setup and Hold times must be met to guarantee recognition at the clock edge in question. If they are not met with respect to a clock edge, the Z80S183 may or may not recognize the new state of the signal at that edge. If it does not, and the signal remains in the same state, the Z80S183 recognizes the new state 1 clock period later.

TABLE 158. AC CHARACTERISTICS, $T_A = -40^\circ\text{C}$ TO $+85^\circ\text{C}$, $C_L = 100\text{ pF}$

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|----|--------------------|-------------------------------------------------------------------------|--------|-----|--------|-------|------------------|
| | | | Min | Max | Min | Max | |
| 1 | f_{OSC} | Crystal Frequency | | 20 | | 33.33 | MHz |
| 2 | t_{EXCYC} | External Clock Cycle Time (EXTAL) | 50 | DC | 30 | DC | ns |
| 3 | t_{CHW} | External Clock High Time (EXTAL) | | 15 | | 10 | ns |
| 4 | t_{CLW} | External Clock Low Time (EXTAL) | | 15 | | 10 | ns |
| 5 | t_{EXr} | External Clock Rise Time (EXTAL) | | 10 | | 5 | ns |
| 6 | t_{EXf} | External Clock Fall Time (EXTAL) | | 10 | | 5 | ns |
| 7 | t_{CHW} | PHI High Time | 20 | | 12 | | ns |
| 8 | t_{CLW} | PHI Low Time | 20 | | 12 | | ns |
| 9 | t_{CYC} | PHI Cycle Time | 50 | DC | 30 | DC | ns |
| 10 | t_{RES} | $\overline{\text{RESET}}$ Setup to PHI Fall ² | 10 | | 8 | | ns |
| 11 | t_{REH} | $\overline{\text{RESET}}$ Hold from PHI Fall ² | 5 | | 5 | | ns |
| 12 | t_{Rr} | $\overline{\text{RESET}}$ Fall Time ¹ | | 50 | | 50 | ms |
| 13 | t_{RL} | $\overline{\text{RESET}}$ Low Time | 6 | | 6 | | t_{CYC} |
| 14 | t_{Rf} | $\overline{\text{RESET}}$ Rise Time ¹ | | 50 | | 50 | ms |
| 15 | t_{ir} | Input Fall Time (except EXTAL, $\overline{\text{RESET}}$) ¹ | | 50 | | 50 | ns |
| 16 | t_{if} | Input Rise Time (except EXTAL, $\overline{\text{RESET}}$) ¹ | | 50 | | 50 | ns |
| 17 | t_{AV} | PHI Rise to Address Valid | | 15 | | 5 | ns |

Memory Read

TABLE 158. AC CHARACTERISTICS, $T_A = -40^\circ\text{C}$ TO $+85^\circ\text{C}$, $C_L = 100\text{ pF}$ (CONTINUED)

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|---------------------|-------------|----------------------------------------------------------------------------------|----------------------------|-----|----------------------------|-----|-------|
| | | | Min | Max | Min | Max | |
| 18 | t_{M1L} | PHI Rise to $\overline{M1}$ Fall | | 15 | | 15 | ns |
| 19 | t_{ASMR} | Address Valid to $(\overline{ROMRD}$ or $\overline{RAMRD})$ Fall | $t_{CHW}-5$ | | $t_{CHW}-10$ | | ns |
| 20 | t_{MRL} | PHI Fall to $(\overline{ROMRD}$ or $\overline{RAMRD})$ Fall | | 15 | | 15 | ns |
| 21 | t_{WTS} | \overline{WAIT} Setup to PHI Fall ² | 15 | | 15 | | ns |
| 22 | t_{WTH} | \overline{WAIT} Hold from PHI Fall ² | 5 | | 5 | | ns |
| 23 | t_{MRPL} | $(\overline{ROMRD}$ or $\overline{RAMRD})$ Width Low | $2t_{CYC}-15$ | | $2t_{CYC}-10$ | | ns |
| 24 | t_{RDS} | Read Data Setup to PHI Rise ¹ | 15 | | 15 | | ns |
| 25 | t_{M1H} | PHI Rise to $\overline{M1}$ Rise | | 15 | | 15 | ns |
| 26 | t_{RWH} | PHI Fall to $(\overline{ROMRD}$ or $\overline{RAMRD})$ Rise | | 15 | | 15 | ns |
| 27 | t_{RDH} | Read Data Hold from $(\overline{ROMRD}$ or $\overline{RAMRD})$ Rise ¹ | 0 | | 0 | | ns |
| 28 | t_{MRHAC} | $(\overline{ROMRD}$ or $\overline{RAMRD})$ Rise to Address Change | $t_{CLW}-10$ | | $t_{CLW}-10$ | | ns |
| Memory Write | | | | | | | |
| 29 | t_{ASMW} | Address Valid to $(\overline{ROMWR}$ or $\overline{RAMWR})$ Fall | $t_{CYC}-10$ | | $t_{CYC}-10$ | | ns |
| 30 | t_{WDV} | PHI Fall to Write Data Valid | | 20 | | 20 | ns |
| 31 | t_{MWDS} | Write Data Valid to $(\overline{ROMWR}$ or $\overline{RAMWR})$ Fall | $t_{CLW}-15$ | | $t_{CLW}-15$ | | ns |
| 32 | t_{MRL} | PHI Rise to $(\overline{ROMWR}$ or $\overline{RAMWR})$ Fall | | 15 | | 15 | ns |
| 33 | t_{MWPL} | $(\overline{ROMWR}$ or $\overline{RAMWR})$ Width Low | $t_{CYC}+$ $t_{CHW}-10$ | | $t_{CYC}+$ $t_{CHW}-10$ | | ns |
| 34 | t_{WRH} | PHI Fall to $(\overline{ROMWR}$ or $\overline{RAMWR})$ Rise | | 15 | | 15 | ns |
| 35 | t_{WDH} | $(\overline{ROMWR}$ or $\overline{RAMWR})$ Rise to Write Data Change | $t_{CLW}-20$ | | $t_{CLW}-12$ | | ns |
| 36 | t_{WDZ} | PHI Rise to Write Data Float | | 10 | | 10 | ns |
| 37 | t_{MWHAC} | $(\overline{ROMWR}$ or $\overline{RAMWR})$ Rise to Address Change | $t_{CLW}-20$ | | $t_{CLW}-12$ | | ns |
| I/O Read | | | | | | | |
| 38 | t_{ASIR} | Address Valid to \overline{IOR} Fall ($\overline{IOC} = 1$) | $t_{CHW}-10$ | | $t_{CHW}-10$ | | ns |
| 39 | | Address Valid to \overline{IOR} Fall ($\overline{IOC} = 0$) | $t_{CYC}-10$ | | $t_{CYC}-10$ | | ns |
| 40 | t_{IRL} | PHI Fall to \overline{IOR} Fall ($\overline{IOC} = 1$) | | 15 | | 15 | ns |
| 41 | | PHI Rise to \overline{IOR} Fall ($\overline{IOC} = 0$) | | 15 | | 15 | ns |

TABLE 158. AC CHARACTERISTICS, $T_A = -40^\circ\text{C}$ TO $+85^\circ\text{C}$, $C_L = 100\text{ pF}$ (CONTINUED)

