L28 Microcomputer (MCU)

Technical Reference Manual (Preliminary)

Doc. No. 1198
June 8, 1999

CONEXANT

Table of Contents

1 INTRODUCTION 1-1
1.1 SUMMARY 1-1
1.2 FEATURES 1-2
1.3 MCU DESIGN CHANGES 1-5
1.3.1 Design Changes Incorporated in the L28 from the L39 1-5
1.4 REFERENCES 1-5
2 HARDWARE INTERFACE SIGNALS 2-1
2.1 SYSTEM BLOCK DIAGRAM 2-1
2.1.1 PIN ASSIGNMENTS 2-1
2.1.2 PIN SIGNAL DESCRIPTIONS 2-1
2.2 SPECIAL CONSIDERATIONS 2-10
2.2.1 Address Map - PA[6:3] RS232/16550 Signal Interference 2-10
2.2.2 Parallel Host/Serial DTE Interface Mode Operation 2-10
3 SYSTEM ARCHITECTURE 3-1
3.1 OVERVIEW AND MEMORY MAPS 3-1
3.1.1 Block Diagram 3-1
3.1.2 Top Level Memory Map 3-1
3.1.3 I/O Register Bit Assignments 3-9
1.1.4 MCU Reset Initialization 3-13
1.2 CENTRAL PROCESSING UNIT (CPU) 3-16
1.2.1 Index Registers 3-16
1.2.2 Stack Pointer 3-17
1.2.3 Arithmetic and Logic Unit (ALU) 3-17
1.2.4 Accumulator (A) 3-17
1.2.5 Program Counter (PC) 3-17
1.2.6 Instruction Register and Instruction Decode 3-17
1.2.7 W Register (W) 3-17
1.2.8 I Register (I) 3-17
1.2.9 Processor Status Register (PSR) 3-18
1.1.10 CPU Interrupt Logic 3-20
1.3 CLOCK OSCILLATOR 3-20
1.4 LOW POWER OPERATION 3-21
1.4.1 Sleep Mode 3-21
1.4.2 Stop Mode 3-21
1.4.3 Wake-Up 3-21
1.4.4 Low Power Register (LPR) 3-22
1.4.5 Snooze Timer 3-24
1.5 IRQ INTERRUPT LOGIC 3-26
1.5.1 Interrupt Request (IRQ)Vector and Hardware Priority 3-26
1.5.2 Break Command 3-26
1.6 INTERNAL ROM 3-30
1.6.1 Size and Location 3-30
1.6.2 TSTP and Internal ROM Control 3-30

1.7 INTERNAL RAM 3-30
1.7.1 Size and Location 3-30
1.8 MEMORY BANKING 3-31
1.8.1 Bank Select Register (BSR) 3-35
1.8.2 ES Speed Register 3-36
1.9 PARALLEL INPUT/OUTPUT PORTS 3-37
3.9.1. Bidirectional Ports A, C, and E 3-40
1.9.1 Bidirectional and Input Only Port D 3-40
1.9.2 Output Port B 3-40
1.9.3 Port B Select Register (PBS) 3-41
1.9.4 External Interrupt Register (EIR) 3-42
1.1.5 Clear Interrupt Register (CIR) 3-43
1.10 COUNTER/TIMERS 3-44
1.10.1 Timer A Registers 3-44
1.10.2 Timer B Registers 3-44
1.10.3 Timer A Mode Register (TAM) 3-46
1.10.4 Timer B Mode Register (TBM) 3-47
1.10.5 Timer Modes 3-48
1.11 PRECISION TIME GENERATORS 3-51
1.11.1 Precision Time Generator A 3-51
1.11.2 PTGA Mode Register (PAM) 3-53
1.11.3 Precision Time Generator B 3-53
1.11.4 PTGB Mode Register (PBM) 3-54
1.11.5 Example Rates 3-54
1.12 USART 3-55
1.12.1 Enhancements over the L39 3-55
1.12.2 General Operation 3-56
1.12.3 Internal Timing 3-56
1.12.4 USART Registers and Serial Buffer Registers (SB) 3-59
1.1.5 Serial Mode Register (SMR) 3-59
1.1.6 Serial Interrupt Register (SIR) 3-61
1.1.7 Serial Line Control Register (SLCR) 3-62
1.1.8 Serial Status Register (SSR) 3-63
1.1.9 Serial Form Register (SFR) 3-64
1.1.10 Serial Out (RXD) Divider Latch (SODL) 3-65
1.1.11 Serial In (TXD) Divider Latch (SIDL) 3-65
1.13 DUAL PORT RAM - 16550A/16450 INTERFACE 3-66
1.13.1 Host Bus Interface Signals and Registers 3-66
1.13.2 16550A Interface Mode and FIFOs 3-68
1.13.3 16450 Interface Mode and FIFOs 3-69
1.13.4 Host Control Register (HCR) 3-76
1.13.5 Interrupt Enable Register (IER) 3-77
1.13.6 Interrupt Identifier Register (IIR) 3-78
1.13.7 Line Control Register (LCR) 3-79
1.13.8 Modem Control Register (MCR) 3-80

1.13.9 Line Status Register (LSR) ... 3-81
1.13.10 Modem Status Register (MSR) ... 3-82
1.13.11 Divisor Latch Registers ... 3-82
1.13.12 Scratch Register ... 3-82
1.13.13 FIFO Control Register (FCR) ... 3-83
1.13.14 FIFO Interrupt Enable Register (FIER) ... 3-84
1.13.15 FIFO Status Register (FSR) ... 3-84
1.13.16 GP FIFO Status Register (GPFS) ... 3-85
1.13.17 16550 DMA Operation ... 3-86
1.14 INTERNAL DMA OPERATION ... 3-87
1.14.1 DMA Control/Special Features ... 3-90
1.14.2 DMA Enable Register (DER) ... 3-91
1.14.3 DMA Flag Register (DFR) ... 3-92
1.14.4 DMA Status Register (DSR) ... 3-93
1.14.5 DMA Control ... 3-94
1.14.6 DTE Generated Flow Control (DTE In-bound) ... 3-95
1.14.7 Modem Generated Flow Control (DTE Out-bound) ... 3-96
1.14.8 Tx CHAR3 Escape Sequence ... 3-97
1.14.9 Register Description ... 3-98
1.14.10 DMA Timing ... 3-100
1.14.11 Example DMA Setup ... 3-102
1.14.12 Non DMA Setup ... 3-102
1.14.13 DMA Initialization ... 3-102
1.14.14 Resetting the DMA ... 3-104
1.15 EXPANSION BUS ... 3-105
1.16 TEST MODE ... 3-105
1.17 MASK OPTIONS ... 3-105
1.17.1 Selectable Options ... 3-105
1.17.2 Mask Option Register (MOR) ... 3-105
1.18 CYCLIC REDUNDANCY CHECK (CRC) HARDWARE ... 3-106
4 GENERAL SPECIFICATIONS ... 4-1
4.1 OPERATING CONDITIONS AND MAXIMUM RATINGS ... 4-1
4.1.1 Operating Conditions ... 4-1
4.1.2 Maximum Ratings ... 4-1
4.1.3 Handling CMOS Devices ... 4-1
4.2 ELECTRICAL CHARACTERISTICS ... 4-2
4.3 POWER DISSIPATION ... 4-3
5 TIMING CHARACTERISTICS ... 5-1
5.1 TEST CONDITIONS ... 5-1
5.2 OSCILLATOR TIMING ... 5-1
5.3 HOST BUS INTERFACE ... 5-2
5.4 EXPANSION BUS TIMING ... 5-5
5.5 RESP, NMIP, AND TSTP ASYNCHRONOUS INPUTS ... 5-7
6 PACKAGE DIMENSIONS ... 6-1

List of Figures

Figure 2-1. MCU Block Diagram 2-2
Figure 2-2. MCU Pin Signals for 80-Pin PQFP 2-3
Figure 3-1. MCU Block Diagram 3-2
Figure 3-2. MCU Memory Map Overview 3-3
Figure 3-3. MCU Memory Map Breakdown by 8K-byte Blocks 3-4
Figure 3-4. MCU Memory Map: 0-07FFh 3-5
Figure 3-5. CPU Registers and Data Flow 3-16
Figure 3-6. CPU Clock Oscillator Input Options 3-20
Figure 3-7. Low Power Mode Logic and Timing 3-23
Figure 3-8. Timer C Snooze Timer 3-24
Figure 3-9. IRQ Interrupt Logic Interface 3-27
Figure 3-10. Memory Map - RAM Banking 3-33
Figure 3-11. Banking Logic 3-34
Figure 3-12. Counter/Timer A Block Diagram 3-45
Figure 3-13. Counter/Timer B Block Diagram 3-45
Figure 3-14. Timer Mode 0 (Interval Timer) Waveforms 3-49
Figure 3-15. Timer Mode 2 (Event Counter) Waveforms 3-49
Figure 3-16. Timer Mode 3 (Pulse Width Measurement) Waveforms 3-50
Figure 3-17. Precision Time Generator A Block Diagram 3-52
Figure 3-18. Precision Time Generator B Block Diagram 3-52
Figure 3-19. USART Block Diagram 3-57
Figure 3-20. MCU/Host Interface 3-67
Figure 3-21. TX FIFO Block Diagram 3-73
Figure 3-22. FIFO UART Timing Simulator 3-74
Figure 3-23. RX FIFO Block Diagram 3-75
Figure 3-24. Internal DMA Operation 3-88
Figure 3-25. DMA Operation - External RAM 3-89
Figure 3-26. DMA State Machine Diagram 3-101
Figure 3-27. 16-bit CRC 3-106
Figure 3-28. CRC-16 Polynomial 3-107
Figure 5-1. Host Bus Read Waveforms 5-2
Figure 5-2. Host Bus Write Waveforms 5-3
Figure 5-3. Host Bus HINT Waveforms 5-4
Figure 5-4. Expansion Bus Read Waveforms 5-5
Figure 5-5. Expansion Bus Write Waveforms 5-6
Figure 5-6. RESP, NMIP, and TSTP Input Waveforms 5-7
Figure 5-7. Power On RESP Timing 5-7
Figure 6-1. Package Dimensions - 80-Pin PQFP 6-2

List of Tables

Table 2-1. MCU Pin Signals for 80-Pin PQFP 2-4
Table 2-2. MCU Interface Signal Description 2-6
Table 3-1. MCU Memory Map: 0-7FFh 3-6
Table 3-2. Register Bit Assignments: 0-05FFh 3-9
Table 3-3. MCU Reset Initialization Values 3-13
Table 3-4. Processor Status Register (PSR) 3-18
Table 3-5. Register Bit Assignments: Low Power Register (LPR) - 0009h 3-22
Table 3-6. Snooze Timer Bit Assignments 3-25
Table 3-7. LPR Bits Affected by TCM3 and TCM4 3-25
Table 3-8. Interrupt Request (IRQ) Vector and Hardware Priority 3-28
Table 3-9. Clearing Source of IRQ Interrupt 3-29
Table 3-10. TSTP and Internal ROM Control 3-30
Table 3-11. Register Bit Assignments: Bank Select Register i (BSRi) - 0018h-001Fh 3-35
Table 3-12. Register Bit Assignments: ES Speed Register - 0033h 3-36
Table 3-13. I/O Port Special Purpose Functions 3-37
Table 3-14. I/O Port Special Purpose Function Control 3-38
Table 3-15. Port B Select Register (PBS) 3-41
Table 3-16. External Interrupt Register (EIR) - $000A 3-42
Table 3-17. Clear Interrupt Register (CIR)- $000B 3-43
Table 3-18. Register Bit Assignments: 0010h - 0013h 3-46
Table 3-19. Register Bit Assignments: 0014h - 0017h 3-47
Table 3-20. Register Bit Assignments: PTGA - 0034 3-53
Table 3-21. Register Bit Assignments: PTGB - 000Ch 3-54
Table 3-22. PTGA or PTGB Generated Standard Data Rates 3-54
Table 3-23. USART Generated Standard Data Rates 3-58
Table 3-24. Register Bit Assignments: USART - 0038h - 003Fh 3-59
Table 3-25. MCU Dual Port RAM Signal Equivalence 3-67
Table 3-26. MCU Host Bus Interface Memory Map - 16550A/16450 Mode 3-70
Table 3-27. Register Bit Assignments: 0020h - 0032h 3-71
Table 3-28. 16550 Register Initialization 3-72
Table 3-29. Register Bit Assignments: DMA 05D0, 05F5-05F6 3-90
Table 3-30. DMA Registers 3-99
Table 3-31. Register Bit Assignments: Mask Option Register - 0008h 3-105
Table 3-32. Register Bit Assignments: 05FEh - 05FFh 3-106
Table 5-1. Host Bus Read Timing 5-2
Table 5-2. Host Bus Write Timing 5-3
Table 5-3. Host Bus HINT Timing 5-4
Table 5-4. Expansion Bus Read Timing 5-5
Table 5-5. Expansion Bus Write Timing 5-6
Table 5-6. Power On RESP, NMIP, and TSTP Timing 5-7

This page is intentionally blank.

1 INTRODUCTION

1.1 SUMMARY

The Conexant L28 Microcontroller (MCU) is a complete 8-bit microcontroller fabricated on a single chip using a CMOS silicon gate process. This MCU complements an industry standard line of R6500 and R65C00 microprocessors, as well as R6500*/, R65CXX, C19, C29, C39, C40, and L39 microcomputers, and their compatible peripheral devices. This MCU family has a wide range of controller applications where high 8-bit performance, minimal chip count and low power consumption are required.

The basic MCU features include an enhanced 6502 Central Processing Unit (CPU), 8k bytes of internal mask programmable read only memory (ROM), 1456 bytes of internal random access memory (RAM), three 16-bit counter/timers, two 17-bit precision timing generators, an asynchronous/synchronous USART port, a 16 byte dual port RAM with FIFO which can emulate a 16450 or 16550 device, a four channel DMA/buffer manager, a 16-bit cyclic redundancy check (CRC), a 16-bit address/8-bit data expansion bus, an 8-byte banking RAM, and 38 general purpose input/output ports.

The MCU contains a software controlled ultra low power stop mode with the oscillator turned off or sleep mode with the oscillator remaining on and internal operating clocks off (Idle).

The MCU operates on +5 V.

The MCU provides a clock output with the same frequency as the crystal input and a +3.3V regulated voltage output for use by a Conexant +3.3V modem data pump (MDP). Parallel host or serial DTE interface operation is selected via a dedicated input.

The innovative architecture and the demonstrated high performance of the R65C02 CPU, as well as instruction simplicity, result in system cost effectiveness and a wide range of computational power. These features make the L28 a leading candidate for low power single chip microcontroller applications.

1.2 FEATURES

- Single-chip microcomputer
  - Enhanced R6502 CPU
  - 8k bytes internal read-only memory (ROM)
  - 1456 bytes internal random access memory (RAM)
  - Three 16-bit counter/timers
  - Two 17-bit precision time generators
  - Universal asynchronous/synchronous receiver transmitter (USART)
  - Host bus for 16550A interface or scratchpad RAM interface with 16-byte FIFO and DMA
  - Eight levels of prioritized, vectored interrupts
  - High speed operation: Up to 35 MHz
  - Low power sleep mode
  - Ultra-low power stop mode
  - Snooze timer for automatic wake-up from low power mode
- Enhanced R6502 CPU
  - 12 new bit manipulation and branching instructions to shorten code and speed up execution
  - 21 new arithmetic processing instructions to optimize arithmetic processing
  - 10 new direct threaded code instructions to support high level languages that compile linked machine instructions
  - R6502 instruction compatible except “(indirect,X)” addressing mode changed to “(Indirect)” and “(Indirect),Y” addressing mode changed to “(Indirect),X”
- 38 General purpose input/output (GPIO), output (GPO), or input (GPI) lines
  - 26 GPIO lines with data latches and direction registers: Ports A (0-7), C (0-7), D (0-3), and E (0-4, 7)
  - 8 GPO lines with data latches: Port B (0-7)
  - 4 GPI lines with data latches: Port D (4-7)
- Two 16-bit programmable counter/timers (A and B) with latches
  - Timer A supports four modes: Interval Timer, Pulse Generation, Pulse Width Measurement, Event Counter
  - Timer B supports one mode: Interval Timer
  - Selectable divide-by-32 prescaler
  - Timer interrupt can be vectored to either ROM or page 1 RAM
  - I/O port interface
- One 16-bit programmable timer (Timer C) with latches which can be used for Snooze Timer.
  - Timer C supports four Snooze Clock modes: Clock Off, Internal Oscillator C2 Rate, Internal Oscillator Divide by 127 Rate, and Ring Oscillator Rate (about 100 kHz)
- Two 17-bit precision time generators (PTGs) with latches
  - Increment/decrement capability with counter option (clear accumulator on overflow)
  - Interrupt enables

- 16550A interface (application software dependent)
  - 7 bytes for 16550A registers
  - 5 bytes for scratch pad RAM
  - 1 byte for dual port scratch pad RAM
  - 1 byte for receiver and transmitter FIFO interface (16-byte deep FIFO)
  - 3 bytes for FIFO status
  - DMA interface
- USART Serial I/O
  - Common asynchronous/synchronous features
    - Full double buffering
    - Serial in and serial out individually enabled
    - Interrupt enables for receiver buffer full and transmitter buffer empty
    - Echo modes
    - Timer B or PTGA/PTGB timing
    - Speed recognition
  - Asynchronous features
    - 5-, 6-, 7-, or 8-bit characters
    - Even, odd, stuff or no parity bit generation and detection
    - 1, 1-1/2 or 2 stop bit generation with 3/4 or 7/8 stop bit control
    - False start bit detection
    - Interrupt enables
    - Line break generation and detection
  - Synchronous features
    - Transmit data (TXD) serial input timing – internal, external TXCLK, or external TXREF
    - Received data (RXD) serial output timing – internal or external RXCLK
    - 5-, 6-, 7-, or 8-bit characters
    - Automatic word sync on first 1 to 0 transition
    - Interrupt enables for serial input clock (TXCLK) and serial output clock (RXCLK)
- Internal DMA
  - RS-232/16550 host interface optionally supported by DMA
  - 32 bytes of RAM address assignable to RAM or DMA registers
- Expansion Bus
  - Built-in memory banking allows up to 512k bytes of external memory to be addressed
  - Eight Bank Select Registers independently control 8k-byte memory banks
  - Each BSR defines address translation (A13-A15), A16 and A17 control, and chip select (ES0, ES2-ES3)

- Eight levels of prioritized, vectored interrupts
  - RESP (highest priority)
  - Non-mask interrupt (NMI)
  - Six prioritized interrupt requests (IRQ1-IRQ6)
    - Six IRQ ROM vectors
    - Two software selectable Timer IRQ page 1 RAM vectors
- Internal clock with crystal or clock input
  - Internal divide-by-1 input frequency divider
  - Input Frequency: 1 MHz to 35 MHz
  - Sleep mode, software enabled,
    - Low power dissipation
    - Normal operation resumption within 1.5 clock cycles upon wake up condition
  - Stop mode, software enabled
    - Ultra-low power dissipation
    - Normal operation resumption after hardware delay following power on stabilization upon wake-up condition
  - Flexible wake up conditions from Sleep or Stop mode
    - Awakened by a low level on NMIP
    - Awakened by a high level detected on PA0, software enabled
    - Awakened by a low level detected on PA2, software enabled
    - Awakened by a low level detected PD4, software enabled
    - Awakened by a low level detected PD5 software enabled
    - Awakened by a low level detected on PD4 or on PD5, software enabled
    - Awakened by a snooze timer - Sleep Mode (120 sec maximum programmed delay @ 15 MHz), software enabled
    - Awakened by a snooze timer - Stop Mode (70 - 280 sec* maximum programmed delay), software enabled
    - * Programmed delay can be calibrated by software.
- Clock output with frequency equal to crystal input
- Internal voltage regulator supplies +3.3V output
- Packaged in 80-pin PQFP
- Power supply: +5 V ±5%

1.3 MCU DESIGN CHANGES

MCU changes effecting input/output operation are summarized below.

1.3.1 Design Changes Incorporated in the L28 from the L39.

1. Added Counter.
2. Increased maximum clock speed.
3. Added DMA logic.
4. Added Clock output (replaced a GND pin with Clock output).
5. Added +3.3V output (replaces PE5).
6. Added parallel host/serial DTE interface mode selection input (replaces PE6).
7. Modified masked Boot Loader to allow flash loading of blank external memory.
8. Revised host bus write timing.

1.4 REFERENCES

1. L28 Microcontroller P01 Specification, Rev. A, July 17, 1997.

This page is intentionally blank.

2 HARDWARE INTERFACE SIGNALS

2.1 SYSTEM BLOCK DIAGRAM

The MCU external signals and interfacing internal functions are shown in Figure 2-1.

2.1.1 PIN ASSIGNMENTS

The pin assignments for the MCU in 80-pin PQFP are shown in Figure 2-2 and are listed in Table 2-1.

2.1.2 PIN SIGNAL DESCRIPTIONS

The pin signals are described in Table 2-2.

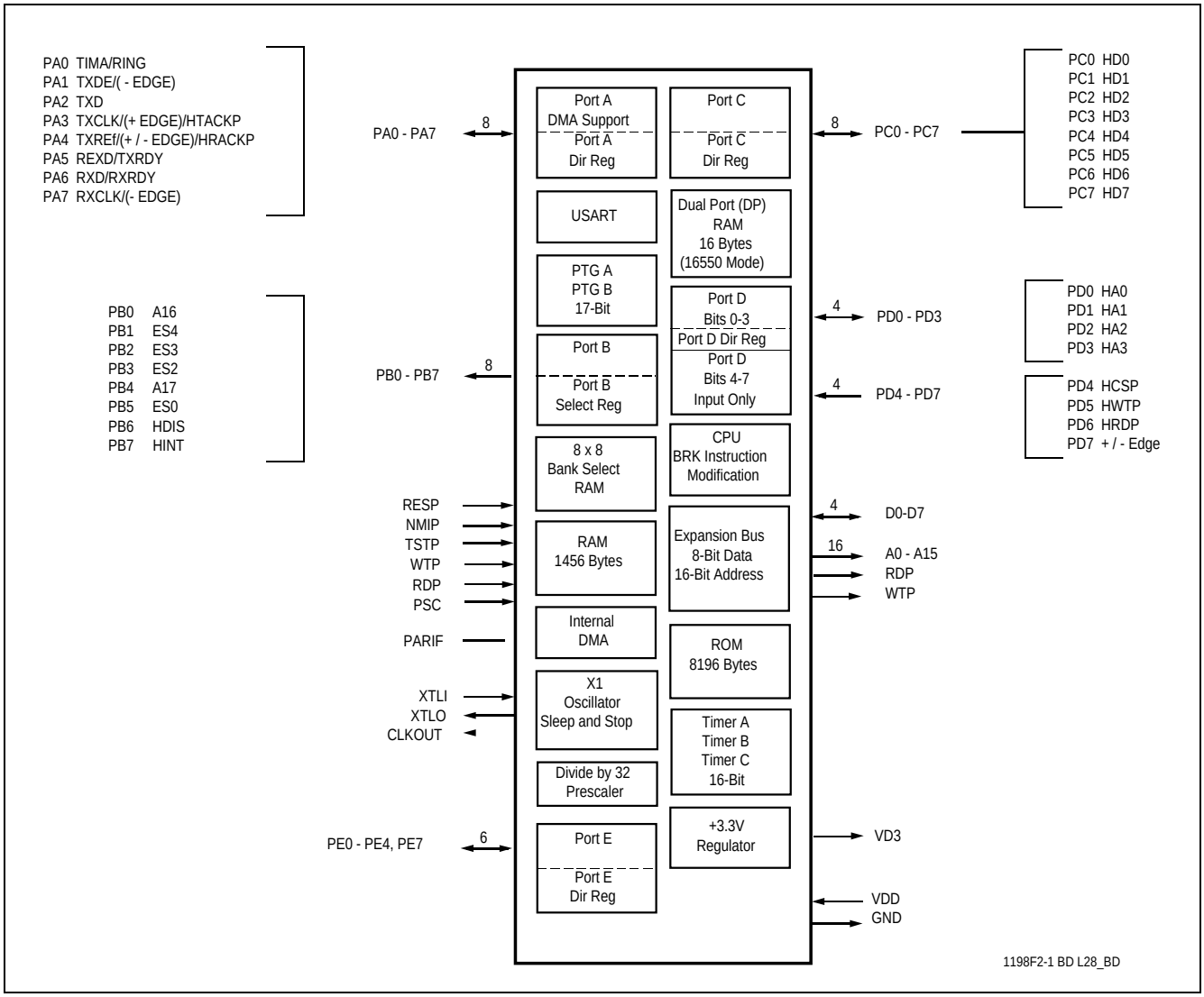


Figure 2-1. MCU Block Diagram

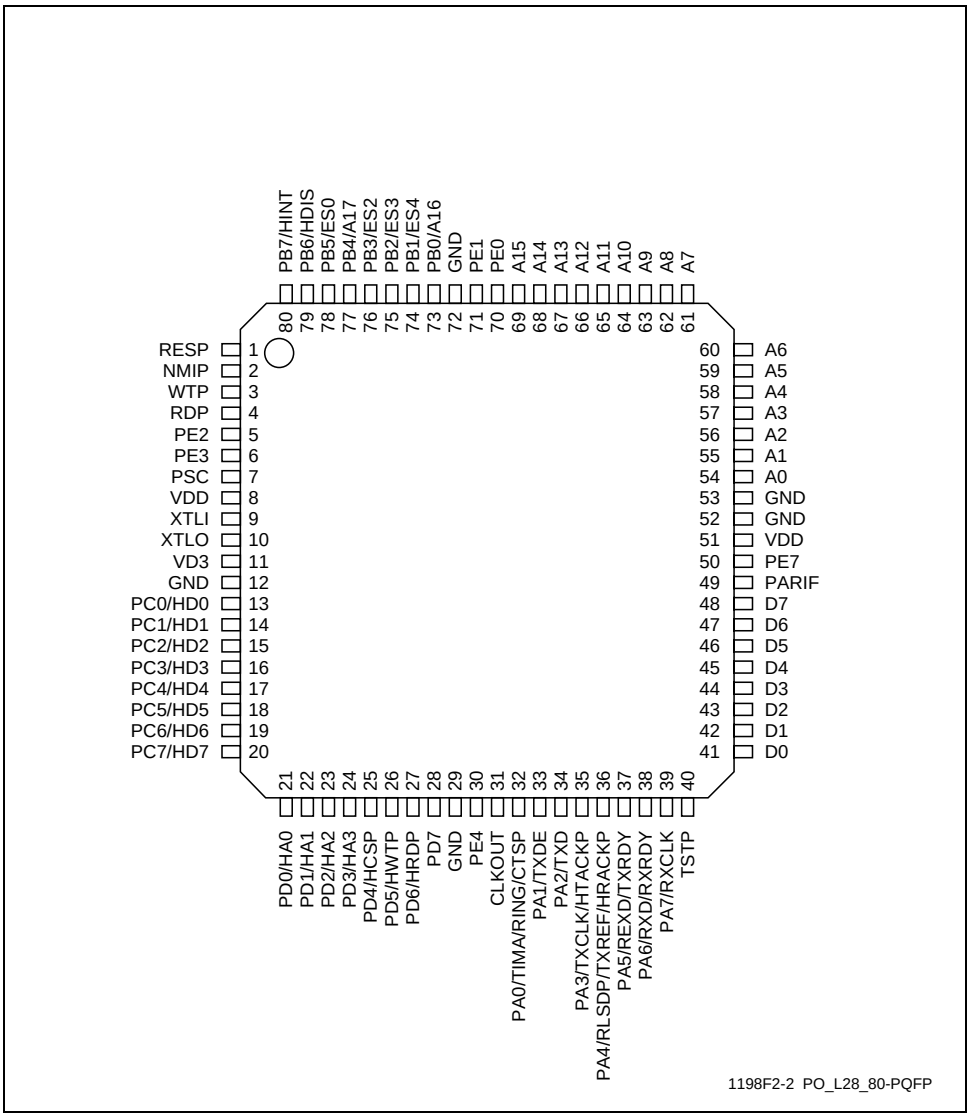


Figure 2-2. MCU Pin Signals for 80-Pin PQFP

Table 2-1. MCU Pin Signals for 80-Pin PQFP

Pin No.	Signal Name	I/O	Pull-up	Receiver Type	Pin No.	Signal Name	I/O	Pull-up	Receiver Type
1	RESP	I		HCMOS	41	D0	I/O		HTTLid
2	NMIP	I	PU	HCMOS	42	D1	I/O		HTTLid
3	WTP	Oh			43	D2	I/O		HTTLid
4	RDP	Oh			44	D3	I/O		HTTLid
5	PE2	I/O		TTL	45	D4	I/O		HTTLid
6	PE3	I/O		TTL	46	D5	I/O		HTTLid
7	PSC	I	PD	CMOS	47	D6	I/O		HTTLid
8	VDD				48	D7	I/O		HTTLid
9	XTLI	I		CMOS	49	PARIF	I	PU	TTL
10	XTLO				50	PE7	I/O		TTL
11	VD3	O		Modem Data Pump	51	VDD			
12	GND				52	GND			
13	PC0/HD0	I/O		HTTL	53	GND			
14	PC1/HD1	I/O		HTTL	54	A0	Or		
15	PC2/HD2	I/O		HTTL	55	A1	Or		
16	PC3/HD3	I/O		HTTL	56	A2	Or		
17	PC4/HD4	I/O		HTTL	57	A3	Or		
18	PC5/HD5	I/O		HTTL	58	A4	Or		
19	PC6/HD6	I/O		HTTL	59	A5	Or		
20	PC7/HD7	I/O		HTTL	60	A6	Or		
21	PD0/ HA0	I/O I		HTTL	61	A7	Or		
22	PD1/ HA1	I/O I		HTTL	62	A8	Or		
23	PD2/ HA2	I/O I		HTTL	63	A9	Or		
24	PD3/ HA3	I/O I		HTTL	64	A10	Or		
25	PD4/HCSP	I		HTTL	65	A11	Or		
26	PD5/HWTP	I		HTTL	66	A12	Or		
27	PD6/HRDP	I		HTTL	67	A13	Or		
28	PD7	I		HTTL	68	A14	Or		
29	GND				69	A15	Or		
30	PE4	I/O		TTL	70	PE0	I/O		TTL
31	CLKOUT	O		TTL	71	PE1	I/O		TTL
32	PA0/TIMA/ RING/CTSP	I/O I		TTL	72	GND			
33	PA1/ TXDE	I/O O		TTL	73	PB0/A16	Or		
34	PA2/ TXD	I/O I		TTL	74	PB1/ES4	Or		
35	PA3/TXCLK/ HTACKP	I/O I	PU	TTL	75	PB2/ES3	Or		
36	PA4/ RLSDP/ HRACKP	I/O/ O I	PU	TTL	76	PB3/ES2	Or		
37	PA5/ REXD/ TXRDY	I/O I O		TTL	77	PB4/A17	Or		
38	PA6/ RXD/RXRDY	I/O O		TTL	78	PB5/ES0	Or		
39	PA7/RXCLK	I/O		TTL	79	PB6/HDIS	Or		
40	TSTP	I	PU	CMOS	80	PB7/HINT	Or		

Table 2-1. MCU Pin Signals for 80-Pin PQFP (Cont'd)

I/O Types:	
HTTLid	Receiver turned off during Idle and stop mode
HTTL	hysteresis TTL Receiver
HCMOS	Hysterics CMOS Receiver
I/O	Output floats when MRESP=0
Or	Output floats when MRESP=0
Oh	Output is forced high when MRESP=0
O	Output active

Table 2-2. MCU Interface Signal Description

Symbol	Pin	I/O	Name/Function
Input/Output Ports			
PA0-PA7	32-39	I/O	**Port A. General Purpose 8-bit I/O Port.** All pins can be assigned special functions under software control. The Port A Direction Register (PAD) must be set appropriately for all pins in both general and special use. PA1, PA3, PA4 and PA7 have edge detect circuitry that remains active regardless of data direction. PA3 and PA4 each have an internal pull-up.
TIMA (PA0)	32	I/O	**Timer A I/O.** An input in Timer A Event Counter or Pulse Width Measurement mode and an output in Timer A Pulse Generation mode.
RING (PA0)	32	I	**Ring.** An unclocked input whose positive level is used to clear the low power mode. Ring is enabled by LPR5.
CTSP (PA0)	32	I	**Clear To Send.** The negative edge of an external CTSP signal can be used to initialize the USART TXD synchronous mode word counter (SM1 = 1).
TXDE (PA1)	33	O	**Serial Input Passthrough.** PA2 input is connected to the PA1 output when the serial passthrough is selected (SF7 = 1).
Falling Edge (PA1)	33	I/O	**PA1 Negative Edge Detect.** The PA1 edge detect flag (EI7) is set on each detected edge and is cleared by writing a zero to register bit CI7. When the PA1 edge detect flag is asserted (EI7 = 1) it will generate an IRQ5 interrupt if its enable is set (EI4 = 1).
TXD (PA2)	34	I	**TXD Serial Input.** PA2 input is connected to the USART as the serial input stream when USART TXD mode is selected (SM6 = 1).
TXCLK (PA3)	35	I/O	**TXCLK Clock.** PA3 is internally connected to the serial input clock when synchronous USART operation is selected (SM4) = 1). TXCLK can be an external signal, an internal signal, or can be internally connected to TXREF (PA4). TXD is clocked on the rising edge of TXCLK.
Rising Edge (PA2-PA3)	34-35	I/O	**PA3/PA2 Positive Edge Detect.** The PA3/PA2 edge detect flag (SI6) is set on each detected edge and is cleared by writing a zero to register bit CI3. When the PA3/PA2 edge detect flag is asserted (SI6 = 1) it will generate an IRQ4 interrupt if its enable is set (SI4 = 1). A PA3 edge is detected when SF3 = 0 and a PA2 edge is detected when SF3 = 1.
HTACKP (PA3)	35	I	**Host Transmit Acknowledge.** PA3 is an active low transmit acknowledge signal asserting the host 16550 DMA TX FIFO address and chip select as a software option (HCR2 = 1). Non-operational in the 16450 mode or when serial mode is selected (SM7 = 1 or SM4 = 1). Operational only in 16550 mode when serial mode is not selected (SM7 = 0 and SM4 = 0).
TXREF (PA4)	36	I	**External TXD Reference Clock.** PA4 input is connected internally to PA3 output when synchronous USART operation is selected (SM4 = 1) and TXREF is selected as the TXCLK source (SM2 = 1).
Rising or Falling Edge (PA4)	36	I/O	**PA4 Positive/Negative Edge Detect.** The PA4 edge detect flag (EI6) is set on each detected edge and is cleared by writing a zero to register bit CI6. When the PA4 edge detect flag is asserted (EI6 = 1) it will generate an IRQ2 interrupt if its enable is set (EI3 = 1). When EI1 = 0 negative edges are detected and when EI1 = 1 positive edges are sensed.
RLSDP (PA4)	36	O	**Receiver Line Signal Detect.** The negative edge of an external RLSDP signal can be used to initialize the USART RXD synchronous mode word counter (SM1 = 1).
HRACKP (PA4)	36	I	**Host Receive Acknowledge.** PA4 is an active low receive acknowledge signal asserting the host 16550 DMA RX FIFO read address and chip select as a software option (HCR2 = 1). Non-operational in the 16450 mode or when serial mode is selected (SM7 = 1 or SM4 = 1). Operational only in 16550 mode when serial mode is not selected (SM7 = 0 and SM4 = 0).
REXD (PA5)	37	I	**Serial Output Passthrough.** PA5 input is internally connected to the PA6 output when serial out passthrough mode is selected (SM7 = 1, SF5 = 0, and SF6 = 1).
TXRDY (PA5)	37	O	**Transmit Ready.** PA5 is an active high transmit ready output in the host 16550 DMA TX FIFO write mode and the 16450 mode as a software option (HCR2 = 1). Non-operational in the GP host mode or when serial mode is selected (SM7 = 1 or SM4 = 1). Operational only in 16550 mode when serial mode is not selected (SM7 = 0 and SM4 = 0).
RXD (PA6)	38	O	**RXD Serial Output.** PA6 is connected as the USART serial output stream when RXD ON is selected (SM7 = 1). When loopback is selected (SF5 = 1) the PA2 signal is internally connected to the PA6 output. PA6 is also modified by the serial output pass through mode (SF6 = 1).
RXRDY (PA6)	38	O	**Receive Ready.** PA6 is an active high transmit ready output in the host 16550 DMA RX FIFO read mode and the 16450 mode as a software option (HCR2 = 1). Non-operational in the GP host mode or when serial mode is selected (SM7 = 1 or SM4 = 1). Operational only in 16550 mode when serial mode is not selected (SM7 = 0 and SM4 = 0).

Table 2-2. MCU Interface Signal Description (Cont'd)

Symbol	Pin	I/O	Name/Function
RXCLK (PA7)	37	I/O	**RXD Serial Output Clock.** PA7 output is internally connected to the USART in the synchronous mode (SM7 = 1 and SM4 = 1). RXCLK can be an external input signal or can be generated internally and become an output signal. Serial out is clocked on the negative edge of RXCLK.
Falling Edge (PA7)	37	I/O	**PA7 Negative Edge Detect.** The PA7 edge detect flag (SI5) is set on each detected edge and is cleared by writing a zero to register bit CI4. When the PA7 edge detect flag is asserted (SI5 = 1) it will generate an IRQ6 interrupt if its enable is set (SI3 = 1).
PB0-PB7	73-80	O	**Port B.** General Purpose 8-bit Output-Only Port. All pins can be assigned special functions under software control. The Port B Selection Register (PBS) must be set appropriately for all pins in both general and special use. All Port B outputs float during reset active low.
ES4 (PB1)	74	O	**ES4.** When PBS1 = 0, the PB1 pin becomes the ES4 chip select signal. ES4 is an active low output from addresses $0600 to $07FF.
ES3 (PB2)	75	O	**ES3.** When PBS2 = 0, the PB2 pin becomes the ES3 chip select signal. ES3 is an active low output controlled by bit 7 of the selected or active (i) Banking RAM register (BRi7).
ES2 (PB3)	76	O	**ES2.** When PBS3 = 0, the PB3 pin becomes the ES2 chip select signal. ES2 is an active low output controlled by bit 6 of the selected or active (i) Banking RAM register (BRi6). PB3 is a chip select option and may not be wire bonded out in some packages.
ES0 (PB5)	78	O	**ES0.** When PBS5 = 0, the PB5 pin becomes the ES0 chip select signal. ES0 is an active low output controlled by bit 4 of the selected or active (i) Banking RAM register (BRi4).
HDIS (PB6)	79	O	**Host Bus Driver Disable.** Active high output asserted when the host is selected (HCR2 = 1) and is reading data from the host data bus. Used to control an external transceiver driver (PBS6 = 0).
HINT (PB7)	80	O	**Host Bus Interrupt.** Active high output asserted in both the general purpose (GP) host interface (HC2 = 1 and HC1 = 0) or the 16450 host interface (HC2 = 1 and HC1 = 1) under various conditions (PBS7 = 0).
PC0-PC7	13-20	I/O	**Port C.** General Purpose 8-bit I/O Port. All pins can be assigned in a group to host bus data lines by a software option. The Port C Direction Register (PCD) must be set appropriately for all pins.
HD0-HD7 (PC0-PC7)	13-20	I/O	**Host Bus Data Lines.** PC0-PC7 are dedicated to the host bus interface bidirectional data lines HD0-HD7, respectively, as a software option (HC2 = 1). The Port C direction register must be set to the output mode when the host bus software option is selected.
PD0-PD7	21-28	I/O	**Port D.** General Purpose 8-bit Port. PD0-PD3 are general purpose I/O pins while PD4- PD7 are input only. PD0-PD3 output latch is controlled by writing bits 0-3 at address $0003, while their direction control is contained in bits 4-7 respectively. PD0 - PD6 can be assigned in a group to host bus address and control lines by a software option.
HA0-HA3 (PD0-PD3)	21-24	I	**Host Bus Address Lines.** PD0-PD3 are dedicated to the host bus interface as address lines HA0-HA3, respectively, as a software option (HC2 = 1, HC1 = 0). PD3 remains a GP pin in the 16450/16550 mode (HC2 = 1, HC1 = 1).
HCSP (PD4)	25	I	**Host Bus Chip Select.** PD4 is dedicated to the host bus interface as the active low chip select input as a software option (HC2 = 1).
HWTP (PD5)	26	I	**Host Bus Write.** PD5 is dedicated to the host bus interface as an active low write control input as a software option (HC2 = 1).
HRDP (PD6)	27	I	**Host Bus Read.** PD6 is dedicated to the host bus interface as an active low read control input as a software option (HC2 = 1).
Rising or Falling Edge (PD7)	28	I/O	**PD7 Positive/Negative Edge Detect.** The PD7 edge detect flag (EI5) is set on each detected edge and is cleared by writing a zero to register bit CI5. When the PD7 edge detect flag is asserted (EI5 = 1) it will generate an IRQ1 interrupt if its enable is set (EI2 = 1). When EI0 = 0 negative edges are detected and when EI0 = 1 positive edges are sensed.
PE0-PE1, PE2-PE3, PE4, PE7	70-71, 5-6, 30, 50	I/O	**Port E.** General Purpose I/O Port. The Port E Direction Register (PED) must be set appropriately for all pins.