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|----------------------------|-------------|--------------------------------------------------------------------|---------------------------|-----|---------------------------|-----|-----------|
| | | | Min | Max | Min | Max | |
| 42 | t_{IRPL} | \overline{IORD} Width Low ($\overline{IOC} = 1$) | $3t_{CYC} - 15$ | | $3t_{CYC} - 10$ | | |
| 43 | | \overline{IORD} Width Low ($\overline{IOC} = 0$) | $2t_{CYC} + t_{CHW} - 15$ | | $2t_{CYC} + t_{CHW} - 10$ | | |
| 44 | t_{RWH} | PHI Fall to \overline{IORD} Rise | | 15 | | 15 | ns |
| 45 | t_{RDH} | Read Data Hold from \overline{IORD} Rise ¹ | 0 | | 0 | | ns |
| 46 | t_{IRHAC} | \overline{IORD} Rise to Address Change | $t_{CLW} - 20$ | | $t_{CLW} - 12$ | | ns |
| I/O Write | | | | | | | |
| 47 | t_{ASOW} | Address Valid to \overline{IOWR} Fall | $t_{CYC} - 10$ | | $t_{CYC} - 10$ | | |
| 48 | t_{OWDS} | Write Data Valid to \overline{IOWR} Fall | $t_{CLW} - 10$ | | $t_{CLW} - 10$ | | |
| 49 | t_{WRL} | PHI Rise to \overline{IOWR} Fall | | 15 | | 15 | ns |
| 50 | t_{OWPL} | \overline{IOWR} Width Low | $2t_{CYC} + t_{CHW} - 15$ | | $2t_{CYC} + t_{CHW} - 10$ | | ns |
| 51 | t_{WRH} | PHI Fall to \overline{IOWR} Rise | | 15 | | 15 | ns |
| 52 | t_{WDH} | \overline{IOWR} Rise to Write Data Change | $t_{CLW} - 20$ | | $t_{CLW} - 12$ | | ns |
| 53 | t_{OWHAC} | \overline{IOWR} Rise to Address Change | $t_{CLW} - 20$ | | $t_{CLW} - 12$ | | ns |
| Bus Exchange Timing | | | | | | | |
| 54 | t_{BRS} | \overline{BUSREQ} Setup to PHI Fall ² | 10 | | 10 | | ns |
| 55 | t_{BRH} | \overline{BUSREQ} Hold after PHI Fall ² | 10 | | 10 | | ns |
| 56 | t_{RLAL} | \overline{BUSREQ} Low to \overline{BUSACK} Low | $t_{CYC} + t_{CLW}$ | | $t_{CYC} + t_{CLW}$ | | |
| 57 | t_{BAL} | PHI Rise to \overline{BUSACK} Fall | | 15 | | 15 | ns |
| 58 | t_{BZ} | PHI Rise to Bus Float | | 10 | | 10 | ns |
| 59 | t_{RHAH} | \overline{BUSREQ} High to \overline{BUSACK} High | 1 | 2 | 1 | 2 | t_{CYC} |
| 60 | t_{BAH} | PHI Fall to \overline{BUSACK} Rise | | 15 | | 15 | ns |
| 61 | t_{BV} | PHI Rise to Bus Valid | | 30 | | 20 | ns |
| Interrupt Timing | | | | | | | |
| 62 | t_{INTS} | $\overline{INT0}$ Setup to PHI Fall ² | 15 | | 15 | | ns |
| 63 | t_{INTH} | $\overline{INT0}$ Hold after PHI Fall ² | 5 | | 5 | | ns |
| 64 | t_{INTS} | $\overline{INT1-2}$ Setup to PHI Fall (level sense) ² | 15 | | 15 | | ns |
| 65 | t_{INTH} | $\overline{INT1-2}$ Hold after PHI Fall (level sense) ² | 5 | | 5 | | ns |
| 66 | t_{INTL} | $\overline{INT1-2}$ Pulse Width (edge sense) ¹ | 15 | | 15 | | ns |

TABLE 158. AC CHARACTERISTICS, $T_A = -40^\circ\text{C}$ TO $+85^\circ\text{C}$, $C_L = 100\text{ pF}$ (CONTINUED)

| No | Symbol | Parameter | 20 MHz | | 33 MHz | | Units |
|---------------------|-------------------|---------------------------------------------------------------------------|------------------|-----|------------------|-----|-------|
| | | | Min | Max | Min | Max | |
| 67 | t_{NMIL} | NMI Width low ¹ | 35 | | 25 | | ns |
| DMA Timing | | | | | | | |
| 68 | t_{DROS} | $\overline{\text{DREQ0-1}}$ Setup to PHI Rise (level sense) ² | 15 | | 15 | | ns |
| 69 | t_{DROH} | $\overline{\text{DREQ0-1}}$ Hold from PHI Rise (level sense) ² | 5 | | 5 | | ns |
| 70 | t_{DRQL} | $\overline{\text{DREQ0-1}}$ Low Width (edge sense) ¹ | t_{CYC} | | t_{CYC} | | |
| 71 | t_{DRQH} | $\overline{\text{DREQ0-1}}$ High Width (edge sense) ¹ | t_{CYC} | | t_{CYC} | | |
| CSI/O Timing | | | | | | | |
| 72 | t_{STDI} | CKS Low to TXS Valid | | 10 | | 10 | ns |
| 73 | t_{SRSI} | RXS Valid to CKS high ¹ | 15 | | 15 | | ns |
| 74 | t_{SRHI} | CKS High to RXS Invalid ¹ | 5 | | 5 | | ns |

Notes:

1. These timing requirements must be met to assure correct device operation.
2. These Setup and Hold times must be met to guarantee recognition at the clock edge in question. If they are not met with respect to a clock edge, the Z80S183 may or may not recognize the new state of the signal at that edge. If it does not, and the signal remains in the same state, the Z80S183 recognizes the new state 1 one clock period later.

CAPACITANCE

The capacitance of each pin on the Z80S183 depends on whether the pin is an input, output, or both. The total capacitance associated with outputs affects their AC characteristics (switching time) as described in “Characteristic Curves” on page 178.

| | |
|--------|-------|
| Input | 5 pF |
| Output | 10 pF |
| I/O | 15 pF |

TIMING DIAGRAMS

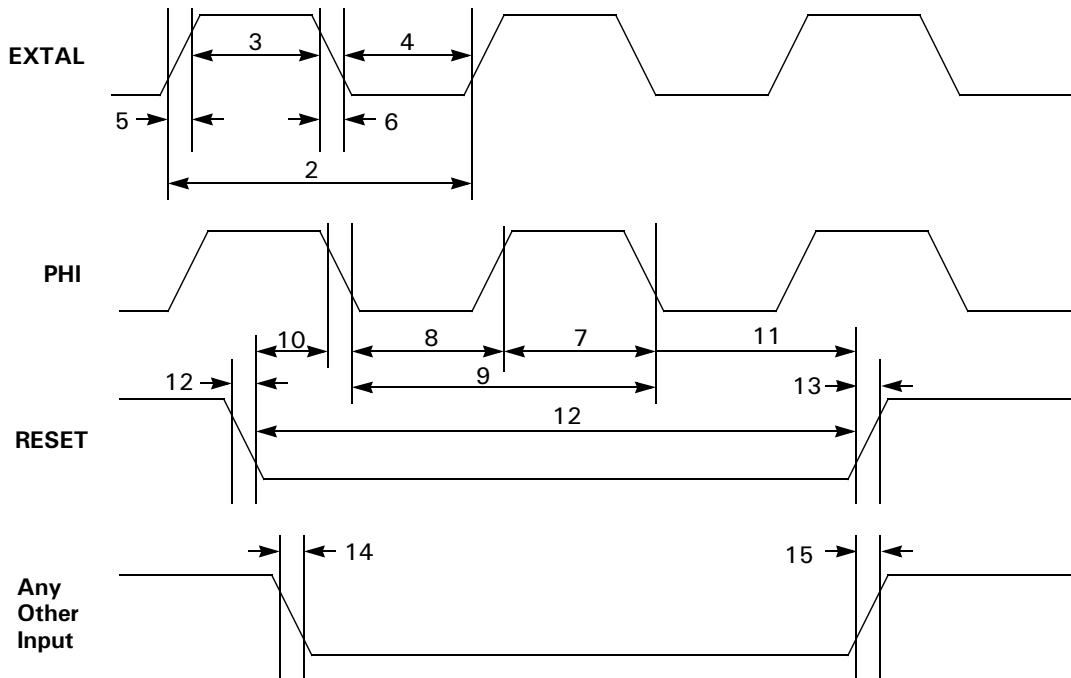


FIGURE 17. BASIC TIMING

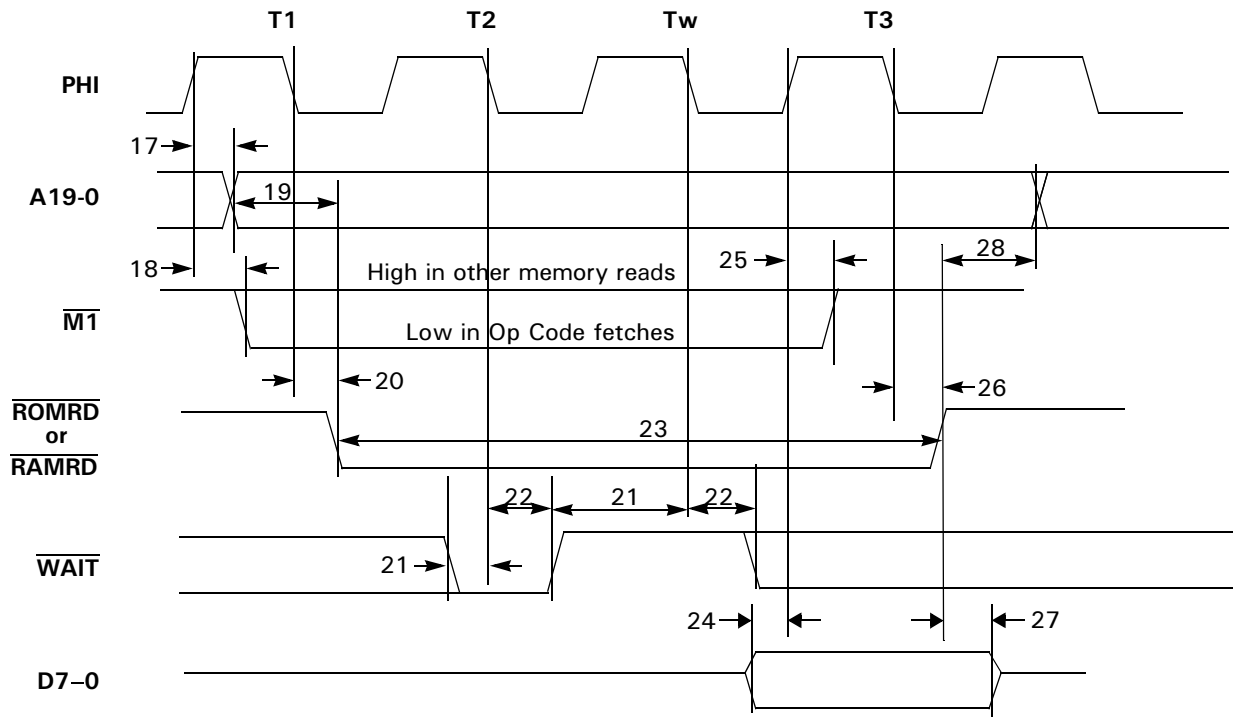


FIGURE 18. MEMORY READ TIMING (ONE WAIT STATE)

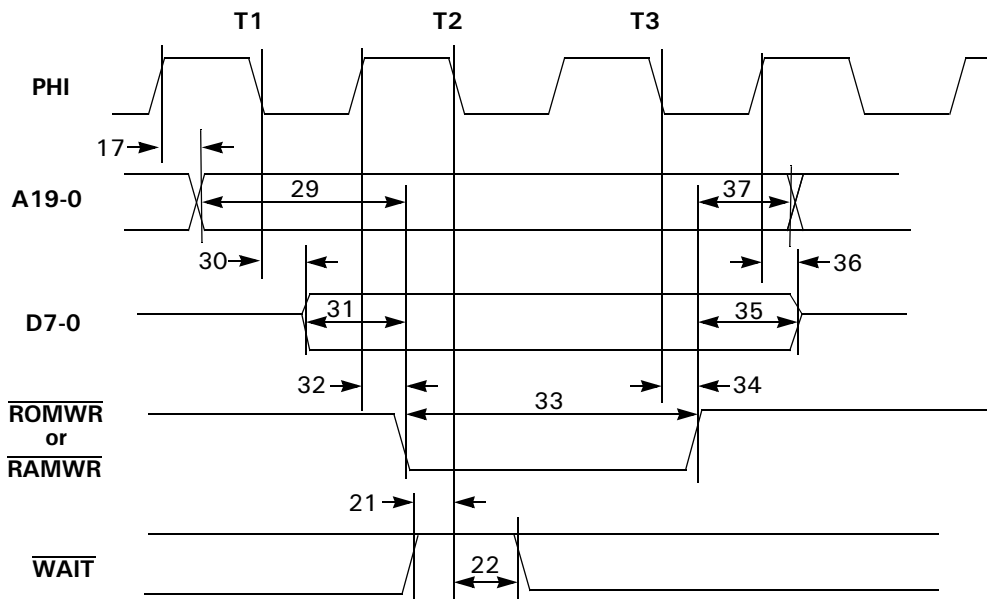


FIGURE 19. MEMORY WRITE TIMING (NO WAIT STATES)

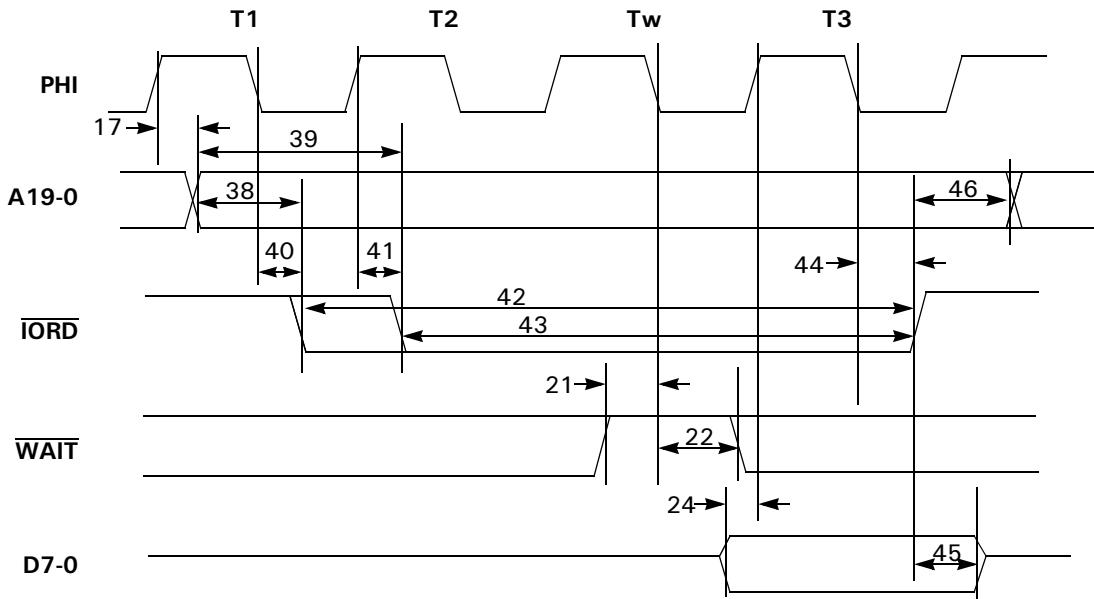


FIGURE 20. I/O READ TIMING (AUTO WAIT STATE)

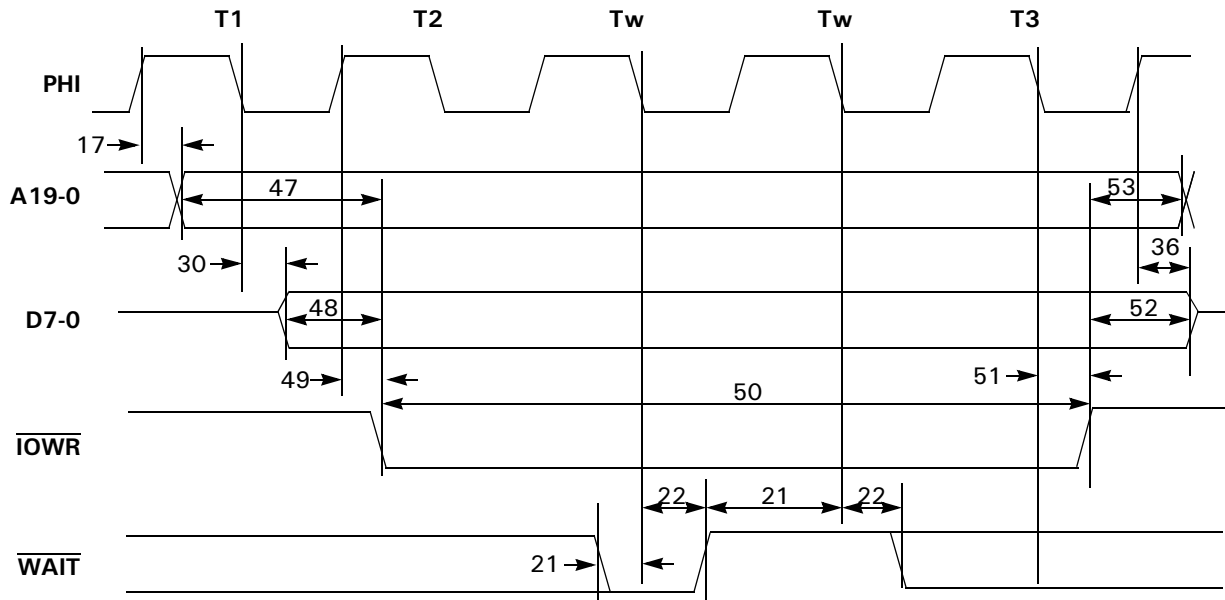


FIGURE 21. I/O WRITE TIMING (AUTO + ONE WAIT STATE)