Table 2-2. MCU Interface Signal Description (Cont'd)

Symbol	Pin	I/O	Name/Function
Expansion Memory Bus			
A0-A12	54-66	O	**A0-A12.** Address lines for external memory bus. These lines float during reset active.
A13-A15	67-69	O	**A13-A15.** Address lines for external memory bus. These lines float during reset active. Address lines A13 - A15 are generated by bits 0 - 2, respectively, of the active or selected Banking RAM register.
A16 (PB0)	73	O	**A16.** When PBS0 = 0, the PB0 pin becomes the A16 address function from the selected or active (i) Banking RAM register bit 4 (BRi4). (See PB0.)
A17 (PB4)	77	O	**A17.** When PBS4 = 0, the PB4 pin becomes the A17 address function from the selected or active (i) Banking RAM register (BRi5). (See PB4.)
D0-D7	41-48	I/O	**Expansion Bus Data.** D0-D7 provide an 8-bit bi-directional data bus for interfacing to external memories. D0-D7 float during reset active. The receivers are automatically disabled for low power during low power operation.
WTP	3	O	**Expansion Bus Write.** Active low write strobe for the expansion bus. When WTP is asserted low data on the expansion data bus is to be written into the selected peripheral device. WTP remains high during controller reset active.
RDP	4	O	**Expansion Bus Read.** Active low read strobe for the expansion bus. When RDP is asserted low the selected peripheral device is to read external data onto the expansion data bus. RDP remains high during controller reset active.
Wake-Up Operation			
LPWU NMIP	2	I	**Low Power Wake Up - NMIP.** When the controller is operating in the low power mode, a low level on the NMIP pin will cause normal operation to resume.
LPWU Ring (PA0)	32	I	**Low Power Wake Up - Ring.** When the controller is operating in the low power mode, a high level on PA0 will cause normal operation to resume if the PA0 enable term is set (LPR5 = 1).
LPWU HBW (PD4 + PD5)P	25 + 26	I	**Low Power Wake Up - Host Bus Write.** When the controller is operating in the low power mode, a low level on both PD4 and PD5 will cause normal operation to resume if the host write enable term is set (LPR4 = 1).
LPWU TXD (PA2)P	34	I	**Low Power Wake Up - USART Start Bit.** When the controller is operating in the low power mode, a low level on PA2 will cause normal operation to resume if the USART start bit enable term is set (LPR3 = 1).
LPWU DTR (PD4)P	25	I	**Low Power Wake Up - DTR.** When the controller is operating in the low power mode a low level on PD4 will cause normal operation to resume if the host write enable term is set (LPR2 = 1).
LPWU AL (PD5)P	26	I	**Low Power Wake Up - AL.** When the controller is operating in the low power mode a low level on PD5 will cause normal operation to resume if the host write enable term is set (LPR1 = 1).

Table 2-2. MCU Interface Signal Description (Cont'd)

Symbol	Pin	I/O	Name/Function
System Signals			
RESP	1	I	**Reset.** Active low input resets all internal circuits and registers to their initial state. Controller operation resumes when the RESP input goes high; program execution starts at the location defined by the reset address vector. RESP must be held low for at least 10 cycles or until the crystal oscillator has stabilized.
NMIP	2	I	**Non-Maskable Interrupt.** Non-Maskable negative edge sensitive interrupt input with an internal pull-up. Interrupts program execution upon completion of current instruction and jumps to interrupt service subroutine starting at location defined by the NMI address vector.
TSTP	40	I	**Emulator Mode.** TSTP is an input to the controller device. This pin has an internal pull-up. TSTP when asserted high places the controller device in a standard operating mode. 1. Internal ROM is enabled. 2. RDP and WTP reflect CPU activity. 3. Internal CPU read activity on pages 0, 1, 2, 3, 4 or 5 (internal RAM and registers) do not appear on the expansion bus. TSTP when asserted low places the controller device in a special emulation mode. 1. Internal ROM is disabled and all ROM fetches are transferred to the expansion bus. 2. Internal bus activity on pages 0, 1, 2, 3, 4 or 5 impact D0-D7. An internal read from pages 0-5 cause the internal data to appear on D0-D7.
PARIF	49	I	**Parallel Host Interface.** This pin selects either the parallel interface mode (high or float) or serial DTE Interface Mode (low or GND). Internal pullup.
CLKOUT	31	O	**Clock Output.** CLKOUT is a clock output of the same frequency as the crystal input. It is turned off in low power STOP mode. It is not affected by extended cycle operation or by DMA cycle stealing.
VD3	11	O	**Regulated +3.3V Output Power.** This voltage source and its regulator are to be used for powering an external Conexant +3.3V Modem Data Pump (MDP). There is no internal use of this +3.3V supply.
XTLI	9	I	**Crystal Oscillator Input Pin.** Input connection from crystal or external clock circuit.
XTLO	10	O	**Crystal Return.** Oscillator output connection to an external crystal.
PSC	7	I	**Power Supply Control.** Leave open.
VDD	8, 51	I	**Power. +5 VDC.**
GND	12, 29, 52-53, 72	I	**Ground.** Signal and power ground.

2.2 SPECIAL CONSIDERATIONS

2.2.1 Address Map - PA[6:3] RS232/16550 Signal Interference

Interference between the RS232 and 16550 modes occur on PA3, PA4, PA5, and PA6. These four pins are used to support the parallel 16550 DMA request and acknowledge signals. In the MCU, the conflict is resolved by defeating the 16550 DMA request and acknowledge signals whenever the RS232 mode is simultaneously active. DMA is only supported in the host 16550 mode when the serial RS232 mode is deselected.

Whenever the RS232 mode is active (SM6 or SM7 = 1), PA3, and PA4 will not support the 16550 DMA HTACT and HRACT acknowledge signals. However, these two pins can be used for TXREF (PA4) or TXCLK (PA3) or as general purpose I/O pins when TXREF or TXCLK are not required.

If PAD5 =1 (PA5 output driver active), and the 16550 mode is active, PA5 will output the 16550 DMA TXRDY output signal. If PAD5 = 0 (Hi Z mode), PA5 can be used as a RS232 REXD input signal or as a general purpose input only pin.

Whenever the RS232 mode is active (SM7 = 1), the 16550 DMA RXRDY output signal is not supported and PA6 will only output the RXD serial signal.

2.2.2 Parallel Host/Serial DTE Interface Mode Operation

The L28 can operate in either parallel host or serial DTE interface mode using the same masked code. The interface mode is selected by the level on pin 49 (PARIF): open or high = Parallel Host, GND or low = Serial DTE.

3 SYSTEM ARCHITECTURE

3.1 OVERVIEW AND MEMORY MAPS

3.1.1 Block Diagram

A block diagram of the MCU is shown in Figure 3-1.

3.1.2 Top Level Memory Map

The top level memory map in Figure 3-2 shows major boundaries between I/O, internal ROM, internal RAM, and external addresses. The breakdown of memory in 8K-byte blocks is shown in Figure 3-3. The MCU Memory Map Overview for addresses 0 - 07FFh is shown in Figure 3-4. The individual registers are identified in Table 3-1.

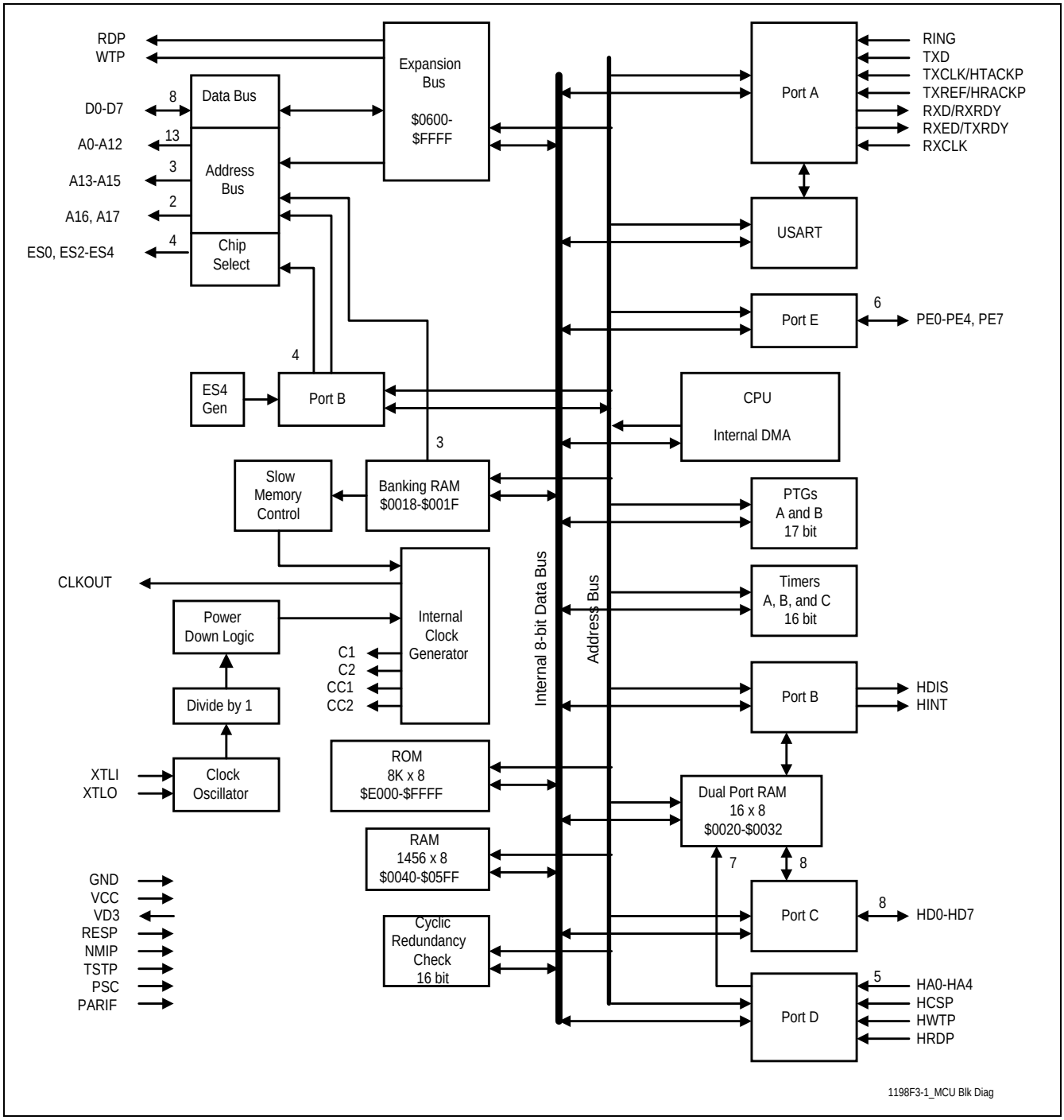


Figure 3-1. MCU Block Diagram

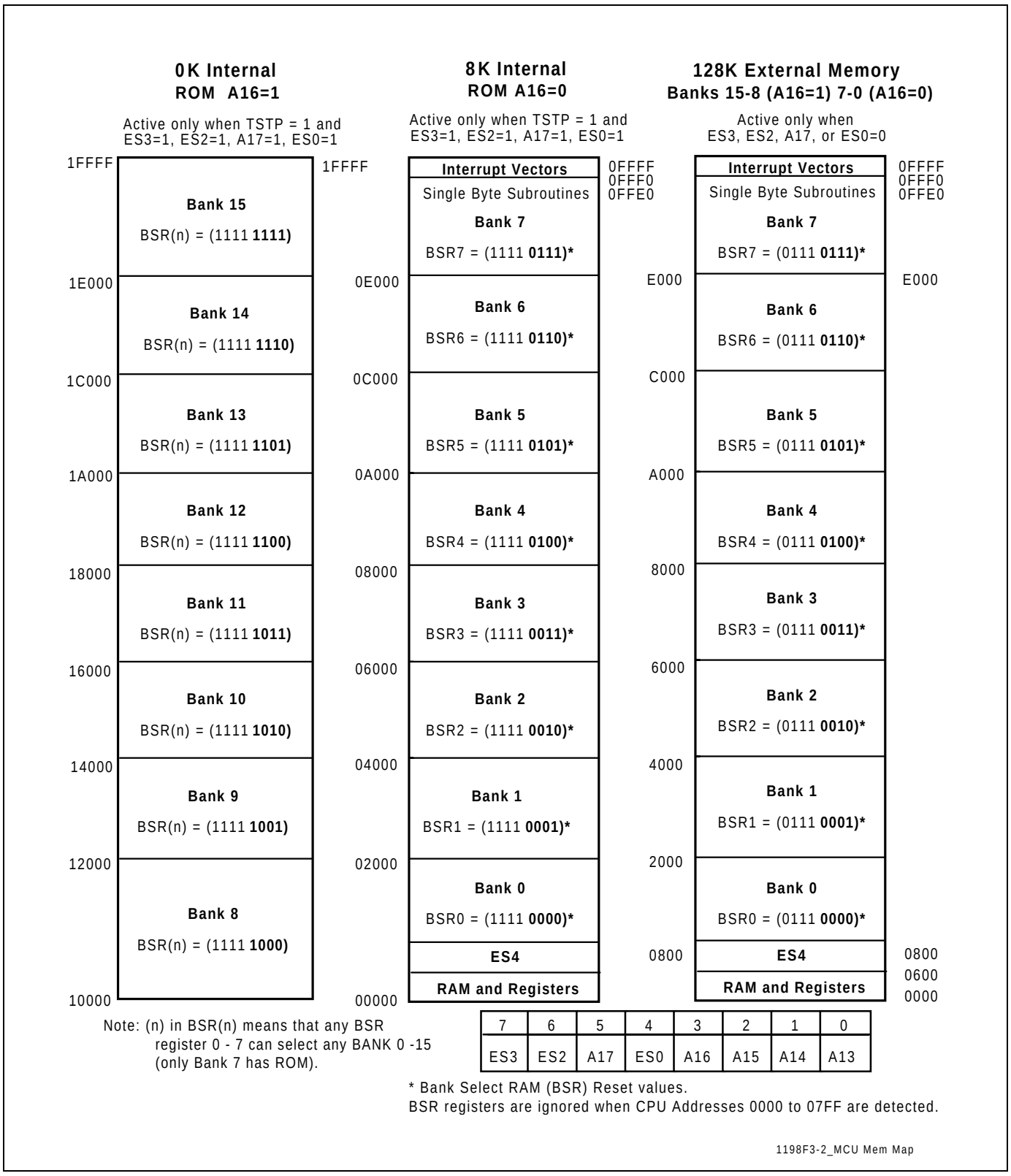


Figure 3-2. MCU Memory Map Overview

Bank Select Register	Address (Hex)	
BSR7 (8K)	FFE0 - FFFF	Interrupt/JSR Vectors (32)
	FFDF	
	F000	Internal ROM (8K-32), TSTP High
	E000	Default ESS Expansion, TSTP Low
BSR6 (8K)	DFFF	
	D000	Default
	C000	ES3 Expansion (32K)
BSR5 (8K)	BFFF	
	B000	
	A000	
BSR4 (8K)	9FFF	
	9000	
	8000	
BSR3 (8K)	7FFF	
	7000	Default
	6000	ES1 Expansion (16K)
BSR2 (8K)	5FFF	
	5000	
	4000	
BSR1 (8K)	3FFF	Default
	3000	ES1 Expansion (8K)
	2000	
BSR0 (8K)	17FF	Default
	1000	ES0 Expansion (6K)
	0800	
	0600 - 07FF	ES4 (512)
	0000 - 05FF	RAM and Registers (1536)

Figure 3-3. MCU Memory Map Breakdown by 8K-byte Blocks

Addr. (Hex)	Function
07FF 0700	Page 7 (256) Available Externally (Part of ES4 when PBS1 = 0)
06FF 0600	Page 6 (256) Available Externally (Part of ES4 when PBS1 = 0)
05FE - 05FF	CRC Control (2)
05F4-05FD	Reserved (10)
05F0-05F3	Timer C (Snooze Timer) (4)
05EF 0500	Page 5 RAM (240)
04FF 0400	Page 4 RAM (256)
03FF 0300	Page 3 RAM (256)
02FF 0200	Page 2 RAM (256)
01FF 0100	Page 1 RAM (256)
00FF 0040	Page 0 RAM (192)
0038 - 003F	USART Registers (8)
0034 - 0037	Precision Time Generator A (4)
0033	ES Speed Select
0020 - 0032	16550 I/F & DP RAM (19)
0018 - 00FF	Bank Select Registers (8)
0010 - 0017	Timer/Counters A &B (8)
000C - 000F	Precision Time Generator B (4)
0000 - 000B	Ports A to E and Misc. Registers (12)

Figure 3-4. MCU Memory Map: 0-07FFh

Table 3-1. MCU Memory Map: 0-7FFh

Address (Hex.)	Read	Write
0000	Port A	Port A
0001	--	Port B (Output Only)
0002	Port C	Port C
0003	Port D	Port D (0-3), Direction (4-7)
0004	--	Port A Direction
0005	Port B Select	Port B Select
0006	--	Port C Direction
0007	Port E	Port E
0008	Mask Option Register	Port E Direction
0009	Low Power Register (LPR)	Low Power Register (LPR)
000A	External Interrupt Register (EIR)	External Interrupt Register (EIR)
000B	Clear External Interrupt	Clear External Interrupt
000C	PTG B Mode (PBM)	PTG B Mode (PBM)
000D	PBB	PBB
000E	PBUL	PBUL, PBB to PBLL
000F	PBUL, Clear Flag	PBUL,PBB to PBLL, Download, Clear Flag
0010	Timer A Mode (TAM)	Timer A Mode (TAM)
0011	TALC, TAUC to TAS	TALL
0012	TAS	TAUL
0013	TAS, Clear Flag	TAUL, Download, Clear Flag
0014	Timer B Mode (TBM)	Timer B Mode (TBM)
0015	TBLC, TBUC to TBS	TBLL
0016	TBS	TBUL
0017	TBS, Clear Flag	TBUL, Download, Clear Flag
0018	Bank 0	Bank 0
0019	Bank 1	Bank 1
001A	Bank 2	Bank 2
001B	Bank 3	Bank 3
001C	Bank 4	Bank 4
001D	Bank 5	Bank 5
001E	Bank 6	Bank 6
001F	Bank 7	Bank 7

Table 3-1. MCU Memory Map: 0-7FFh (Cont'd)

Address (Hex)	Read	Write
Host General Purpose Mode Only		
0020	TX FIFO Buffer	RX FIFO Buffer
0021	SP RAM 1	SP RAM 1
0022	SP RAM 2	SP RAM 2
0023	SP RAM 3	SP RAM 3
0024	SP RAM 4	SP RAM 4
0025	SP RAM 5	SP RAM 5
0026	SP RAM 6	SP RAM 6
0027	SP RAM 7	SP RAM 7
0028	SP RAM 8	SP RAM 8
0029	SP RAM 9	SP RAM 9
002A	SP RAM A	SP RAM A
002B	SP RAM B	SP RAM B
002C	SP RAM C	SP RAM C
002D	SP RAM D	SP RAM D
002E	GP FIFO Status (GPFS)	--
002F	Host Handshake Register	Host Handshake Register
0030	Transmitter FIFO Status Register	Transmitter FIFO Status Register
0031	Receiver FIFO Status Register	Receiver FIFO Status Register
0032	Host Control Register (HCR)	Host Control Register (HCR)
Host 16550 Emulation Mode Only		
0020	TX FIFO Buffer	RX FIFO Buffer
0021	Line Status Register (LSR)	LSR1 Only
0022	Modem Status Register MSR)	Modem Status Register (MSR)
0023	Line Control Register (LCR)	Line Control Register (LCR)
0024	Modem Control Register (MCR)	--
0025	FIFO Control Register (FCR)	--
0026	SP RAM 6	SP RAM 6
0027	SP RAM 7	SP RAM 7
0028	Divisor Latch LSB	Divisor Latch LSB
0029	Divisor Latch MSB	Divisor Latch MSB
002A	SP RAM A	SP RAM A
002B	SP RAM B	SP RAM B
002C	SP RAM C	SP RAM C
002D	SP RAM D	SP RAM D
002E	GP FIFO Status (GPFS)	GPFS3 Only (TuClk Off)
002F	Host Handshake Register	Host Handshake Register
0030	FIFO Status Register (FSR)	FIFO Status Register (FSR)
0031	FIFO Interrupt Enable Register (FIER)	FIFO Interrupt Enable Register (FIER)
0032	Host Control Register (HCR)	Host Control Register (HCR)
Common		
0033	Chip Select Fast/Slow	Chip Select Fast/Slow
0034	PTG A Mode (PAM)	PTG A Mode (PAM)
0035	PAB	PAB
0036	PAUL	PAUL, PAB to PALL
0037	PAUL, Clear Flag	PAUL,PAB to PALL, Download, Clear Flag
0038	Serial In Buffer (SIB)	Serial Out Buffer (SOB)
0039	Serial Interrupt Enable (SIR)	Serial Interrupt Enable (SIR)
003A	Serial Mode Register (SMR)	Serial Mode Register (SMR)
003B	Serial Line Control Register (SLCR)	Serial Line Control Register (SLCR)
003C	Serial Status Register (SSR)	Serial Status Register (SSR)
003D	Serial Form Register (SFR)	Serial Form Register (SFR)
003E	--	SOUT (RXD) Divider Latch (SODL)
003F	--	SIN (TXD) Divider Latch (SIDL)

Table 3-1. MCU Memory Map: 0-7FFh (Cont'd)

Address (Hex.)	Read	Write
0040 . 00FF	Page 0 RAM (192 bytes)	
0100 . 01FF	Page 1 RAM (256 bytes)	
0200 . 02FF	Page 2 RAM (256 bytes)	
0300 . 03FF	Page 3 RAM (256 bytes)	
0400 . 04FF	Page 4 RAM (256 bytes)	
0500 . 05CF	Page 5 RAM (208 bytes)	
05D0 . 05EF	Page 5 RAM (32 bytes) when DSR0 = 0 DMA Register File when DSR0 = 1	
05F0	Timer CC Mode (TCM)	Timer CC Mode (TCM)
05F1	TCLC, TCUC TO TCS	TCLL
05F2	TCS	TCUL
05F3	TCS	TCUL, DNLD, CLEAR, FLAG
05F4	DMA Get Register (DGR)	DMA Get Register (DGR)
05F5	DMA Flag Register (DFR)	DMA Flag Register (DFR)
05F6	DMA Status Register (DSR)	DMA Status Register (DSR)
05F7 . 05FD	Reserved (7 bytes)	
05FE	CRC-L	CRC Input Buffer
05FF	CRC-H	Initialize CRC
0600 . 07FF	ES4 Active (PBS1 = 0) (512 bytes)	ES4 Active (PBS1 = 0) (512 bytes)
0600 . 07FF	ES4 Inactive (PBS1 = 1) (512 bytes mapped to internal ROM: FE00 - FFFF)	ES4 Active (PBS1 = 1) (Reserved)

3.1.3 I/O Register Bit Assignments

The individual I/O signal and data bits are identified in Table 3-2.

Table 3-2. Register Bit Assignments: 0-05FFh

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0000	Port A Data (0-7; I/O)	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
0001	Port B Data (0-7; Output Only)	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0002	Port C Data (0-7; I/O)	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
0003	Port D Data (0-2; I/O), Port D Direction (4-6); Port D Data (3-4; O): Port D Data (5-7; I)	PD7 (I)	PD6 (I); PDD2	PD5 (I); PDD1	PD4 (O); PDD0	PD3 (O)	PD2 (I/O)	PD1 (I/O)	PD0 (I/O)
0004	Port A Direction (0-7)	PAD7	PAD6	PAD5	PAD4	PAD3	PAD2	PAD1	PAD0
0005	Port B Select (0-7)	**PB7/HINT Sel** 1 = PB7 0 = HINT	**PB6/HDIS** 1 = PB6 0 = HDIS	**PB5/ES0 Sel** 1 = PB5 0 = ES0	**PB4/A17 Sel** 1 = PB4 0 = A17	**PB3/ES2 Sel** 1 = PB3 0 = ES2	**PB2/ES3 Sel** 1 = PB2 0 = ES3	**PB1/ES4 Sel** 1 = PB1 0 = ES4	**PB0/A16 Sel** 1 = PB0 0 = A16
0006	Port C Direction (0-7)	PCD7	PCD6	PCD5	PCD4	PCD3	PCD2	PCD1	PCD0
0007	Port E Data (0-4, 7)	PE7	--	--	PE4	PE3	PE2	PE1	PE0
0008	Mask Option Register (MSR) (Read Only)	MOR7	MOR6	MOR5	MOR4	MOR3	MOR2	MOR1	MOR0
0008	Port E Direction (0-4, 7) (Write Only)	PED7	--	--	PED4	PED3	PED2	PED1	PED0
0009	Low Power Register (LPR)	Low Power Mode Enable 1 = Enable	Low Power Mode 0 = Sleep 1 = Stop	PA0 (Ring) Wake up Enable 1 = Enable	PD4 & PD5 (Host Wrt) Wake up Enable 1 = Enable	PA2 (TXD) Wake up Enable 1 = Enable	PD4 (DTR) Wake up Enable 1 = Enable	PD5 (AL) Wake up Enable 1 = Enable	Not used
0009 (Write Only)	Low Power Register (LPR) Write to TCM3 and TCM4	Bits Not Modified			If TCM = 1, Write LPR 7 to TCM4	If TCM = 1, Write LPR 6 to TCM3	Bits Not Modified		
000A	External Interrupt Register (EIR)	PA1 Ext. Int. Flag 1 = Flag	PA4 Ext. Int. Flag 1 = Flag	PD7 Ext. Int. Flag 1 = Flag	PA1 Ext. Int. Enable 1 = Enable	PA4 Ext. Int. Enable 1 = Enable	PD7 Ext. Int. Enable 1 = Enable	PA4 Edge Detect Polarity 1 = Rising ↑ 0 = Falling ↓	PD7 Edge Detect Polarity 1 = Rising ↑ 0 = Falling ↓
000B	Clear Interrupt Register (CIR)	PA1Clear Int. Flag	PA4 Clear Int. Flag	PD7 Clear Int. Flag	PA7 Clear Int. Flag	PA3 Clear Int. Flag	ES4 Fast Mem Cycle	Not Used	Not Used
000C	Precision Time Generator B (PTGB) Mode (PBM)	PTG B Int. Flag 1 = Flag	PTG B Int. Enable 1 = Enable	Not Used					PTG B Timer Mode 1 = Timer 0 = PTG
000D	PTGB Buffer (PBB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
000E	PTGB Upper Latch (PBUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
000F	PTGB Upper Latch (PBUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0010	Timer A Mode (TAM)	Timer A Int. Flag 1 = Flag	Timer A Int. Enable 1 = Enable	RAM Int. Vector Select 1 = RAM 0 = ROM	Not Used		Timer A Div by 32 Prescale 1 = Div by 32	Timer A Mode 00 = Interval Timer 01 = Pulse Generator 10 = Event Counter 11 = Pulse Width Measurement	
0011	Timer A Lower Latch	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0012	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0013	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0014	Timer B Mode (TBM)	Timer B Int. Flag 1 = Flag	Timer B Int. Enable 1 = Enable	RAM Int. Vector Select 1 = RAM 0 = ROM	Not Used		Timer B Div by 32 Prescale 1 = Div by 32	Timer B Mode 00 = Interval Timer 01 = Pulse Generator 10 = Event Counter 11 = Pulse Width Measurement	
0015	Timer B Latch Low	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0016	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0017	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Table 3-2. Register Bit Assignments: 0-05FFh (Cont'd)

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0018	Bank Select Register 0 (BSR0)	ES3 (0)	ES2 (0)	A17 (0)	ES0 (0)	A16 (0)	A15 (0)	A14 (0)	A13 (0)
0019	Bank Select Register 1 (BSR1)	ES3 (1)	ES2 (1)	A17 (1)	ES0 (1)	A16 (1)	A15 (1)	A14 (1)	A13 (1)
001A	Bank Select Register 2 (BSR2)	ES3 (2)	ES2 (2)	A17 (2)	ES0 (2)	A16 (2)	A15 (2)	A14 (2)	A13 (2)
001B	Bank Select Register 3 (BSR3)	ES3 (3)	ES2 (3)	A17 (3)	ES0 (3)	A16 (3)	A15 (3)	A14 (3)	A13 (3)
001C	Bank Select Register 4 (BSR4)	ES3 (4)	ES2 (4)	A17 (4)	ES0 (4)	A16 (4)	A15 (4)	A14 (4)	A13 (4)
001D	Bank Select Register 5 (BSR5)	ES3 (5)	ES2 (5)	A17 (5)	ES0 (5)	A16 (5)	A15 (5)	A14 (5)	A13 (5)
001E	Bank Select Register 6 (BSR6)	ES3 (6)	ES2 (6)	A17 (6)	ES0 (6)	A16 (6)	A15 (6)	A14 (6)	A13 (6)
001F	Bank Select Register 7 (BSR7)	ES3 (7)	ES2 (7)	A17 (7)	ES0 (7)	A16 (7)	A15 (7)	A14 (7)	A13 (7)

Table 3-2. Register Bit Assignments: 0-05FFh (Cont'd)

Addr.	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0020	RX FIFO Buffer (Host Read/MCU Write)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0020	TX FIFO Buffer (Host Write/MCU Read)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0021	Line Status Register (LSR)	RX FIFO Error 1 = Error	XMTR Empty (TEMT) 1 = Empty	XMTR Holding Register Empty (THRE) 1 = Empty	Break Interrupt (BI)	Framing Error (FE) 1 = Error	Parity Error (PE) 1 = Error	Overrun Error (OE) 1 = Error	RX Data Ready (DR) 1 = Error
0022	Modem Status Register (MSR)	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear to Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge of Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear-to Send (DCTS)
0023	Line Control Register (LCR)	DLAB	Set Break	Stick Parity	Even Parity	Parity Enable	Number Stop Bits	Word Length (WLS1)	Word Length (WLS0)
0024	Modem Control Register (MCR)	0	0	0	Loop	Out 2	Out 1	Request to Send (RTS)	Data Terminal Ready (DTR)
0025	FIFO Control Register (FCR)	RCVR Trigger MSB	RCVR Trigger LSB	Reserved	Reserved	DMA Mode Select	TX FIFO Reset	RX FIFO Reset	FIFO Enable
Host Rd/Wt @ x1	Interrupt Enable Register (IER)	0	0	0	0	Modem Status Int. Enable (EDSSI)	RX Line Status Int. Enable (ELSI)	TX Hold Empty Int. Enable (ETBEI)	RX Data Avail Int. Enable (ERBFI)
Host Read @ x2	Interrupt Identifier Register (IIR)	FIFO Enable (FCR0)	FIFO Enable (FCR0)	0	0	Interrupt ID Bit 2	Interrupt ID Bit 1	Interrupt ID Bit 0	0 if Interrupt Pending
0027	Scratch Register (SCR)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0028	Divisor Latch LSB	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0029	Divisor Latch MSB	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
002E	GP FIFO Status (GPFS)	Tx Empty Int. Enable 1 = Enable	Tx Empty Int. Flag 1 = Flag	Tx FIFO Half Empty 1 =	Rx Trig Level Select (RTL1)	Rx Trig Level Select (RTL0) [GP Mode] TuClk Off [16550]	Rx Trig Level Int. Enable 1 = Enable	Rx Trig Level Int. Flag 1 = Flag	RCNE [GP Mode] Rx FIFO Data Avail [16450/ 16550]
002F	Host Handshake Register (HHR)	Not supported							
0030	FIFO Status Register (FSR)	TX FIFO Half Full (TCHF)	TX FIFO Data Avail (TCDA)	RX FIFO Freeze	RX FIFO Break Interrupt (BI)	RX FIFO Framing Error (FE)	RX FIFO Parity Error (PE)	RX FIFO Empty Flag (RCEMT)	RX FIFO Half Empty Flag (RCHE)
0031	FIFO Int. Enable Register (FIER)	TX FIFO Half Full Int. Enable (TCHFE)	TX FIFO Data Avail Int. Enable (TCDAE)	UART Timing Select	RUCLK Off	TX FIFO Reset	RX FIFO Reset	RX FIFO Empty Int. Enable (RCEMTE)	RX FIFO Half Empty Int. Enable (RCHEE)
0032	Host Control Register (HCR)	TX FIFO Interrupt	MCR Write Flag	LCR Write Flag	Divisor Latch Write Flag	RX FIFO Interrupt	Host Mode Select	16450/ 16550 Mode	16450/ 16550 Int. Enable

Table 3-2. Register Bit Assignments: 0-05FFh (Cont'd)

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0033	ES Speed (ESS)	ES3-1	ES3-0	ES2-1	ES2-0	ES1-1	ES1-0	ES0-1	ES0-0
0034	Precision Time Generator A (PTGA) Mode (PAM)	PTG A Int. Flag 1 = Flag	PTG A Int. Enable 1 = Enable	Not Used				SIN (TXD) Select 1 = PTGA 0 = PTGB	PTGA Timer Mode 1 = Timer 0 = PTG
0035	PTGA Buffer (PAB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0036	PTGA Upper Latch (PAUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0037	PTGA Upper Latch (PAUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0038	Serial In Buffer (SIB) / Serial Out Buffer (SOB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0039	Serial Interrupt Register (SIR)	TXD Status Int. Flag 1 = Flag	TXCLK (PA3) ↑ Int. Flag 1 = Flag	RXCLK (PA7) ↓ Int. Flag 1 = Flag	TXCLK (PA3) ↑ Int. Enable	RXCLK (PA7) ↓ Int. Enable	TXD Status Int. Enable 1 = Enable	RXD Buffer Empty Int. Enable (BE) 1 = Enable	TXD Buffer Full Int. Enable (BF) 1 = Enable
003A	Serial Mode Register (SMR)	RXD On	TXD On	Timing Select 1 = TIMB 0 = PRG	Sync Mode	TXD Sync Bit	TXREF Clock Select	CTSP/ RLSDP Sync Enable	Not Used
003B	Serial Line Control Register (SLC)	Parity Stuff Bit	Set Break	Stuff Parity	Even Parity	Enable Parity	Two Stop Bits	Word Length Bit 1 (SL1)	Word Length Bit 0 (SL0)
003C	Serial Status Register (SSR)	TXD Parity Bit	RXD Underrun (UR)	RXD Buffer Empty (BE)	TXD Break Int. (BI)	TXD Framing Error (FE)	TXD Parity Error (PE)	TXD Overrun Error (OE)	TXD Buffer Full (BF)
003D	Serial Form Register (SFR)	TXD/ TXDE Echo	REXD/TXD Echo	TXD/RXD Echo	TXD to TIMA Input Test	TXD to PA↑ Edge Test	Not used	7/8 Short Stop Bit	3/4 Short Stop Bit
003E	SOUT (RXD) Divider Latch (SODL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
003F	SIN (TXD) Divider Latch (SIDL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
05F0	Timer C Mode Control (TCM)	TIM C Underflow Flag 1 = Flag	TIM C Div by 217 Overflow Flag 1 = Flag	TIM C Div by 217 Snooze Enable 1 = Enable	Snooze Clock Mode 00 = Clock Off 01 = Int. Osc. C2 Rate 10 = Int. Osc. Div by 127 11 = Ring Osc. Rate		Not Used		
05F1	Timer C “Lower’ Latch	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
05F2	Timer C ‘Upper’ Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
05F3	Timer C ‘Upper’ Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
05D0 (DS0 =1)	DMA Enable Register (DER)	Tx CHAR3 Int. Enable 1 = Enable	Tx CHAR3 DFR (0) Enable 1 = Enable	Tx Hi-Lo Watermark Int. Enable 1 = Enable	Tx Hi-Lo Watermark Auto Insert	Tx CHAR1/2 Int. Enable 1 = Enable	Tx CHAR1/2 Save to Buffer	Mode 1 = RS-232 0 = 16550	Not used
05F5	DMA Flag Register (DFR)	Tx CHAR3 Int. Flag 1 = Flag	Tx Hi-Lo Watermark Int. Flag 1 = Flag	Tx CHAR1/2 Int. Flag 1 = Flag	DTE - Tx Channel Enable 1 = Enable	Rx - DTE Channel Enable 1 = Enable	CHAR3 Seq. Flag 3rd Detect 1 = Detect	CHAR3 Seq. Flag 2nd Detect 1 = Detect	CHAR3 Seq. Flag 1st Detect 1 = Detect
05F6	DMA Status Register (DSR)	Tx RAM Buffer Hi Watermark Status	Tx RAM Buffer Full 1 = Full	Rx RAM Buffer Exceed ROFF	Rx RAM Buffer Empty 1 = Empty	CHAR1(0) CHAR2(1) Detected	Tx_BF "GET" Status	Rx_BE "PUT" Status	RAM(0) DMA(1) Select
05FE	CRC Low Byte (Read only)	CRC 7	CRC 6	CRC 5	CRC 4	CRC 3	CRC 2	CRC 1	CRC 0
05FE	CRC Data In (Write only)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
05FF	CRC High Byte (Read only)	CRC 15	CRC 14	CRC 13	CRC 12	CRC 11	CRC 10	CRC 9	CRC 8
05FF	Initialize CRC (Write only)	A write to $05FF initializes the CRC to $FFFF.							

3.1.4 MCU Reset Initialization

MCU reset initialization values are listed in Table 3-3. Registers not shown are uninitialized by reset.

Table 3-3. MCU Reset Initialization Values

Address (Hex)	Function	Reset Value								Comments
		7	6	5	4	3	2	1	0	
0000	Port A[7:0]	Z	Z	Z	Z	Z	Z	Z	Z	Input Hi Z Mode
0001	Port B[7:0]	0	0	1	1	1	1	1	0	Output Only
0002	Port C[7:0]	Z	Z	Z	Z	Z	Z	Z	Z	Input Hi Z Mode
0003	Port D[7:0]	Z	Z	Z	Z	Z	Z	Z	Z	Input Hi Z Mode
0003	Port D Direction	0	0	0	0					Control bits [3:0]
0004	Port A Direction	0	0	0	0	0	0	0	0	
0005	Port B Select	0	0	0	0	0	0	0	0	
0006	Port C Direction	0	0	0	0	0	0	0	0	
0007	Port E[7, 4:0]	Z	-	-	Z	Z	Z	Z	Z	Input Hi Z Mode
0008	Port E Direction	0	-	-	0	0	0	0	0	
0009	Low Power Register	0	0	0	0	0	0	0	-	
000A	External Interrupt Register	-	-	-	0	0	0	0	0	
000B	Clear External Interrupt	-	-	-	-	-	0	-	-	
000C	Precision Timer B Mode	0	0	-	-	-	-	-	0	
0010	Timer A Mode	0	0	0	-	-	0	0	0	
0014	Timer B Mode	0	0	0	-	-	0	0	0	
0018	Bank 0 Dual Port RAM	1	1	1	0	0	0	0	0	TSTP = 0
0018	Bank 0 Dual Port RAM	1	1	1	1	0	0	0	0	TSTP = 1
0019	Bank 1 Dual Port RAM	1	1	0	1	0	0	0	1	TSTP = 0
0019	Bank 1 Dual Port RAM	1	1	1	1	0	0	0	1	TSTP = 1
001A	Bank 2 Dual Port RAM	1	0	1	1	0	0	1	0	TSTP = 0
001A	Bank 2 Dual Port RAM	1	1	1	1	0	0	1	0	TSTP = 1
001B	Bank 3 Dual Port RAM	1	0	1	1	0	0	1	1	TSTP = 0
001B	Bank 3 Dual Port RAM	1	1	1	1	0	0	1	1	TSTP = 1
001C	Bank 4 Dual Port RAM	0	1	1	1	0	1	0	0	TSTP = 0
001C	Bank 4 Dual Port RAM	1	1	1	1	0	1	0	0	TSTP = 1
001D	Bank 5 Dual Port RAM	0	1	1	1	0	1	0	1	TSTP = 0
001D	Bank 5 Dual Port RAM	1	1	1	1	0	1	0	1	TSTP = 1
001E	Bank 6 Dual Port RAM	0	1	1	1	0	1	1	0	TSTP = 0
001E	Bank 6 Dual Port RAM	1	1	1	1	0	1	1	0	TSTP = 1
001F	Bank 7 Dual Port RAM	0	1	1	1	0	1	1	1	TSTP = 0
001F	Bank 7 Dual Port RAM	1	1	1	1	0	1	1	1	TSTP = 1

Table 3-3. MCU Reset Initialization Values (Cont'd)

Address (Hex)	Function	Reset Value								Comments
		7	6	5	4	3	2	1	0	
0021	Host Line Status Register	0	1	1	0	0	0	0	0	LSR
0022	Host Modem Status	X	X	X	X	0	0	0	0	MSR
0024	Host Modem Control	0	0	0	0	0	0	0	0	MCR
0025	Host FIFO Control	0	0	-	-	0	0	0	0	FCR
-	Host Interrupt Enable	0	0	0	0	0	0	0	0	IER
-	Host Interrupt Identifier	0	0	0	0	0	0	0	1	IIR
002E	Host GP FIFO Status	0	1	1	0	0	0	0	0	GPFS
002F	Host Handshake Register	0	0	0	0	0	0	0	0	HHR
0030	Host FIFO Status	0	0	0	0	0	0	1	1	FSR
0031	Host FIFO Interrupt Enable	0	0	0	0	0	0	0	0	FIER
0032	Host Control Register	0	0	0	0	0	0	0	0	HCR
0033	ES Speed Control	0	0	0	0	0	0	0	0	ESS
0034	Precision Timer A Mode	0	0	-	-	-	-	-	0	PAM
0039	USART Interrupt Register	0	0	0	0	0	0	0	0	RS232 - SI
003A	USART Mode Control	0	0	0	0	0	0	0	-	RS232 - SM
003B	USART Line Control	0	0	0	0	0	0	0	0	RS232 - SL
003C	USART Status Register	0	1	1	0	0	0	0	0	RS232 - SS
003D	USART Form Register	0	0	0	0	0	0	0	0	RS232 - SF

Table 3-3. MCU Reset Initialization Values (Cont'd)

Address (Hex)	Function	Reset Value								Comments
		7	6	5	4	3	2	1	0	
05D0	ENAB	0	0	0	0	0	0	0	0	DMA
05D1	CHAR1	0	0	0	0	0	0	0	0	DMA
05D2	CHAR2	0	0	0	0	0	0	0	0	DMA
05D3	CHAR3	0	0	0	0	0	0	0	0	DMA
05D4	Low Water Mark	0	0	0	0	0	0	0	0	DMA
05D5	High Water Mark	0	0	0	0	0	0	0	0	DMA
05D6	DXM_L	0	0	0	0	0	0	0	0	DMA
05D7	DXM_H	0	0	0	0	0	0	0	0	DMA
05D8	DXB_L	0	0	0	0	0	0	0	0	DMA
05D9	DXB_H	0	0	0	0	0	0	0	0	DMA
05DA	DXZ_L	0	0	0	0	0	0	0	0	DMA
05DB	DXZ_H	0	0	0	0	0	0	0	0	DMA
05DC	RXI_L	0	0	0	0	0	0	0	0	DMA
05DD	DXI_H	0	0	0	0	0	0	0	0	DMA
05DE	DXO_L	0	0	0	0	0	0	0	0	DMA
05DF	DXO_H	0	0	0	0	0	0	0	0	DMA
05E5	ROFF Water Mark	0	0	0	0	0	0	0	0	DMA
05E6	DRM_L	0	0	0	0	0	0	0	0	DMA
05E7	DRM_H	0	0	0	0	0	0	0	0	DMA
05E8	DRB_L	0	0	0	0	0	0	0	0	DMA
05E9	DRB_H	0	0	0	0	0	0	0	0	DMA
05EA	DRZ_L	0	0	0	0	0	0	0	0	DMA
05EB	DRZ_H	0	0	0	0	0	0	0	0	DMA
05EC	RRI_L	0	0	0	0	0	0	0	0	DMA
05ED	DRI_H	0	0	0	0	0	0	0	0	DMA
05EE	DRO_L	0	0	0	0	0	0	0	0	DMA
05EF	DRO_H	0	0	0	0	0	0	0	0	DMA
05F0	Timer C Mode	0	0	0	0	0	0	-	-	Snooze Timer
05F4	DMA GET Register	0	0	0	0	0	0	0	0	DMA - Read only
05F4	DMA PUT Register	0	0	0	0	0	0	0	0	DMA - Write only
05F5	DMA Flag Register	0	0	0	0	0	0	0	0	DMA
05F6	DMA Status Register	0	0	0	1	0	0	1	0	DMA

3.2 CENTRAL PROCESSING UNIT (CPU)

The central processing unit (CPU) is an enhanced 8-bit 6502 CPU. The CPU executes stored instructions fetched from memory (usually external ROM) sequentially unless a jump to a new location is specified in the instruction or an interrupt occurs. Operation of the CPU instructions are described in Appendix A. The CPU is 6502 instruction compatible except "(indirect, X)" addressing mode changed to "(indirect)," and "(indirect), Y" changed to "(indirect), X".

The CPU registers are the same as the 6502 CPU with the addition of the W-register and I-register.

The data flow for the CPU registers is illustrated in Figure 3-5.

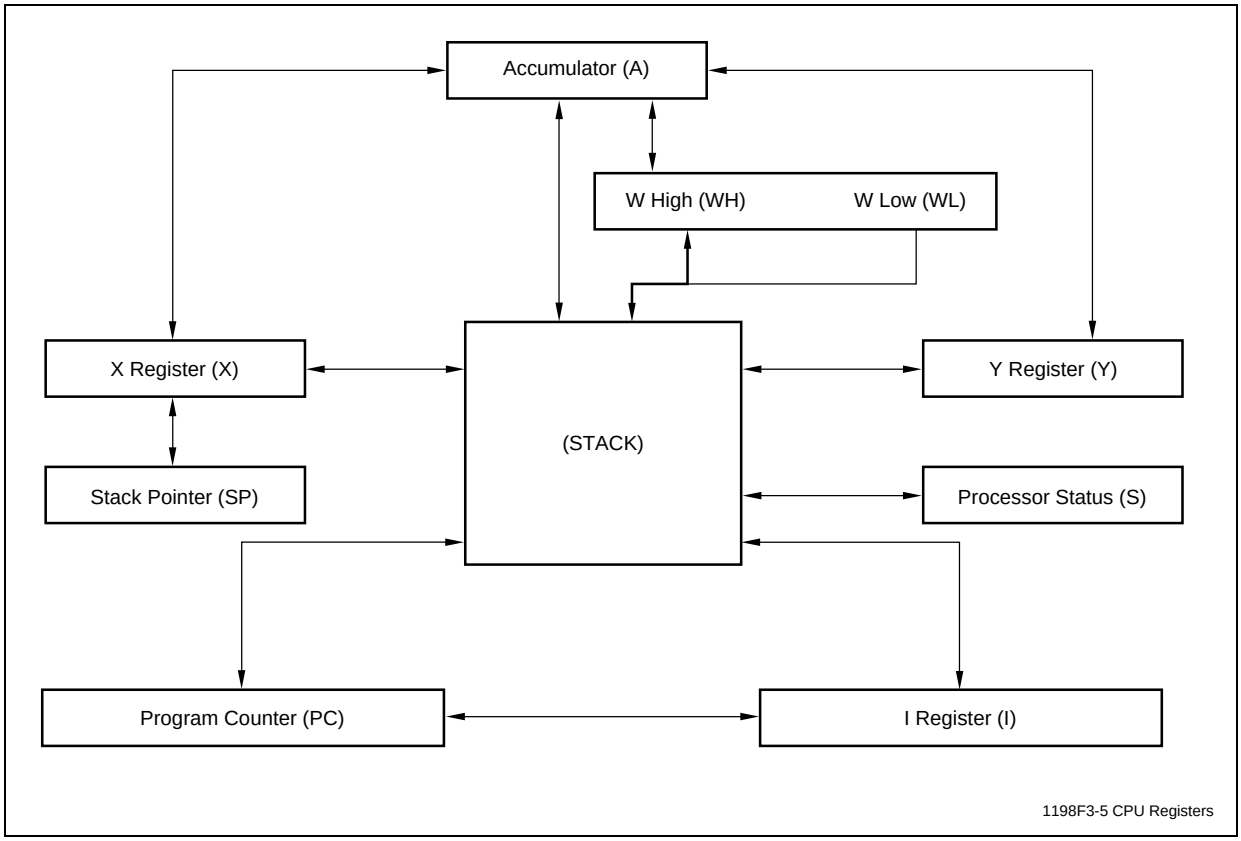


Figure 3-5. CPU Registers and Data Flow

3.2.1 Index Registers

There are two 8-bit index registers: X and Y. Either index register can be used as a base to modify the program counter contents and thus obtain a new address—the sum of the program counter contents and the index register contents. When executing an instruction which specifies indexed addressing, the CPU fetches the op code and the address, and modifies the address from memory by adding the index register to it prior to loading or storing the value of memory.

3.2.2 Stack Pointer

The Stack Pointer is an 8-bit register that controls access to the stack. The stack is initialized under software control usually to the top of Page 1 RAM. The stack length can be up to 256 bytes ($1FF down to $100).

The Stack Pointer is automatically incremented and decremented under control of the CPU to perform stack manipulation in response to program instructions, a reset, a non-maskable interrupt (NMI), an internally generated interrupt request (IRQ), or execution of the CPU Break (BRK) instruction. The Stack Pointer must be initialized by the user program. The JSR, JPI, PIA, BRK, RTI, and RTS instructions use the stack and the Stack Pointer.

3.2.3 Arithmetic and Logic Unit (ALU)

All arithmetic and logic operations take place in the ALU, including incrementing and decrementing internal registers (except the Program Counter). The ALU cannot store data for more than one cycle. If data is placed on the inputs to the ALU at the beginning of a cycle, the result is always gated into one of the storage registers or to memory during the next cycle.

Each bit of the ALU has two inputs. These inputs can be tied to various internal buses or to a logic 0; the ALU then generates the function (AND, OR, SUM, and so on) using the data on the two inputs.

3.2.4 Accumulator (A)

The Accumulator (A) is a general purpose 8-bit register that stores the results of most arithmetic and logic operations. In addition, the Accumulator usually contains one of the two data bytes used in these operations.

3.2.5 Program Counter (PC)

The 16-bit Program Counter (PC) provides the addresses that step the processor through sequential instructions in a program. Each time the processor fetches an instruction from the program memory, the least significant byte of the Program Counter (PCL) is placed on the eight low-order lines of the internal address bus and the most significant byte of the Program Counter (PCH) is placed on the eight high-order lines of the internal address bus. The Program Counter is incremented each time an instruction or data byte is fetched from program memory.

3.2.6 Instruction Register and Instruction Decode

Instructions are fetched from ROM or RAM and gated onto the internal data bus. These instructions are latched then decoded along with timing and interrupt signals to generate control signals for the various registers.

3.2.7 W Register (W)

The 16-bit W register is used exclusively to perform the accumulate function during execution of the Multiply Accumulate (MPA) instruction.

3.2.8 I Register (I)

The 16-bit I register is used for threaded code instructions. Note that the I register should not be confused with the CPU Instruction register which is not addressable.

3.2.9 Processor Status Register (PSR)

The 8-bit Processor Status Register (Table 3-4) contains seven status flags. Some of these flags are controlled by the user program; others may be controlled both by the user program and the CPU. The instruction set contains a number of conditional branch instructions which allow testing of these flags. Each of the seven processor status flags is described in the following paragraphs.

Table 3-4. Processor Status Register (PSR)

Function	Bit							
	7	6	5	4	3	2	1	0
Processor Status Register (PSR)	Negative (N)	Overflow (V)	Not Used	Break (B)	Decimal Mode (D)	IRQ Interrupt Disable (I)	Zero (Z)	Carry (C)

Bit 7: **Negative (N).** The Negative (N) bit copies the arithmetic sign bit value resulting from a data movement or an arithmetic operation. If the sign bit is set, the resulting values of the data movement or arithmetic operation is negative and the N bit is a logic 1; if the sign bit is cleared, the result of the data movement or arithmetic operation is positive and the N bit is a logic 0. There are no instructions that set or clear the N bit since the N bit represents only the status of a result. Instructions that affect the state of the N bit are: ADC, ADD, AND, ASL, ASR, BIT, CMP, CPX, CPY, DEC, DEX, DEY, EOR, INC, INX, INY, LAB, LAI, LAN, LDA, LDX, LDY, LSR, MPA, MPY, NEG, ORA, PIA, PLA, PLP, RND, ROL, ROR, RTI, SBC, TAW, TAX, TAY, TSX, TWA, TXA, and TYA.

1 = Negative value

0 = Positive value

Bit 6: **Overflow (V).** The Overflow (V) bit indicates that the result of a signed, binary addition or subtraction operation is a value that cannot be contained in seven bits (-128 • n • +127). The V bit only has meaning when signed arithmetic (sign and seven magnitude bits) is performed. When the ADC or SBC instruction is performed, the V bit is set to logic 1 if the polarity of the sign bit (bit 7) is changed because the result exceeds +127 or -128; otherwise the V bit is cleared to logic 0. The V bit may also be cleared under program control by the Clear Overflow (CLV) instruction. There is no instruction to set V.

The Overflow bit may also be used with the BIT instruction. The BIT instruction, which may be used to sample interface devices, allows the V bit to reflect the condition of bit 6 in the sampled field. During a BIT instruction, the V bit is set to equal to the content of bit 6 of the data tested with the BIT instruction. When used in this mode, the V bit has nothing to do with signed arithmetic, but is just another sense bit for the CPU. Instructions which affect the V flag are ADC, ADD, BIT, CLV, CLW, MPA, MPY, PLP, RND, RTI and SBC.

1 = Overflow set

0 = Overflow cleared

Bit 5: **Not Used.**

Bit 4: **Break (B).** The Break (B) bit indicates the condition which caused the IRQ service routine to be entered. If the IRQ service routine was entered because the CPU executed a BRK instruction, the B bit is set to logic 1. If the IRQ routine was entered as the result of an IRQ occurrence, the B bit is cleared to logic 0. There are no instructions which can set or clear this bit.

1 = BRK instruction

0 = No BRK instruction

Bit 3: **Decimal Mode (D).** The Decimal Mode (D) bit controls the arithmetic mode of the CPU. When the D bit is a logic 1, the adder operates as a decimal adder. When this bit is a logic 0, the adder operates as a straight binary adder. The adder mode is controlled only by two instructions. The Set Decimal Mode (SED) instruction sets the D bit; the Clear Decimal Mode (CLD) instruction clears it. The PLP and RTI instructions also affect the D bit. The D bit is cleared by assertion of RESP thus initially establishing binary mode.

1 = Decimal mode selected

0 = Binary mode selected

Bit 2: **IRQ Interrupt Disable (I).** The Interrupt Disable (I) bit controls the servicing of the internal interrupt request (IRQ). If the I bit is a logic 0, the IRQ will be serviced. If the bit is a logic 1, the IRQ will be ignored. The CPU sets the I bit to logic 1 if the external NMIP, external RESP, or the internal IRQ input (with the I bit set to a logic 0) is asserted. The I bit is restored by the Pull Processor Status from Stack (PLP) instruction, or as the result of executing a Return from Interrupt (RTI) instruction (provided the Interrupt Disable bit was cleared prior to the interrupt). The Interrupt Disable bit may be set or cleared under program control using a Set Interrupt Disable (SEI) or a Clear Interrupt Disable (CLI) instruction, respectively.

1 = IRQ interrupt disabled

0 = IRQ interrupt enabled

Bit 1: **Zero (Z).** The Zero (Z) bit is set to logic 1 by the CPU during any data movement or by any calculation which sets all eight bits of the result to zero. This bit is cleared to logic 0 when the resultant eight bits of a data movement or calculation operation are not all zero. The instruction set contains no instruction to specifically set or clear the Zero bit. The Z bit is, however, affected by the following instructions: ADC, ADD, AND, ASL, ASR, BIT, CMP, CPX, CPY, DEC, DEX, DEY, EOR, INC, INX, INY, LAB, LAI, LAN, LDA, LDX, LDY, LSR, NEG, ORA PIA,, PLA, PLP, ROL, ROR, RTI, SBC, TAW, TAX, TAY, TSX, TXA, TWA, and TYA.

1 = Zero result

0 = Non-zero result

Bit 0: **Carry (C).** The Carry (C) bit can be considered as the ninth bit of an arithmetic operation. It is set to logic 1 if a carry from the eighth bit has occurred, or cleared to logic 0 of no carry occurred, as the result of arithmetic operations. The Carry bit may be set or cleared under program control by use of the Set Carry (SEC) or Clear Carry (CLC) instruction, respectively. Other operations which affect the C bit are ADC, ADD, ASL, CMP, CPX, CPY, LSR, PLP, ROL, ROR, RTI and SBC.

1 = Carry set

0 = Carry cleared

3.2.10 CPU Interrupt Logic

CPU interrupt logic controls the sequencing of the RESP, NMIP, and IRQ activated interrupts and the CPU BRK instruction.

RES Sequencing

A low-to-high transition on RESP causes the Interrupt Disable (I) bit in the Processor Status Register to be set and program execution to begin at the address fetched from the RES vector ($FFFE and $FFFF).

NMI Sequencing

At the first operation code fetch following the high-to-low transition of the NMIP input, the interrupt logic forces execution of the Break (BRK) instruction and subsequent execution from the address vector stored at $FFFC and $FFFD. Simultaneous with the execution of the BRK instruction, the Interrupt Disable bit in the Processor Status Register is set to disable an IRQ.

IRQ Sequencing

An IRQ interrupt occurs when the Interrupt Disable (I) bit of the Process Status Register is cleared (0) and IRQ has been asserted from the IRQ Interrupt Logic. Upon IRQ interruption, the BRK instruction is forced and subsequent program execution begins at the IRQ interrupt service subroutine location specified by the IRQ interrupt vector corresponding to the IRQ number (1-6). The IRQ vector is located in one of six locations in ROM ($FFF0-$FFFB) or one of four locations in RAM ($0102-$0103, $0106-$0107, $0108-$0109, or $010A-$010B). The page 1 RAM IRQ vectors are Timer A and Timer B options, respectively.

The I bit is set to inhibit further IRQ interruption until completion of the IRQ interrupt service subroutine, at which time the I bit is automatically cleared by the RTI instruction. The I bit can also be cleared under program control with the CLI instruction.

For each IRQ that has multiple sources of interruption, the IRQ service subroutine must determine the source of the interrupt by examining applicable interrupts flags. The interrupt flag causing the IRQ should also be cleared after processing the interrupt and before returning to the interrupted routine.

3.3 CLOCK OSCILLATOR

The Clock Oscillator provides the basic timing signals used by the MCU internal circuits. The reference frequency can be supplied by either a parallel resonant crystal or a clock input. The input frequency is divided by 1 to generate the internal Ø2 clock. Typical Clock Oscillator input circuits are shown in Figure 3-6.

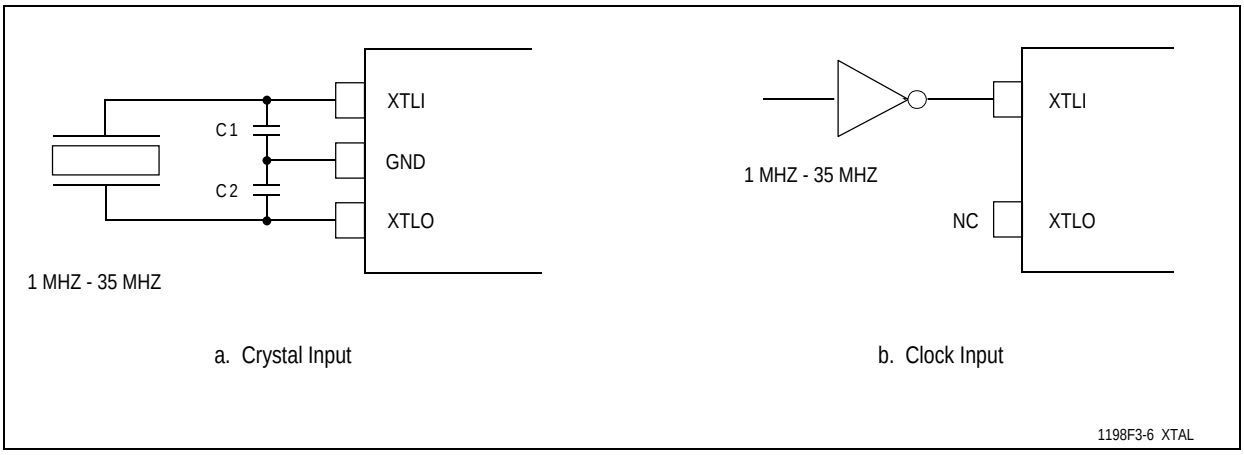


Figure 3-6. CPU Clock Oscillator Input Options

3.4 LOW POWER OPERATION

The MCU supports two low power modes (Sleep Mode and Stop Mode) with wake-up selectable from five externally detectable conditions.

In addition, a snooze timer is available that can be programmed to automatically wake up the MCU following activation of Sleep Mode or Stop Mode.

3.4.1 Sleep Mode

The Sleep Mode provides low power consumption during periods of inactivity, e.g., no host/modem interface activity and no telephone line activity, with ability to resume normal operation within 1 and 1/2 clock cycles of detection of a wake-up condition. In Sleep Mode, internal clocks are turned off but the internal oscillator is running to support resumption of normal operation within 1 and 1/2 clock cycles of detection of a wake-up condition.

The MCU enters Sleep Mode when Low Power Mode is enabled (LPR7 = 1) and Sleep Mode is selected (LPR6 = 0). LPR7 is reset to 0 by the MCU following wake-up.

3.4.2 Stop Mode

The Stop Mode allows the host to completely turn off MCU operation except for wake up capability during periods of no MCU activity as determined by the host, with ability to resume normal operation within several milliseconds of detection of a wake-up condition. In Stop Mode, internal clocks and the internal oscillator are turned off. Wake-up from the Stop Mode takes several milliseconds.

The MCU enters Stop Mode when Low Power Mode is enabled (LPR7 = 1) and Stop Mode is selected (LPR6 = 1). LPR7 is reset to 0 by the MCU following wake-up.

3.4.3 Wake-Up

The MCU will wake up and resume normal operation whenever an enabled wake-up condition occurs (see below) or the NMIP input is asserted. Wake-up from the Sleep Mode takes one and one-half clock cycles. Wake-up from the Stop Mode takes several milliseconds. An additional twelve cycles must elapse following wake-up before the MCU can be put back in the low power mode. Also, a high-to-low-to high transition on the NMIP input will cause the MCU to wake-up.

The wake-up conditions are:

LPR5 = 1	A high RING (PA0) signal occurs (RING high)
LPR4 = 1	The host writes to the device (HWTP low and HCSP low)
LPR3 = 1	A low TXD (PA2) signal occurs (TXD low)
LPR2 = 1	A high-to-low DTR (PD4) signal occurs (DTR high > low)
LPR1 = 1	A low AL (PD5) signal occurs (AL low)

3.4.4 Low Power Register (LPR)

The Low Power Register (LPR), located at $0009, controls the low power mode (LPM) enable, mode selection, and wake-up condition enables (Table 3-5). All bits are cleared to logic 0 upon reset. All bits can be written and read. However, neither bit 6 nor 7 can be read in the logic 1 state since the CPU will be in the Sleep or Stop mode. Figure 3-7 illustrates the MCU low power mode logic and timing.

Table 3-5. Register Bit Assignments: Low Power Register (LPR) - 0009h

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0009	Low Power Register (LPR)	Low Power Mode Enable 1 = Enable	Low Power Mode 0 = Sleep 1 = Stop	PA0 (Ring) Wake up Enable 1 = Enable	PD4 & PD5 (Host Wrt) Wake up Enable 1 = Enable	PA2 (TXD) Wake up Enable 1 = Enable	PD4 (DTR) Wake up Enable 1 = Enable	PD5 (AL) Wake up Enable 1 = Enable	Not used

Bit 0: **Not used.**

Bit 1: **Wake-Up Enable for PD5 (AL).** Control bit.

1 = Enables wake up from low power mode when a low is detected on PD5.

0 = Disables wake up from low power mode when a low is detected on PD5.

Bit 2: **Wake-Up Enable for PD4 (DTR).** Control bit.

1 = Enables wake up from low power mode when a low is detected on PD4.

0 = Disables wake up from low power mode when a low is detected on PD4.

Bit 3: **Wake-Up Enable for PA2 (TXD Start Bit).** Control bit.

1 = Enables wake up from low power mode when a low is detected on PA2.

0 = Disables wake up from low power mode when a low is detected on PA2.

Bit 4: **Wake-Up Enable for PD4 (MCU Chip Select) and PD5 (Host Bus Write).** Control bit.

1 = Enables wake up from low power mode when a low is detected on both PD4 and PD5.

0 = Disables wake up from low power mode when a low is detected on both PD4 and PD5.

Bit 5: **Wake Up Enable for PA0 (Ring).** Control bit.

1 = Enables wake up from low power mode when a high is detected on PA0.

0 = Disables wake up from low power mode when a high is detected on PA0.

Bit 6: **Low Power Mode.** Control bit.

1 = Invokes the Stop low power mode if low power mode is enabled (LPR7 = 1).

0 = Invokes the Sleep low power mode if low power mode is enabled (LPR7 = 1).

Bit 7: **Low Power Mode Enable.** Control bit.

1 = Enables the low power mode selected by LPR6 to be entered when low power conditions exist. LPR7 is cleared by low power mode wake up.

0 = Disables entry into the low power mode.

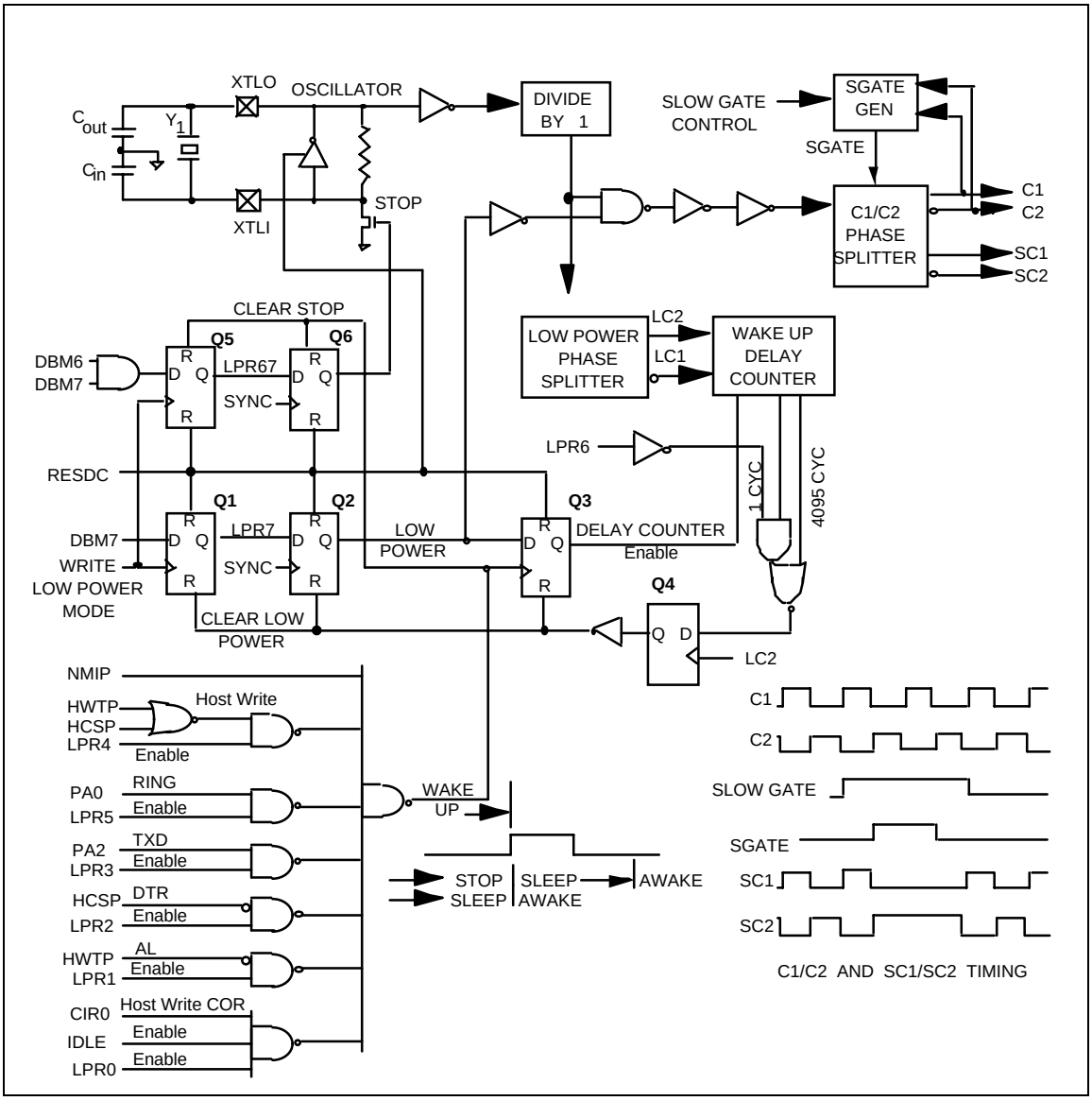


Figure 3-7. Low Power Mode Logic and Timing

3.4.5 Snooze Timer

The low power wake up Snooze Timer (Timer C) provides a means for the MCU to be automatically awakened from Sleep Mode or Stop Mode under timed control by the CPU. Timer C can be programmed and interrogated by the CPU writing and reading CPU addresses $05F0 through $05F3 (Table 3-6).

Timer C is composed of a 16-bit latch, a 16-bit counter and an 8-bit snapshot register (Figure 3-8). The latch consists of two 8-bit registers, Timer C Upper Latch (TCUL) and Timer C Lower Latch (TCLL). The counter also consists of two 8-bit registers, Timer C Upper Counter (TCUC) and Timer C Lower counter (TCLC). The snapshot register is referred as Timer C Snapshot (TCS). Timer C operation is controlled and monitored using the Timer C Mode Register.

The Timer C can also be used as a third 16-bit programmable timer during normal operation. It is recommended that Timer C be stopped prior to reading the Timer C lower counter value.

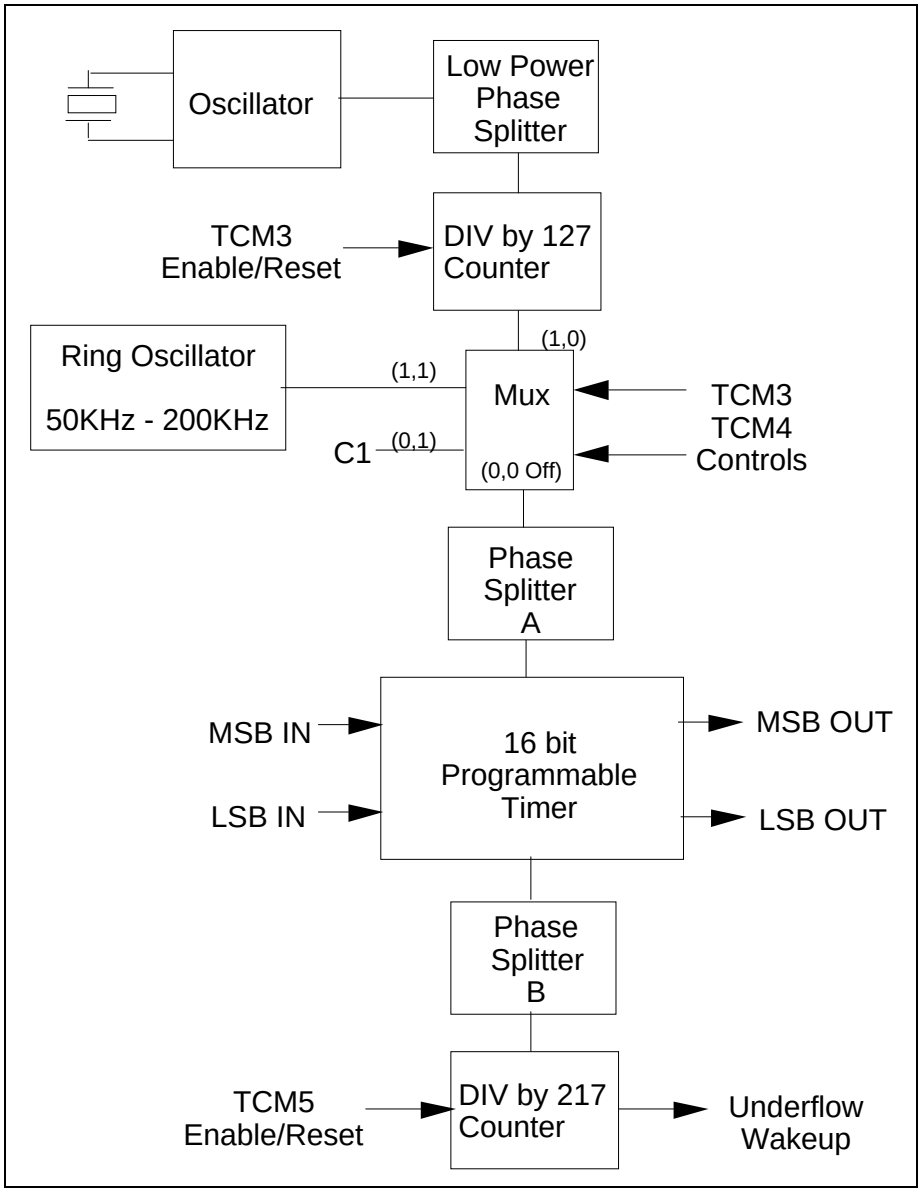


Figure 3-8. Timer C Snooze Timer

Table 3-6. Snooze Timer Bit Assignments

Addr.	Function	Bit 7	6	5	4	3	2	1	0
$05F0	Timer C Mode Control (TCM)	TIM C Underflow Flag	Div by 217 Overflow Flag	Snooze Enable	Snooze Clock Mode TCM4	Snooze Clock Mode TCM3	Not Used		
$05F1	Timer C “Lower’ Latch	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
$05F2	Timer C ‘Upper’ Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
$05F3	Timer C ‘Upper’ Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

All bits are cleared by reset. All bits can be read at address $05F0.

Table 3-7. LPR Bits Affected by TCM3 and TCM4

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0009 (Write Only)	Low Power Register (LPR) Write to TCM3 and TCM4	Bits Not Modified			If TCM = 1, Write LPR7 to TCM4	If TCM = 1, Write LPR6 to TCM3	Bits Not Modified		

Timer Mode Control (TCM) Register

Bits 0-2 Not Used.

Bits 3-4: **Snooze Clock Mode.** The Snooze Clock Mode bits select the Snooze Timer clock mode of operation.

TCM4	TCM3	Clock Mode
0	0	Clock Off Mode
0	1	Internal Oscillator C2 rate
1	0	Internal Oscillator Divide by 127 rate
1	1	Ring Oscillator rate - approximately 100 kHz

TCM4 and TCM3 can be set and reset by writing ones or zeros to bits 3 and 4, respectively. The divide by 217 counter is initialized to zero when both TCM3 and TCM4 are zero.

If Snooze is enabled (TCM5 = 1); the CPU can also write to TCM4 and TCM3 by a write to the Low Power Register at address $0009. In this case, data bit 7 is copied into TCM4 and data bit 6 is copied into TCM3. This feature automatically selects the ring oscillator clock source for stop mode snooze wake-up and the internal oscillator and divide by 127 timer for idle mode snooze wake-up (Table 3-7).

The following sequence should be followed when enabling the snooze timer low power wake-up features:

1. Write 20h to $05F0 to enable snooze wake-up.
2. Write the TCUL value to $05F3 to initialize Timer C and clear the TIMC underflow flag. This assumes that TCLL was previously set.
3. Write either an $80 (Sleep Mode) or $C0 (Stop Mode) to the LPR at $0009.

Bit 5: **Snooze Enable.** This bit can be set or cleared by writing a one or zero to $05F0. When set, TCM5 enables the divide by 217 timer to sequence with each Timer C underflow.

Bit 6: **DIV by 217 Overflow.** TCM6 is set each time that the divide by 217 timer overflows. TCM6 can be cleared by writing a zero to TCM6.

Bit 7: **Timer C Underflow.** TCM7 is set when Timer C transitions from $0000 to the Timer C latch values. TCM7 is cleared by a write to $05F3 (TCM download).

3.5 IRQ INTERRUPT LOGIC

3.5.1 Interrupt Request (IRQ)Vector and Hardware Priority

The IRQ Interrupt Logic prioritizes the individual interrupt requests (IRQ1-IRQ6) from the various sources and passes a single IRQ along with an IRQ number (1-6) and the IRQ vector page indicator to the CPU Interrupt Logic. Figure 3-9 illustrates the IRQ Interrupt Logic interface. Table 3-8 shows the IRQ interrupt levels, sources and vector addresses. Table 3-9 describes the interrupt clearing procedure.

If simultaneous IRQs occur on IRQ1-IRQ6 lines, the number of the highest priority IRQ (1 = highest) is passed to the CPU. When the interrupt flag causing the IRQ is cleared by the IRQ interrupt service subroutine, the IRQ number of the highest pending IRQ is passed.

The selection of ROM or RAM IRQ interrupt vectors for Timer A (IRQ5) and Timer B (IRQ3) is determined by bits 5 and 6 in the Timer A Mode and Timer B Mode registers, respectively.

3.5.2 Break Command

The BRK command causes the processor to go through an interrupt request sequence under program control. The address in the program counter (which points to the location of the BRK command + 1) is pushed on the stack, along with the processor status at the beginning of the BRK instruction. The processor then transfers control to the NMI interrupt vector ($FFFC and $FFFD). Note that the BRK command cannot be masked by setting the I flag.