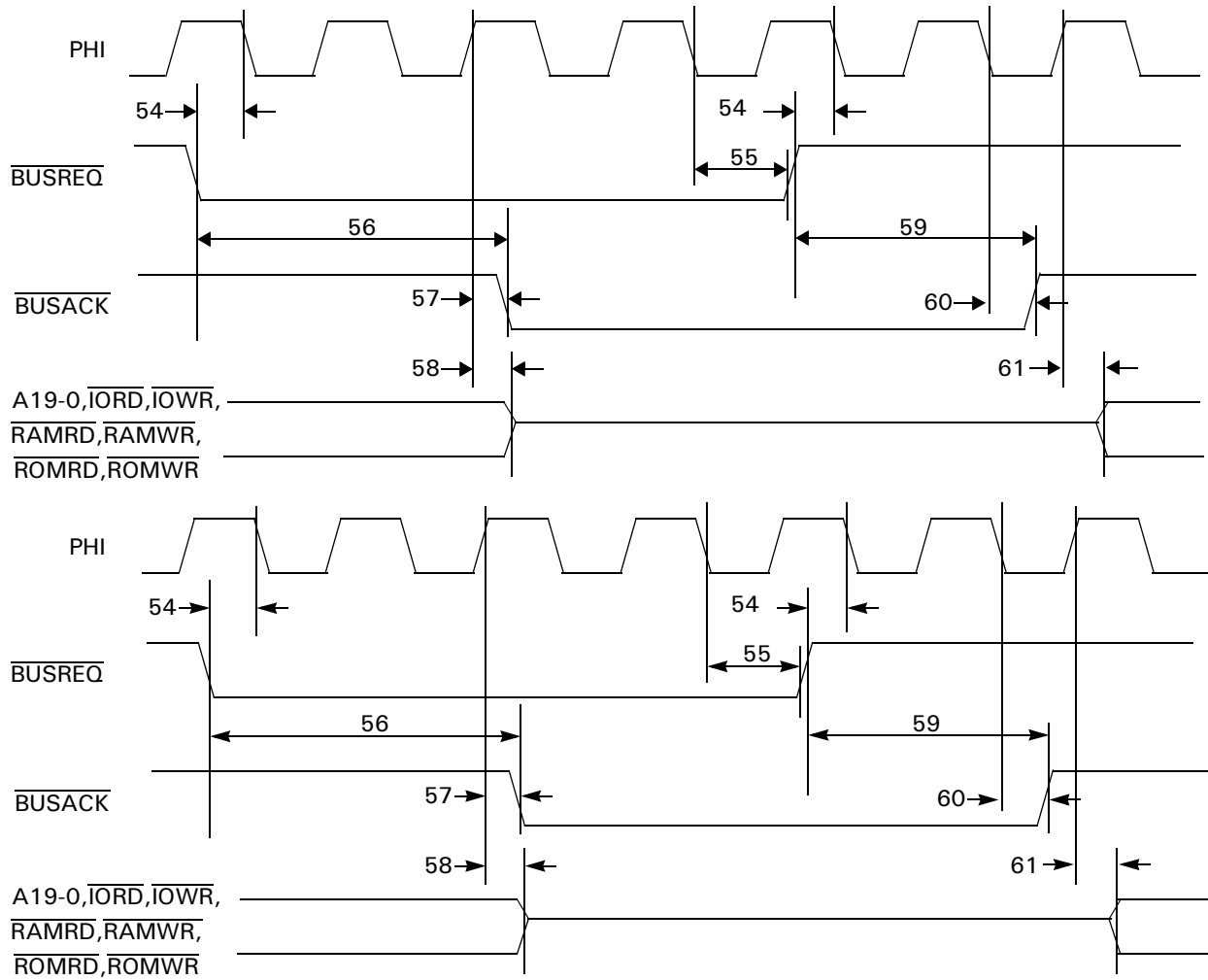


FIGURE 22. BUS EXCHANGE TIMING

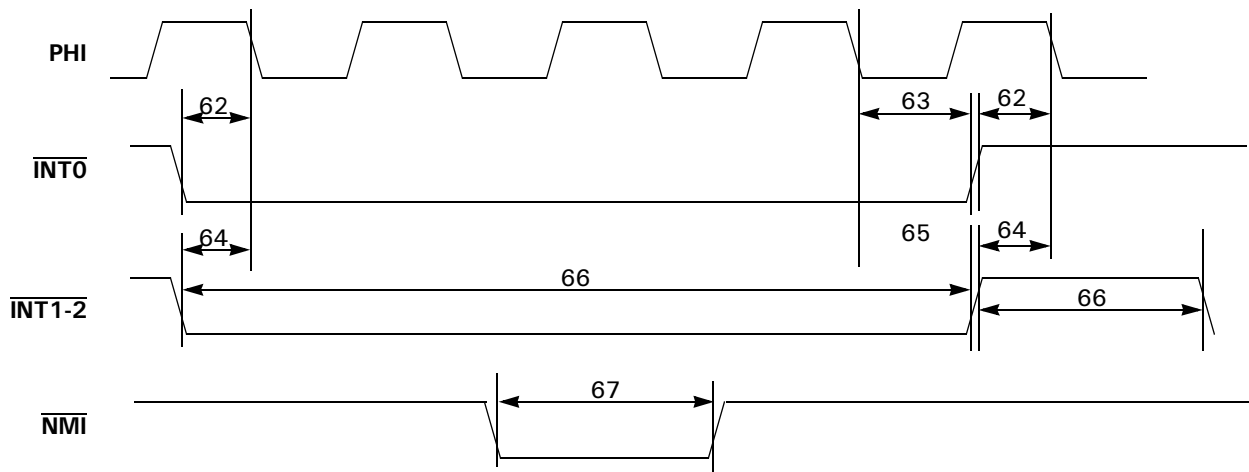


FIGURE 23. INTERRUPT TIMING

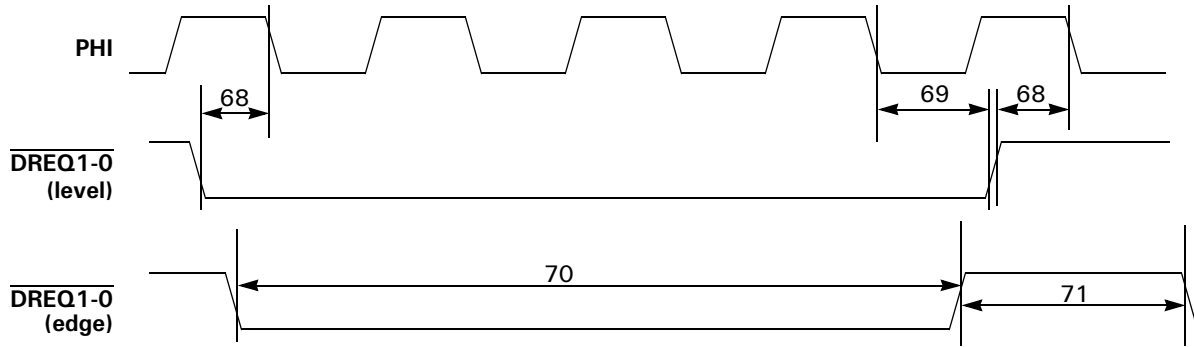


FIGURE 24. DMA TIMING

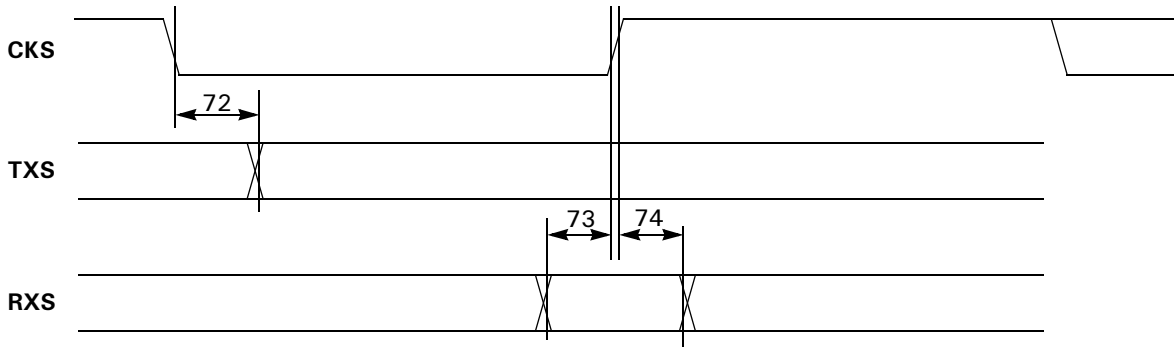


FIGURE 25. CSI/O TIMING

CHARACTERISTIC CURVES

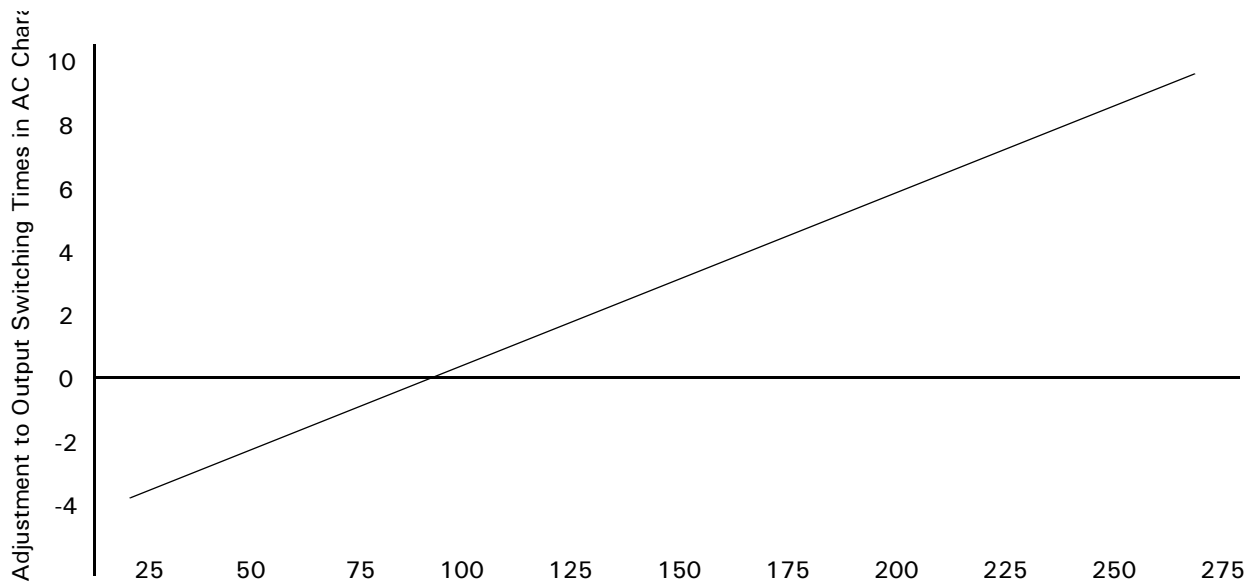


FIGURE 26. CAPACITIVE LOAD C_L VS. SWITCHING TIME

SYSTEM DESIGN CONSIDERATIONSD

ERRATA

ZiLOG expects Revision BA of the Z80S183 to fix the following problems. The errata apply to revision AB of the Z80S183, which is identified by a 0 in the Device Revision register, I/O address 003DH.

Schmitt Triggers

The $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, ZCL and ZDA pins are specified to include Schmitt triggers, but these were not included on Rev AB. The 50 mS max rise and fall time specified for $\overline{\text{RESET}}$ thus becomes much shorter, 50 nS or less. Since $\overline{\text{RESET}}$ is an open-drain output, perhaps the best workaround is to simply connect a pullup resistor, and not drive $\overline{\text{RESET}}$ externally.

No RTC Alarm

The Alarm function of the Real Time Clock is not operative.

PWRSWTCH Negative Logic

The PWRSWTCH pin is specified as positive logic, but in Rev AB is implemented as negative logic (1 is Low, 0 is High).

Output Control Register Reset

The OCR is specified to reset to 0xx0 0000, but instead resets to 0xx1 1000.

No Reset From OPMOD1

A rising edge on OPMOD1 does not trigger a Power On Reset sequence.

PIOS Port Initialization

The Programmable I/O Sequencer enables Port C outputs when it is enabled. Software needs to define the state of Port C outputs before enabling the PIOS, by writing to the Port C Data Register.

PRT Race Condition