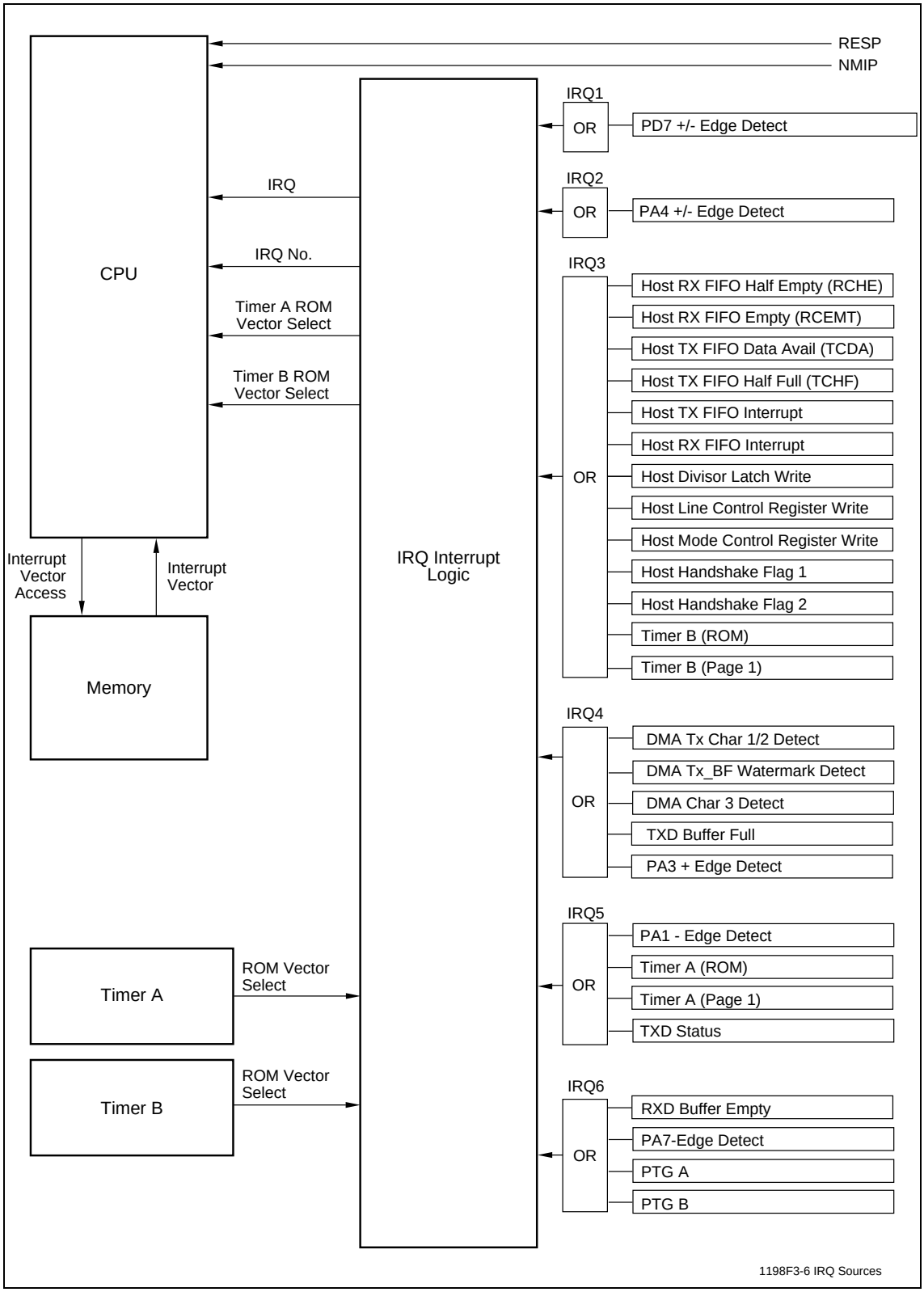


Figure 3-9. IRQ Interrupt Logic Interface

Table 3-8. Interrupt Request (IRQ) Vector and Hardware Priority

Source	Location		Interrupt Vector and Priority Level					
	Flag	Enable	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6
PD7 Edge Detect (+/-)	EIR5	EIR2	FFFA,B					
PA4 Edge Detect (+/-)	EIR6	EIR3		FFF8,9				
PA1 Edge Detect (-)	EIR7	EIR4					FFF2,3	
Host RX FIFO Half Empty (RCHE)	FSR0	FIER0			FFF6,7			
Host RX FIFO Empty (RCEMT)	FSR1	FIER1			FFF6,7			
Host TX FIFO Data Avail (TCDA)	FSR6	FIER6			FFF6,7			
Host TX FIFO Half Full (TCHF)	FSR7	FIER7			FFF6,7			
Host TX FIFO Interrupt	HCR7	FIER6, FIER7			FFF6,7			
Host RX FIFO Interrupt	HCR3	FIER0, FIER1			FFF6,7			
Host Divisor Latch Write	HCR4	HCR0			FFF6,7			
Host Line Control Register Write	HCR5	HCR0			FFF6,7			
Host Mode Control Register Write	HCR6	HCR0			FFF6,7			
Host Flag #1	HSR7	HSR3			FFF6,7			
Host Flag #2	HSR6	HSR2			FFF6,7			
Timer A (ROM)	TAM7	TAM6					FFF2,3	
Timer A (Page 1)	TAM6	TAM6					0102,3	
Timer B (ROM)	TBM7	TBM6			FFF6,7			
Timer B (Page 1)	TBM6	TBM6			0106,7			
DMA Tx_CHAR1/2 Detect	DFR5	DER3				FFF4,5		
DMA Tx_BF Watermark Detect	DFR6	DER5				FFF4,5		
DMA CHAR3 Detect	DFR7	DER7				FFF4,5		
TXD Buffer Full	SS0	SI0				FFF4,5		
TXD Status	SI7	SI2					FFF2,3	
PA3 Edge Detect (+)	SI6	SI4				FFF4,5		
RXD Buffer Empty	SS5	SI1						FFF0,1
PA7 Edge Detect (-)	SI5	SI3						FFF0,1
PTGA	PAM7	PAM6						FFF0,1
PTGB	PBM7	PBM6						FFF0,1

Table 3-9. Clearing Source of IRQ Interrupt

	Location		
Source	**Flag**	**Enable**	**Clear Interrupt Procedure**
PD7 Edge Detect (+/-)	E15	E12	Write a 0 to CI-5 bit, $000B; writing a 1 has no impact.
PA4 Edge Detect (+/-)	E16	E13	Write a 0 to CI-6 bit, $000B; writing a 1 has no impact.
PA1 Edge Detect (-)	E17	E14	Write a 0 to CI-7 bit, $000B; writing a 1 has no impact.
Host RX FIFO Half Empty (RCHE)	FSR0	FIER0	Write to Rx FIFO, $0020, until contents are 8 or more.
Host RX FIFO Empty (RCEMT)	FSR1	FIER1	Write to Rx FIFO, $0020.
Host TX FIFO Data Avail (TCDA)	FSR6	FIER6	Read Tx FIFO, $0020, until FSR6 = 0.
Host TX FIFO Half Full (TCHF)	FSR7	FIER7	Read Tx FIFO, $0020, until FSR7 = 0, can read at least 8 times.
Host TX FIFO Interrupt	HCR7	FIE6,7	Same as FSR6 and FSR7.
Host RX FIFO Interrupt	HCR3	FIE0,1	Same as FSR0 and FSR1.
Host Divisor Latch Write	HCR4	HCR0	Write a 0 to HCR-4 bit, $0032; writing a 1 has no impact.
Host Line Control Register Write	HCR5	HCR0	Write a 0 to HCR-5 bit, $0032; writing a 1 has no impact.
Host Mode Control Register Write	HCR6	HCR0	Write a 0 to HCR-6 bit, $0032; writing a 1 has no impact.
Host Flag #1	HSR7	HSR3	Write a 0 to HSR-7 bit, $002F; writing a 1 has no impact.
Host Flag #2	HSR6	HSR2	Write a 0 to HSR-6 bit, $002F; writing a 1 has no impact.
Timer A (ROM)	TAM7	TAM6	Read TAS at $0013 or write TAUL value to $0013.
Timer A (Page 1)	TAM6	TAM6	Read TAS at $0013 or write TAUL value to $0013.
Timer B (ROM)	TBM7	TBM6	Read TBS at $0017 or write TBUL value to $0017.
Timer B (Page 1)	TBM6	TBM6	Read TBS at $0017 or write TBUL value to $0017.
DMA Tx_CHAR1/2 Detect	DFR5	DER3	Read DFR at $05F5, recommend saving value in RAM.
DMA Tx_BF Watermark Detect	DFR6	DER5	Read DFR at $05F5, recommend saving value in RAM.
DMA CHAR3 Detect	DFR7	DER7	Read DFR at $05F5, recommend saving value in RAM.
TXD Buffer Full	SS0	SI0	Read RS232 buffer at $0038
TXD Status	SI7	SI2	Read RS232 status register at $0038 (TXD status bits)
PA3 Edge Detect (+)	SI6	SI4	Write a 0 to CI-3 bit, $000B; writing a 1 has no impact.
RXD Buffer Empty	SS5	SI1	Write to RS232 buffer at $0038.
PA7 Edge Detect (-)	SI5	SI3	Write a 0 to CI-4 bit, $000B; writing a 1 has no impact.
PTGA	PAM7	PAM6	Read or write PAUL at $0037.
PTGB	PBM7	PBM6	Read or write PBUL at $000F.

3.6 INTERNAL ROM

The internal Read Only Memory (ROM) contains program instructions and other fixed constants that validate the presence of proper firmware in external ROM and allow flash loading of blank external ROM. These program instructions and constants are mask-programmed during fabrication.

3.6.1 Size and Location

The internal ROM size is 8196 (8k) bytes and is mapped from $E000 - $FFFF (see the memory map in Figure 3-2).

3.6.2 TSTP and Internal ROM Control

The relationship between TSTP and internal ROM control is described in Table 3-10.

Table 3-10. TSTP and Internal ROM Control

TSTP State	Mode
High	**NORMAL MODE (Using Internal ROM).** Internal ROM is enabled and Program starting vector is obtained from default ROM address $FFFE and $FFFF. After any write to Bank Select Register 7, the internal ROM is enabled whenever all the external selects are disabled (ES0=1, A17=1, ES2=1, ES3=1,); otherwise external accesses are enabled (any ES0, A17, ES2, ES3 set to zero). Upon power turn-on or reset, the internal code looks for a copyright message at a specific location in the external ROM. If found, the code will jump through the reset vector to the next instruction located in external ROM.
Low	**EMULATION MODE (Internal ROM Disabled).** Internal ROM is disabled and the Program starting vector is obtained from external memory address $FFFE and $FFFF via the expansion data bus. This mode should not be used.

3.7 INTERNAL RAM

The internal Random Access Memory (RAM) contains the program stack and is used for scratch pad memory during system operation. This RAM is completely static in operation and requires no clock or dynamic refresh. The data contained in RAM is read out nondestructively with the same polarity as the input data. In the event that execution stops, RAM data is retained until execution resumes.

3.7.1 Size and Location

The internal RAM size is 1424 or 1456 bytes, depending if RS232/16550 DMA is used. A block of 192 bytes is assigned to page 0 ($40 to $FF). Four blocks of 256 bytes each are assigned to pages 1 through 4 ($100 to $1FF, $200 to $2FF, $300 to $3FF, and $400 to $4FF). A sixth block of 208 bytes is assigned to page 5 ($500 to $5CF). Another 32 bytes can be assigned to page 5 RAM ($05D0 to $05EF) or to the DMA register file when DRS0 = 1.

3.8 MEMORY BANKING

The MCU has a dual port RAM that allows the user to bank select external memory (Figure 3-10). The dual port bank select RAM consists of 8 bytes of internal RAM shown below from address locations $0018 through $001F. These bytes are called Bank Select Registers 0-7 (BSR0-BSR7). Each Bank Select Register contains the following bits:

7	6	5	4	3	2	1	0
ES3	ES2	A17	ES0	A16	A15	A14	A13

Each time the CPU provides addresses AD15, AD14, and AD13, one of the eight BANK select RAM is selected.

Address	CPU Address Line			Bank Select	Reset	Default	
Map	AD15	AD14	AD13	Register	Values	Select	Speed
$0018	0	0	0	0	1110 0000	ES0	X1
$0019	0	0	1	1	1101 0001	A17	X1
$001A	0	1	0	2	1011 0010	ES2	X1
$001B	0	1	1	3	1011 0011	ES2	X1
$001C	1	0	0	4	0111 0100	ES3	X1
$001D	1	0	1	5	0111 0101	ES3	X1
$001E	1	1	0	6	0111 0110	ES3	X1
$001F	1	1	1	7	0111 0111	ES3	X1

The MCU outputs active low chip select lines on PB2, PB3, PB4, and PB5 according to the logic-0 bits of the selected BSR bits 4 through 7 (ES3, ES2, A17, and ES0), respectively, when its respective port B select registers PBS5 = 0, PBS4 = 0, PBS3 = 0, or PBS2 = 0. When neither ES3, ES2, A17, or ES0 is a logic "0" (default during RESP=0 and TSTP=1), the internal ROM is selected. PB1 outputs active low chip select ES4 for address space $0600 to $0800 when PBS1 = 0. Address space $0600 to $0800 is mapped to internal ROM address $FE00 to $FFFF when PBS1=1 is selected.

The MCU outputs address lines A15, A14, and A13 and selects the internal ROM bank according to the contents of the selected BSR bits 2-0, respectively. PB0 becomes bit 3 (A16) of the contents of the selected BSR when the port B select register PBS0 = 0. Address A16 allows addressing 8k bytes memory banks above 64k bytes of memory.

The user can alter the contents of each BSR byte by writing to the individual register address or use its respective reset value shown above.

The selected banking RAM's speed choice for chip selects ES3, ES2, and ES0, and address line A17, are reflected in the ESS speed register shown below. During power up, bank select 7 is automatically selected with chip select ES3 running at the highest X1 speed. When external ROM is interrogated for proper copyright message, ES3 is selected to run at the slowest speed (X ¼).

The selected BSR's speed choice for chip selects ES3, ES2, ES1, and ES0 are reflected in the ESS speed register shown below. During power up, bank select 7 is automatically selected with chip select ES3 running at the highest X1 speed. The user can alter the contents of the ESS register or use the highest X1 speed default value shown.

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0033	ES Speed (ESS)	ES3		ES2		A17		ES0	
		ESS7	ESS6	ESS5	ESS4	ESS3	ESS2	ESS1	ESS0
		MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB

Default = $00 (X1 - 1 clock time selected.)

The following table shows the memory speed selected for the chip selects chosen in the selected banking RAM.

ES Speed Control		
MSB	**LSB**	**Speed**
0	0	X 1
0	1	X 1/2
1	0	X 1/3
1	1	X 1/4

Priority logic makes the memory speed take on the choice of the highest active chip select ES3 or ES2 when two are enabled.

Memory banking is performed using internal address translation using eight Bank Select Registers (3.8.1). Each Bank Select Register (BSR) controls the address translation of A[15:13], controls the A16 and A17 pseudo-address bit, and defines an active bank select from ES3, ES2, and ES0 for a fixed 8k-byte region of linear address. This logic enables flexible banking while simultaneously eliminating the need for the external programmable logic device normally required to support banking. In addition, stretched internal clock cycles can be assigned independently to each BSR chip select range using the ES Speed Register (3.8.2) to allow the use of slower memories. The address translation allows several banks with the same logical address range to be physically relocated so that they can all reside within the same memory device with different physical address ranges.

This banking technique becomes more easily understood if one views the addition of A16, A17, and ES3, ES2, and ES0 separately from the internal address translation. Although A16 and A17 function identically to ES3, ES2, and ES0, in that they are manually controlled bits in each BSR and drive external pins, emulation support mandates that these signals not be used as chip selects. The new expanded address field A[17:0] maps directly to A[17:0] of a 256K memory device.

The address translation feature does not increase the amount of banked memory beyond this theoretical limit, but it allows several banks with the same logical address range to be mapped within the same physical device.

Note that ES3, ES2, ES0, A17, and A16 are Port B special purpose outputs and must be enabled using the Port B Select (PBS) register.

The memory banks are illustrated within the overall memory map in Figure 3-10. The MCU banking logic is shown in Figure 3-11. The bank select registers are shown in Table 3-11.

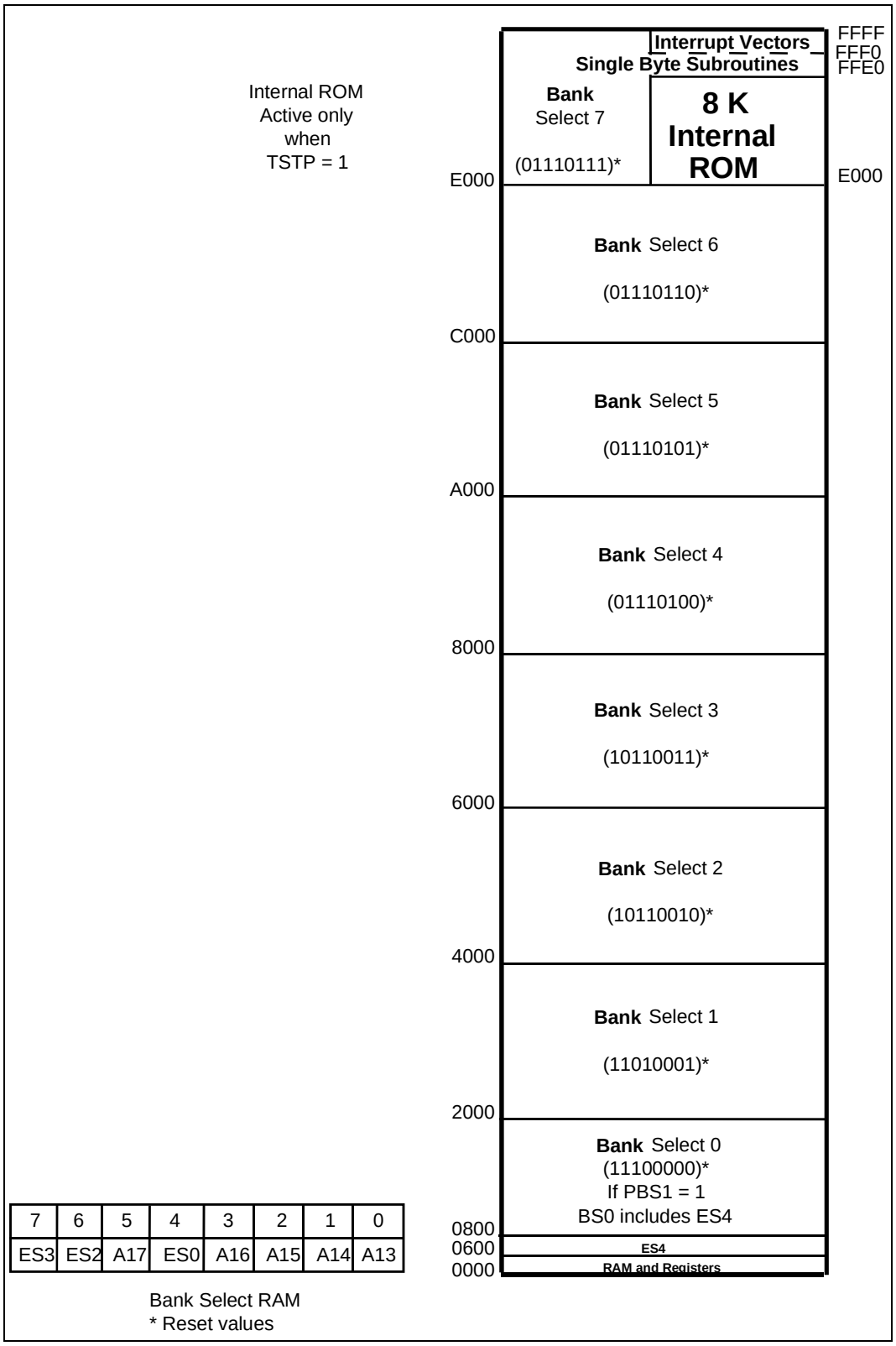


Figure 3-10. Memory Map - RAM Banking

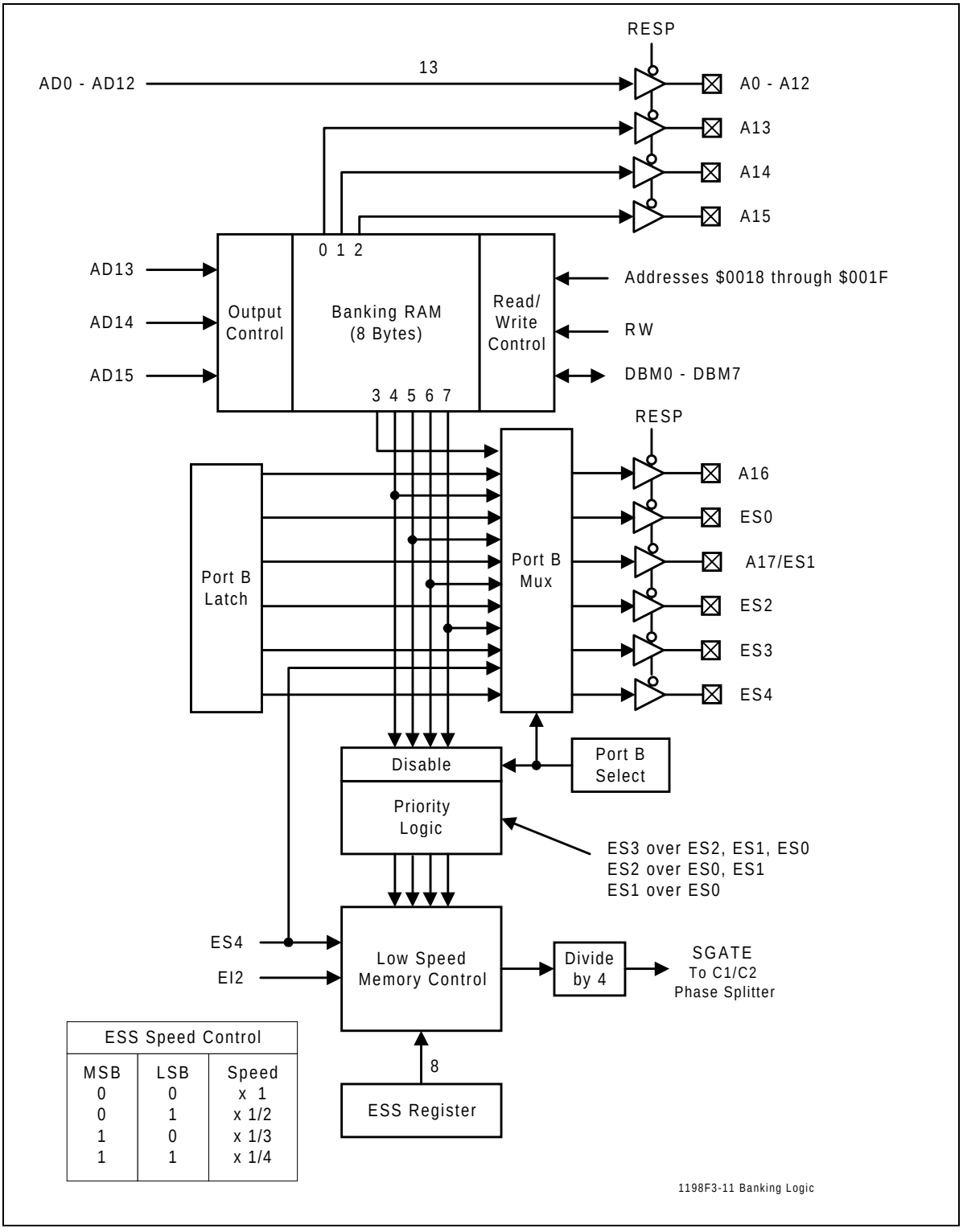


Figure 3-11. Banking Logic

Table 3-11. Register Bit Assignments: Bank Select Register i (BSRi) - 0018h-001Fh

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0018	Bank Select Register 0 (BSR0)	ES3 (0)	ES2 (0)	A17 (0)	ES0 (0)	A16 (0)	A15 (0)	A14 (0)	A13 (0)
0019	Bank Select Register 1 (BSR1)	ES3 (1)	ES2 (1)	A17 (1)	ES0 (1)	A16 (1)	A15 (1)	A14 (1)	A13 (1)
001A	Bank Select Register 2 (BSR2)	ES3 (2)	ES2 (2)	A17 (2)	ES0 (2)	A16 (2)	A15 (2)	A14 (2)	A13 (2)
001B	Bank Select Register 3 (BSR3)	ES3 (3)	ES2 (3)	A17 (3)	ES0 (3)	A16 (3)	A15 (3)	A14 (3)	A13 (3)
001C	Bank Select Register 4 (BSR4)	ES3 (4)	ES2 (4)	A17 (4)	ES0 (4)	A16 (4)	A15 (4)	A14 (4)	A13 (4)
001D	Bank Select Register 5 (BSR5)	ES3 (5)	ES2 (5)	A17 (5)	ES0 (5)	A16 (5)	A15 (5)	A14 (5)	A13 (5)
001E	Bank Select Register 6 (BSR6)	ES3 (6)	ES2 (6)	A17 (6)	ES0 (6)	A16 (6)	A15 (6)	A14 (6)	A13 (6)
001F	Bank Select Register 7 (BSR7)	ES3 (7)	ES2 (7)	A17 (7)	ES0 (7)	A16 (7)	A15 (7)	A14 (7)	A13 (7)

3.8.1 Bank Select Register (BSR)

Bits 0-3: A13 (i) - A16(i). Address Translation Bits for Bank i.

Bits 4: ES0(i). Memory Bank Select for Bank i.

Bits 5: A17(i). Address Translation Bits for Bank i.

Bits 6: ES2(i). Memory Bank Select for Bank i.

Bits 7: ES3(i). Memory Bank Select for Bank i.

Notes:

1. When 0 is not present in BSR7 (ES3, ES2, ES0), the internal ROM is enabled. This allows a bootstrap routine in internal ROM to be run, e.g., to configure the ES Speed register to select the effective clock width during external memory access or flash loading of external ROM. When the internal ROM is enabled, the external transceivers are disabled for addresses within the BSR7 range.

   When the TSTP pin is high, the internal ROM is enabled following reset until a new value is written into BSR7.

3.8.2 ES Speed Register

The ES Speed register at address $0033 (Table 3-12) selects the number clock times that the internal clock is stretched during a memory access in an address range corresponding to an ES3, ES2, ES0 chip select or the A17 address line.

Table 3-12. Register Bit Assignments: ES Speed Register - 0033h

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0033	ES Speed (ESS)	ES3-1	ES3-0	ES2-1	ES2-0	A17-1	A17-0	ES0-1	ES0-0

Default = $00 (X1 - 1 clock time selected.)

Bits 0-1: **Chip Select 0 (ES0) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the ES0 chip select is active.

Bit 1 (ES0-1)	Bit 0 (ES0-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

Bits 2-3: **Address 17 (A17) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the A17chip select is active.

Bit 3 (A17-1)	Bit 2 (A17-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

Bits 4-5: **Chip Select 2 (ES2) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the ES2 chip select is active.

Bit 5 (ES2-1)	Bit 4 (ES2-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

Bits 6-7: **Chip Select 3 (ES3) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the ES3 chip select is active.

Bit 7 (ES3-1)	Bit 6 (ES3-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

3.9 PARALLEL INPUT/OUTPUT PORTS

The MCU parallel input/output interface consists of five 8-bit ports: A, B, C, D, and E.

Ports A, C and E contain 24 bidirectional lines with the data direction determined by the direction registers. Port D has 4 bidirectional lines (0-3) with the data direction determined by its direction register, and 4 input only lines (4-7). Port B supports 8 output-only lines.

All port lines can be used for general purpose functions. Most I/O lines can be assigned special functions under software control. The special purpose and mask option functions of the port lines are identified in Table 3-13. Table 3-14 further defines the special purpose applications along with software control and direction register requirements. Port read and write timing is described in Section 5.

Output Mode. The data written to each output pin is loaded into an output data latch. The data will remain in the output latch until new data is written to the port address or until power is removed. The output latches are individually connected to output drivers. The output drivers are double-ended, push-pull type. The drivers force the output pins high (+ 2.4V) if the output data bit is a logic 1, or low (+0.4V) if the output data bit is a logic 0. The output drivers are TTL compatible. The External Interrupt Register (EIR) and Clear Interrupt Register (CIR) are associated with port B and D lines. The CIR bits are cleared by writing zero to the register.

Input Mode. For each input port line, either permanently or direction register assigned as an input, the data is sampled by an input synchronizer. A low input level (+0.8 v) is interpreted as a logic 0 and a high input level (+2.0 V) is interpreted as a logic 1. Input data is sampled during internal C2 clock time and then temporarily held from C1 to C1 clock time. When the CPU reads the input port, the data is transferred to the CPU during C2 time and represents data sampled during the previous C2 time.

Table 3-13. I/O Port Special Purpose Functions

	Port				
Port Bit	**A**	**B**	**C**	**D**	**E**
0	TIM A, LPWU RING, CTSP	A16	HD0	HA0	None
1	TXDE or - Edge	ES4	HD1	HA1	None
2	TXD, LPWU TXD	ES3	HD2	HA2	None
3	TXCLK, +Edge, or HTACKP	ES2	HD3	PWRDWNP (Internal)	None
4	TXREF, ± Edge, RLSDP, or HRACKP	A17	HD4	MCSP, LPWU DTR, or LPWU HCSP (Internal)	None
5	REXD or TXRDY	ES0	HD5	HWTP, LPWU AL, or LPWU HWTP	No port available
6	RXD or RXRDY	HDIS*	HD6	HRDP	No port available
7	RXCLK or - Edge	HINT	HD7	± Edge	None

* Not recommended for use.

Table 3-14. I/O Port Special Purpose Function Control

Port Line	Function	Option Selected by	I/O	Direction Reg.
Port A				
PA0	TIMA - Timer A Event Counter Input	TAM1 = 1, TAM0 = 0	I	PAD0 = 0
	TIMA - Timer A Pulse Width Measurement Input	TAM1 = 1, TAM0 = 1	I	PAD0 = 0
	RING - PA0 High (Ring)	LPR5 = 1	I	PAD0 = 0
	CTSP - USART TXD Sync to Falling Edge	SMR6 = 1, SMR1 = 1	I	PAD0 = 0
PA1	TXDE - USART PA1 Output Copies PA2 Input	SF7 = 1	O	PAD1 = 1
	PA1 - Negative Edge Detect	None, see EI7	I/O	PAD1 = X
PA2	TXD - USART Serial Input	SMR6 = 1	I	PAD2 = 0
	TIMA - Input TXD Test	TAM1 = 1, TAM0 = 1, SFR4 = 1	I	PAD2 = 0
	LPWU - PA2 Low (TXD)	LPR3 = 1	I	PAD2 = 0
PA3	TXCLK - USART Internal Timing		O	PAD3 = 1
	TXCLK - USART External Timing		I	PAD3 = 0
	TXCLK - USART Copy TXCLK (PA4)	SMR4 = 1	I/O	PAD3 = X
	PA3 - Positive Edge Detect	None, see SIR3	I/O	PAD3 = X
	HTACKP - TX Acknowledge Input	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0	I	PAD3 = 0
PA4	TXREF - External Clock Input	SMR4 = 1 and SMR2 = 1		PAD4 = 0
	PA4 - Positive and Negative Edge Detect	None, see IER6	I/O	PAD4 = X
	RLSDP USART RXD Sync to Falling Edge	SMR7 = 1, SMR1 = 1		PAD4 = 0
	HRACKP - 16550A RX Acknowledge Input	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0	I	PAD4 = 0
PA5	REXD - USART Serial Input	SM7 = 1, SF5 = 0, SF6 = 1		PAD5 = 0
	TXRDY - TX Ready Output	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0		PAD5 = 1
PA6	RXD - USART Serial Output	SMR7 = 1		PAD6 = 0
	RXRDY - RX Ready Output	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0		PAD6 = 1
PA7	RXCLK - RXD Serial Output Clock	SMR7 = 1, SMR4 = 1	O	PAD7 = 1
	PA7 - Falling Edge Detect	None, see SIR5	I	PAD7 = X
Port B				
PB0	A16 - External Bus Address Line A16	PBS0 = 0	O	Output Only
PB1	ES4 - External Bus Chip Select	PBS1 = 0	O	
PB2	ES3- External Bus Chip Select	PBS2 = 0	O	
PB3	ES2- External Bus Chip Select	PBS3 = 0	O	
PB4	A17- External Bus Address Line A17	PBS4 = 0	O	
PB5	ES0- External Bus Chip Select	PBS5 = 0	O	
PB6	HDIS - Host Bus Driver Disable*	PBS6 = 0, HCR2 = 1	O	
PB7	HINT - Host Bus Interrupt Line	PBS7 = 0, HCR2 = 1	O	

* Not recommended for use.

Table 3-14. I/O Port Special Purpose Function Control (Cont'd)

Port Line	Function	Option Selected by	I/O	Direction Reg.
Port C				
PC0	HD0 - Host Bus Data Line 0	HCR2 = 1	I/O	Controlled by HWTP and HRDP
PC1	HD1 - Host Bus Data Line 1			
PC2	HD2- Host Bus Data Line 2			
PC3	HD3- Host Bus Data Line 3			
PC4	HD4- Host Bus Data Line 4			
PC5	HD5- Host Bus Data Line 5			
PC6	HD6- Host Bus Data Line 6			
PC7	HD7- Host Bus Data Line 7			
Port D				
PD0	HA0 - Host Bus Address Line 0	HCR2 = 1	I	PD0 = 0
PD1	HA1- Host Bus Address Line 1	HCR2 = 1	I	PD1 = 0
PD2	HA2- Host Bus Address Line 2	HCR2 = 1	I	PD2 = 0
PD3*	PWRDWNP (Internal)		O	
PD4*	MCSP (MCU Chip Select)/ HCSP (Host Bus Chip Select) (Internal)	HCR2 = 1	O	PD4 = 1
	LPWU - PD4 Low (DTR)	LPR2 = 1		
	LPWU - PD4 (HCSP) Low and PD5 Low (HWTP)	LPR4 = 1		
PD5	HWTP - Host Bus Write Enable	HCR2 = 1	I	PD4 = 0
	LPWU - PD5 Low (AL)	LPR1 = 1		
	LPWU - PD4 (HCSP) Low and PD5 Low (HWTP)	LPR4 = 1		
PD6	HRDP - Host Bus Read Enable	HCR2 = 1	I	PD6 = 0
PD7	PA7 Falling or rising edge detect	None, see IER5	I	N/A
* Internal connection				

3.9.1. Bidirectional Ports A, C, and E

Ports A, C, and E are general purpose bidirectional input/output lines. The data direction for each I/O line is controlled by an associated direction register bit. For each direction register bit that is a logic 1, the corresponding port line is an output. Conversely, a 0 in a direction register bit defines the corresponding port line as an input. The direction register bits are initialized to a 0 by reset causing the I/O ports to be inputs.

All port A, and C lines can be assigned to special purpose functions during operation under software control. The port A lines can be assigned to special functions under software control (Table 3-13 and Table 3-14). PA3 and PA7 have associated edge detect logic that can generate an IRQ interrupt.

Seven port B lines can be assigned to special functions under software control and two lines can be permanently masked to a special functions (Table 3-13 and Table 3-14). PB2 and PB3 have associated edge detect logic that can generate an IRQ interrupt.

All eight port C lines can be assigned to host bus data lines under software control (Table 3-13 and Table 3-14).

3.9.1 Bidirectional and Input Only Port D

Port D contains three I/O pins with three corresponding data directional control bits, two special purpose output only ports to internal connections, and three input only pins (Table 3-13 and Table 3-14).

PD0-PD2 are general purpose bidirectional input/output pins, with internal pull-downs. The PD0-PD2 output level is controlled by writing to bits 0-2 at address $0003, while their direction control is contained in bits 4-6, respectively. PD0-PD2 can be assigned in a group to host bus address (HA0-HA2) when Host Mode is enabled (HCR2 = 1).

PD3 and PD4 are output pins reflecting internal power down status and modem chip select signals, respectively.

PD5-PD6 are input only pins, with internal pull-ups.

PD7 is an input only pin.

3.9.2 Output Port B

The 8 port B lines (PB0-PB7) are general or special purpose output only (Table 3-13). Lines PB0-PB5 can be assigned to A16 and A17 address lines and the ES0, ES2, ES3, and ES4 chip select functions controlled by the Bus Select Registers and PB6 and PB7 can be assigned to HDIS and HINT functions (Table 3-14). Port B output latches are initialized low upon reset. The Port B output drivers tri-state (float) during reset active low.

3.9.3 Port B Select Register (PBS)

The Port B Select Register (PBS) bits are identified in Table 3-15.

Table 3-15. Port B Select Register (PBS)

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0005	Port B Select (0-7)	**PB7/HINT Sel** 1 = PB7 0 = HINT	**PB6/HDIS** 1 = PB6 0 = HDIS	**PB5/ES0 Sel** 1 = PB5 0 = ES0	**PB4/A17** 1 = PB4 0 = A17	**PB3/ES2 Sel** 1 = PB3 0 = ES2	**PB2/ES3 Sel** 1 = PB2 0 = ES3	**PB1/ES4 Sel** 1 = PB1 0 = ES4	PB0/A16 Sel 1 = PB0 0 = A16

Bit 0: **PB0/A16 Select. Control bit.** Control bit.

0 = Selects PB0 to operate as A16.

1 = Selects PB0 to operate as a general purpose port.

Bit 1: **PB1/ES4 Select.** Control bit.

0 = Selects PB1 to operate as ES4.

1 = Selects PB1 to operate as a general purpose port.

Bit 2: **PB2/ES3 Select.** Control bit.

0 = Selects PB2 to operate as ES3.

1 = Selects PB2 to operate as a general purpose port.

Bit 3: **PB3/ES2 Select.** Control bit.

0 = Selects PB3 to operate as ES2.

1 = Selects PB3 to operate as a general purpose port.

Bit 4: **PB4/A17 Select.** Control bit.

0 = Selects PB4 to operate as A17.

1 = Selects PB4 to operate as a general purpose port.

Bit 5: **PB5/ES0 Select.** Control bit.

0 = Selects PB5 to operate as ES0.

1 = Selects PB5 to operate as a general purpose port.

Bit 6: **PB6/HDIS Select.** Control bit.

0 = Selects PB6 to operate as HDIS (not recommended for use).

1 = Selects PB6 to operate as a general purpose port.

Bit 7: **PB7/HINT Select.** Control bit.

0 = Selects PB7 to operate as HINT.

1 = Selects PB7 to operate as a general purpose port.

3.9.4 External Interrupt Register (EIR)

The External Interrupt Register (EIR) enables and reports interrupts associated with ports PA1, PA4, and PD7 (Table 3-16). All bits are cleared to zero by reset. The CPU can set or reset bits 0-4 by writing to address $000A. Bits 5-7 are cleared by writing a 0 to the corresponding bit position in the CIR. The CPU can monitor all bits by reading address $000A.

Table 3-16. External Interrupt Register (EIR) - $000A

Addr.	Function	Bit 7	6	5	4	3	2	1	0
000A	External Interrupt Register (EIR)	PA1 Ext. Int. Flag 1 = Flag	PA4 Ext. Int. Flag 1 = Flag	PD7 Ext. Int. Flag 1 = Flag	PA1 Ext. Int. Enable 1 = Enable	PA4 Ext. Int. Enable 1 = Enable	PD7 Ext. Int. Enable 1 = Enable	PA4 Edge Detect Polarity 1 = Rising ↑ 0 = Falling ↓	PD7 Edge Detect Polarity 1 = Rising ↑ 0 = Falling ↓

Bit 0: **PD7 Positive Edge Detect.** Control bit.

1 = Enables positive or rising (low-to-high) edge detection on PD7.

0 = Enables negative or falling (high-to-low) edge detection on PD7.

Bit 1: **PA4 Positive Edge Detect.** Control bit.

1 = Enables positive or rising (low-to-high) edge detection on PA4.

0 = Enables negative or falling (high-to-low) edge detection on PA4.

Bit 2: **PD7 Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ1 when EIR5 is set to a logic 1.

0 = Disables assertion of IRQ1 due to EIR5.

Bit 3: **PA4 Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ2 when EIR6 is set to a logic 1.

0 = Disables assertion of IRQ2 due to EIR6.

Bit 4: **PA1 Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ5 when EIR7 is set to a logic 1.

0 = Disables assertion of IRQ5 due to EIR7.

Bit 5: **PD7 Interrupt Flag.** Status bit.

1 = A positive (EIR0 = 1) or negative (EIR0 = 0) edge has been detected on PD7. This bit is cleared by writing a logic 0 to CIR5.

0 = An edge has not been detected on PD7.

Bit 6: **PA4 Interrupt Flag.** Status bit.

1 = A positive (EIR1 = 1) or negative (EIR1 = 0) edge has been detected on PA4. This bit is cleared by writing a logic 0 to CIR6.

0 = An edge has not been detected on PA4.

Bit 7: **PA1 Interrupt Flag.** Status bit.

1 = A negative edge has been detected on PA1 This bit is cleared by writing a logic 0 to CIR7.

0 = An edge has not been detected on PA1.

3.9.5 Clear Interrupt Register (CIR)

The Clear Interrupt Register (CIR) at address $000B (Table 3-17) is used to clear five interrupt flags (PA3, PA7, PD7, PA4, and PA1) and set the fast/slow operation of ES4 expansion bus addressing. CIR0, CIR1 and CIR2 are initialized to 0 by reset.

Table 3-17. Clear Interrupt Register (CIR)- $000B

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
000B	Clear Interrupt Register (CIR)	PA1Clear Int. Flag (EIR7)	PA4 Clear Int. Flag (EIR6)	PD7 Clear Int. Flag (EIR5)	PA7 Clear Int. Flag (SIR5)	PA3 Clear Int. Flag (SIR6)	ES4 Fast Mem Cycle	Not Used	Not Used

Bits 0-1: Not Used.

Bit 2: **ES4 Expansion Bus Fast Memory Cycle.** Control and status bit. This bit can be set or reset by the CPU. The CPU can monitor the bit by reading address $000B. The bit is initialized to zero by reset.

1 = Enables ES4 Expansion Bus addresses to operate at fast speed, i.e., with one ø2 clock cycle per memory access cycle.

0 = Enables ES4 Expansion Bus addresses to operate at slow speed, i.e., with two ø2 clock cycles per memory access cycle.

Bit 3: **Clear PA3 Interrupt Flag.** Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA3 Interrupt Flag (SIR6) to a logic 0.

Bit 4: **Clear PA7 Interrupt Flag.** Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA7 Interrupt Flag (SIR5) to a logic 0.

Bit 5: **Clear PD7 Interrupt Flag.** Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PD7 Interrupt Flag (EIR5) to a logic 0.

Bit 6: **Clear PA4 Interrupt Flag.** Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA4 Interrupt Flag (EIR6) to a logic 0.

Bit 7: **Clear PA1 Interrupt Flag.** Writing a logic 0 to this bit position resets the PA1 Interrupt Flag (EIR7) to a logic 0. Writing a logic 1 to this bit position has no effect. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA1 Interrupt Flag (EIR7) to a logic 0.

3.10 COUNTER/TIMERS

There are two separate 16-bit counter/timer systems in the MCU: Counter/Timer A (called Timer A) and Counter/Timer B (called Timer B). Timer A operates in one of three modes and one input port. Timer B operates only in the interval timer mode. Otherwise, operation of the two counter/timers is similar except for register addresses, the generated IRQ (and priority level) and the interfacing I/O port. The operation of Timer A is described in detail followed by a description of Timer B differences. Block diagrams of Timer A and Timer B are shown in Figure 3-12 and Figure 3-13, respectively.

A divide-by-32 counter connected to Ø2 clock is shared by both timers. The counter provides a Ø2/32 clock that can be individually selected by each timer.

3.10.1 Timer A Registers

Timer A is composed of a 16-bit latch, a 16-bit counter and an 8-bit snapshot register (Figure 3-12). The latch consists of two 8-bit registers, Timer A Upper Latch (TAUL) and Timer A Lower Latch (TALL). The counter also consists of two 8-bit registers, Timer A Upper Counter (TAUC) and Timer A Lower counter (TALC). The snapshot register is referred as Timer A Snapshot (TAS). Timer A operation is controlled and monitored using the Timer A Mode Register (Table 3-18).

TALL is loaded by the CPU writing to address $0011. TAUL can be loaded by writing to either $0012 or $0013. When the CPU writes to address $0013, the contents of TALL and TAUL are also downloaded into TALC and TAUC, respectively, and the Timer A Interrupt Flag (TAM7) is cleared. The contents of TALC can be monitored at any time by reading $0011. Reading $0011 also causes the contents of TAUC to transfer into TAS. The contents of TAS can be monitored by reading either $0012 or $0013.

When Timer A underflows, the Timer A Interrupt Flag bit in the (TAM7) is set to a logic 1. This bit can be used to assert IRQ5.

3.10.2 Timer B Registers

Timer B is similar to Timer A except only the interval timer mode is supported (Figure 3-13). Timer B registers are located at $0014-$0017. When the Timer B Underflow Flag is set (TBM7) and enabled (TBM5 and TBM6), IRQ3 is asserted. Timer B interfaces with I/O port PB0 rather than PA0.

Timer B is composed of a 16-bit latch, a 16-bit counter and an 8-bit snapshot register. The latch consists of two 8-bit registers, Timer B Upper Latch (TBUL) and Timer B Lower Latch (TBLL). The counter also consists of two 8-bit registers, Timer B Upper Counter (TBUC) and Timer B Lower counter (TBLC). The snapshot register is referred as Timer B Snapshot (TBS).

TBLL is loaded by the CPU writing to address $0015. TBUL can be loaded by writing to either $0016 or $0017. When the CPU writes to address $0017, the contents of TBLL and TBUL are also downloaded into TBLC and TBUC, respectively, and the Timer B Interrupt Flag (TBM7) is cleared. The contents of TBLC can be monitored at any time by reading $0015. Reading $0015 also causes the contents of TBUC to transfer into TBS. The contents of TBS can be monitored by reading either $0016 or $0017.

When Timer B underflows, the Timer B Interrupt Flag bit in the (TBM7) is set to a logic 1. This bit can be used to assert IRQ3.

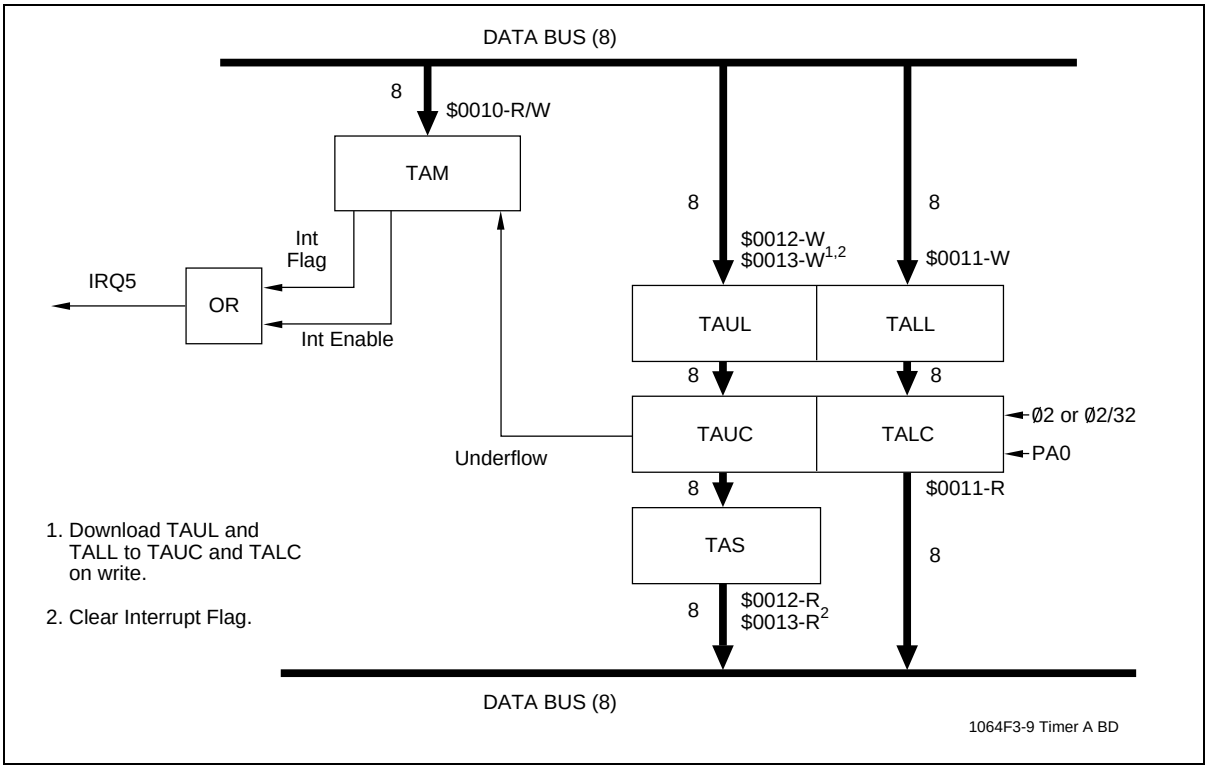


Figure 3-12. Counter/Timer A Block Diagram

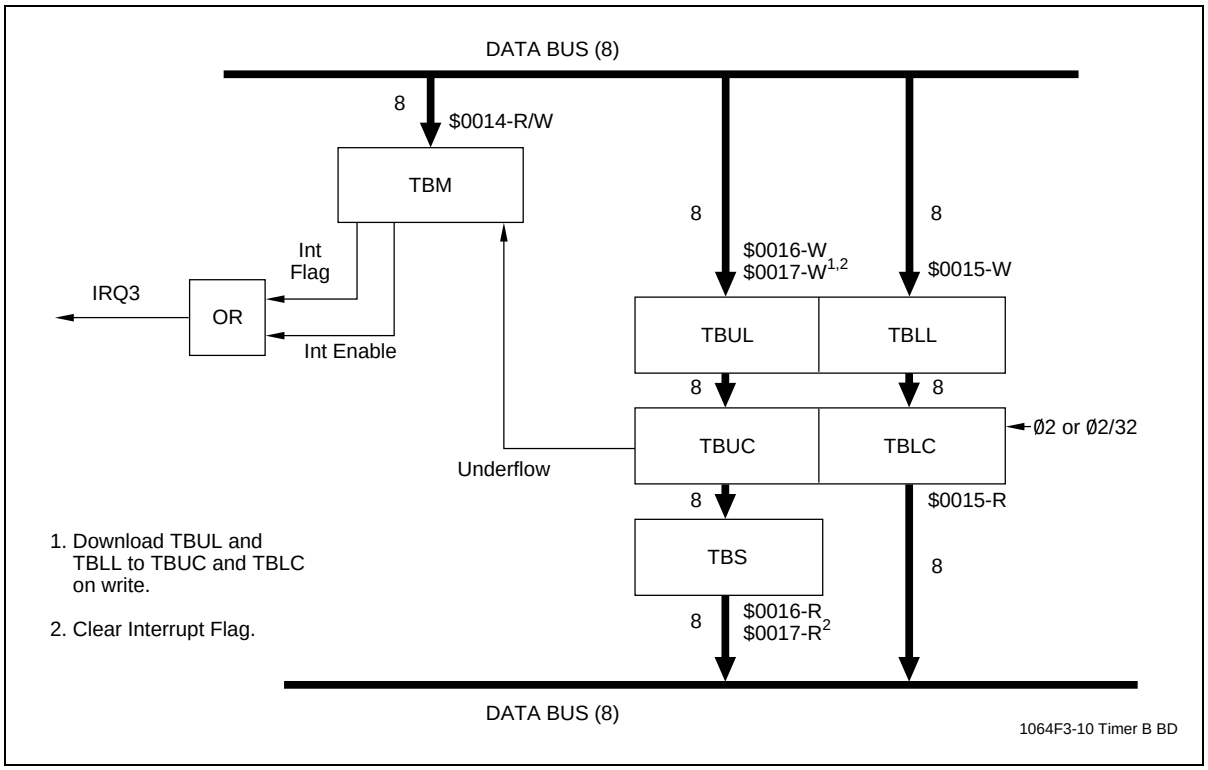


Figure 3-13. Counter/Timer B Block Diagram

3.10.3 Timer A Mode Register (TAM)

The Timer A Mode Register (TAM) selects the Timer A operating mode selection, and controls and reports the Timer A interrupt (Table 3-18). Bits 0-2, 5, and 6 are cleared to zero by reset, and can be reset or set by the CPU writing to address $0010. Bits 0-2, and 5-7 can be read by the CPU.

Table 3-18. Register Bit Assignments: 0010h - 0013h

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0010	Timer A Mode (TAM)	Timer A Int. Flag 1 = Flag	Timer A Int. Enable 1 = Enable	RAM Int. Vector Select 1 = RAM 0 = ROM	Not Used		Timer A Div by 32 Prescale 1 = Div by 32	Timer A Mode 00 = Interval Timer 01 = Pulse Generator 10 = Event Counter 11 = Pulse Width Measurement	
0011	Timer A Lower Latch	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0012	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0013	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bits 0-1: **Timer A Mode Select.** These two bits select the Timer A operating mode. The interval timer mode is selected upon reset since these bits are reset to logic 0.

Bit 1	Bit 0	Mode
0	0	Interval Timer
0	1	Pulse Generator (Not supported; invalid mode)
1	0	Event Counter
1	1	Pulse Width Measurement

Bits 2: **Timer A Divide by 32 Prescale.** Control bit.

1 = The Timer A ø2 clock is divided by 32.

0 = The Timer A ø2 clock is not divided by 32.

Bits 3-4: **Not used.**

Bit 5: **Timer A IRQ5 RAM Vector Enable.** Control bit.

1 = When the Timer A Interrupt Enable (TAM6) is a logic 1, IRQ5 is asserted through the Timer A IRQ5 RAM vector.

0 = When the Timer A Interrupt Enable (TAM6) is a logic 1, IRQ5 is asserted through the Timer A IRQ5 ROM vector.

Bit 6: **Timer A Interrupt Enable.** Control bit.

1 = Enables IRQ5 to be asserted when the Timer A Interrupt Flag (TAM7) is set to a logic 1.

0 = Disables IRQ5 assertion due to TAM7.

Bit 7: **Timer A Interrupt Flag.** Status bit. Reading or writing $0013 clears the Timer A Interrupt Flag (TAM7).

1 = Timer A counter underflow has occurred, i.e., it decremented from 0 to - 1 (0000 to $FFFF).

0 = Timer A counter underflow has not occurred.

3.10.4 Timer B Mode Register (TBM)

The Timer B Mode Register (TBM) controls and reports the Timer B interrupt (Table 3-19). Bits 0-2, 5, and 6 are cleared to zero by reset, and can be reset or set by the CPU writing to address $0014. Bits 0-2, and 5-7 can be-read by the CPU.

Table 3-19. Register Bit Assignments: 0014h - 0017h

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0014	Timer B Mode (TBM)	Timer B Int. Flag 1 = Flag	Timer B Int. Enable 1 = Enable	RAM Int. Vector Select 1 = RAM 0 = ROM	Not Used		Timer B Div by 32 Prescale 1 = Div by 32	Timer B Mode 00 = Interval Timer 01 = Pulse Generator 10 = Event Counter 11 = Pulse Width Measurement	
0015	Timer B Latch Low	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0016	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0017	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bits 0-1: **Timer B Mode Select. Timer B Mode Select.** These two bits select the Timer B operating mode. Since the interval timer mode is the only valid mode supported, these two bits must be set to 0.

Bit 1	Bit 0	Mode
0	0	Interval Timer
0	1	Pulse Generator (Not supported; invalid mode)
1	0	Event Counter
1	1	Pulse Width Measurement)

Bits 2: **Timer B Divide by 32 Prescale.** Control bit.

1 = The Timer B ø2 clock is divided by 32.

0 = The Timer B ø2 clock is not divided by 32.

Bits 3-4: **Not used.**

Bit 5: **Timer B IRQ3 RAM Vector Enable.** Control bit.

1 = When the Timer B Interrupt Enable (TBM6) is a logic 1, IRQ3 is asserted through the Timer B IRQ3 RAM vector.

0 = When the Timer B Interrupt Enable (TBM6) is a logic 1, IRQ3 is asserted through the Timer B IRQ3 ROM vector.

Bit 6: **Timer B Interrupt Enable.** Control bit.

1 = Enables IRQ3 to be asserted when the Timer B Interrupt Flag (TBM7) is set to a logic 1.

0 = Disables IRQ3 assertion due to TBM7.

Bit 7: **Timer B Interrupt Flag.** Status bit. Reading $0017 clears the Timer B Interrupt Flag (TBM7).

1 = Timer B counter underflow has occurred, i.e., it decremented from 0 to - 1 (0000 to $FFFF).

0 = Timer B counter underflow has not occurred.

3.10.5 Timer Modes

Timer operation is described for the four Timer A modes. The Mode 0 - Interval Timer operation also applies to Timer B (the Timer B parameters are denoted in parentheses). The waveforms for the timer modes are illustrated in Figure 3-14, Figure 3-15, and Figure 3-16.

Mode 0 - Interval Timer

Writing to TAUL (TBUL) transfers the 16-bit latch value to the counter. The counter counts down at the Ø2 or Ø2/32 rate. When the counter counts through zero, the TAIF (TBIF) is set to a 1, the value in the latches is transferred to the counter and the counter continues to count down. (See Figure 3-14.)

Mode 1 - Pulse Generation

Not supported.

Mode 2 - Event Counter

The PAD0 direction register bit must be set to 0 to establish PA0 as an input pin. The TAM2 clock divide-by-32 bit must be set to a 0 to select divide-by-1. The counter is initialized with the latch value when the TAUL value is written to address $0013. The timer decrements by 1 at each positive transition on input port PA0. TAIF is set to a 1 when the counter counts through zero. At the same time the latch value is reloaded into the counter. The maximum rate of the signal of PA0 is one-half the timer clock rate. (See Figure 3-15.)

Mode 3 - Pulse Width Measurement

The PAD0 direction register bit must be set to an 0 to establish PA0 as an input pin. Writing to TAUL at $0013 transfers the 16-bit latch value to the counter. The value in the timer is decremented at the Ø2 or Ø2/32 rate when the PA0 signal is low. Each time the PA0 signal goes high, the counter stops and then continues when the signal is low again. If the counter counts through zero, TAIF is set to a 1 and the latch value transfers to reinitialize the counter. The countdown continues as long as PA0 is low. (See Figure 3-16.)

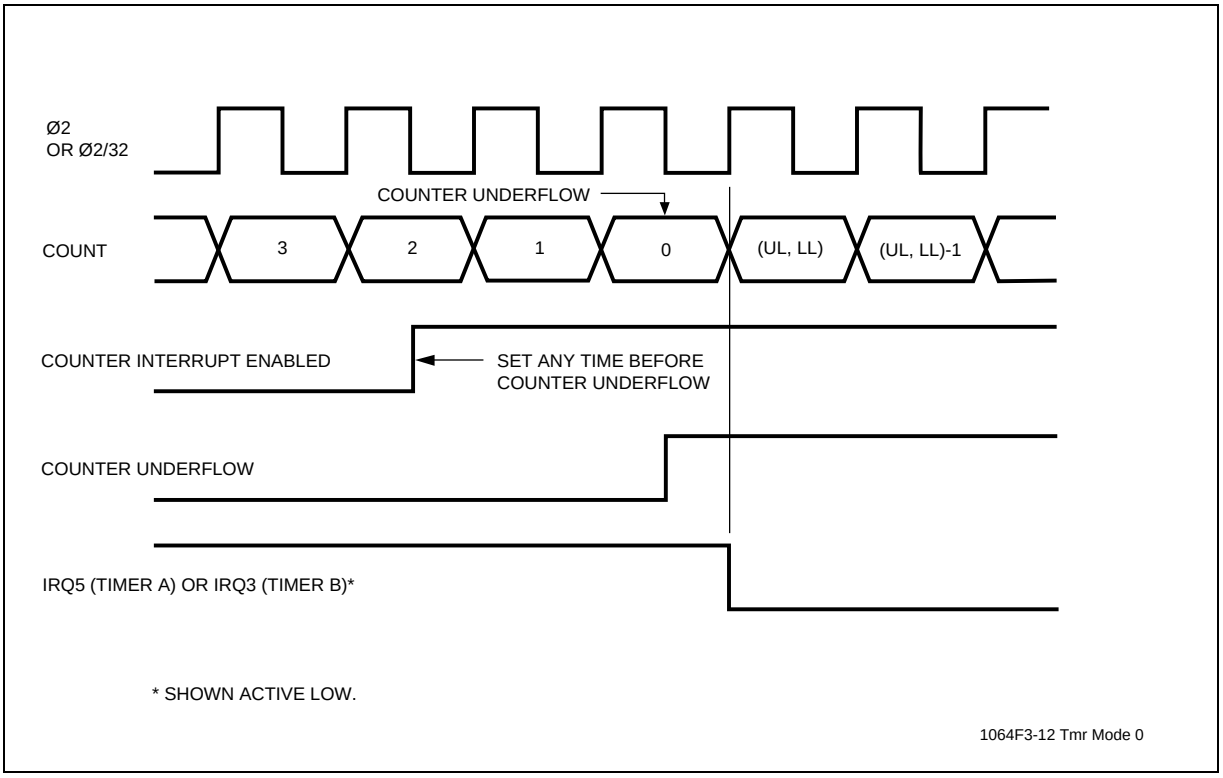


Figure 3-14. Timer Mode 0 (Interval Timer) Waveforms

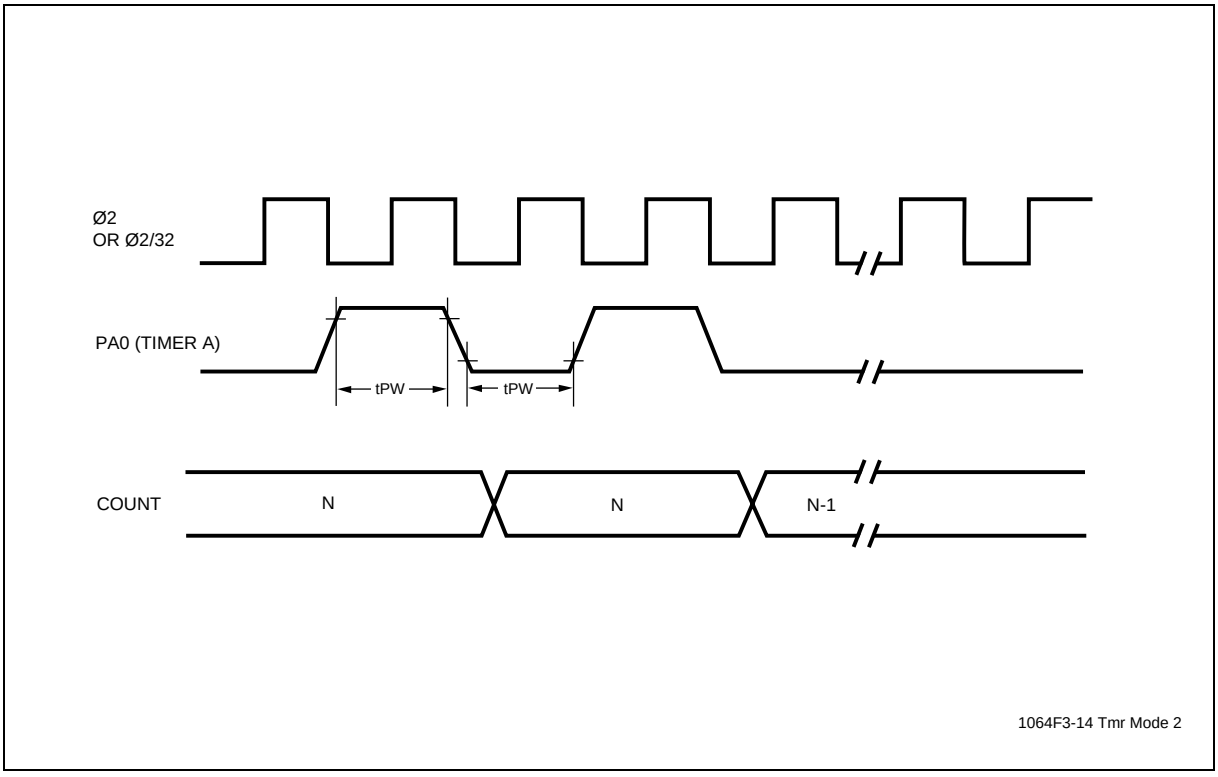


Figure 3-15. Timer Mode 2 (Event Counter) Waveforms

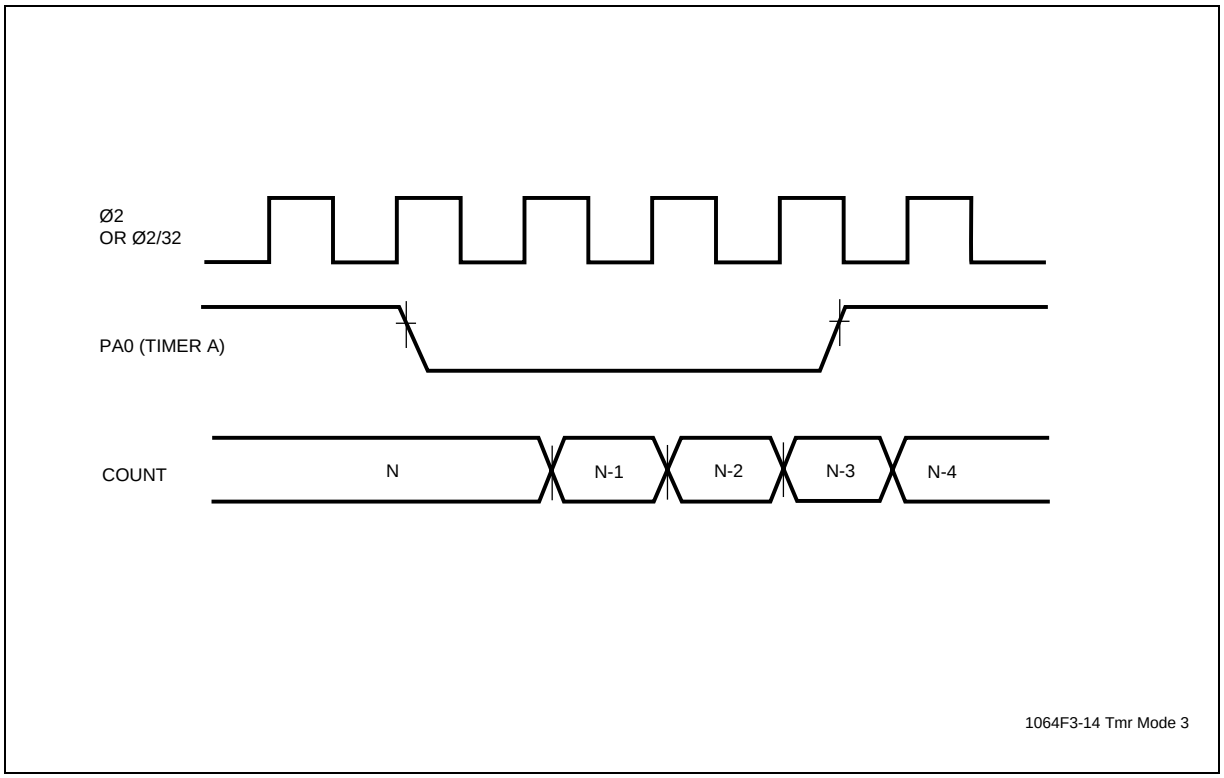


Figure 3-16. Timer Mode 3 (Pulse Width Measurement) Waveforms

3.11 PRECISION TIME GENERATORS

There are two identical 17-bit precision time generators: Precision Time Generator A (PTGA) and Precision Time Generator B (PTGB). Each PTG can be used for such functions as timing event interrupts, generating an external pulse train or as a source for synchronous USART timing.

Only PTGA is discussed in detail since both precision timer generators are identical in structure. Only the differences in I/O port addresses and IRQ interfaces are described. Block diagrams of PTGA and PTGB are shown in Figure 3-17 and Figure 3-18, respectively.

3.11.1 Precision Time Generator A

PTGA consists of five 8-bit registers and a 17-bit pulse accumulator (Figure 3-17). The three input registers—PTGA Buffer (PAB), PTGA Lower Latch (PALL) and PTGA Upper Latch (PAUL)—are all 8-bit. There are two output registers: an 8-bit PTGA Lower Residue (PALR) and a 9-bit PTGA Upper Residue (PAUR). The PTGA Accumulator (PAAC)is 17-bits long. Operation is controlled by the PTGA Mode Register (PAM) located at $0034.

The CPU can read or write the contents of PAB using address $0035. The CPU can read or write the contents of PAUL at either address $0036 or $0037. Whenever the CPU writes to $0036 or $0037 the contents of PAB are transferred to PALL. This allows a simultaneous 16-bit update of the input latches. The 17th input bit to the pulse accumulator is always a logic 0. When the CPU writes to address $0037, PAB is transferred to PALL and the new contents of PALL and PAUL are downloaded into PALR and PAUR, respectively, and the most significant bit of PAUR is set to a logic 0. This feature is particularly helpful during testing.

Operation of the precision time generator is governed by the equation:

$$Rate = Latch*(Ø2/2^{17})$$

where:

Rate = Pulse rate in Hz

Latch = Latch value

Ø2 = Internal clock rate in Hz

For example, if Ø2 = 4 MHz and the latch is loaded with 7550 ($107E), the resulting rate is 230408 Hz.

Conversely. the latch value can be computed using the equation

$$Latch = Rate*(2^{17}/Ø2)$$

For example, if the desired rate = 342857 Hz, ø2 = 6 MHz, the required latch value is 7489.8 ($1D42).

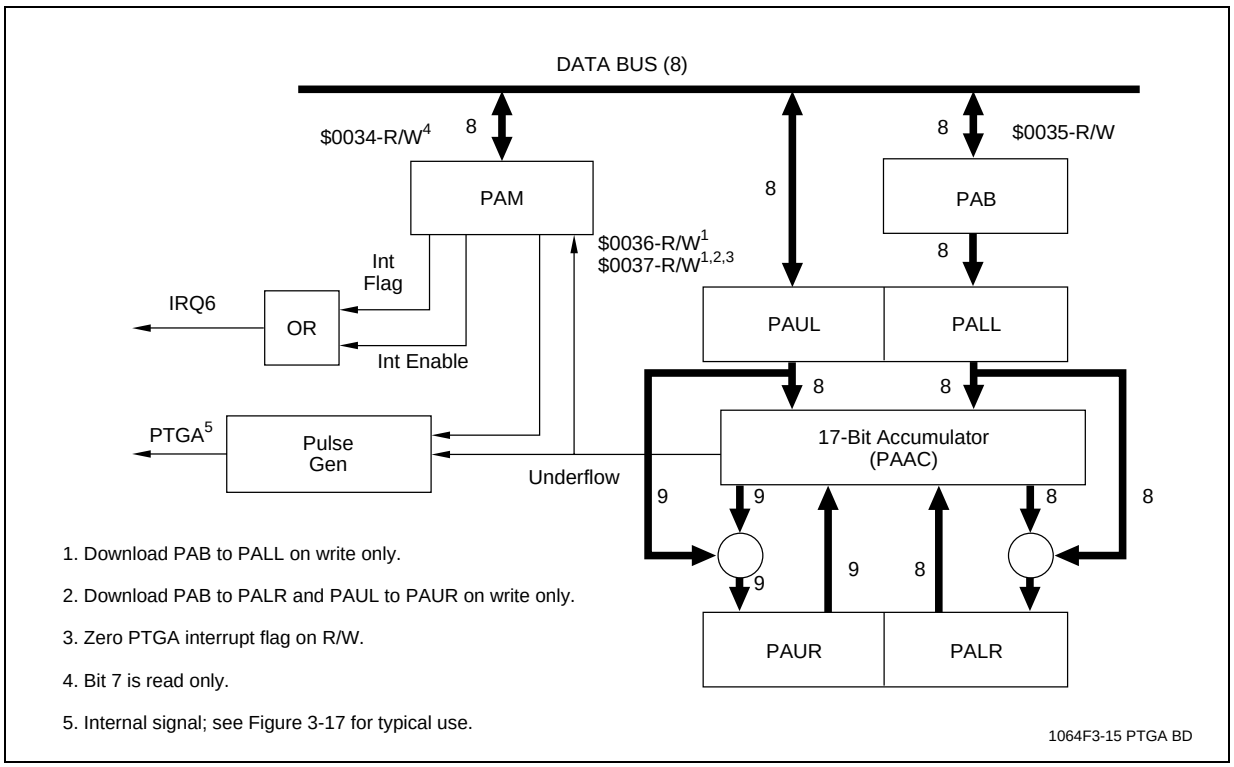


Figure 3-17. Precision Time Generator A Block Diagram

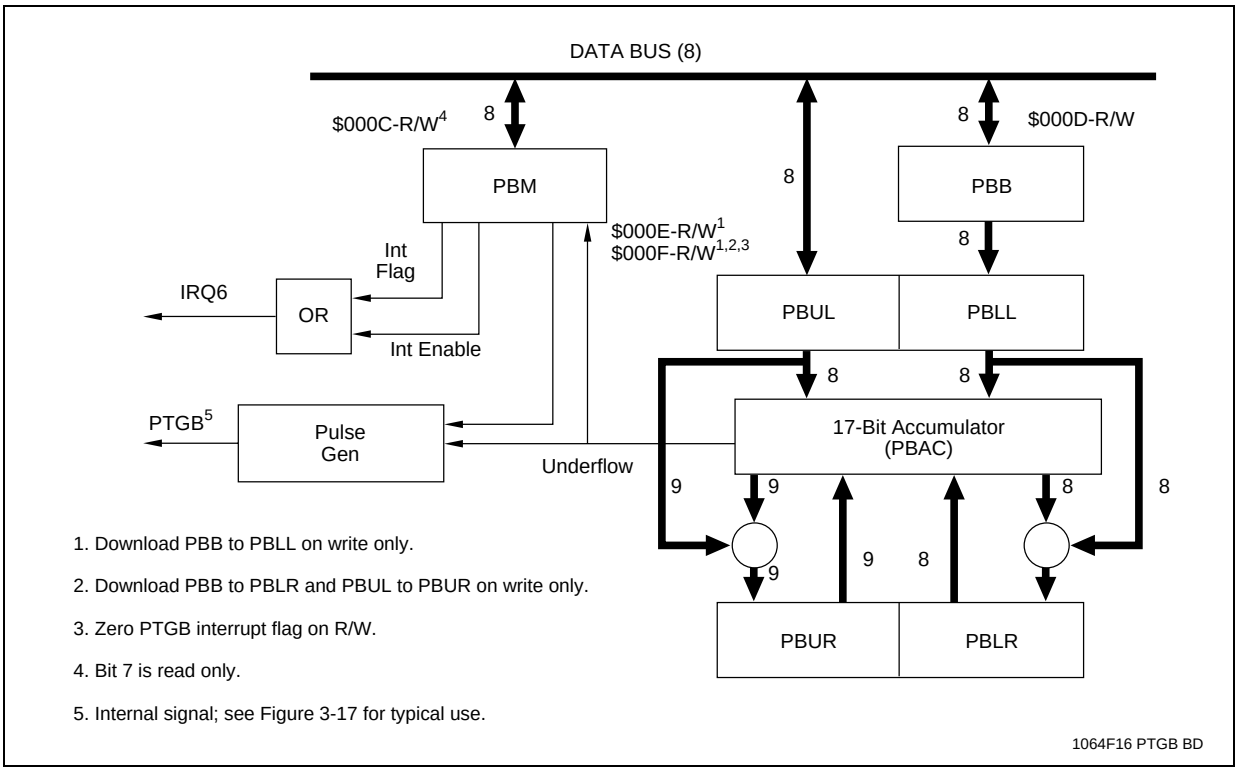


Figure 3-18. Precision Time Generator B Block Diagram

3.11.2 PTGA Mode Register (PAM)

The PTGA Mode Register (PAM) selects the PTGA timer and port options, and controls and reports the PTGA interrupt (Table 3-20). Bits 0, 1, and 6 are cleared by reset and can be set to a logic 1 or 0 by writing to address $0037. All bits can be read by the CPU.

Table 3-20. Register Bit Assignments: PTGA - 0034

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0034	Precision Time Generator A (PTGA) Mode (PAM)	PTG A Int. Flag 1 = Flag	PTG A Int. Enable 1 = Enable	Not Used				SIN (TXD) Select 1 = PTGA 0 = PTGB	PTGA Timer Mode 1 = Timer 0 = PTG
0035	PTGA Buffer (PAB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0036	PTGA Upper Latch (PAUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0037	PTGA Upper Latch (PAUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bit 0: **PTGA Timer Mode Select.** Control bit.

1 = PTGA operates as a timer.

0 = PTGA operates as a precision time generator.

Bit 1: **SIN (TXD) PTG Select.** Control bit.

1 = When SMR5 is a logic 1, selects PTGA as the input to the SINC Programmable Counter. The PTGA input may be divided by 3 (see SFR2).

0 = When SMR5 is a logic 1, selects PTGB as the input to the SINC Programmable Counter when SMR5 is a logic 1. The PTGB input may be divided by 3 (see SFR2).

Bit 2-5: **Not Used.**

Bit 6: **PTGA Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the PRM A Interrupt Flag (PAM7) is set to a logic 1. Note that PTGB can also generate IRQ6. If both precision time generator interrupts are enabled, the IRQ6 interrupt service subroutine should examine the interrupt flag bit in both PTGA and PTGB mode registers to determine the IRQ6 source.

0 = Disables IRQ6 assertion due to PAM7.

Bit 7: **PTGA Interrupt Flag.** Status bit, read only. Reading or writing to address $0037 clears the PTGA Interrupt Flag (PAM7).

1 = PTGA accumulator overflow has occurred.

0 = PTGA accumulator overflow has not occurred.

3.11.3 Precision Time Generator B

PTGB consists of five 8-bit registers and a 17-bit pulse accumulator (Figure 3-18). The three input registers—PTGB Buffer (PBB), PTGB Lower Latch (PBLL) and PTGB Upper Latch (PBUL)—are all 8-bit. There are two output registers: an 8-bit PTGB Lower Residue (PBLR) and a 9-bit PTGB Upper Residue (PBUR). The PTGB Accumulator (PBAC) is 17-bits long. Operation is controlled by the PTGB Mode Register (PBM) located at $000F.

Operation of PTGB is identical to PTGA with the exception of register addresses, and the port B interface line (Figure 3-18).

3.11.4 PTGB Mode Register (PBM)

The PTGB Mode Register (PBM) selects the PTGB timer and port options, and controls and reports the PTGB interrupt (Table 3-21). Bits 0, 1, and 6 are cleared by reset and can be set to a logic 1 or 0 by writing to address $001C. All bits can be read by the CPU. Bit 7 is read only.

Table 3-21. Register Bit Assignments: PTGB - 000Ch

Addr.	Function	Bit 7	6	5	4	3	2	1	0
000C	Precision Time Generator B (PTGB) Mode (PBM)	PTG B Int. Flag 1 = Flag	PTG B Int. Enable 1 = Enable	Not Used					PTG B Timer Mode 1 = Timer 0 = PTG
000D	PTGB Buffer (PBB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
000E	PTGB Upper Latch (PBUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
000F	PTGB Upper Latch (PBUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bit 0: **PTGB Timer Mode Select.** Control bit.

1 = PTGB operates as a timer.

0 = PTGB operates as a precision time generator.

Bit 1-5: **Not Used.**

Bit 6: **PTGB Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the PRM B Interrupt Flag (PBM7) is set to a logic 1. Note that PTGA can also generate IRQ6. If both precision time generator interrupts are enabled, the IRQ6 interrupt service subroutine should examine the interrupt flag bit in both PTGA and PTGB mode registers to determine the IRQ6 source.

0 = Disables IRQ6 assertion due to PBM7.

Bit 7: **PTGB Interrupt Flag.** Status bit, read only. Reading or writing to address $001F clears the PTGB Interrupt Flag (PBM7).

1 = PTGB accumulator overflow has occurred.

0 = PTGB accumulator overflow has not occurred.

3.11.5 Example Rates

Some examples of standard rates are shown in Table 3-22.

Table 3-22. PTGA or PTGB Generated Standard Data Rates

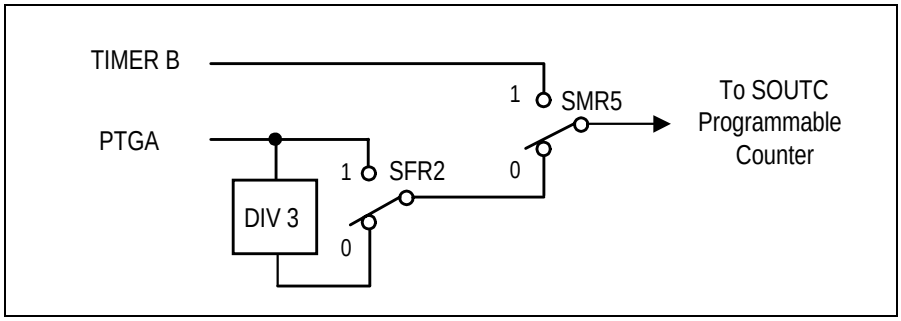
Rate	Ø2 = 6 MHz		Ø2 =8 MHz		Ø2 = 28.224 MHz	
	Latch (Hex)	Actual	Latch (Hex)	Actual	Latch (Hex)	Actual
576 KHz	3127	576004.03 Hz	24DD	575988.77 Hz	0A72	575797.31 Hz
500 KHz	2AAB	500015.26 Hz	2000	500000.00 Hz	0911	499786.67 Hz
250 KHz	1555	249984.74 Hz	1000	250000.00 Hz	0488	249785.16 Hz
230.4 KHz	13A9	230392.46 Hz	0EBF	230407.71 Hz	042D	230189.94 Hz
200 KHz	1111	199996.95 Hz	0CCD	200012.21 Hz	03A0	199828.13 Hz
100 KHz	0889	100021.36 Hz	0666	99975.59 Hz	01D0	99914.06 Hz

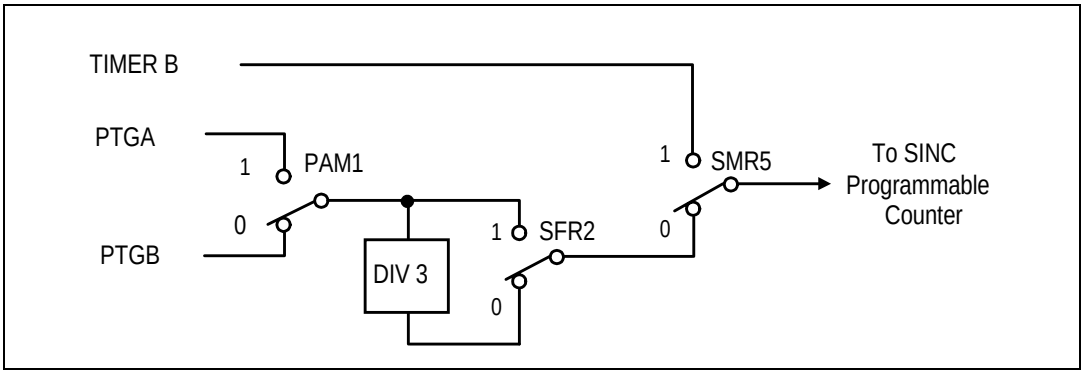
3.12 USART

The MCU provides a full-duplex serial universal synchronous/asynchronous receiver/transmitter (USART) interface with programmable operating modes and data rates. Serial-to-parallel conversion is performed on data characters received from an external device and parallel-to-serial conversion is performed on data characters received from the MCU internal data bus. A block diagram of the USART is shown in Figure 3-19.

3.12.1 Enhancements over the L39

The MCU allows the user additional flexibility in selecting the timing sources for the USART. This is illustrated in the two figures shown below for generating the clock sources for SINC and SOUTC PROGRAMMABLE COUNTERS. As before, bit 5 of the Serial Mode Control Register (SMR5) selects between Timer B and Pulse Rate Multiplier timing. A new control, bit 2 of the Serial Form Register (SFR2), allows the user to bypass the divide-by-3 counter in the pulse rate multiplier path. When SFR2 is set to 0, the user can select PRMA to source the SINC Programmable Counter by setting bit 1 of the PRMA Mode Control Register (PAM1). This causes PRMA to source both SINC and SOUTC timing, freeing PRMB for other tasks.





3.12.2 General Operation

Internal timing for both asynchronous and synchronous operation can be referenced to either Timer B or the Precision Timing Generators under software control. Synchronous transmit data (TXD) timing can also be derived externally by an external transmit clock (TXCLK) input on PA3 or an external transmit reference clock (TXREF) input on PA4. Synchronous received data (RXD) timing can be generated from an external receive clock (RXCLK) input on PA7. Note that the direction registers for PA1 through PA7 (PAD1 - PAD7) must be set correctly for the mode selected. Table 3-23 shows how standard data rates can be generated internally using either Timer B or the Precision Timing Generators A and B.

The serial interface registers are located at addresses $0034-$003F (Table 3-24). The CPU may read or write any of the serial interface registers with the exception of Serial Out Divider Latch (SODL) and Serial In Divider Latch (SIDL) which are write only. The Serial Status Register is read-only. Reading and/or writing to some of the registers also causes clearing of interrupt bits or data downloading actions.

3.12.3 Internal Timing

Since internal timing for TXD and RXD is similar, the following discussion covers only TXD timing. It differs only when precision timing generators (PTGs) are selected. TXD uses PTGB and RXD uses PTGA.