If software writes to the PRT Reload registers while the PRT is operating, and a write coincides with a window near the end of the PRT's countdown cycle, the down counter may end up reloaded with the wrong value.

50/60 Hz RTC Clock Taken From Wrong Pin

The RTC specs read that if bit 4 of the RTC Control/Status Register is 1, the RTC takes a 50 or 60 Hz clock from the PC0 pin. Rev AB makes the frequency adjustment when this bit is 1, but takes the clock from LFEXTAL, as when the bit is 0.

Register Write Enable State Indeterminate After Reset

The Register Write Enable state that is readable as bit 0 of the Watch-Dog Timer Master register is specified to reset to 1. However, its value is indeterminate after power up and unchanged by other Resets. Reset-initialization software writes a 0BH to the Watch-Dog Timer Command Register to set this bit, before writing any of the System Configuration, Power Control, Port Data Direction, or Real Time Clock registers.

APPLICATION NOTES/DEVELOPMENT TOOLS

ZILOG DEBUG INTERFACE

The Z80S183 includes this serial interface to allow ZiLOG and 3rd-party development systems and emulators to control and monitor the processor and other on-chip resources, during application development and debugging. This 2-wire interface is intended to be a standard feature of ZiLOG processors, obviating the need for expensive and cumbersome pods and clip-on emulation equipment.

To use the ZiLOG Developer Studio (ZDS) or equivalent equipment with an application or target board, include a standard right angle, 0.1 in spaced, 0.025 in square post, six pin header on the board (Berg P/N 75867-131 or equivalent). Connect the ZCL and ZDA pins of the Z80S183 to pins 4 and 6 of this header as illustrated in Figure 27, which is a top view.