Case 1 - Asynchronous Timer B. Whenever short stop bits are selected (SFR0 = 1 or SFR1 = 1), or 5-bit operation is selected (SLCR0 = 0 and SLCR1 = 0), the user must program the SOUT (RXD) Divider Latch (SODL) to $0F. This restriction does not apply to the SIN (TXD) Divider Latch (SIDL).

$$UIB = øB/[(n + 1)(l + 1)]$$

where UIB is the TXD bit rate, øB is the internal clock rate of Timer B, n is the decimal value loaded into the Timer B Latch, and l is the decimal value loaded into the SIDL latch. If ø2 = 6 MHz and the DIV BY 32 option for Timer B is not selected, øB = 6 MHz. If n = 155 ($009B) and l = 15 ($0F), then

$$UIB = 6\ MHz/[(155 + 1)(15 + 1)] = 2403.85\ Hz.$$

Case 2 - Synchronous Timer B. An additional DIV BY 2 is required to generate the 50% duty cycle TXCLK (PA3) required for synchronous operation. Using the same values as in Case 1,

$$UIB = øB/[2(n + 1)(l + 1)] = 6\ MHz/[2\ (155 + 1)(15 + 1)] = 1201.92\ Hz$$

Case 3 - Asynchronous PTGB. Assume that PTGB is generating a rate of 230.4 KHz and that the ø2 internal timing is 6 MHz. From Table 3-23, the PTGB latch is loaded with $13A9. Also, from Table 3-23, PTGB = 230.39246 KHz. If l = 15 ($0F), then:

$$UIB = PTGB/\{[3(l + 1)]\} = 230392.46/[3\ (15 + 1)] = 4799.84\ Hz$$

This assumes that the USART DIV BY 2 mask option (Figure 3-19) is selected. If short stop bits or 5-bit operation is selected, the user must program the SOUT Divider Latch (SODL) to decimal 15 ($0F).

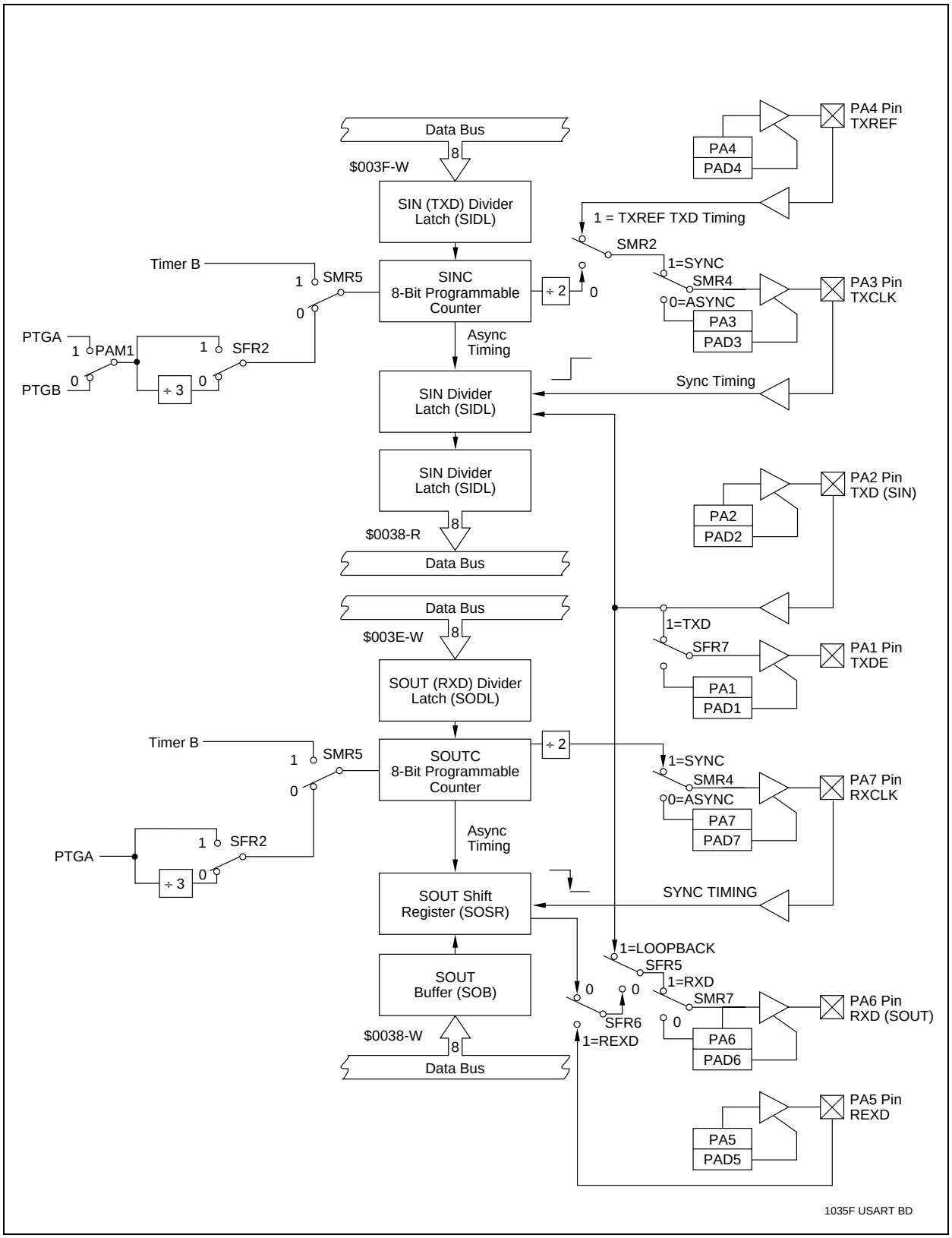


Figure 3-19. USART Block Diagram

Table 3-23. USART Generated Standard Data Rates

Standard Data Rate	Asynchronous, Ø2 = 6 MHz (SIDL = SODL = $06)				Synchronous, Ø2 = 6 MHz			
	Timer B		PTGA and PTGB		TIMB Latch = 009B, Rate = 38461.54 Hz PTG Rate = 230.4 KHz, PTG Latches = $13A9			
	TIMB Latch	Bit Rate	PTG Latch	Bit Rate	SIDL, SODL	Bit Rate	SIDL, SODL	Bit Rate
50 Hz	$1D4B	50.00	$0034	49.59	–	–	–	–
75 Hz	$1387	75.00	$004F	75.34	$FF	75.12	–	–
110 Hz	$0D50	110.00	$0073	109.76	–	–	–	–
150 Hz	$09C3	150.00	$009D	149.73	$7F	150.24	$FF	150.00
300 Hz	$04E1	300.00	$013B	300.41	$3F	300.48	$7F	299.99
600 Hz	$0270	600.00	$0275	599.86	$1F	600.96	$3F	599.98
1200 Hz	$0137	1201.92	$04EA	1199.72	$0F	1201.92	$1F	1199.96
2400 Hz	$009B	2403.85	$09D5	2400.40	$07	2403.85	$0F	2399.92
4800 Hz	$004D	4807.69	$13A9	4799.84	$03	4807.69	$07	4799.84
9600 Hz	$0026	9615.38	$2752	9599.69	$01	9615.38	$03	9599.69
14400 Hz	$0019	14423.08	$3AFB	14399.53	–	–	–	–
19200 Hz	$0013	18750.00	$4EA5	19200.32	$00	19230.77	$01	19199.37
38400 Hz							$00	38398.74

Case 4 - Synchronous PTGB. An extra DIV BY 2 also occurs in the synchronous mode. Using the same setup as in Case 3 and again assuming that the USART DIV BY 2 mask option (Figure 3-19) is selected.

UIB = PTGB/2[3 (I + 1)]= 230392.46/[6 (15 + 1)]= 2399.92 Hz.

3.12.4 USART Registers and Serial Buffer Registers (SB)

The USART control and status registers are shown in Table 3-24. The serial buffer (SB) registers provide double buffering for serial in and serial out data. Both buffers are located at address $0038. The Serial In Buffer is a read-only register and the Serial Out Buffer is a write-only register. All data is shifted least significant bit first. When odd or even parity is used with 5-, 6- or 7-bit character operation, the unused most significant bits in the Serial Out Buffer must be loaded with zeros. Zeros are automatically inserted in the Serial In Buffer.

Table 3-24. Register Bit Assignments: USART - 0038h - 003Fh

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0038	Serial In Buffer (SIB) / Serial Out Buffer (SOB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0039	Serial Interrupt Register (SIR)	TXD Status Int. Flag 1 = Flag	TXCLK (PA3) ↑ Int. Flag 1 = Flag	RXCLK (PA7) ↓ Int. Flag 1 = Flag	TXCLK (PA3) ↑ Int. Enable	RXCLK (PA7) ↓ Int. Enable	TXD Status Int. Enable 1 = Enable	RXD Buffer Empty Int. Enable (BE) 1 = Enable	TXD Buffer Full Int. Enable (BF) 1 = Enable
003A	Serial Mode Register (SMR)	RXD On	TXD On	Timing Select 1 = TIMB 0 = PRG	Sync Mode	TXD Sync Bit	TXREF Clock Select	CTSP/ RLSDP Sync Enable	Not Used
003B	Serial Line Control Register (SLC)	Parity Stuff Bit	Set Break	Stuff Parity	Even Parity	Enable Parity	Two Stop Bits	Word Length Bit 1 (SL1)	Word Length Bit 0 (SL0)
003C	Serial Status Register (SSR)	TXD Parity Bit	RXD Underrun (UR)	RXD Buffer Empty (BE)	TXD Break Int. (BI)	TXD Framing Error (FE)	TXD Parity Error (PE)	TXD Overrun Error (OE)	TXD Buffer Full (BF)
003D	Serial Form Register (SFR)	TXD/ TXDE Echo	REXD/TXD Echo	TXD/RXD Echo	TXD to TIMA Input Test	TXD to PA↑ Edge Test	Not used	7/8 Short Stop Bit	3/4 Short Stop Bit
003E	SOUT (RXD) Divider Latch (SODL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
003F	SIN (TXD) Divider Latch (SIDL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0

3.12.5 Serial Mode Register (SMR)

The Serial Mode Register (SMR), located at $003A, controls basic serial mode and timing selection. All bits are cleared to zero by reset and can be set or reset by the CPU. Bits 6 and 7 enable the TXD and the RXD modes, respectively, for both asynchronous and synchronous modes of operation. Bits 2-4 control synchronous functions. Bits 2-4 must be set to zero for asynchronous operation.

Bits 0: Not used.

Bits 1: **CTSP/RLSDP Enable.** Control bit.

1 = When TXD Mode is selected (SMR6 = 1), the USART TXD synchronous mode word counter is initialized upon detection of a negative edge on PA0 (normally connected to CTSP). When RXD Mode is selected (SMR7 = 1), the USART RXD synchronous mode word counter is initialized upon detection of a negative edge on PA4 (normally connected to RLSDP).

0 = When set to a logic 0, PA0 and PA4 are not used for USART support.

Bit 2: **Select TXREF Source.** Control bit.

1 = Selects the TXREF on PA4 clock for TXD timing.

0 = Selects internal TXD timing.

Bit 3: **Enable TXD Sync Detect.** Control bit.

1 = TXD data line is monitored for a high-to-low (1-0) character transition. This transition causes serial in word synchronization following the 0 character and resets bit SMR3.

0 = TXD data line is not monitored for a high-to-low (1-0) character transition.

Bit 4: **Sync Mode Select.** Control bit.

1 = Selects synchronous mode operation; PA7 is assigned to RXCLK and PA3 is assigned to TXCLK.

0 = Selects asynchronous mode operation; PA7 and PA3 are general purpose I/O pins.

Bit 5: **Internal Timing Reference.** Control bit.

1 = Selects Timer B for both TXD (SIN) and RXD (SOUT).

0 = Selects PTGA or PTGB for TXD as selected by the SIN (TXD) PTG Select bit in the PTGA Mode Register (PAM1) and selects PTGA for RXD. This allows PTGA to be selected for both TXD and RXD timing thus freeing PTGB for other use. The input from the PTG can be divided by 3 as selected by the PTG Divide by 3 bit in the Serial Form Register (SFR2).

Bit 6: **TXD Mode Select.** Control bit.

1 = TXD timing and shift register are operational. PA2 is assigned to TXD.

0 = PA2 is not assigned to TXD.

Bit 7: **RXD Mode Select.** Control bit.

1 = RXD timing and shift register are operational. PA6 is assigned to RXD.

0 = PA6 is not assigned to RXD.

3.12.6 Serial Interrupt Register (SIR)

The Serial Interrupt Register (SIR) is located at address $0039. The SIR contains five enable bits (bits 0-4) and three interrupt flag bits (bits 5-7). The CPU can read all bits but can write only to the five interrupt enable bits. All SIR bits are cleared to zero by reset. An interrupt enable bit, when set to a logic 1, permits the corresponding condition to assert the associated IRQ.

Bit 0: **TXD (Serial In) Buffer Full Interrupt Enable.** Control bit.

1 = Enables IRQ4 to be asserted when the Serial In Buffer Full bit (SSR0) is a logic 1.

0 = Disables IRQ4 assertion based on SSR0.

Bit 1: **RXD (Serial Out) Buffer Empty Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the Serial Out Buffer Empty bit (SSR5) is a logic 1.

0 = Disables IRQ6 assertion based on SSR5.

Bit 2: **Serial In Status Interrupt Enable.** Control bit.

1 = Enables IRQ5 to be asserted when the Serial In Status bit (SIR7) is a logic 1.

0 = Disables IRQ5 assertion based on SIR7.

Bit 3: **PA7 Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the PA7 Interrupt Flag (SIR5) is a logic 1.

0 = IRQ6 will not be asserted based on SIR5.

Bit 4: **PA3 Interrupt Enable.** Control bit.

1 = Enables IRQ4 to be asserted when the PA3 Interrupt Flag (SIR6) is a logic 1.

0 = Disables IRQ4 assertion based on SIR6.

Bit 5: **PA7 Interrupt Flag.** Control bit.

1 = A high-to-low transition has been detected on the PA7. RXCLK is connected to this pin during synchronous serial output operation. Writing a 0 to bit 4 of the Clear Interrupt Register clears this bit to a logic 0.

0 = A high-to-low transition has not been detected on the PA7.

Bit 6: **PA3 Interrupt Flag.** Status bit.

1 = A low-to-high transition has been detected on PA3 (SFR3 = 0) or on PA2 (SFR3 = 1). TXCLK is normally connected to PA3 and TXD is normally connected to PA2 in synchronous serial input operation. Writing a 0 to bit 3 of the Clear Interrupt Register clears this bit to a logic 0.

0 = A low-to-high transition has not been detected on PA3 (SFR3 = 0) or on PA2 (SFR3 = 1).

Bit 7: **Serial In Status Interrupt Flag.** Status bit.

1 = Any of the following three flags have been set to a logic 1 in the Serial Status Register (SSR): Framing Error (SSR3), Parity Error (SSR2), or Overrun Error (SSR1). The Serial In Status Interrupt Flag (SIR7) is also set to a logic one whenever the Break Interrupt (SSR4) changes state, i.e., whenever it changes from a 0 to a 1 or from a 1 to a 0. SIR7 is reset by writing to the Serial Status Register (SSR) at address $003C. Writing to $003C clears SIR7 but does not alter any bits in the Serial Status Register (SSR).

0 = None of the following three flags have been set to a logic 1 in the Serial Status Register (SSR): Framing Error (SSR3), Parity Error (SSR2), or Overrun Error (SSR1).

3.12.7 Serial Line Control Register (SLCR)

The Serial Line Control Register (SLCR) at $003B specifies the word length and parity generation and checks in asynchronous mode. Each bit can be set or reset by the CPU. All SLCR bits are initialized to zero by reset. The SLCR must be set to $03 for 8-bit synchronous operation. The Parity Enable bit (SLCR3) must be a zero in the synchronous mode. Both asynchronous and synchronous modes support 5-, 6-, 7-, and 8-bit word lengths.

Bits 0-1: **Word Length.** These two control bits specify the number of bits in each serial in or serial out character. The encoding of bits 0 and 1 is:

Bit 1	Bit 0	Word Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

Whenever 5 bits are selected in the asynchronous mode the transmitter will generate 1 and 1/2 stop bits. This requires that the SOUT Divider Latch (SODL) be loaded with a decimal 15 ($0F).

Bit 2: **Number Stop Bits.** Control bit.

1 = One and a half stop bits are generated when a 5-bit word length is selected; two stop bits are generated when a 6-, 7-, or 8-bit word length is selected.

0 = One stop bit is generated regardless of word length.

Bit 3: **Enable Parity.** Control bit.

1 = Enables a parity bit to be inserted in the serial out data stream and to be checked for in the serial in data stream. The parity bit is located between the last data bit and the first stop bit.

0 = Disables parity generation and checking.

Bit 4: **Even Parity Select.** Control bit.

1 = When parity is enabled (SLCR3 = 1) and parity stuff is disabled (SLCR5 = 0), a one or zero is automatically inserted into the serial out parity position such that the total number of ones in the data and parity fields is even.

0 = When parity is enabled (SLCR3 = 1) and parity stuff is disabled (SLCR5 = 0), a one or zero is automatically inserted into the serial out parity position such that the total number of ones in the data and parity fields is odd.

Bit 5: **Enable Parity Stuff.** Control bit.

1 = When parity is enabled (SLCR3 = 1) and parity stuff is enabled (SLCR5 = 1), the parity stuff bit value (SLCR7) is copied into the serial out parity position.

0 = The parity stuff bit value (SLCR7) is never copied into the serial out parity position.

Bit 6: **Set Break.** This bit is the Break Control bit. The Break Control bit acts only on RXD and has no effect on the serial in logic.

1 = The received data output (RXD) is forced to the space (logic 0) state.

0 = The break is disabled, i.e., the received data output (RXD) is normal.

Bit 7: **Parity Stuff Bit.** Control bit.

1 = A logic 1 is copied into the RXD parity bit when both parity is enabled (SLCR3 =1) and parity stuff is enabled (SLCR5 = 1).

0 = A logic 0 is copied into the RXD parity bit when both parity is enabled (SLCR3 =1) and parity stuff is enabled (SLCR5 = 1).

3.12.8 Serial Status Register (SSR)

The Serial Status Register (SSR) at $003C provides serial-in and serial-out status to the CPU. It is a read-only register. All bits are initialized by reset to a logic 0, except Serial Out Buffer Empty (SSR5), which is initialized to a logic 1. Writing to the Serial Status Register will not alter any bits in the register, but it will reset the Serial In Status Interrupt Flag (SIR7).

Bit 0: **Serial In Buffer Full (BF).** Status bit.

1 = A complete incoming character, asynchronous or synchronous, has been received and transferred to the Serial In Buffer. This bit is reset to a logic 0 when the CPU reads the Serial In Buffer.

0 = A complete incoming character, asynchronous or synchronous, has not been received and transferred to the Serial In Buffer.

Bit 1: **Overrun Error (OE).** Status bit.

1 = Data in the Serial In Buffer was not read by the CPU before the next received character was transferred into the Serial In Buffer, thereby destroying the previous character. This bit is reset as soon as a received data character can be safely transferred into an empty Serial In Buffer.

0 = A complete incoming character, asynchronous or synchronous, has not been received and transferred to the Serial In Buffer.

Bit 2: **Parity Error (PE).** Status bit.

1 = The received data character does not have the correct even or odd parity, as selected by the Even Parity Select bit (SLCR4). The PE bit is set to a logic 1 upon detection of a parity error. It is reset to a logic 0 as soon as a received data character is found with a correct parity bit.

0 = The received data character has a correct parity bit.

Bit 3: **Framing Error (FE).** Status bit.

1 = The received character did not have a valid stop bit. The FE is set to a logic 1 whenever the stop bit following the last data bit or parity bit is detected as a zero bit (space level). The FE bit is reset as soon as a received data character is found with a correct first stop bit.

0 = The received data character has a correct first stop bit.

Bit 4: **Break Interrupt (BI).** Status bit.

1 = The received data input (serial in) is held in the space (logic 0) state continuously from the start bit to the first stop bit. The BI bit is cleared when a mark bit (logic 1) is detected on TXD.

0 = A mark bit (logic 1) is detected on TXD.

Bit 5: **Serial Out Buffer Empty (BE).** Status bit.

1 = The Serial Out Buffer is empty and ready to accept a new character for transmission. When this bit is true simultaneously with the Serial Out Interrupt Enable bit (SIR1), IRQ6 is asserted. The BE flag is set to a logic 1 when the contents of the Serial Out Buffer are transferred into the Serial Out Shift Register. BE is reset to a logic 0 when the CPU writes to the Serial Out Buffer.

0 = The Serial Out Buffer is not empty

Bit 6: **Serial Out Underrun (UR).** Status bit.

1 = The Serial Out Shift Register has emptied and the Serial Out Buffer has not been reloaded by the CPU. When the UR bit is a logic 1, the Serial Out Shift Register will output mark bits (logic 1). It is reset to a logic 0 when the CPU writes to the Serial Out Buffer.

0 = Serial out underrun has not occurred.

Bit 7: **Serial In Parity Bit.** This bit copies the received serial in parity bit.

3.12.9 Serial Form Register (SFR)

The Serial Form Register (SFR), located at $003D, controls TXD testing and specifies special routing of serial signals and stop bit length adjustment. Bits 5-7 can support local and remote loopback operation. The CPU can read and write all SFR bits except bit 2.

Bits 0-1: **Short Stop Bits.** When set, these bits shorten the width of the final transmitter stop bit. A normal stop bit width occurs when both bits are set to 0. Stop bit control requires the user to load the Serial Out Divider Latch (SODL) with a decimal 15 ($0F).

Bit 1	Bit 0	Final Stop Bit Width 5-bit Word Length, 2 Stop Bit Mode	Final Stop Bit Width All Other Modes
0	0	1/2 Bit	1.0 Bit
0	1	1/4 Bit	3/4 Bit
1	0	3/8 Bit	7/8 Bit
1	1	1/8 Bit	5/8 Bit

Bit 2: **Not Used**

Bit 3: **TXD to PA3 Positive Edge Test.** Control bit.

1 = PA2 (TXD) is routed to the PA3 positive edge detect logic (see SI3).

0 = PA3 (TXCLK) is routed to the PA3 positive edge detect logic.

Bit 4: **TXD to Timer A Input Test.** Control bit.

1 = PA2 (TXD) input is routed to the Timer A input detect logic. SFR3 and SFR4, when set, can be used to easily measure the TXD start pulse width.

0 = PA0 is routed to the Timer A input detect logic.

Bit 5: **Loopback.** Control bit.

1 = PA2 (TXD) is routed back to RXD instead of the normal serial out data.

0 = PA 6 (RXD) will copy the SOUT shift register when SFR6 = 0 or SOP (PA6) when SFR6 = 1.

Bit 6: **Serial Out Passthrough on PA5.** Control bit

1 = The serial out input on PA5 is routed to RXD output on PA6. Note that serial out mode must be selected (SMR7 = 1) and serial loopback must not be selected (SFR5 = 0).

0 = PA5 is a general purpose I/O line.

Bit 7: **Serial In Passthrough on PA1.** Control bit.

1 = PA2 (TXD) input on is routed to the serial in output on PA1.

0 = PA1 is a general purpose I/O line.

3.12.10 Serial Out (RXD) Divider Latch (SODL)

The Serial Out Divider Latch (SODL) is a CPU write-only register at $003E that controls operation of the RXD (Figure 3-19). The SODL must be loaded with the value corresponding to the desired data. Table 3-23 contains standard values for asynchronous and synchronous operation. Note that the SODL must be loaded with $0F for asynchronous operation whenever Short Stop Bit operation is required. Each time that the CPU writes to the SODL, the new latch value is downloaded to the counter.

3.12.11 Serial In (TXD) Divider Latch (SIDL)

The Serial In Divider Latch (SIDL) is also a CPU write-only register at $003F that controls operation of the TXD timing (Figure 3-19). Its operation is similar to that of the SODL, except that it is not restricted to a value of $0F for asynchronous operation.

3.13 DUAL PORT RAM - 16550A/16450 INTERFACE

The Dual Port RAM interface can be configured to operate a 16550A register compatible mode with transmit/receive 16-byte FIFOs or a 16450 register compatible mode.

The MCU/host interface diagram is shown in Figure 3-20. Signal equivalence between the MCU and each mode is shown in Table 3-25.

The selectable host bus interface provides a 6500 or RDP/WTP bus compatible interface between the MCU and a host microprocessor. This interface allows the MCU to act like a standard peripheral device connected to the host bus under control of the host processor. Under MCU software control, this interface can emulate a 16550A/16450 UART interface.

Built-in hardware registers and control signals allow a 16550A/16450 UART compatible interface to be presented to the host bus. More supporting 16550A/16450 interface functions are provided in MCU hardware than in earlier models in order to relieve the MCU application firmware from having to provide some time critical and overhead functions when servicing the interface. Supporting MCU application firmware functions are required to fully implement the 16550A/16450A interface, however.

3.13.1 Host Bus Interface Signals and Registers

When the host bus interface is selected (HCR2 = 1), the following host bus signals are supported instead of the general purpose I/O lines on ports A (4 lines), B (2 lines), C (8 lines), and D (7 lines):

8-bit bidirectional data lines (HD0-HD7)

4-bit address inputs (HA0-HA3), HA3 remains a GP I/O in the 16550A/16450A mode.

1 chip select input (HCSP)

2 bus timing inputs (HWTP and HRDP, or Hø2 and HR/WP)

1 host interrupt output (HINT)

2 data ready outputs (RXRDY and TXRDY) if 16550A/16450 DP RAM is selected (HCR1 = 1), FIFO is enabled (FCR0 = 1), and serial mode is not selected (SMR7 = 0 and SMR4 = 0)

2 data acknowledge inputs (RXACKP and TXACKP) if 16550A/16450 DP RAM is selected (HCR1 = 1), FIFO is enabled (FCR0 = 1), and serial mode is not selected (SMR7 = 0 and SMR4 = 0)

The host bus waveforms and timing are described in Section 5.

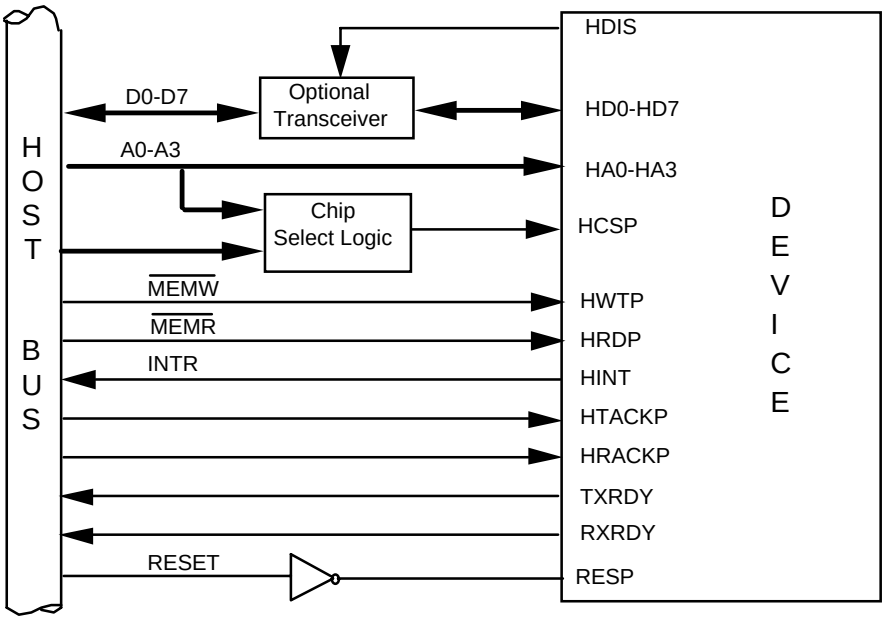


Figure 3-20. MCU/Host Interface

Table 3-25. MCU Dual Port RAM Signal Equivalence

MCU Port Name	MCU Signal Name	16450 Mode	16550A Mode
PB6	HDIS*	DDIS	DDIS
PB7	HINT	INTR	INTR
PC0-PC7	HD0-HD7	D0-D7	D0-D7
PD0-PD2	HA0-HA2	A0-A2	A0-A2
PD3	PWRDWNP (Internal)	-	-
PD4	MCSP (HCSP) (Internal)	CS2P	CS2P
PD5	HWTP	WTP	WTP
PD6	HRDP	RDP	RDP
PA3	HTACKP	HTACKP	HTACKP
PA4	HRACKP	HRACKP	HRACKP
PA5	TXRDY	TXRDY	TXRDY
PA6	RXRDY	RXRDY	RXRDY
RESP	RESP	NR	MR

* Not recommended for use.

3.13.2 16550A Interface Mode and FIFOs

When the FIFO mode is selected (HCR1 = 1, HCR2 = 1, and FCR0 = 1), the MCU provides a 16550A register compatible interface mode. The 16550A interface registers are described in Table 3-26 and Table 3-27. The 16550A interface uses the hardware signals shown in Figure 3-20 except the A3 signal is not used. The operation of the FIFOs is illustrated in Figure 3-21, Figure 3-22, and Figure 3-23.

FIFO UART Timing Simulator. The FIFO UART Timing Simulator (Figure 3-22) provides both an RUCLK and a TUCLK to simulate the 16550 UART word rate. These rates are driven by either TIMB or PTGB through divide by 16 counters.

RUCLK is automatically re-synchronized to one sixteenth of the word rate by either the termination of Freeze (FSR5) or by the first MCU RX FIFO write to occur when both RUPTR and RXPTR are empty. TUCLK will be re-synchronized by the first host write to the TX FIFO to occur when both TUPTR and TXPTR are empty. When the host writes to an empty FIFO the first TUCLK does not occur until a UART delay has occurred. Note that host can read data from the RX FIFO and write data to the TX FIFO faster than the baud rate by monitoring the 16550 interface interrupt flags rather than waiting for the baud clock to elapse. This timer “override” feature allows much faster data throughput rates than normal baud timing. Note also that when using this technique, the buffers must be completely emptied for the MCU to properly re-synchronize upon receipt of the next byte at the word rate.

TX FIFO. In the 16550A mode, the host can burst data into the TX FIFO Buffer until the TX FIFO is full. This can be accomplished under host control or by a supported DMA mode using TXRDY and TXACKP lines.

Both TXPTR and TUPTR advance together as the host fills the TX FIFO. The CPU unloads the TX FIFO using TXPTR. TUPTR simulates the UART by decrementing the TUCLK rate until TX FIFO is emptied. This causes THRE to be asserted which signals the host that the TX FIFO is empty. In the event the host does not write to the TX FIFO before the next TUCLK, TEMT is asserted. Two interrupt request conditions with enables are provided to the CPU: TCDA (TX FIFO data available) and TCHF (TX FIFO half full). If the TX FIFO contains between one and seven bytes of data and TCHF is enabled and neither the host or CPU have accessed the TX FIFO for three or four TUCLK intervals, a TCTO character time-out interrupt is provided to the CPU. Both TXPTR and TUPTR are reset whenever the TX FIFO bit (FCR2) is set to a 1 by the host, or whenever FIER3 is set to a 1 by the CPU, or, whenever the host changes the state of FCR0. These reset controls are self clearing.

RX FIFO. In the 16550A mode, the host can burst read data from the RX FIFO Buffer until the RX FIFO is empty. This can be accomplished under host control or by a supported DMA mode using RXRDY and RXACKP lines.

The RX FIFO contains 16 eleven-bit words. Each eleven bits contain 8 bits of data and a 3-bit error field. The error field is established by the CPU loading PE, FE and BI data into the RX FIFO Status Register (FSR) bits 2 - 4. The next IWRX clock causes the error data stored in the FSR to be inserted together with CPU data into the RX FIFO. The error bits PE, FE, and BI in the FSR can be cleared only by the CPU writing 0s to these bit positions. IWRX also processes or shifts older information contained in the RX FIFO. RXPTR always points to the oldest unread data and error bits contained in the RX FIFO. The RX FIFO error indication (LSR7) is asserted whenever there is an unread error in the RX FIFO error field. Both RXPTR and RUPTR are reset whenever the RX FIFO bit (FCR1) is set to a 1 by the host, or FIER2 is set to a 1 by the MCU, or whenever the host changes the state of FCR0. These reset control bits are self clearing. The RXRDY signal supports a host RX FIFO DMA Real Mode.

RXPTR permits the MCU to burst-fill the error and data fields of the RX FIFO. The error field can be ignored by leaving BI, FE and PE in the FIFO Status Register in their zero state. Two CPU interrupt request conditions with separate enables are provided: RCEMT (RX FIFO empty) and RCHE (RX FIFO half empty) (see Figure 3-23). Bit 0 of the GP status register provides an RX FIFO full indication.

The RXPTR pointer goes from empty to not empty when the CPU writes the first data into an empty RX FIFO. The RX FIFO is considered empty when both RXPTR and RUPTR are at zero. When the CPU writes to an empty RX FIFO, the RUCLK timer is resynchronized. The RUPTR transition from empty to not empty is delayed until the first RUCLK interval following the resynchronization. At this point the RUPTR pointer increments at the RUCLK rate until it reaches the value in RXPTR. If the trigger level has not been reached and RUPTR reaches the level in RXPTR, a receiver time-out event is started. Once started, if neither an IWRX or ERRX occur for three or four RUCLK intervals, an RXTO time out event is used to generate a host character time-out interrupt (IIR3). The host can set the RX FIFO trigger level at 1, 4, 8 or 14 using the FIFO Control Register.

3.13.3 16450 Interface Mode and FIFOs

When the 16450 mode is selected (HCR1 = 1, HCR2 = 1, and FCR0 = 0), the MCU provides a 16450 register compatible interface mode. The 16450 interface uses the hardware signals shown in Figure 3-20 except the A3, TXRDY, RXRDY, HTACKP, or HRACKP signals are not used. The 16450 interface operation is a subset of the 16550A interface operation.

The 16450 interface registers are described in Table 3-26 and Table 3-27. All registers are initialized as shown in Table 3-28 by MCU reset. The CPU has access to both FIFOs while in the 16450 mode.

FIFO UART Timing Simulator. The FIFO UART Timing Simulator (Figure 3-22) provides both an RUCLK and an TUCLK to simulate the 16450 UART word rate. These rates are driven by either TIMB or PTGB through divide by 16 counters.

RUCLK is automatically re-synchronized to one sixteenth of the word rate by either the termination of Freeze (FSR5) or by the first MCU RX FIFO write to occur when both RUPTR and RXPTR are empty. TUCLK will be re-synchronized by the first host write to the TX FIFO to occur when both TUPTR and TXPTR are empty. Note that host can read data from the RX FIFO and write data to the TX FIFO faster than the baud rate by monitoring the 16550 interface interrupt flags rather than waiting for the baud clock to elapse. This timer “override” feature allows much faster data throughput rates than normal baud timing. Note also that when using this technique, the buffers must be completely emptied for the MCU to properly re-synchronize upon receipt of the next byte at the word rate.

TX FIFO. In the 16450 mode, the host is limited to writing a single byte of data with each THRE interrupt. When the host writes to an empty TX FIFO the THRE bit is cleared and a TUCLK is generated; after a short delay THRE is set true. The host responds by sending new data to the TX FIFO Buffer which then clears THRE. TUCLK will cause THRE to be asserted one word time after the first host write to the empty TX FIFO Buffer. In the event that THRE has not been reset prior to the next TUCLK, TEMT is asserted. TUPTR simulates the responses of a 16450 UART while TXPTR provides normal TX FIFO support for the MCU.

RX FIFO. In the 16450 mode, the CPU has full use of the RX FIFO. The RX FIFO trigger level is set at one. Following a CPU write to an empty RX FIFO Buffer, the DR flag is not set until after an RUCLK delay. A host RX FIFO read, ERRX, decrements RUPTR and resets DR. As long as the RX FIFO contains data, the DR flag will continue to be set following the next RUCLK delay.

Bit 0 of the GP status register provides an RX FIFO full indication.

The RXRDY and RXTO (time-out) signals are non-operational in the 16450 mode.