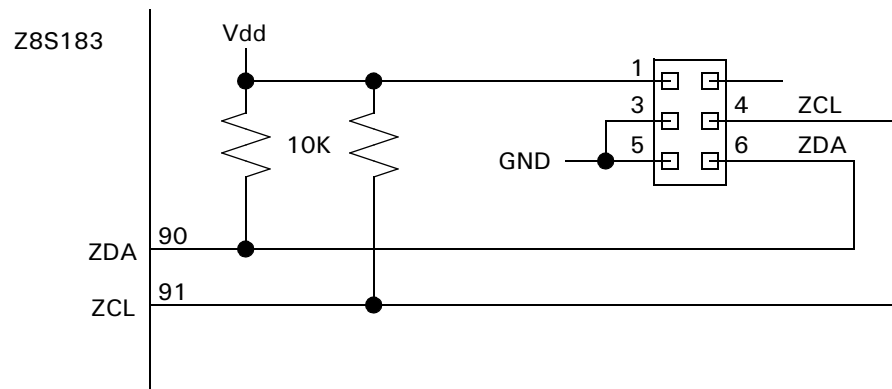


FIGURE 27. ZDI CONNECTOR ON TARGET BOARD

PACKAGING

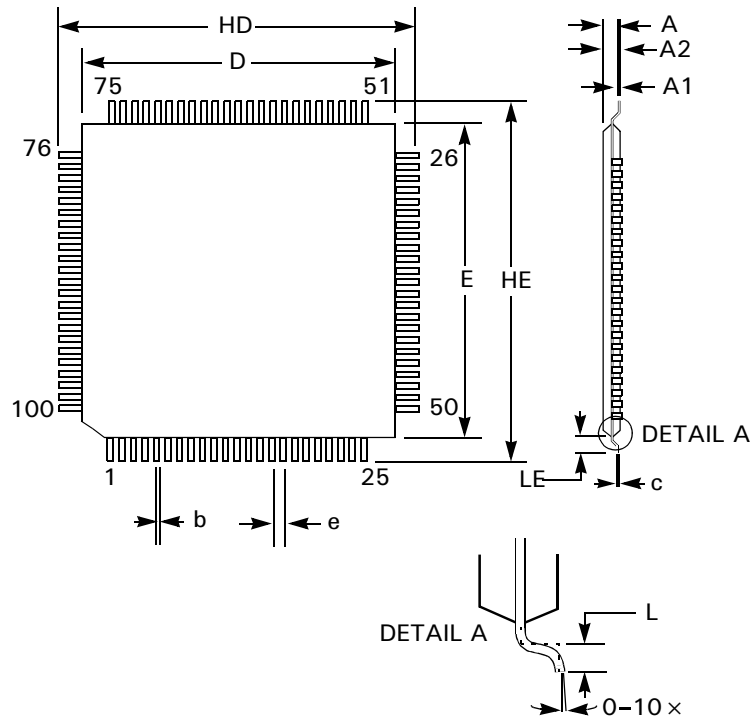


FIGURE 28. 100 PIN THIN QFP

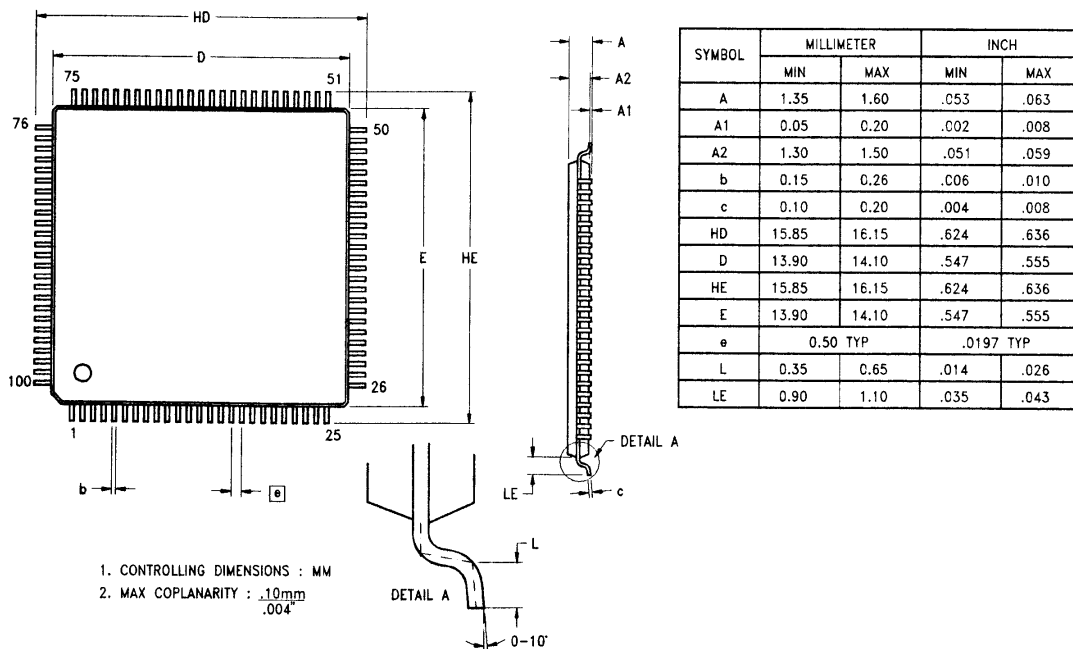


FIGURE 29. 100-PIN VQFP PACKAGE DIAGRAM

ORDERING INFORMATION

| |
|-----------------------------|
| Z80S183 (33 MHz) |
| Standard Temperature |
| 100 Pin QFP |
| Z80S18333ASCRxxxx |
| Extended Temperature |
| 100 Pin QFP |
| Z80S18333AECRxxxx |
| Z8L183 (20 MHz) |
| Standard Temperature |
| 100 Pin QFP |
| Z8L18320ASCRxxxx |
| Extended Temperature |
| 100 Pin QFP |
| Z8L18320AECRxxxx |

For fast results, contact your nearest ZiLOG sale office or the ZiLOG Customer Support Center at 877-945-6427 (web site: www.zilog.com) for assistance in ordering parts.

PART NUMBER DESCRIPTION

ZiLOG part numbers consist of a number of components.

Example:

Part number Z80S183 33 A S C, a Z80S183, 33 MHz, Thin Quad Flat Pack, 0 to 70 C, Plastic Standard Flow, is made up of the codes described in the following table.

| | |
|-------|--------------------|
| Z | ZiLOG prefix |
| 8S183 | Product Number |
| 33 | Speed |
| A | Package |
| S | Temperature |
| C | Environmental Flow |

ROM CODE SUBMISSION

ROM Code Submission Instructions

Outlook Public Folders: All Public Folders: ZiLOG Corporate:
 ROM Submission Forms: Z8S183-4.2.

ZiLOG Z8S183 CODE SUBMISSION FORM

To submit a ROM Code:

1. Complete ROM code submission form.
2. Send this form and the hex file (in Intel Hex format) as separate attachments in an E-mail to: `codes@zilog.com`.

Company Name: _____ Disty/Subcon: _____ Date: _____
ZiLOG P/N: _____ Package Code Legend: _____
Checksum _____ Rev. (input by ZiLOG) _____
Company P/N: _____ File P Input: (input by ZiLOG) _____
Expected Annual volume in Units: _____
Application: _____

SPECIAL INSTRUCTIONS: (Optional)

ZiLOG Sales Office (or your City and Country): _____
Send ROM Verification to: _____
Phone: _____ E-mail: _____ Fax: _____

TOPMARK OF PARTS

To submit a topmark:

1. Check box for default or custom for the preferred package.
2. On default, 9999 indicates ROM number assigned to part by ZiLOG.
3. If custom topmark is selected, enter characters in the space provided. ZiLOG adds the date code (XXYY BB) as bottom line and align as shown on default topmark.
4. If you do not use all the lines on a custom topmark you must leave the top line blank.
5. For TM (Trademark) symbol, place lowercase *tm*.
6. For © (Copyright) symbol, place lowercase *c* followed by a space.
7. For custom logo, attach BMP, JPEG or GIF file to this form.
8. To use **ZiLOG** in a custom topmark, type: pZiLOG. The p is a placemark for the ZiLOG name.

The Z logo is used when applicable. (See Default)

VQFP 100 PIN Custom _____
11 Char Max. _____ Default _____

DISCLAIMER

©1999 by ZiLOG, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval of ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses are conveyed, implicitly or otherwise, by this document under any intellectual property rights.

PRECHARACTERIZATION PRODUCT

The product represented by this document is newly introduced and ZiLOG has not completed the full characterization of the product. The document states what ZiLOG knows about this product at this time, but additional features or nonconformance with some aspects of the document may be found, either by ZiLOG or its customers, in the course of further application and characterization work. In addition, ZiLOG cautions that delivery may be uncertain at times, because of start-up yield issues.

ZiLOG, Inc.
910 East Hamilton Avenue
Campbell, CA 95008
Telephone (408) 558-8500
FAX 408 558-8300
Internet: [HTTP://WWW.ZILOG.COM](http://www.ZiLOG.COM)

DOCUMENT INFORMATION**CHANGE LOG**

| Rev | Date | Purpose | By |
|------------|-------------|----------------|-----------|
| 00 | 09/13/99 | Original issue | ggamble |

A

| | |
|--------------------------------|-----|
| Absolute maximum ratings | 165 |
| AC characteristics | |
| Extended temperature range | 171 |
| Normal temperature range | 168 |
| ADC | |
| Control register 0 | 126 |
| Control register 1 | 127 |
| Result register | 128 |
| ADC instruction | 148 |
| ADD instruction | 148 |
| Addressing modes | |
| Direct | 21 |
| Indexed | 21 |
| Register indirect | 21 |
| Relative | 21 |
| Alarm hours registers | 124 |
| Alarm minutes register | 123 |
| Alarm seconds register | 123 |
| Alarm, RTC | 180 |
| Analog/digital converter (ADC) | |
| Channel selection | 50 |
| Interrupts | 51 |
| Power save | 52 |
| Programming | 51 |
| Restarting | 52 |
| Sampling | 52 |
| AND instruction | 148 |
| Architectural overview | 1 |
| Arithmetic instructions | 142 |
| ASCII | |
| ASYNCR reception | 59 |
| ASYNCR transmission | 58 |
| BRG and clock mode | 60 |
| Clock mode | 58 |
| Clocking | 57 |
| Features | 57 |
| ASCII options | |
| 7 or 8 bit | 60 |
| Parity | 60 |
| Transmit Stop bits | 61 |
| ASCII0 | |
| Control register A | 131 |
| Control register B | 132 |
| Extension control register | 137 |
| Rx data register | 136 |
| Status register | 134 |
| Time constant high register | 139 |

| | |
|----------------------------|-----|
| Time constant low register | 138 |
| Tx data register | 135 |

ASCII

| | |
|----------------------------|-----|
| Control register A | 132 |
| Control register B | 133 |
| Extension control register | 138 |
| Rx data register | 136 |
| Status register | 135 |
| Time constant low register | 139 |
| Tx data register | 136 |
| Assembly language syntax | 147 |

B

| | |
|-------------------------------|-----|
| Bank base register | 82 |
| Basic timing diagram | 175 |
| Baud rate generator (BRG) | 57 |
| Bit instruction | 148 |
| Bit manipulation instructions | 143 |
| Block diagram | 2 |
| Illustrated | 3 |
| Block transfer instructions | 143 |
| Break conditions status | 61 |
| Bus exchange timing diagram | 178 |

C

| | |
|------------------------------------------------|-----|
| Call instruction | 148 |
| Cancellation of transmit or receive capability | 69 |
| Capacitance | 175 |
| Capacitive load | 179 |
| CCF instruction | 148 |
| Century register | 123 |
| Change log | 185 |
| Channel completion | 42 |
| Channels | |
| DMA | 36 |
| DMA basics | 36 |
| Characteristics, electrical | |
| Absolute maximum ratings | 165 |
| AC | 168 |
| Capacitive load | 179 |
| Curves | 179 |
| DC | 166 |
| Standard test conditions | 165 |
| Chip select, I/O | 26 |
| Classes of instructions | 142 |

| | |
|------------------------------------------------|-----|
| Clock | |
| ASCI | 57 |
| Circuit description | 27 |
| Crystal specifications | 29 |
| CSI/O selection | 67 |
| Direct vs. divide-by-2 option | 27 |
| Input | 68 |
| Output | 68 |
| Reduced oscillator drive option | 29 |
| Selection | 27 |
| Clock control register | 73 |
| Command register | 112 |
| Common bank area register | 83 |
| Common base register | 82 |
| Condition codes | 145 |
| Control register | 140 |
| Converters | |
| Analog/digital (ADC) | 49 |
| Digital/Analog (DAC) | 48 |
| CP instruction | 149 |
| CPD instruction | 149 |
| CPDR instruction | 149 |
| CPI instruction | 149 |
| CPIR instruction | 149 |
| CPL instruction | 149 |
| CPU control register | 73 |
| CSI/O | |
| Byte reception | 69 |
| Byte transmission | 69 |
| Cancellation of transmit or receive capability | 69 |
| Clock selection | 67 |
| Data register | 141 |
| Input clock | 68 |
| Operation | 68 |
| Output clock | 68 |
| Curves, characteristic | 179 |
| D | |
| DAA instruction | 149 |
| DAC | |
| Control register | 124 |
| Data register | 125 |
| Data bits, 7 or 8 option | 60 |
| Data direction registers | 34 |
| Data registers | 34 |
| Date register | 121 |
| Day of week register | 121 |
| DC characteristics | 166 |
| Extended temperature range | 167 |
| Normal temperature range | 166 |
| Debug interface | 181 |
| DEC instruction | 149 |
| Decoding | |
| External RAM | 23 |
| External ROM/Flash | 23 |
| Description | |
| Operational | 10 |
| Processor | 10 |
| Design considerations | 180 |
| Device and version ID registers | 13 |
| DI instruction | 149 |
| Digital/Analog converter (DAC) | 48 |
| Direct addressing mode | 21 |
| Disclaimers | 185 |
| Divide-by-2 | 27 |
| DJNZ instruction | 149 |
| DMA | |
| Basics | 36 |
| Channel completion | 42 |
| Channels | 36 |
| Edge-sensitive requests | 37 |
| Handling interrupts | 43 |
| Interrupts | 39 |
| Level-sensitive requests | 37 |
| Memory-to-memory modes | 39 |
| Mode register | 107 |
| Processor/DMA operation | 38 |
| Requests | 37 |
| Setting up transfers | 40 |
| Source and destination modes | 41 |
| Status register | 106 |
| Timing diagram | 179 |
| Wait control register | 109 |
| DMA0 | |
| Byte count high register | 102 |
| Byte count low register | 101 |
| Destination address B register | 101 |
| Destination address high register | 100 |
| Destination address low register | 100 |
| Source address B register | 99 |
| Source address high register | 99 |
| Source address low register | 98 |

| | |
|-------------------------------------------|-----|
| DMA1 | |
| Byte count high register | 106 |
| Byte count low register | 105 |
| I/O address high register | 103 |
| I/O address low register | 103 |
| I/O address register B | 104 |
| Memory address B register | 103 |
| Memory address high register | 102 |
| Memory address low register | 102 |
| E | |
| Edge-sensitive requests | 37 |
| EI instruction | 150 |
| Electrical characteristics | |
| AC | 168 |
| Capacitance | 175 |
| Capacitive load vs. switching time | 179 |
| Curves | 179 |
| DC | 166 |
| Timing diagrams | 175 |
| Errata | 180 |
| Error message vs. restart | 13 |
| EX AF AF' instruction | 150 |
| EX instruction | 150 |
| Exchange instructions | 142 |
| Extending the instruction set | 13 |
| EXX instruction | 150 |
| F | |
| Features | 1 |
| ASCI | 57 |
| Reduced drive/low noise | 33 |
| Flag | |
| Condition codes | 146 |
| Description | 11 |
| R register | 145 |
| Settings definitions | 145 |
| Free-running register | 72 |
| H | |
| HALT instruction | 150 |
| Hours register | 120 |
| I | |
| I/O 91 | |
| Chip select 26 | |
| Control register 77 | |
| Data direction registers | 34 |
| Data registers | 34 |
| Iinstructions | 24 |
| Instructions | 144 |
| IORD timing | 26 |
| Port A alternate function select (AFSA) | 34 |
| Port A alternate function select register | 88 |
| Port A data direction register | 88 |
| Port A data register | 87 |
| Port A output control register | 89 |
| Port B alternate function select (AFSB) | 35 |
| Port B data direction register | 90 |
| Port B data register | 89 |
| Port B output control register | 92 |
| Port B weak latch disable feature | 35 |
| Port C alternate function register | 94 |
| Port C alternate function select (AFSC) | 35 |
| Port C data direction register | 93 |
| Port C data register | 92 |
| Port C output control register | 94 |
| Port D alternate function select (AFSD) | 35 |
| Port D alternate function select register | 96 |
| Port D data direction register | 95 |
| Port D data register | 95 |
| Port D output control register | 97 |
| Ports | 33 |
| Read timing diagram | 177 |
| Relocating the 180 registers | 25 |
| Space | 10 |
| Waits | 26 |
| Write enable/lock | 25 |
| Write timing diagram | 177 |
| IEF1, IEF2 bits | 14 |
| Illegal instruction traps | 12 |
| Error message vs. restart | 13 |
| Extending the instruction set | 13 |
| Trap handling | 12 |
| IM instruction | 150 |
| IN instruction | 150 |
| IND instruction | 150 |
| Indexed addressing mode | 21 |
| INDR instruction | 150 |
| INI instruction | 150 |
| INIR instruction | 151 |
| INO instruction | 150 |

| | | | |
|--------------------------|-----|-----------------------|--------|
| Instruction | | | |
| ADC | 148 | OTR | 154 |
| ADD | 148 | OUT | 154 |
| AND | 148 | OUT0 | 154 |
| Assembly language syntax | 147 | OUTD | 154 |
| Bit | 148 | OUTI | 154 |
| Call | 148 | POP | 154 |
| CCF | 148 | PUSH | 154 |
| CP | 149 | RES | 154 |
| CPD | 149 | RET | 154 |
| CPDR | 149 | RETI | 155 |
| CPI | 149 | RETN | 155 |
| CPIR | 149 | RL | 155 |
| CPL | 149 | RLA | 155 |
| DAA | 149 | RLC | 155 |
| DEC | 149 | RLCA | 155 |
| DI | 149 | RLD | 155 |
| DJNZ | 149 | RLRL | 148 |
| EI | 150 | RR | 155 |
| EX | 150 | RRA | 155 |
| EX AF,AF' | 150 | RRC | 155 |
| Extending | 13 | RRCA | 156 |
| EXX | 150 | RRD | 156 |
| HALT | 150 | RST | 156 |
| IM | 150 | SBC | 156 |
| IN | 150 | SCF | 156 |
| IN0 | 150 | SET | 156 |
| IND | 150 | SLA | 156 |
| INDR | 150 | SLP | 156 |
| INI | 150 | SRA | 156 |
| INIR | 151 | SRL | 156 |
| I/O | 24 | SUB | 157 |
| JP | 151 | TST | 157 |
| JR | 151 | TSTIO | 157 |
| LD | 151 | XOR | 157 |
| LDD | 152 | Instruction Set | |
| LDDR | 152 | Summary | 148 |
| LDI | 152 | Symbols | 146 |
| LDIR | 153 | Interrupts | |
| MLT | 153 | A/D | 51 |
| NEG | 153 | DMA | 39 |
| NOP | 153 | Edge register | 14, 81 |
| OR | 153 | high address register | 11 |
| OTDM | 153 | I register | 14 |
| OTDMR | 153 | IEF1, IEF2 | 14 |
| OTDR | 153 | IL register | 14 |
| OTIM | 153 | Registers | 80 |
| OTIMR | 154 | Timing diagram | 178 |
| | | Trap control register | 14, 80 |

| | | | |
|------------------------------|-----|-----------------------------------|--------|
| Vector low register | 80 | Month register | 122 |
| IORD timing | 26 | N | |
| J | | No DRAM Refresh | 24 |
| JP instruction | 151 | NOP instruction | 153 |
| JR instruction | 151 | O | |
| L | | On-chip | |
| LD instruction | 151 | Observing read data | 12 |
| LDD instruction | 152 | RAM | 22 |
| LDDR instruction | 152 | ROM | 22 |
| LDI instruction | 152 | Op code map | |
| LDIR instruction | 153 | 1st op code | 158 |
| Level-sensitive requests | 37 | 2nd op code after 0CBH | 159 |
| Load instructions | 142 | 2nd op code after 0DDH | 160 |
| logic, negative | 180 | 2nd op code after 0EDH | 161 |
| Logical instructions | 142 | 2nd op code after 0FDH | 162 |
| M | | 4th byte, after 0DDH, 0CBH, and d | 163 |
| Memory Management Unit (MMU) | 10 | 4th byte, after 0FDH, 0CBH, and d | 164 |
| Memory | | Operating mode control register | 76 |
| Read timing diagram | 176 | Operation | |
| Write timing diagram | 176 | CSI/O | 68 |
| Memory-to-memory modes | 39 | Status | 61, 63 |
| MLT instruction | 153 | Operational overview | 2 |
| Modes | | OR instruction | 153 |
| Addressing | 21 | Order information | 183 |
| ASCI clock | 58 | Part number description | 183 |
| BRG | 63 | ROM code submission | 183 |
| Burst | 39 | Topmark submission | 184 |
| Cycle steal | 39 | OTDM instruction | 153 |
| DMA | 39 | OTDMR instruction | 153 |
| DMA0 source and destination | 41 | OTDR instruction | 153 |
| DMA1 operating | 42 | OTIM instruction | 153 |
| Edge-sensitive request | 36 | OTIMR instruction | 154 |
| Falling-edge | 37 | OTR instruction | 154 |
| Idle | 32 | OUT instruction | 154 |
| INT0 | 16 | OUT0 instruction | 154 |
| Isochronous | 58 | OUTD instruction | 154 |
| Low-power | 30 | OUTI instruction | 154 |
| Memory-to-memory | 39 | Output control register | 77 |
| Multiprocessor | 65 | Output control register reset | 180 |
| Nine-bit | 65 | Overview | 1 |
| Sleep | 32 | P | |
| Standby | 32 | Packaging | 182 |
| Standby with Quick recovery | 32 | Parity option | 60 |
| System stop | 32 | Part number description | 183 |