Table 3-26. MCU Host Bus Interface Memory Map - 16550A/16450 Mode

MCU (Internal) Access				Host (External) Access			
Addr.	**HCR1**	**Read**	**Write**	**DLAB**	**Addr.**	**Read**	**Write**
0020	X	TX FIFO Buffer	RX FIFO Buffer	0	0	RX FIFO Buffer	TX FIFO Buffer
0021	1	Line Status Register (LSR)	LSR1 only	X	5	Line Status Register	Line Status Register
0022	1	Modem Status Register (MSR)	Modem Status Register	X	6	Modem Status Register	Modem Status Register
0023	X	Line Control Register (LCR)	Line Control Register	X	3	Line Control Register	Line Control Register
0024	1	Modem Control Register (MCR)	—	X	4	Modem Control Register	Modem Control Register
—	—	—	—	0	1	Interrupt Enable Register (IER)	Interrupt Enable Register
—	—	—	—	X	2	Interrupt Identifier Register (IIR)	—
0025	1	FIFO Control Register (FCR)	—	X	2	—	FIFO Control Register
0026	X	SP RAM 6	SP RAM 6	—	—	—	—
0027	X	SP RAM 7	SP RAM 7	X	7	Scratch Register	Scratch Register
0028	X	Divisor Latch LSB (DLL)	Divisor Latch LSB	1	0	Divisor Latch LSB	Divisor Latch LSB
0029	X	Divisor Latch MSB (DLM)	Divisor Latch MSB	1	1	Divisor Latch MSB	Divisor Latch MSB
002A	X	SP RAM A	SP RAM A	—	—	—	—
002B	X	SP RAM B	SP RAM B	—	—	—	—
002C	X	SP RAM C	SP RAM C	—	—	—	—
002D	X	SP RAM D	SP RAM D	—	—	—	—
002E	X	GP FIFO Status	GPFS3 Only	—	—	—	—
002F	X	Host Handshake Register (HHR)	Host Handshake Register	—	—	—	—
0030	X	FIFO Status Register (FSR)	FIFO Status Register	—	—	—	—
0031	X	FIFO Interrupt Enable (FIER)	FIFO Interrupt Enable	—	—	—	—
0032	X	Host Control Register (HCR)	Host Control Register	—	—	—	—

Table 3-27. Register Bit Assignments: 0020h - 0032h

Addr.	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0020	RX FIFO Buffer (Host Read/MCU Write)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0020	TX FIFO Buffer (Host Write/MCU Read)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0021	Line Status Register (LSR)	RX FIFO Error 1 = Error	XMTR Empty (TEMT) 1 = Empty	XMTR Holding Register Empty (THRE) 1 = Empty	Break Interrupt (BI)	Framing Error (FE) 1 = Error	Parity Error (PE) 1 = Error	Overrun Error (OE) 1 = Error	RX Data Ready (DR) 1 = Error
0022	Modem Status Register (MSR)	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear to Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge of Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear-to Send (DCTS)
0023	Line Control Register (LCR)	DLAB	Set Break	Stick Parity	Even Parity	Parity Enable	Number Stop Bits	Word Length (WLS1)	Word Length (WLS0)
0024	Modem Control Register (MCR)	0	0	0	Loop	Out 2	Out 1	Request to Send (RTS)	Data Terminal Ready (DTR)
0025	FIFO Control Register (FCR)	RCVR Trigger MSB	RCVR Trigger LSB	Reserved	Reserved	DMA Mode Select	TX FIFO Reset	RX FIFO Reset	FIFO Enable
Host Rd/Wt @ x1	Interrupt Enable Register (IER)	0	0	0	0	Modem Status Int. Enable (EDSSI)	RX Line Status Int. Enable (ELSI)	TX Hold Empty Int. Enable (ETBEI)	RX Data Avail Int. Enable (ERBFI)
Host Read @ x2	Interrupt Identifier Register (IIR)	FIFO Enable (FCR0)	FIFO Enable (FCR0)	0	0	Interrupt ID Bit 2	Interrupt ID Bit 1	Interrupt ID Bit 0	0 if Interrupt Pending
0027	Scratch Register (SCR)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0028	Divisor Latch LSB	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0029	Divisor Latch MSB	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
002E	GP FIFO Status (GPFS)	Tx Empty Int. Enable 1 = Enable	Tx Empty Int. Flag 1 = Flag	Tx FIFO Half Empty 1 =	Rx Trig Level Select (RTL1)	Rx Trig Level Select (RTL0) [GP Mode] TuClk Off [16550]	Rx Trig Level Int. Enable 1 = Enable	Rx Trig Level Int. Flag 1 = Flag	RCNE [GP Mode] Rx FIFO Data Avail [16450/ 16550]
002F	Host Handshake Register (HHR)	Not supported							
0030	FIFO Status Register (FSR)	TX FIFO Half Full (TCHF)	TX FIFO Data Avail (TCDA)	RX FIFO Freeze	RX FIFO Break Interrupt (BI)	RX FIFO Framing Error (FE)	RX FIFO Parity Error (PE)	RX FIFO Empty Flag (RCEMT)	RX FIFO Half Empty Flag (RCHE)
0031	FIFO Int. Enable Register (FIER)	TX FIFO Half Full Int. Enable (TCHFE)	TX FIFO Data Avail Int. Enable (TCDAE)	UART Timing Select	RUCLK Off	TX FIFO Reset	RX FIFO Reset	RX FIFO Empty Int. Enable (RCEMTE)	RX FIFO Half Empty Int. Enable (RCHEE)
0032	Host Control Register (HCR)	TX FIFO Interrupt	MCR Write Flag	LCR Write Flag	Divisor Latch Write Flag	RX FIFO Interrupt	Host Mode Select	16450/ 16550 Mode	16450/ 16550 Int. Enable

Table 3-28. 16550 Register Initialization

Register	Bit							
	7	6	5	4	3	2	1	0
Line Status Register (LSR)	0	1	1	0	0	0	0	0
Modem Status Register (MSR)	X	X	X	X	0	0	0	0
Line Control Register (LCR)	X	X	X	X	X	X	X	X
Modem Control Register (MCR)	0	0	0	0	0	0	0	0
Interrupt Enable Register (IER)	0	0	0	0	0	0	0	0
Interrupt Identifier Register (IIR)	0	0	0	0	0	0	0	1
FIFO Control Register (FCR)	0	0	X	X	0	0	0	0
GP FIFO Status (GPFS)	0	1	1	0	0	0	0	0
Host Handshake Register (HHR)	0	0	0	0	0	0	0	0
FIFO Status Register (FSR)	0	0	0	0	0	0	1	1
FIFO Interrupt Enable (FIER)	0	0	0	0	0	0	0	0
Host Control Register (HCR)	0	0	0	0	0	0	0	0

TX FIFO Block Diagram

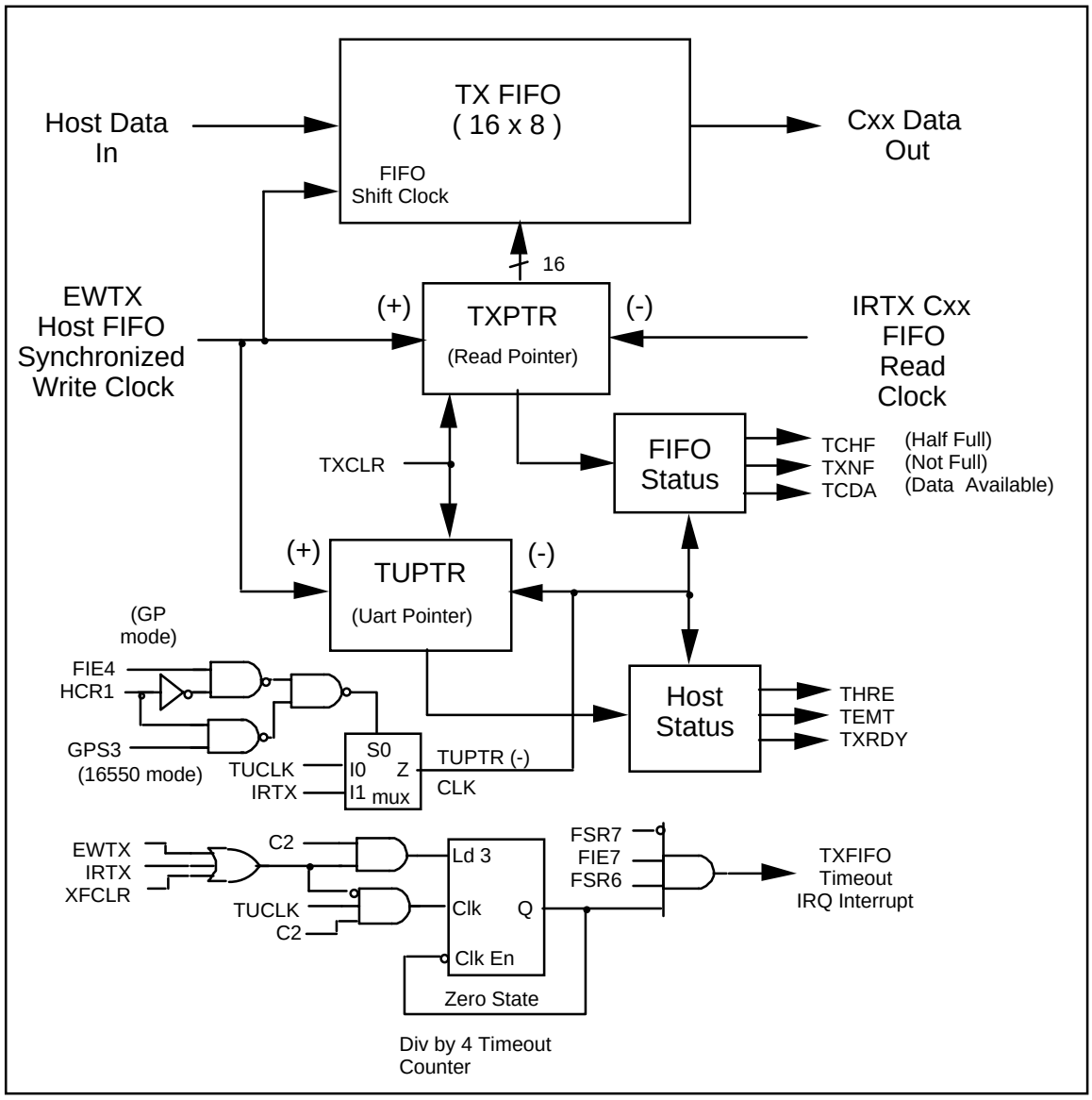


Figure 3-21. TX FIFO Block Diagram

FIFO UART Timing Simulator

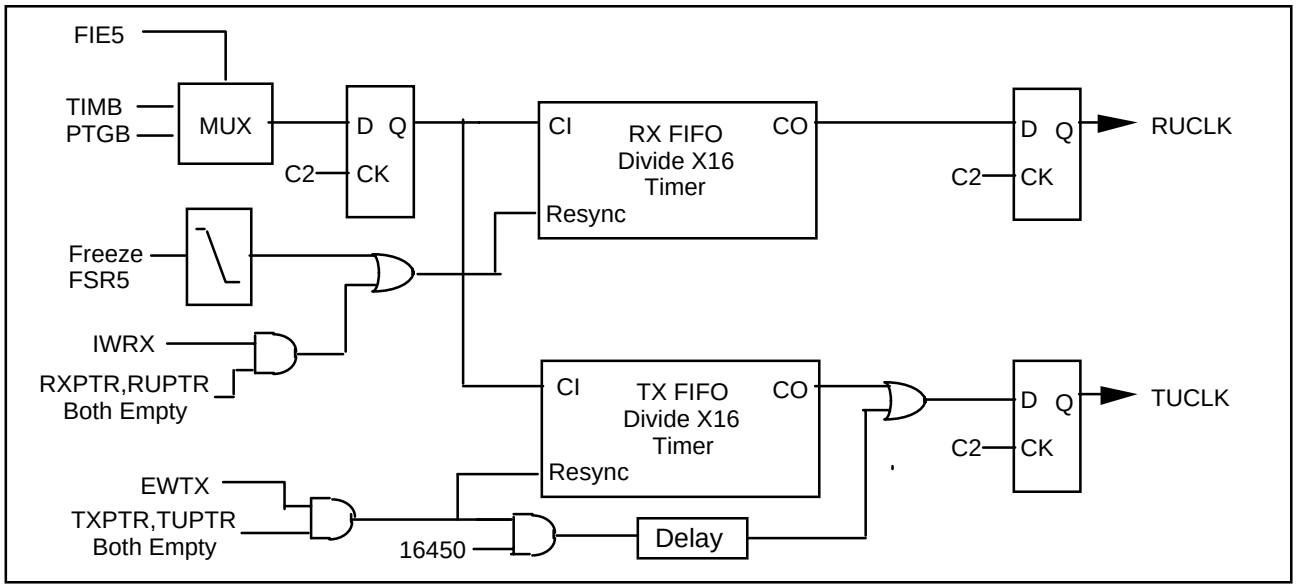


Figure 3-22. FIFO UART Timing Simulator

RX FIFO Block Diagram

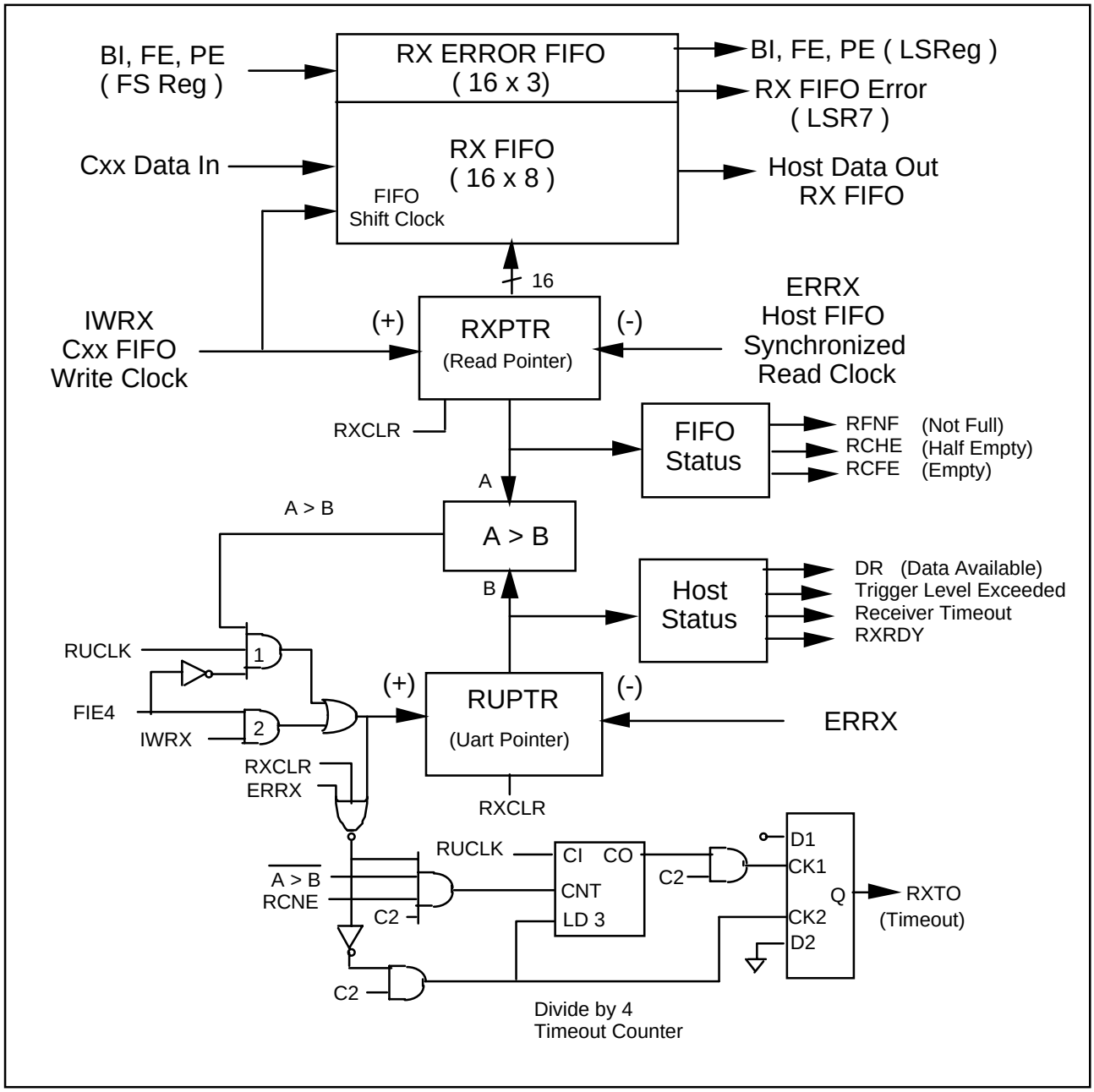


Figure 3-23. RX FIFO Block Diagram

3.13.4 Host Control Register (HCR)

The Host Control Register (HCR) controls and monitors the operation of the 16550A/16450 interface. This register cannot be accessed by the host. All bits can be read by the CPU Bits 0-2 can be written to either state by the CPU. Bits 4-6 can be cleared by the CPU (by writing 0s to the corresponding bit locations) but cannot be set by the CPU. All HCR bits are cleared by MCU reset.

Bit 0: **16550A/16450 Interrupt Enable**. Control bit.

1 = Enables IRQ3 to be asserted if any interrupt flag in HCR4 through HCR6 is set.

0 = Disables IRQ3 assertion due to any interrupt flag in HCR4 through HCR6.

Bit 1: **16550A/16450 DP RAM Mode.** Control bit.

1 = Selects 16550A/16450 dual port (DP) RAM mode if the host mode is enabled (HCR2 = 1).

0 = Selects General Purpose DP RAM mode if the host mode is enabled (HCR2 = 1) (Not valid for the L28).

Bit 2: **Host Mode Enable**. Control bit.

1 = Enables I/O port host mode (PA3-PA6, PB6-PB7, PC0-PC7, and PD0-PD6 operate as dedicated host I/O pins). This bit must be set for DP RAM to operate in GP DP RAM or 16550A/16450 DP RAM mode (see HCR1).

0 = Disables I/O port host mode (PA3-PA6, PB6-PB7, PC0-PC7, and PD0-PD6 operate as general purpose I/O lines).

Bit 3: **RX FIFO Interrupt**. Status bit.

1 = IRQ3 has been asserted due the RX FIFO Half Empty Flag (FSR0) being set or the RX FIFO Empty Flag (FSR1) being set. This bit is set whenever the following logic term is true:

(FSR0)(FIER0) + (FSR1P)(FIER1)

0 = IRQ3 has not been asserted due FSR0 or FSR1 being set.

Bit 4: **Divisor Latch Write Flag**. This bit is set when the host writes to either the Divisor Latch LSB or Divisor Latch MSB register. IRQ3 is asserted if HCR0 is a 1 and this bit is a 1. This bit is cleared by the CPU writing a 0 to this bit position; writing a 1 has no effect.

Bit 5: **Line Control Register Write Flag**. This bit is set when the host writes to the Line Control Register. IRQ3 is asserted if HCR0 is a 1 and this bit is a 1. This bit is cleared by the CPU writing a 0 to this bit position; writing a 1 has no effect.

Bit 6: **Mode Control Register Write Flag**. This bit is set when the host writes to the Mode Control Register. IRQ3 is asserted if HCR0 is a 1 and this bit is a 1. This bit is cleared by the CPU writing a 0 to this bit position; writing a 1 has no effect.

Bit 7: **TX FIFO Interrupt**. Status bit.

1 = IRQ3 has been asserted due the TX FIFO Data Available Flag (FSR6) being set, TX FIFO Half Full Flag (FSR7) being set, or TX FIFO Time-out (FSR7)(FSR6)(TX FIFO Time-out) occurring. A TX time out occurs when neither a EWTX, IRTX or a TXCLR occur for three or four TUCLK intervals. The TX time out is cleared by EWTX, IRTX or a TXCLR. This bit is set whenever the following logic term is true:

(FSR6)(FIER6) + (FSR7)(FIER7) + (FSR7P)(FSR6)(FIER7)(TX Time out).

0 = IRQ3 has not been asserted due FSR6, FSR7, or TX Time out.

3.13.5 Interrupt Enable Register (IER)

The host can read or write bits 0 through 3. The CPU can neither read nor write these bits. These bits are cleared by MCU reset.

The IER enables five types of interrupts that can separately assert the HINT output signal. A selected interrupt can be enabled by setting the corresponding enable bit to a logic 1, or disabled by setting the corresponding enable bit to a logic 0. Disabling all interrupts inhibits the Interrupt Identifier Register (IIR) and inhibits assertion of the HINT output.

The typical use of these bits in a 16550A/16450A interface application is:

Bit 0: **Enable Receiver Buffer Full Interrupt (ERBFI) and Character Time-out in FIFO Mode.** Control bit.

- 1 = Enables assertion of the HINT output when the Data Ready bit in the Line Status Register (LSR0) is set to a logic 1 or character time-out occurs in FIFO mode.
- 0 = Disables assertion of HINT due to LSR0 or character time-out.

Bit 1: **Enable Transmitter Buffer Empty Interrupt (ETBEI).** Control bit.

- 1 = Enables assertion of the HINT output when the Transmitter Empty bit in the Line Status Register (LSR5) is set to a logic 1.
- 0 = Disables assertion of HINT due to LSR5.

Bit 2: **Enable Receiver Line Status Interrupt (ELSI).** Control bit.

- 1 = Enables assertion of the HINT output when bit 1, 2, 3, or 4 in the Line Status Register (LSR) is a logic 1.
- 0 = Disables assertion of HINT due to setting of any of LSR1-LSR4 bits.

Bit 3: **Enable Modem Status Interrupt (EDSSI).** Control bit.

- 1 = Enables assertion of the HINT output when bit 0, 1, 2, or 3 in the Modem Status Register (MSR) is a logic 1.
- 0 = Disables assertion of HINT due to setting of any of the MSR0-MSR3 bits.

Bits 4-7: **Not used.** Always 0.

3.13.6 Interrupt Identifier Register (IIR)

The host can read all bits but cannot write to the IIR. All bits are controlled by MCU hardware. The IIR is not accessible to the CPU.

The IIR identifies the existence and type of five prioritized pending interrupts. Four priority levels are set to assist interrupt processing in the host. When accessed, the IIR freezes the highest priority interrupt pending and acknowledges no other interrupts until the particular interrupt is serviced by the host.

Bits 4 and 5 are not used and always read low. Bits 6 and 7 copy the state of FCR0, the host-controlled FIFO Enable bit. The IIR is initialized to 001b by MCU reset.

Bit 0: **Interrupt Pending.** Status bit. This bit can be used in a hardwired prioritized or polled environment to indicate whether an interrupt is pending. If an interrupt is pending. the IIR contents can be used as a pointer to the appropriate interrupt service routine in the host.

1 = An interrupt is not pending.

0 = An interrupt is pending.

Bits 1-3: **Highest Priority Pending Interrupt.** These three bits identify the highest priority pending interrupt when bit 0 is a logic 0:

Bit 3	Bit 2	Bit 1	Priority Level	Pending Interrupt
0	1	1	1 (highest)	Receiver Line Status
0	1	0	2	Receiver Buffer Full
1	1	0	2	Character Timeout
0	0	1	3	Transmitter Buffer Empty
0	0	0	4	Modem Status

Bits 4-5: **Not used.** Always 0.

Bits 6-7: **FIFO Mode.** These two bits copy FCR0.

3.13.7 Line Control Register (LCR)

All Line Control Register (LCR) bits can be read or written to either by the host or the CPU. The LCR is not initialized by MCU reset.

HCR bit 5 is automatically set to a logic 1 whenever the host writes to the LCR, providing a flag to the CPU that the LCR has been updated. User-provided software must read and interpret the LCR contents.

Bit 7 is duplicated as a separate hardware latch and provides the DLAB address extension bit used for register access. This bit cannot be written to by the CPU.

The typical use of these bits in a 16550A/16450 interface application is:

Bits 0-1: **Word Length Select (WLS0 and WLS1).** These two bits specify the number of bits in each transmitted or received serial character (word):

Bit 1	Bit 0	Word Length	No. of Stop Bits (LCR2 = 1)
0	0	5 Bits	1 1/2
0	1	6 Bits	2
1	0	7 Bits	2
1	1	8 Bits	2

Bit 2: **Number Stop Bits (STB).** This control bit specifies the number of stop bits in each transmitted and received in each serial character. The receiver logic checks the first stop bit only regardless of the number of stop bits selected.

1 = The number of stop bits generated depends on the word length (see bits 0 and 1).

0 = One stop bit is generated.

Bit 3: **Parity Enable (PEN).** Control bit.

1 = Enables parity generation and checking. An even or odd Parity bit is generated in the transmitted data or checked in the received data in accordance with Even Parity Select (LCR4) and Stick Parity (LCR5) bits. The parity bit is located between the last data bit and the first Stop bit.

0 = Disables parity generation and checking.

Bit 4: **Even Parity Select (EPS).** Control bit.

1 = Selects even parity. When parity is enabled (LCR3 = 1) and stick parity is disabled (LCR5 = 0),an even number of total number of ones in the data and parity fields is generated or checked.

0 = Selects odd parity. When parity is enabled (LCR3 = 1) and stick parity is disabled (LCR5 = 0), an odd number of total number of ones in the data and parity fields is generated or checked.

Bit 5: **Stick Parity.** Control bit.

1 = Enables stick parity. When parity is enabled (LCR3 = 1), the parity bit is generated and check in accordance with the inverted state of LCR4 (parity bit = 0 when LCR4 = 1; parity bit = 1 when LCR4 = 0).

0 = Disables stick parity.

Bit 6: **Set Break.** Break Control bit. The Break Control bit acts only on RXD and has no effect on the transmitted data.

1 = Sets break. The received data output is forced to the space (logic 0) state.

0 = Disables break.

Bit 7: **Divisor Latch Access Bit (DLAB).** Control bit.

1 = Enables access to the divisor latches of the baud generator during a read or write operation.

0 = Enables access to the RX FIFO Buffer, TX FIFO Buffer, or Interrupt Enable Register during a read or write operation.

3.13.8 Modem Control Register (MCR)

Bits 0-4 of the Modem Control Register (MCR) can be read or written to by the host. The CPU can only read the MCR. When the host mode is selected, the HINT output driver is placed in a three-state mode when either Out2 is false (MCR3 = 0) or Loop is selected (MCR4 = 1). In addition, when the Loop is selected, the operation of the MSR is modified as described under Modem Status Register. The MCR is cleared by MCU reset.

The typical use of these bits in a 16550A/16450A interface application is:

Bit 0: **Data Terminal Ready (DTR).** This bit controls the Data Terminal Ready (DTR) function.

1 = DTR is on.

0 = DTR is off.

Bit 1: **Request to Send (RTS).** This bit controls the Request to Send (RTS) function.

1 = RTS is on.

0 = RTS is off.

Bit 2: **Output 1.** Control bit. This bit is used in local loopback (see MCR4).

Bit 3: **Output 2.** Control bit.

1 = HINT is enabled.

0 = HINT is in the high impedance state.

Bit 4: **Loop.** Control bit.

1 = Enables loop. The diagnostic mode is selected and the following occurs:

1. Data written to the TX FIFO Buffer is looped back to the RX FIFO Buffer.
2. The contents of MCR bits 0-3 are internally connected to the four MSR bits during a host read of the MSR as follows:

Signal	MCR Bit	MSR Bit
DTR	MCR0	MSR5
RTS	MCR1	MSR4
Out1	MCR2	MSR6
Out2	MCR3	MSR7

0 = Disables loop.

Bit 5-7: Not Used. (Always 0).

3.13.9 Line Status Register (LSR)

The host can read and write all bits in this register, however it is recommended that the host not write to the LSR register. The CPU can read all bits but can only write to LSR1 (Overrun Error).

The typical use of these bits in a 16550A/16450A interface application is:

Bit 0: **Receiver Data Ready (DR).** This bit is cleared by MCU reset. DR is set by hardware whenever the RX UART simulator pointer, RUPTR, goes from empty to not empty. Cleared when the RUPTR pointer indicates empty. When Freeze (FSR5) is set, LSR0 is no longer controlled by the state of the RUPTR. When FSR5 = 1, the DR bit can be cleared by reading the RX FIFO Buffer.

Bit 1: **Overrun Error (OE).** This bit is cleared by RX FIFO reset (FCR1 = 1) or MCU reset. OE is set or reset by CPU write. This bit is reset when the host reads LSR. This bit is cleared by MCU reset.

Bit 2-4: **Receive Line Status**. These bits are cleared by RX FIFO reset (FCR1 = 1) or MCU reset. They are updated automatically from the error field of RX FIFO following the first CPU write or each host read to the RX FIFO. LSR bits 2, 3, 4 and 7 and the RX FIFO output data latch are all updated at this time. LSR error data must be read prior to reading the RX FIFO output or the error data is lost. These bits are cleared to zero when the host reads LSR. These bits are cleared by MCU reset.

Bit 2: **Parity Error (PE).** See bits 2-4.

Bit 3: **Framing Error (FE).** See bits 2-4.

Bit 4: **Break Interrupt (BI)**. See bits 2-4.

Bit 5: **Transmitter Holding Register Empty (THRE)**. This bit is set by MCU reset. THRE is set by hardware each time the transmitter UART pointer, TUPTR, transitions from not empty to empty. It is reset by a host write to the TX FIFO. If a HINT interrupt occurs and IIR identifies THRE as the source, the interrupt is cleared while THRE remains set. Whenever the host writes to the Interrupt Enable Register enabling a THRE interrupt and THRE is set, a THRE interrupt is regenerated.

Bit 6: **Transmitter Empty (TEMT)**. Set by MCU reset or by hardware when the TX FIFO has been empty for at least one TUCLK time. Reset when the host writes to the TX FIFO.

Bit 7: **RX FIFO Error**. Set by MCU hardware if at least one error bit is set in the unread error field of the RX FIFO. Reads zero in the 16450 mode.

3.13.10 Modem Status Register (MSR)

All Modem Status Register (MSR) bits can be read or written to by either the host or the CPU. The CPU can write all bits, however the host can only write to bits 0-3. Bits 0-3 are set to a 1 by the CPU writing a 0 to the particular bit. Bits 0-3 are unaffected when the MCU writes a 1 to a bit position. Bits 0-3 are cleared when the host reads the MSR. It is recommended that the host not write to the MSR. A host MSR read is modified when Loop is selected (MCR4 = 1). Bits 0-3 are cleared by MCU reset.

When Loop is not selected (MCR4 = 0), a host MSR read generates the following data:

(MSR7/MSR6/MSR5/MSR4/MSR3/MSR2/MSR1/MSR0).

When Loop is selected (MCR4 = 1), a host MSR read generates the following data:

(MCR3/MCR2/MCR0/MCR1/MSR3/MSR2/MSR1/MSR0).

The typical use of these bits in a 16550A/16450A interface application is:

The Modem Status Register (MSR) reports current state and change information of the modem. Bits 4-7 supply current state and bits 0-3 supply change information. The change bits are set to a logic 1 whenever a control input from the modem changes state from the last MSR read by the host. Bits 0-3 are reset to logic 0 when the host reads the MSR or upon reset. Whenever bits 0, 1, 2, or 3 are set to a logic 1, a Modem Status Interrupt is generated.

Bit 0: **Delta Clear to Send (DCTS).** This bit is set to a logic 1 when the CTS bit has changed since the MSR was last read by the host.

Bit 1: **Delta Data Set Ready (DDSR).** This bit is set to a logic 1 when the DSR bit has changed since the MSR was last read by the host.

Bit 2: **Trailing Edge of Ring Indicator (TERI).** This bit is set to a logic 1 when the RI bit changes from a 1 to a 0 state since the MSR was last read by the host.

Bit 3: **Delta Data Carrier Detect (DDCD).** This bit is set to a logic 1 when the DCD bit changes state since the MSR was last read by the host.

Bit 4: **Clear to Send (CTS).** This bit indicates the logic state of the CTS output. If Loopback is selected (MCR4 = 1), this bit reflects the state of RTS in the MCR (MCR1).

Bit 5: **Data Set Ready (DSR).** This bit indicates the logic state of the DSR output. If Loopback is selected (MCR4 = 1), this bit reflects the state of DTR in the MCR (MCR0).

Bit 6. **Ring Indicator (RI).** This bit indicates the logic state of the RI output. If Loopback is selected (MCR4 = 1), this bit reflects the state of OUT1 in the MCR (MCR2).

Bit 7. **Data Carrier Detect (DCD).** This bit indicates the logic state of the DCD output. If Loopback is selected (MCR4 = 1), this bit reflects the state of OUT2 in the MCR (MCR3).

3.13.11 Divisor Latch Registers

The Divisor Latch LSB (least significant byte) and Divisor Latch MSB (most significant byte) are two read-write registers. HCR bit 4 is set automatically whenever the host writes to either divisor latch byte, thus providing a flag to the CPU that the Divisor Latch has been updated. User-provided software must read and interpret the Divisor Latch contents.

All bits in both registers can be read or written to by either the host or the CPU. This register is not initialized by MCU reset.

3.13.12 Scratch Register

All Scratch Register bits can be read or written to by either the host or the CPU. There are no flag bits associated with this register. This register is not initialized by MCU reset.

3.13.13 FIFO Control Register (FCR)

The host can write all bits but cannot read this register. The CPU can read all bits but cannot write to this register. Bits 1, 2, 4 and 5 read high to the CPU.

The typical use of these bits in a 16550A/16450A interface application is:

Bit 0: **FIFO Enable.** Control bit.

1 = FIFO mode (16550A mode) is selected and both FIFOs are enabled.

0 = 16450 mode is selected and all bits are cleared in both FIFOs.

Bit 1: **RX FIFO Reset.** Control bit.

1 = All bytes in the RX FIFO are cleared. This bit is cleared automatically by the CPU.

Bit 2: **TX FIFO Reset.** Control bit.

1 = All bytes in the TX FIFO are cleared. This bit is cleared automatically by the CPU.

Bit 3: **DMA Mode Select.** Control bit. Selects or deselects the DMA mode when FIFO mode is selected (FCR0 = 1).

1 = Selects the DMA operation when FIFO mode is selected (FCR0 = 1).

0 = Selects non-DMA operation when FIFO mode is selected (FCR0 = 1). Non-DMA operation is also selected when 16450 mode is selected (FCR0 = 0).

DMA operation in FIFO mode. RXRDY is asserted when the number of characters in the RX FIFO exceeds the value in the RX FIFO Trigger Level (FCR6-7) or the RXTO time-out has occurred. RXRDY will go inactive when there are no more characters in the RX FIFO.

TXRDY is asserted when there one or more unfilled locations in the TX FIFO. TXRDY goes inactive when the TX FIFO is completely full.

Non-DMA operation in FIFO mode. RXRDY is asserted when there are one or more characters in the RX FIFO. RXRDY goes inactive when there are no more characters in the RX FIFO.

TXRDY is asserted when there are no characters in the TX FIFO. TXRDY goes inactive when the first character is loaded into the TX FIFO Buffer.

Bits 4-5: **Not used.**

Bits 6-7: **Receiver FIFO Trigger Level.** FCR7 and FCR6 set the trigger level for the RX FIFO interrupt.

FCR7	FCR6	RX FIFO Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14

3.13.14 FIFO Interrupt Enable Register (FIER)

The CPU can read and write all bits. The host cannot access this register. The FIER is cleared by MCU reset.

Bit 0: **RCHE Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER.

Bit 1: **RCEMT Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER.

Bit 2: **RX FIFO Reset.** This bit is set and cleared by a CPU write to the FIER. When set, FIER2 clears the RXPTR pointer, the RUPTR pointer and the RX time-out counter. FIER2 automatically resets itself to zero.

Bit 3: **TX FIFO Reset.** This bit is set and cleared by a CPU write to the FIER. When set, FIER3 clears the TXPTR pointer, the TUPTR pointer and the TX time-out counter. FIER3 automatically resets itself to zero.

Bit 4: **RUCLK Off**. When FIER4 is set, IWRX rather than RUCLK is used to increment RUPTR (Figure 3-23). As a result, when set, the RXPTR and RUPTR become locked together. In the GP Mode only, setting FIER4 causes both the TXPTR and TUPTR to be decremented by a CPU FIFO read. As a result, the TXPTR and TUPTR are also locked together. This bit is set and cleared by a CPU write to the FIER.

Bit 5: **UART Timing Select**. When FIER5 is set, RUCLK and TUCLK timing is derived from PTGB. When XFS5 is reset RUCLK and TUCLK timing is derived from TIMA (Figure 3-23) This bit is set and cleared by a CPU write to the FIER.

Bit 6: **TCDA Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER. When this bit is a 1 and TCDA (FSR6) is also a 1, IRQ3 is asserted.

Bit 7: **TCHF Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER. When this bit is a 1 and TCHF (FSR7) is a 1, IRQ3 is asserted.

3.13.15 FIFO Status Register (FSR)

The CPU can read all bits but can only write to bits 2 through 5. The host cannot access this register. Except for bit 0 which initializes high, all bits are initialized to zero by MCU reset. Bits 2 - 4 are inputs to the error field of the RX FIFO.

Bit 0: **RX FIFO Half Empty Flag (RCHE)**. This bit is set and cleared by hardware. It provides RX FIFO status to the CPU. RCHE is set whenever the RX FIFO contains between 0 and 7 unread bytes of data. If FIER0 and FSR0 are both set, IRQ3 is asserted.

Bit 1: **RX FIFO Empty Flag (RCEMT)**. This bit is set and cleared by hardware. It provides RX FIFO status to the MCU CPU. RCEMT is set whenever the RX FIFO is empty. If FIER1 and FSR1 are both set, IRQ3 is asserted.

Bit 2: **Receiver Parity Error (PE)**. This bit is set and cleared by a CPU write to the FSR.

Bit 3: **Receiver Framing Error (FE)**. This bit is set and cleared by a CPU write to the FSR.

Bit 4: **Receiver Break Interrupt (BI)**. This bit is set and cleared by a CPU write to the FSR.

Bit 5: **DR Freeze (DRF)**. Freeze halts the assertion of DR by RUPTR logic. Freeze is only used in the 16450 mode. If DR is asserted while Freeze is true, DR is cleared by a host read of the RX FIFO Buffer. This bit is set and cleared by a CPU write to the FSR.

Bit 6: **TX FIFO Data Available Flag (TCDA)**. This bit is set and cleared by hardware. It provides TX FIFO status to the CPU. It is true whenever the TX FIFO contains between 1 and 16 unread bytes of data. If FIER6 and FSR6 are both set, IRQ3 is asserted.

Bit 7: **TX FIFO Half Full Flag (TCHF)**. This bit is set and cleared by hardware. It provides TX FIFO status to the CPU. It is true whenever the TX FIFO contains between 8 and 16 unread bytes of data. If FIER7 and FSR7 are both set, IRQ3 is asserted.

3.13.16 GP FIFO Status Register (GPFS)

All bits can be read by either the host or the CPU (see Table 3-21). The host cannot access this register in either the 16450 or 16550A mode. In the 16550 mode only, the MCU can write to bit 4. The host can write to bits 2, 3, 4 and 7 only. MCU reset initializes bits 0-4 and bit 7 to 0 and initializes bits 5 and 6 to 1.

Bit 0: **RX FIFO Data Available.** When operating in the GP FIFO mode, GPFS0 = 1 indicates the not empty status of RUPTR. When operating in the 16450 or 16550 mode, GPFS0 = 1 indicates that the RX FIFO is full.

Bit 1: **RX FIFO Trigger Level Flag.** GPFS1 = 1 indicates that the RUPTR has reached or exceeds the trigger level specified by RTL0 and RTL1.

Bit 2: **RX FIFO Trigger Level Interrupt Enable.** GPFS2 = 1 enables an HINT interrupt to be automatically generated whenever GPFS1 is set.

Bit 3 - 4: **RUPTR Trigger Level (RTL0 and RTL1).** GPFS3 and GPFS4 set the trigger level for the RX FIFO interrupt.

GPFS4	GPFS3	RUPTR Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14

Bit 3: **16550 Mode TUCLK OFF.** When bit 4 is set in the 16550 mode, the TUPTR is decremented by IRTX rather then TUCLK. As a result the TUPTR becomes locked to the TXPTR value.

Bit 5: **Tx FIFO Half Empty.** GPFS5 is a 1 when there are 0 to 7 data bytes in the TX FIFO.

Bit 6: **Tx FIFO Empty Flag.** GPFS6 is a 1 when there are no data bytes in the TX FIFO.

Bit 7: **Tx FIFO Empty Interrupt Enable.** GPFS7 = 1 enables an HINT interrupt to be automatically generated whenever GPRS6 is a 1.

3.13.17 16550 DMA Operation

Transferring Received Data from the RX FIFO to the Host Bus Memory

The Host bus operation to transfer a received data byte from the MCU I/O (RX FIFO) to Host bus memory is as follows:

1. Select 16550 host interface mode and ensure that the USART serial mode is not selected (SM7 = 0). Note that the RXRDY is available as an output signal only in the 16550 host interface mode when USART serial mode is not selected.
2. The MCU sets RXRDY high when the RX FIFO conditions meet the FIFO Control Register specified limit. This requests a DMA transfer on the channel to which the RXRDY line is connected.
3. The Host bus responds by setting DACK (MCU HRACKP signal) low to indicate that the DMA controller now is controlling the Host data bus.
4. The DMA controller places the memory address location on the address bus to which the data will be transferred and sets the I/O READ and memory WRITE signals low. The MCU places the receive FIFO data byte to be transferred to memory on the data bus. In this mode the MCU ignores the host address bus (assumes it is $00) and host CHIP ENABLE (assumes it is enabled).
5. The DMA controller sets memory WRITE and I/O READ lines high transferring the byte from the MCU I/O (RX FIFO) to Host bus memory in one cycle.
6. The DMA controller sets DACK high and relinquishes the Host bus to the host controller. Dependent upon the FIFO conditions and the FIFO Control Register specified limit, the MCU either initiates another byte transfer immediately by continuing to hold RXRDY high, or sets RXRDY low and waits until the RX FIFO meets the limit to initiate a transfer.

Transferring Transmit Data from Host bus Memory to the TX FIFO

The Host bus (Host bus) operation to transfer a transmit data byte from Host bus memory to the MCU I/O (TX FIFO) is as follows:

1. Select 16550 host interface mode and ensure that the USART serial mode is not selected (SM7 = 0). Note that the TXRDY is available as an output signal only in the 16550 host interface mode when USART serial mode is not selected.
2. When the TX FIFO is empty or has one empty location, the MCU sets TXRDY high. This requests a DMA transfer on the channel to which the TXRDY line is connected.
3. The Host bus responds by setting DACK (MCU HTACKP signal) low to indicate that the DMA controller is now controlling the Host bus.
4. The memory address location of the data byte to be transmitted is placed on the address bus by the DMA controller. The DMA controller sets the I/0 WRITE and the memory READ lines low. In this mode, the MCU ignores the host address bus (assumes it is $00) and host CHIP ENABLE (assumes it is enabled). The memory device places the data byte to be transmitted on the Host data bus.
5. The DMA controller sets memory READ and the I/O WRITE lines high. The MCU inputs the data from the data bus so the byte is transferred from the Host bus memory to the MCU I/O (TX FIFO) in one cycle.
6. The DMA controller sets DACK high and relinquishes the Host bus to the host controller. If the TX FIFO is has at least one empty location, dependent on the FIFO Control Register setup, the MCU continues to hold the TXRDY line high in order to initiate another DMA transfer, or if not sets the TXRDY line low.

3.14 INTERNAL DMA OPERATION

The DMA provides a means of off-loading control of the buffer memory required between the CPU and the RS232/16550 DTE interface. A diagram of the DMA interface is shown in Figure 3-24. When the DMA is disabled (DFR3 = 0 and DFR4 = 0), the RS232/16550 interface to the CPU behaves as before. When DMA is selected (DFR3 = 1 and DFR4 = 1), the user must set the RS232/16550 modes with interrupts disabled, and initialize DMA pointers, registers and control registers.

Separate DMA channels are provided for writing to and for reading from the DTE RS232/16550 interface. The user must select either RS232 or 16550 in DER1. A DMA memory map for the two channels is shown in Figure 3-25. The two channels are basically identical. Each channel has base and top 16-bit address registers, and two 16-bit address pointers for next available data IN and oldest data OUT. The IN and OUT pointers are incremented following their RAM write and read operations. The pointers are automatically wrapped to the base address when they increment past the top address. When the incremented IN pointer equals the OUT pointer, the buffer **full** status is set. When a channel buffer is **full,** that buffer will not accept additional data. Similarly, when the incremented OUT pointer equals the IN pointer, the buffer **empty** status is set. An **empty** channel buffer will not output data**.** DRM and DXM provide a count of the unread locations remaining in RAM. DRM is used to compare with ROFF to set the state of DSR5, the ROFF status bit. DXM is used to compare with the Hi WM and the Lo WM to control STAT(7), the Hi WM status bit.

Once initialized, the firmware no longer directly supports the RS232/16550 interface. The DTE in-bound receive channel is supported by writing data to the DMA "PUT" register. The DMA moves the data from PUT to the Rx RAM Buffer and then from the Rx RAM Buffer to the DTE RS232/16550 interface as data is required. The DTE out-bound transmit channel moves data from the DTE RS232/16550 interface as it becomes available and places it in the Tx RAM Buffer and in a TXBI register internal to the DMA. The oldest Tx RAM Buffer data is used to keep the DMA "GET" register full. The CPU will read GET as new transmit data is required. Flag bits Rx_BE (DSR1), and Tx_BF (DSR2), supply status on PUT and GET, respectively.

The DMA moves data between the DTE RS232/16550 interface and internal/external RAM by executing a cycle steal operation. Once the DMA is ready to move data to or from RAM, it waits until the CPU performs a next instruction fetch (SYNC) operation. During cycle steal, the CPU is placed in an idle, or do-nothing state, requiring one extra clock cycle to complete its current instruction. The DMA gains control of the internal/external address and data busses during the steal operation. The cycle steal clock cycle can be a slow memory cycle under firmware control (see Banking RAM Dual Port RAM slow memory cycle description in Section 3.8).