| | | |
|-----------------------------------------------|---------------------------------------|-----|
| PIOS | Status | 62 |
| address/type register | Reception | |
| Control register | ASCI | 59 |
| Counter high | Reduced drive/low noise feature | 33 |
| Counter low | Refresh control register | 74 |
| Port 1 initialization | Registers | 145 |
| POP instruction | 180 relocation | 25 |
| Port B alternate function select register ... | ADC control 0 | 126 |
| Power control register | ADC control 1 | 127 |
| Power save | ADC result | 128 |
| Precharacterization | Alarm hours | 124 |
| Processor control instructions | Alarm minutes | 123 |
| Processor control registers | Alarm seconds | 123 |
| Flags (F) | ASCI0 control A | 131 |
| Interrupt high address (I) | ASCI0 extension control | 137 |
| Program counter (PC) | ASCI0 Rx data | 136 |
| R counter (R) | ASCI0 status | 134 |
| Stack pointer (SP) | ASCI0 time constant high | 139 |
| Processor description | ASCI0 time constant low | 138 |
| Input/output space | ASCI0 Tx data | 135 |
| Memory ManagementUnit (MMU) | ASCI1 control B | 133 |
| Processor control registers | ASCI1 extension control | 138 |
| Processor/DMA operation | ASCI1 Rx data | 136 |
| Program control instructions | ASCI1 status | 135 |
| Program counter (PC) | ASCI1 time constant low | 139 |
| PRT | ASCI1 Tx data | 136 |
| Race condition | ASCI1control A | 132 |
| PRT0 | Bank base | 82 |
| Data high register | Century | 123 |
| Reload high register | Clock control | 73 |
| Reload low register | Common bank area | 83 |
| PRT1 | Common base | 82 |
| Reload register high | CPU control | 73 |
| Reload register low | CSI/O control register | 140 |
| Timer data high register | CSI/O data | 141 |
| Timer data low register | DAC control | 124 |
| PRTO | DAC data | 125 |
| Data low register | Date | 121 |
| PUSH instruction | Day of week | 121 |
| PWRSWTCH negative logic | Device and version ID | 13 |
| | DMA mode | 107 |
| R | DMA Status | 106 |
| R counter register | DMA wait control | 109 |
| RAM | DMA0 byte count high | 102 |
| Lower boundary register | DMA0 byte count low | 101 |
| Upper boundary register | DMA0 destination address B | 101 |
| Receive | DMA0 destination address high | 100 |
| CSI/O byte | DMA0 destination address low | 100 |

| | |
|--------------------------------------|-----|
| Registers, continued | |
| DMA0 source address B | 99 |
| DMA0 source address high | 99 |
| DMA0 source address low | 98 |
| DMA1 byte count high | 106 |
| DMA1 byte count low | 105 |
| DMA1 I/O address high | 103 |
| DMA1 I/O address low | 103 |
| DMA1 I/O address register B | 104 |
| DMA1 memory address B | 103 |
| DMA1 memory address high | 102 |
| DMA1 memory address low | 102 |
| Flags (F) | 11 |
| Free-running | 72 |
| Hours | 120 |
| I register | 14 |
| Indirect addressing mode | 21 |
| Interrupt edge | 14 |
| I/O control | 77 |
| I/O Port A alternate function select | 88 |
| I/O Port A data | 87 |
| I/O Port A data direction | 88 |
| I/O Port A output control | 89 |
| I/O Port B alternate function select | 91 |
| I/O Port B data | 89 |
| I/O Port B data direction | 90 |
| I/O Port B output control | 92 |
| I/O Port C alternate function select | 94 |
| I/O Port C data | 92 |
| I/O Port C data direction | 93 |
| I/O Port C output control | 94 |
| I/O Port D alternate function select | 96 |
| I/O Port D data | 95 |
| I/O Port D data direction | 95 |
| I/O Port D output control | 97 |
| IL register | 14 |
| Interrupt | 80 |
| Interrupt edge | 81 |
| Interrupt high address (I) | 11 |
| Interrupt trap control | 80 |
| Interrupt vector low | 80 |
| Interrupt/trap control | 14 |
| Month | 122 |
| Operating mode control | 76 |
| Output control | 77 |
| PIOS address/type | 130 |
| PIOS control | 129 |
| PIOS counter high | 130 |
| PIOS counter low | 130 |
| Power control | 78 |
| Processor control | 11 |
| Processor program | 10 |
| PRT0 data high | 113 |
| PRT0 data low | 113 |
| PRT0 reload high | 115 |
| PRT0 reload low | 113 |
| PRT1 reload high | 117 |
| PRT1 reload low | 116 |
| PRT1 timer data high | 116 |
| PRT1 timer data low | 116 |
| R counter (R) | 11 |
| RAM lower boundary | 85 |
| RAM upper boundary | 85 |
| Refresh control | 74 |
| Revision ID | 76 |
| ROM boundary | 84 |
| RTC control/status | 118 |
| System configuration | 79 |
| Timer control | 115 |
| Wait state generator control | 83 |
| Watch-Dog timer command | 112 |
| Watch-Dog timer master | 110 |
| Write enable state | 181 |
| Write enable/lock | 25 |
| Year | 122 |
| Relative addressing mode | 21 |
| Requests, DMA | 37 |
| RES instruction | 154 |
| Reset | 180 |
| RET instruction | 154 |
| RETI instruction | 155 |
| RETN instruction | 155 |
| Revision ID register | 76 |
| RL instruction | 155 |
| RLA instruction | 155 |
| RLC instruction | 155 |
| RLCA instruction | 155 |
| RLD instruction | 155 |
| ROM boundary register | 84 |
| ROM code submission | 183 |
| Rotate and shift instructions | 143 |
| RR instruction | 155 |
| RRA instruction | 155 |
| RRC instruction | 155 |
| RRCA instruction | 156 |
| RRD instruction | 156 |

| | | | |
|-------------------------------------|-----|---------------------------------------|-----|
| RST instruction | 156 | Transfers, DMA | 40 |
| RTC | | Transmission | |
| Clock taken from wrong pin | 180 | ASCI | 58 |
| Control/status register | 118 | Transmit | |
| RTC alarm | 180 | CSI/O byte | 69 |
| Rx overrun status | 61 | Stop bits | 61 |
| | | Trap handling | 12 |
| S | | Triggers, Schmitt | 180 |
| Sampling | 52 | TST instruction | 157 |
| SBC instruction | 156 | TSTIO instruction | 157 |
| SCF instruction | 156 | | |
| Schmitt triggers | 180 | W | |
| SET instruction | 156 | Wait state generator control register | 83 |
| SLA instruction | 156 | Waits | 26 |
| SLP instruction | 156 | Watch-Dog timer | 112 |
| SRA instruction | 156 | Master register | 110 |
| SRL instruction | 156 | | |
| Stack pointer (SP) | 11 | X | |
| Status | | XOR instruction | 157 |
| Break conditions | 61 | | |
| Operation | 61 | Y | |
| Receive | 62 | Year register | 122 |
| Rx overrun | 61 | | |
| SUB instruction | 157 | Z | |
| Symbols, instruction summary | 146 | ZiLOG Debug Interface (ZDI) | 181 |
| system configuration register | 79 | | |
| | | | |
| T | | | |
| Temperature, operating | | | |
| AC extended range | 171 | | |
| AC normal range | 168 | | |
| Temperature, operation | | | |
| DC extended range | 167 | | |
| DC normal range | 166 | | |
| Test conditions | 165 | | |
| Timer control register | 115 | | |
| Timing | | | |
| IORD | 26 | | |
| Timing diagrams | 175 | | |
| Basic timing | 175 | | |
| Bus exchange timing | 178 | | |
| DMA timing | 179 | | |
| I/O read timing | 177 | | |
| I/O write timing | 177 | | |
| Interrupt timing | 178 | | |
| Memory read timing (one wait state) | 176 | | |
| Memory write timing | 176 | | |
| Topmark submission | 184 | | |