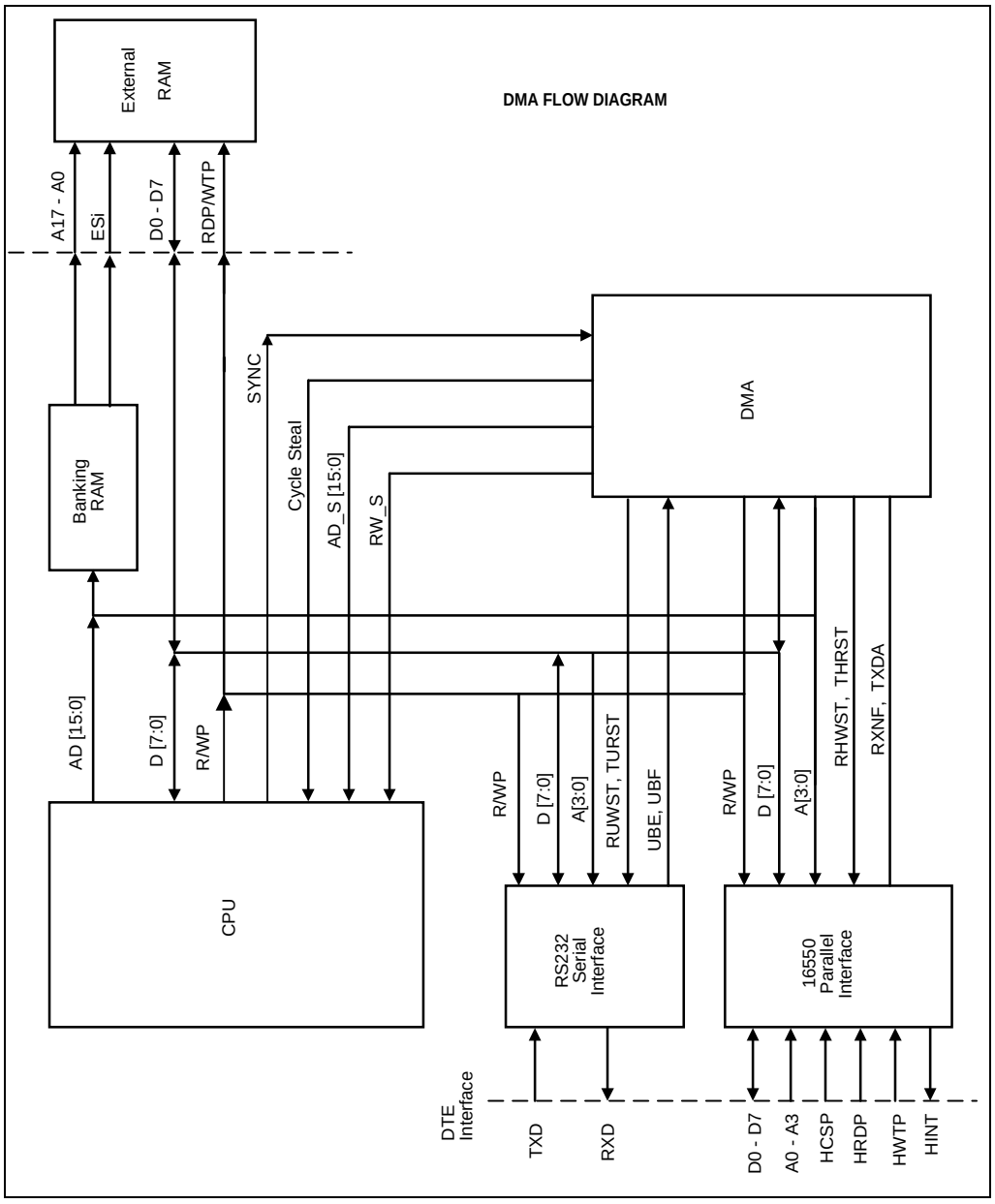


Figure 3-24. Internal DMA Operation

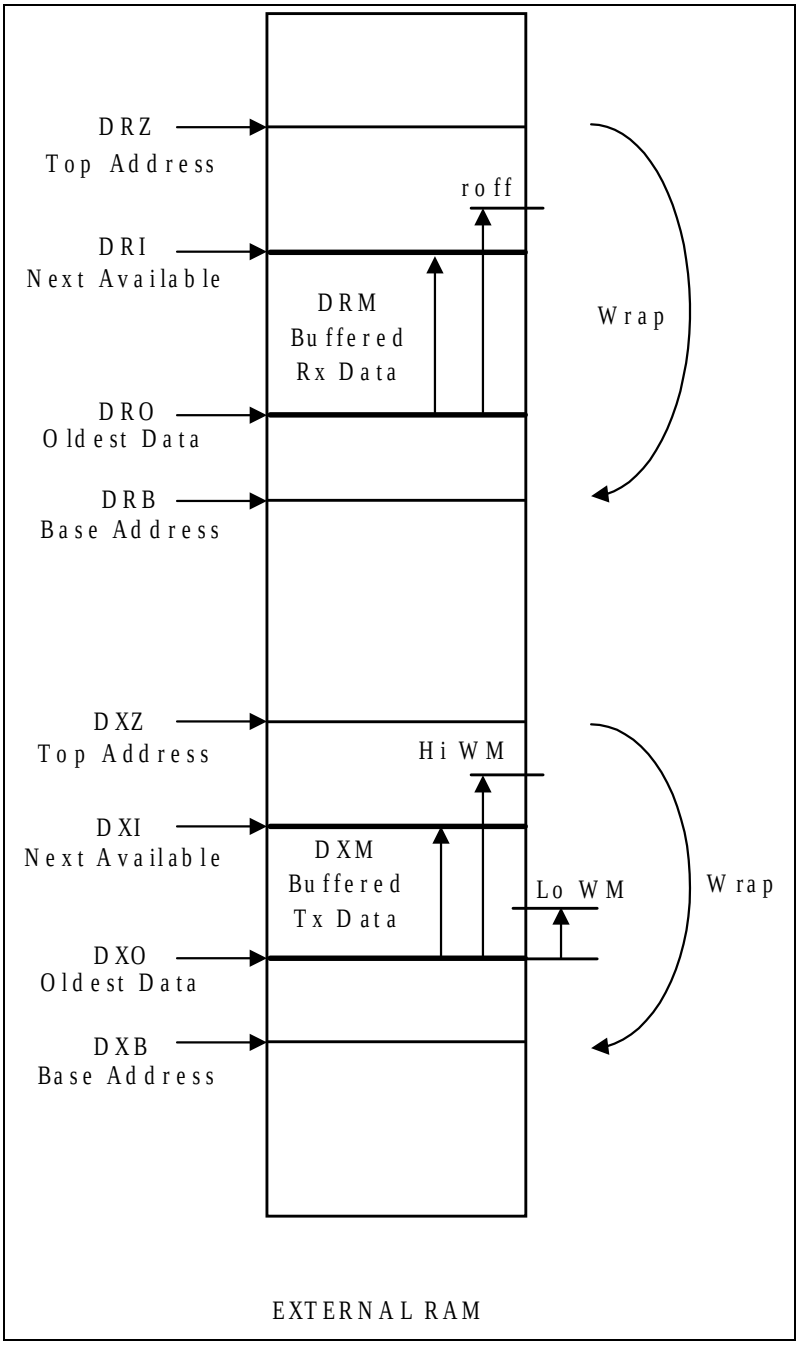


Figure 3-25. DMA Operation - External RAM

3.14.1 DMA Control/Special Features

The three DMA registers shown in Table 3-29 control the various modes of the DMA.

Table 3-29. Register Bit Assignments: DMA 05D0, 05F5-05F6

Addr.	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
05D0 (DS0 =1)	DMA Enable Register (DER)	Tx CHAR3 Int. Enable 1 = Enable	Tx CHAR3 DFR (0) Enable 1 = Enable	Tx Hi-Lo Watermark Int. Enable 1 = Enable	Tx Hi-Lo Watermark Auto Insert	Tx CHAR1/2 Int. Enable 1 = Enable	Tx CHAR1/2 Save to Buffer	Mode 1 = RS-232 0 = 16550	Not used
05F5	DMA Flag Register (DFR)	Tx CHAR3 Int. Flag 1 = Flag	Tx Hi-Lo Watermark Int. Flag 1 = Flag	Tx CHAR1/2 Int. Flag 1 = Flag	DTE - Tx Channel Enable 1 = Enable	Rx - DTE Channel Enable 1 = Enable	CHAR3 Seq. Flag 3rd Detect 1 = Detect	CHAR3 Seq. Flag 2nd Detect 1 = Detect	CHAR3 Seq. Flag 1st Detect 1 = Detect
05F6	DMA Status Register (DSR)	Tx RAM Buffer Hi Watermark Status	Tx RAM Buffer Full 1 = Full	Rx RAM Buffer Exceed ROFF	Rx RAM Buffer Empty 1 = Empty	CHAR1(0) CHAR2(1) Detected	Tx_BF "GET" Status	Rx_BE "PUT" Status	RAM(0) DMA(1) Select

All registers are readable, however only the following bits can be written to:

Register	Write bits	Reset Value
DMA Enable Register (DER)	1 to 7	$00
DMA Flag Register (DFR)	0 to 4	$00
DMA Status Register (DSR)	0 only	$12

A level 4 IRQ interrupt is generated as follows:

IRQ4 = (DER3 and DFR5) or (DER5 and DFR6) or (DER7 and DFR7)

Reading the DMA Flag Register will automatically clear the three flag bits, DFR5, DFR6 and DFR7, terminating the cause of the interrupt.

Note: The DMA Enable Register can be read or written to only when DMA Mode is (DSR0 =1).

3.14.2 DMA Enable Register (DER)

Bit 0: **Not used.**

Bit 1: **DTE Interface Mode.** Control bit.

1 = RS232 mode

0 = 16550 mode.

Bit 2: **Tx CHAR1/2 Save to Buffer.** Control bit.

1 = Enables a detected CHAR1 or CHAR2 character to be saved in the Tx RAM buffer.

0 = Causes all CHAR1 and CHAR2 characters to be discarded.

Bit 3: **Tx CHAR1/2 Interrupt Enable.** Control bit.

1 = Enables assertion of an IRQ4 interrupt whenever the Tx CHAR1/2 Flag (DFR5) is set.

0 = Disables assertion of an IRQ4 interrupt whenever the Tx CHAR1/2 Flag (DFR5) is set.

Bit 4: **Tx Hi-Lo Watermark Auto Insert.** Control bit.

1 = Enables the automatic insertion of CHAR1 or CHAR2 flow control characters in the Rx data stream.

0 = Disables the automatic insertion of CHAR1 or CHAR2 flow control characters in the Rx data stream.

Bit 5: **Tx Hi-Lo Watermark Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ4 interrupt when the Hi-Lo WM Flag (DFR6) is set.

0 = Disables assertion of IRQ4 interrupt when the Hi-Lo WM Flag (DFR6) is set.

Bit 6: **Tx CHAR3 DFR0 Enable.** Control bit.

1 = Enables the 1st Detect Flag (DFR0) to be set when CHAR3 is detected in the Tx data stream.

0 = Disables the 1st Detect Flag (DFR0) from being set when CHAR3 is detected in the Tx data stream.

Bit 7: **Tx CHAR3 Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ4 interrupt when the Tx CHAR3 Flag (DFR7) is set.

0 = Disables assertion of IRQ4 interrupt when the Tx CHAR3 Flag (DFR7) is set.

3.14.3 DMA Flag Register (DFR)

Bit 0: **CHAR3 Sequential Flag - 1st Detect.** Status bit.

1 = See 3.14.8.

0 = See 3.14.8.

Bit 1: **CHAR3 Sequential Flag - 2nd Detect.** Status bit.

1 = See 3.14.8.

0 = See 3.14.8.

Bit 2: **CHAR3 Sequential Flag - 3rd Detect.** Status bit.

1 = See 3.14.8.

0 = See 3.14.8.

Bit 3: **Rx DMA-DTE Channel Enable.** Control bit.

1 = Enables the Rx DMA interface to the DTE RS232/16550.

0 = Disables the Rx DMA interface to the DTE RS232/16550.

Bit 4: **DTE-Tx DMA Channel Enable.** Control bit.

1 = Enables the DTE RS232/16550 interface to Tx DMA.

0 = Disables the DTE RS232/16550 interface to Tx DMA.

Bit 5: **Tx CHAR1/2 Flag.** Status bit.

1 = Set whenever a CHAR1 or CHAR2 character is detected in the Tx channel data stream.

0 = DFR5 is cleared by reading the DFR.

Bit 6: **Hi-Lo Watermark (WM) Flag.** Status bit.

1 = DFR6 is set when either DSR7 = 0 and DXM > Hi WM, (Insert CHAR2), or DSR7 = 1 and DXM < or = Lo WM, (Insert CHAR1).

0 = DFR6 is cleared by reading the DFR.

Bit 7: **Tx CHAR3 Flag.** Status bit.

1 = See Section 3.14.8.

0 = DFR7 is cleared by reading the DFR.

3.14.4 DMA Status Register (DSR)

Bit 0: **DMA Addresses Select.** Control bit.

1 = Assigns addresses 05D0 - 05EF (32 bytes) to DMA registers.

0 = Assigns addresses 05D0 - 05EF (32 bytes) to internal RAM.

Bit 1: **Rx_BE PUT Status**. Status bit.

1 = PUT is empty.

0 = PUT is full.

Bit 2: **Tx_BF GET Status**. Status bit.

1 = GET is full.

0 = GET is empty.

Bit 3: **Tx CHAR1/2 Detected.** Status bit.

1 = The last detection was a CHAR2.

0 = The last detection was a CHAR1.

Bit 4: **Rx RAM Buffer Empty.** Status bit.

1 = The Rx RAM Buffer is empty.

0 = The Rx RAM Buffer is not empty.

Bit 5: **Rx RAM Buffer Exceed ROFF.** Status bit.

1 = DRM > ROFF.

0 = DRM < or = ROFF.

Bit 6: **Tx RAM Buffer Full.** Status bit.

1 = The Tx RAM Buffer is full.

0 = The Tx RAM Buffer is not full.

Bit 7: **Tx RAM Buffer Hi Watermark Status.** Status bit.

1 = DSR7 = 0 and DXM > Hi WM.

0 = DSR7 = 1 and DXM < or = Lo WM.

3.14.5 DMA Control

The bits in bold provide basic DMA control:

Addr.	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
05D0 (DS0=1)	DMA Enable Register (DER)	Tx CHAR3 Int. Enable 1 = Enable	Tx CHAR3 DFR0 Enable 1 = Enable	Tx Hi-Lo Watermark Int. Enable 1 = Enable	Tx Hi-Lo Watermark Auto Insert 1 = Insert	Tx CHAR1/2 Int. Enable 1 = Enable	Tx CHAR1/2 Save to Buffer 1= Save	**DTE Mode 1 = RS232 0 = 16550**	Not used
05F5	DMA Flag Register (DFR)	Tx CHAR3 Int. Flag 1 = Flag	Tx Hi-Lo Watermark Int. Flag 1 = Flag	Tx CHAR1/2 Int. Flag 1 = Flag	**DTE-Tx DMA Ch Enable 1 = Enable**	**Rx DMA DTE Ch Enable 1 = Enable**	CHAR3 Seq. Flag 3rd Detect 1 = Detect	CHAR3 Seq. Flag 2nd Detect 1 = Detect	CHAR3 Seq. Flag 1st Detect 1 = Detect
05F6	DMA Status Register (DSR)	Tx RAM Buffer Hi Watermark Status	**Tx RAM Buffer Full 1 = Full**	**Rx RAM Buffer Exceed ROFF 1 = Exceed**	**Rx RAM Buffer Empty 1 = Empty**	CHAR1/2 Detected 1 = CHAR2 0 = CHAR1	**Tx_BF "GET" Status 1 = Full**	**Rx_BE "PUT" Status 1 = Empty**	**DMA Addr. Select 1 = DMA**

DER1 Selects either the RS232 or 16550 DTE interface.

DFR3 A '1' enables the Rx DMA interface to the DTE RS232/16550.
A '0' disables the Rx DMA interface to the DTE RS232/16550

DFR4 A '1' enables the DTE RS232/16550 interface to Tx DMA.
A '0' disables the DTE RS232/16550 interface to Tx DMA

DSR0 A '1' assigns addresses 05D0 - 05EF to DMA.
A '0' assigns addresses 05D0 - 05EF to internal RAM.

DSR1 Rx_BE = 1 when "PUT" is empty.
Rx_BE = 0 when "PUT" is full.

DSR2 Tx_BF = 1 when "GET" is full.
Tx_BF = 0 when "GET" is empty.

DSR4 Set to a '1' whenever the Rx RAM Buffer is empty.

DSR5 Set to a '1' whenever DRM > ROFF.

DSR6 Set to a '1' whenever the Tx RAM Buffer is full.

3.14.6 DTE Generated Flow Control (DTE In-bound)

The DTE initiates Rx channel flow control by inserting either a CHAR1 or CHAR2 character into the Tx channel. Hardware in the Tx DMA channel compares all Tx data from the RS232/16550 interface to CHAR1 and CHAR2. When a comparison is found, the CHAR1/2 Flag (DFR5) is set, and DSR3 is cleared or set depending on whether the comparison was with CHAR1 or with CHAR2. An IRQ4 interrupt is generated each time DFR5 is set if its enable bit, DER3, is also set. The DMA hardware does not enable or disable the Rx channel as a result of CHAR1/2 comparisons. It is the responsibility of firmware to execute flow control by controlling DFR3. The bits associated with DTE generated flow control are in bold below.

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
05D0 (DS0=1)	DMA Enable Register (DER)	Tx CHAR3 Int. Enable 1 = Enable	Tx CHAR3 DFR0 Enable 1 = Enable	Tx Hi-Lo Watermark Int. Enable 1 = Enable	Tx Hi-Lo Watermark Auto Insert 1 = Insert	**Tx CHAR1/2 Int. Enable 1 = Enable**	**Tx CHAR1/2 Save to Buffer 1= Save**	DTE Mode 1 = RS232 0 = 16550	Not used
05F5	DMA Flag Register (DFR)	Tx CHAR3 Int. Flag 1 = Flag	Tx Hi-Lo Watermark Int. Flag 1 = Flag	**Tx CHAR1/2 Int. Flag 1 = Flag**	DTE-Tx DMA Ch Enable 1 = Enable	**Rx DMA DTE Ch Enable 1 = Enable**	CHAR3 Seq. Flag 3rd Detect 1 = Detect	CHAR3 Seq. Flag 2nd Detect 1 = Detect	CHAR3 Seq. Flag 1st Detect 1 = Detect
05F6	DMA Status Register (DSR)	Tx RAM Buffer Hi Watermark Status	Tx RAM Buffer Full 1 = Full	Rx RAM Buffer Exceed ROFF 1 = Exceed	Rx RAM Buffer Empty 1 = Empty	**CHAR1/2 Detected 1 = CHAR2 0 = CHAR1**	Tx_BF "GET" Status 1 = Full	Rx_BE "PUT" Status 1 = Empty	DMA Addr. Select 1 = DMA

DER2 A '1' enables a detected CHAR1 or CHAR2 character to be saved in the Tx RAM buffer.
A '0' discards all CHAR1/2 characters.

DER3 A '1' enables an IRQ4 interrupt whenever the Tx CHAR1/2 Flag, DFR5 is set.

DFR3 A '1' enables the Rx DMA interface to the DTE RS232/16550.
A '0' disables the Rx DMA interface to the DTE RS232/16550.
Required for firmware Rx channel flow control.

DFR5 The Tx CHAR1/2 Flag is set whenever a CHAR1 or CHAR2 character is detected in the Tx channel data stream.
DFR5 is cleared by reading DFR.

DSR3 Set to a '1' when the last detection was a CHAR2.
Cleared to a '0' when the last detection was a CHAR1.

3.14.7 Modem Generated Flow Control (DTE Out-bound)

The DMA initiates Tx channel flow control by inserting CHAR1 and CHAR2 characters into the Rx channel RS232/16550 data stream. When the DTE senses a CHAR2 character in the Rx data stream, it stops sending Tx channel data. Tx data resumes as soon as the DTE detects a CHAR1 character in the Rx data stream. The Tx channel flow is halted when the Tx RAM Hi WM status bit (DSR7) changes to a 1. DSR7 is forced to a high state when a DXM > Hi WM condition is detected. This condition automatically sets DFR6 and inserts CHAR2 into the Rx channel data path if DER4 is set. DSR7 remains high until a DXM < or = Lo WM condition occurs, indicating that the CPU has emptied the Tx RAM to the Lo WM state. This event sets DFR6 and automatically inserts a CHAR1 into the Rx channel data path if DER4 is set. Whenever the Hi-Lo WM Flag (DFR6), is set, an IRQ4 interrupt is generated if DER5 = 1. The control/status bits for modem generated flow control are in bold below.

Addr.	Function	Bit							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
05D0 (DS0=1)	DMA Enable Register (DER)	Tx CHAR3 Int. Enable 1 = Enable	Tx CHAR3 DFR0 Enable 1 = Enable	**Tx Hi-Lo Watermark Int. Enable 1 = Enable**	**Tx Hi-Lo Watermark Auto Insert 1 = Insert**	Tx CHAR1/2 Int. Enable 1 = Enable	Tx CHAR1/2 Save to Buffer 1= Save	DTE Mode 1 = RS232 0 = 16550	Not used
05F5	DMA Flag Register (DFR)	Tx CHAR3 Int. Flag 1 = Flag	**Tx Hi-Lo Watermark Int. Flag 1 = Flag**	Tx CHAR1/2 Int. Flag 1 = Flag	DTE-Tx DMA Ch Enable 1 = Enable	Rx DMA DTE Ch Enable 1 = Enable	CHAR3 Seq. Flag 3rd Detect 1 = Detect	CHAR3 Seq. Flag 2nd Detect 1 = Detect	CHAR3 Seq. Flag 1st Detect 1 = Detect
05F6	DMA Status Register (DSR)	**Tx RAM Buffer Hi Watermark Status**	Tx RAM Buffer Full 1 = Full	Rx RAM Buffer Exceed ROFF 1 = Exceed	Rx RAM Buffe Empty 1 = Empty	CHAR1/2 Detected 1 = CHAR2 0 = CHAR1	Tx_BF "GET" Status 1 = Full	Rx_BE "PUT" Status 1 = Empty	DMA Addr. Select 1 = DMA

DER4 A '1' enables the automatic insertion of CHAR1 or CHAR2 flow control characters in the Rx data stream. A '0' disables the auto insertion of CHAR1/2 characters in the Rx data stream.

DER5 A '1' enables IRQ4 interrupts when the Hi-Lo WM Flag (DFR6) is set.

DFR6 DFR6 is set when either:
DSR7 = 0 and DXM > Hi WM, (Insert CHAR2), or DSR7 = 1 and DXM < or = Lo WM, (Insert CHAR1).
DFR6 is cleared by reading DFR.

DSR7 DSR7 is set when DSR7 = 0 and DXM > Hi WM.
DSR7 is reset when DSR7 = 1 and DXM < or = Lo WM.

3.14.8 Tx CHAR3 Escape Sequence

An 'escape' detection algorithm is supported in hardware. CHAR3 is compared to TXBI, the Tx channel data received from the DTE RS232/16550 interface. Operation of the DFR flag bits, 0-2 and 7, is described below.

IF TXBI	**= CHAR3**	**not = CHAR3**
THEN	DFR0 = DER6	= 0
	DFR1 = DFR0	= 0
	DFR2 = DFR1	= 0
	DFR7 = 1	= 1 If DFR0 was set.

An IRQ4 interrupt is generated if the TX CHAR3 Interrupt Enable bit (DER7) is set and the Tx CHAR3 Interrupt Flag (DFR7) is also set.

		Bit							
Addr.	**Function**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
05D0 (DS0=1)	DMA Enable Register (DER)	**Tx CHAR3 Int. Enable 1 = Enable**	**Tx CHAR3 DFR0 Enable 1 = Enable**	Tx Hi-Lo Watermark Int. Enable 1 = Enable	Tx Hi-Lo Watermark Auto Insert 1 = Insert	Tx CHAR1/2 Int. Enable 1 = Enable	Tx CHAR1/2 Save to Buffer 1= Save	DTE Mode 1 = RS232 0 = 16550	Not used
05F5	DMA Flag Register (DFR)	**Tx CHAR3 Int. Flag 1 = Flag**	Tx Hi-Lo Watermark Int. Flag 1 = Flag	Tx CHAR1/2 Int. Flag 1 = Flag	DTE-Tx DMA Ch Enable 1 = Enable	Rx DMA DTE Ch Enable 1 = Enable	**CHAR3 Seq. Flag 3rd Detect 1 = Detect**	**CHAR3 Seq. Flag 2nd Detect 1 = Detect**	**CHAR3 Seq. Flag 1st Detect 1 = Detect**

DER6 — A '1' enables the 1st Detect Flag to be set when CHAR3 is detected in the Tx data stream. A '0' prevents the 1st detect flag being set.

DER7 — When set, IRQ4 interrupts are enabled if the Tx CHAR3 Flag (DFR7) is set.

DFR0, DFR1, DFR2 — 1st, 2nd and 3rd CHAR3 sequential detection flags.

DFR7 — Set as shown above. Setting is not dependent on DER6. Cleared by reading DFR.

3.14.9 Register Description

The DMA registers are listed in Table 3-30.

DMA locations 05D0 through 05EF are banked with internal RAM located at these same addresses. Banking is controlled by DSR0. DMA registers located in this address range can only be addressed when DSR0 = 1. All DMA registers located between 05D0 and 05EF are initialized to zero by reset.

Lo WM, Hi WM and ROFF contain 8-bit values that are offset by modulo 16. When they are used to compare against DXM and DXR (16 bit vectors), the Lo WM, Hi WM and ROFF are offset to bits 4 through 11. Bits 0 to 3 remain zero.

Notice that when one writes to the buffer base address, the D*B, the D*I and D*O pointers are also written, see table below. Writing to the base LSB address also clears the associated status bits.

Writing Register	At Address	Also writes to Addresses	And clears status bits:	And sets status bits:
DXB_L	$05D8	$05DC and $05DE	DSR6 and DSR7	Tx Buffer Empty
DXB_H	$05D9	$05DD and $05DF	-	-
DRB_L	$05E8	$05EC and $05EE	DSR5 and Rx Buffer Full	DSR4
DRB_H	$05E9	$05ED and $05EF	-	-

Table 3-30. DMA Registers

ADDRESSES	DSR0 Bit Value	READ	WRITE
05D0 - 05EF	0	**Page 5 RAM (**32 bytes)	**Page 5 RAM** (32 bytes)
05D0	1	**DMA Enable** Register	**DMA Enable** Register
05D1	1	**CHAR1** Character	**CHAR1** Character
05D2	1	**CHAR2** Character	**CHAR2** Character
05D3	1	**CHAR3** Character	**CHAR3** Character
05D4	1	**Lo WM** low water mark	**Lo WM** low water mark
05D5	1	**Hi WM** high water mark	**Hi WM** high water mark
05D6	1	**DXM_L** (LSB)	**DXM_L** (LSB)
05D7	1	**DXM_H** (MSB)	**DXM_H** (MSB)
05D8	1	**DXB_L** (LSB)	**DXB_L**, **DXI_L**, **DXO_L** (LSB)
05D9	1	**DXB_H** (MSB)	**DXB_H**, **DXI_H**, **DXO_H** (MSB)
05DA	1	**DXZ_L** (LSB)	**DXZ_L** (LSB)
05DB	1	**DXZ_H** (MSB)	**DXZ_H** (MSB)
05DC	1	**DXI_L** (LSB)	**DXI_L** (LSB)
05DD	1	**DXI_H** (MSB)	**DXI_H** (MSB)
05DE	1	**DXO_L** (LSB)	**DXO_L** (LSB)
05DF	1	**DXO_H** (MSB)	**DXO_H** (MSB)
05E0	1	-	-
05E1	1	-	-
05E2	1	-	-
05E3	1	-	-
05E4	1	-	-
05E5	1	**ROFF** high water mark	**ROFF** high water mark
05E6	1	**DRM_L** (LSB)	**DRM_L** (LSB)
05E7	1	**DRM_H** (MSB)	**DRM_H** (MSB)
05E8	1	**DRB_L** (LSB)	**DRB_L**, **DRI_L**, **DRO_L** (LSB)
05E9	1	**DRB_H** (MSB)	**DRB_H**, **DRI_H**, **DRO_H** (MSB)
05EA	1	**DRZ_L** (LSB)	**DRZ_L** (LSB)
05EB	1	**DRZ_H** (MSB)	**DRZ_H** (MSB)
05EC	1	**DRI_L** (LSB)	**DRI_L** (LSB)
05ED	1	**DRI_H** (MSB)	**DRI_H** (MSB)
05EE	1	**DRO_L** (LSB)	**DRO_L** (LSB)
05EF	1	**DRO_H** (MSB)	**DRO_H** (MSB)
05F0 - 05F3	x	Reserved - Timer C	Reserved - Timer C
05F4	x	TXB **"GET"** Register	RXB **"PUT"** Register
05F5	x	**DMA Flag** Register	**DMA Flag** Register
05F6	x	**DMA Status** Register	**DMA Status** Register
05F7 - 05F9	x	Reserved - Flash	Reserved - Flash
05FA - 05FB	x	Reserved for Port F Direction/Data	Reserved for Port F Direction/Data
05FC - 05FD	x	Reserved for Port G Direction/Data	Reserved for Port G Direction/Data
05FE - 05FF	x	CRC L/H	CRC Buffer/Init

3.14.10 DMA Timing

A state machine is used to sequence through each DMA operation (Figure 3-26). As each operation ends the other operations are given precedence before an operation can be repeated.

There are four basic operations:

1. **RIN.** If the Rx RAM Buffer is not full, RIN takes data stored in the "PUT" register and stores it in the Rx RAM Buffer at location DRI. There is no bit to indicate that the Rx RAM Buffer is full, however if the Rx_BE status should remain low for at least 43 clock cycles, the user should assume that the buffer is full.

2. **ROUT. R**OUT services the RS232/16550 data request. If the Rx RAM Buffer is not empty, DSR4 = 0, the ROUT routine reads the oldest Rx RAM Buffer data at location DRO and writes it to the RS232/16550 buffer.

3. **TIN.** TIN responds to the RS232/16550 data available signal. If the Tx RAM Buffer is not full, DSR6 = 0,TIN moves the data that has become available at the DTE RS232/16550 interface and stores it in the Tx RAM Buffer at location DXI. The data is also stored internally in the TXBI register. TXBI is used to compare against CHAR1/2/3.

4. **TOUT**. When Tx_BF = 0 and the Tx RAM Buffer is not empty, TOUT moves the oldest Tx RAM data at location DXO into the "GET" register. There is no bit to indicate that the Tx RAM Buffer is empty, however if the Tx_BF flag should remain low for at least 43 clock cycles, the user should assume that the buffer is empty.

The following table gives the timing required to repeat an operation. It assumes that all operations required to increment pointers, check wrap, check empty/full, increment/decrement DXM/DRM, and compare Hi WM, Lo WM and ROFF are performed. The table assumes that each operation must wait one cycle for SYNC, and that cycle steal operates in a divide by 2 slow cycle mode.

Operation	Clock cycles required to repeat operation
RIN	14
ROUT	14
TIN	17
TOUT	14

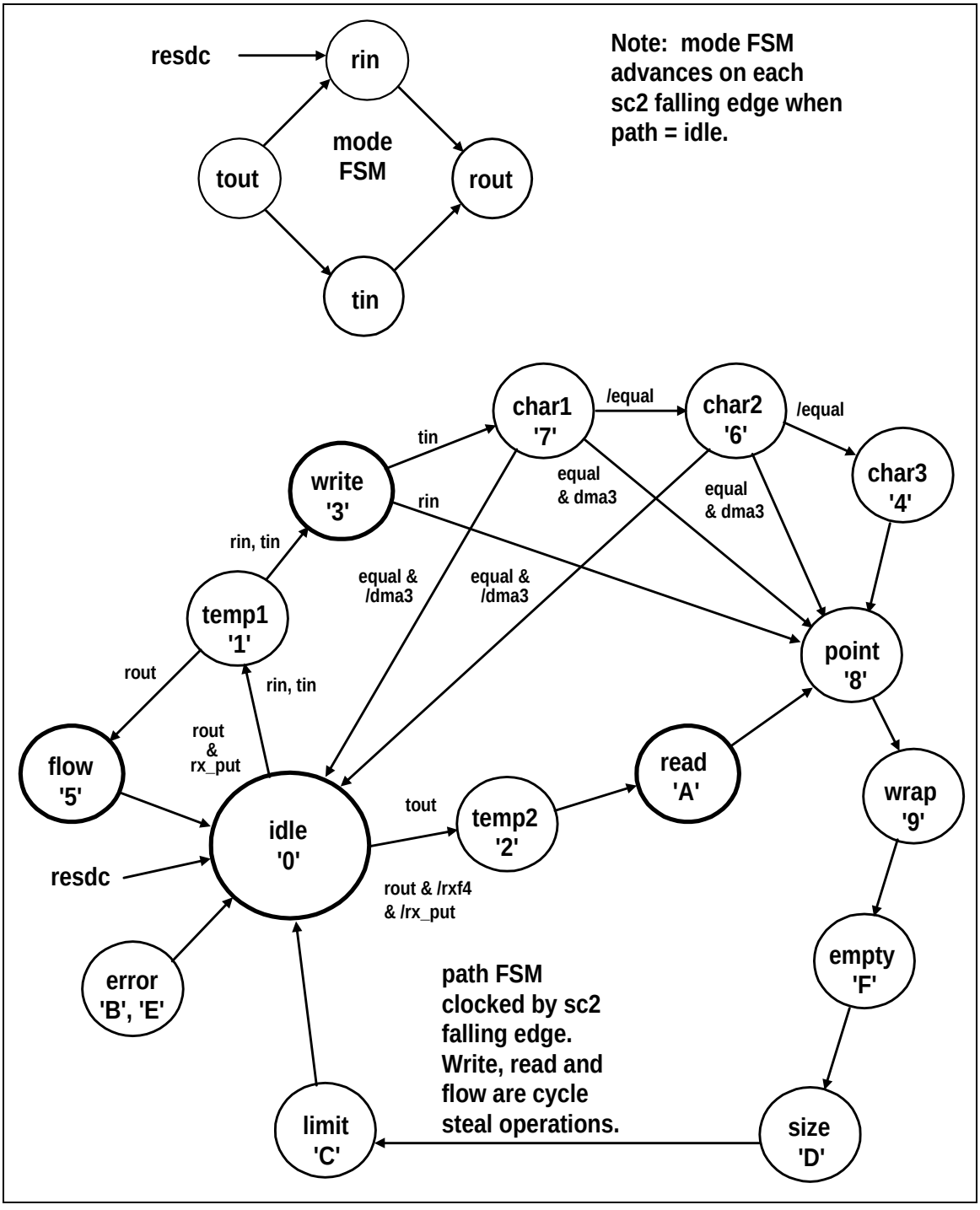


Figure 3-26. DMA State Machine Diagram

3.14.11 Example DMA Setup

Assume that the DMA is to be set up as follows:

- DTE host interface: DER1 = 1.
- Starting external RAM address: $2015
- Tx RAM Buffer size: (475 bytes)$1DB
- Rx RAM Buffer size: (300 bytes) $12C
- Set Hi WM to allow at least 100 remaining bytes.
- Set Lo WM to allow at least 50 remaining bytes.
- Set ROFF to allow at least 25 remaining bytes.
- CHAR1 = $33.
- CHAR2 = $44.
- CHAR3 = $55.
- Set DER7 = 1, DER5 = 1, and DER3 = 1 to enable all 3 DMA IRQ interrupts
- Set the ES1 speed control to a divide by 2 slow mode, cycle steal speed.
- Set DER2 = 0 to not save CHAR1 or CHAR2 characters in the Tx buffer.
- Set DER4 = 1 to enable the auto insertion of CHAR1 and CHAR2 for modem generated flow control.
- Set DER6 = 1 to enable the CHAR3 DFR0 escape sequence,.

3.14.12 Non DMA Setup

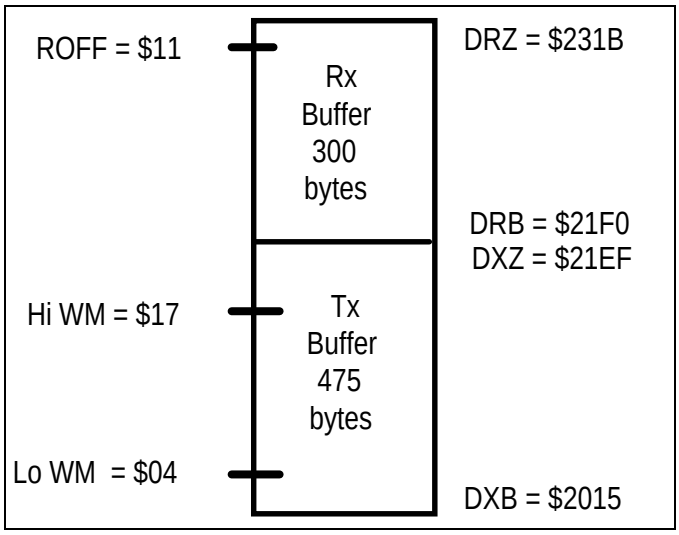
- Initialize the 16550 with all IRQ interrupts disabled.
- Write $04 to ES Speed Register ($0033) to set the ES1 speed control.

3.14.13 DMA Initialization

- Write $01 to DSR ($05F6) to set bank select to DMA.
- Initialize the DMA as shown below:

Parameter	Value
ENAB	$FA
CHAR1	$33
CHAR2	$44
CHAR3	$55
Lo WM	$04
Hi WM	$17
DXB_L - LSB	$15
DXB_H - MSB	$20
DXZ_L - LSB	$EF
DXZ_H - MSB	$21
ROFF	$11
DRB_L - LSB	$F0
DRB_H - MSB	$21
DRZ_L - LSB	$1B
DRZ_H - MSB	$23

- Write $00 to DSR ($05F6) to restore the bank select to internal RAM.
- Write $18 to DFR ($05F5) to enable both the DTE in-bound DMA to 16550 receiver buffer interface, and the DTE out-bound 16550 transmitter buffer to DMA interface. At this point, the DMA is fully activated. The DXI and DXO pointers are initialized to the same address as the Tx buffer base address, TXB. Similarly, the DRI and DRO pointers initialize to the Rx buffer base address DRB. DXM and DRM were initialized by reset to $0000. The values written to CHAR1/2/3 are arbitrary.



DXZ = DXB + 475 -1
= $2015 + $1DB - 1
= $21EF

DRB = DXB + 475
= $2015 + $1DB
= $21F0

DRZ = DRB + 300 -1
= $21F0+ $12C -1
= $231B

Lo WM	= 50 = $032	Since nibble(0) > 0, round nibble(1) up to 4, discard nibble(0).
	= $04	DSR7 transitions low when DXM decrements from 65 to 64.
Hi WM	= 475 -100 = $177	Round down to $170 and discard nibble(0).
	= $17	DSR7 transitions high when DXM increments from 368 to 369, providing a buffer of 107, 475 - 368.
ROFF	= 300 - 25 = $113	Round down to $110 and discard nibble(0).
	= $11	DSR5 = 1 when DRM > 272, providing a buffer of 28 bytes, 300 - 272.

3.14.14 Resetting the DMA

The user can reset or reprogram the DMA at any time. The Rx channel and Tx channels can be reset independently of each other. The following description assumes they are being reset together. Re-configuring the DMA can be accomplished as follows:

1. Disable both the Tx and Rx DTE interfaces by writing $00 to DFR. Wait 20 cycles and verify that Rx_BE = 0 and Tx_BF = 1. These states should exist unless the Rx buffer is full or the Tx buffer is empty. Do not service either the PUT or GET registers.

2. Write new values to the DRB_L, DRB_H and DXB_L, DXB_H registers.

   NOTE: When one writes to the buffer base address, the D*B, the D*I and D*O pointers are also written, see table below. Writing to the base LSB address also clears the associated status bits. The DSR value will be either $12 or $1A.

Writing Register	At Address	Also writes to Addresses	And clears status bits:	And sets status bits:
DXB_L	$05D8	$05DC and $05DE	DSR2, DSR6, DSR7	Tx Buffer Empty
DXB_H	$05D9	$05DD and $05DF	-	-
DRB_L	$05E8	$05EC and $05EE	DSR5 and Rx Buffer Full	DSR1, DSR4
DRB_H	$05E9	$05ED and $05EF	-	-

3. Set the D*Z registers to their new values by writing the DXZ_L, DXZ_H and DRZ_L, DRZ_H registers. This step is not necessary if the values are not changing.

4. Write $00 to DXM_L, DXM_H and DRM_L, DRM_H registers to reset the D*M registers to $0000.

5. Provide as required new values to DMA Enable Register (DER), CHAR1, CHAR2, CHAR3, Lo WM, Hi WM and ROFF.

6. Modify any non DMA setup, RS232, 16550, ES speed control.

7. Write $00 to DSR ($05F6) to restore bank select to internal RAM.

8. Write $18 to DFR ($05F5) to enable both the Rx and Tx DTE interfaces.

3.15 EXPANSION BUS

The expansion bus extends internal address, data and control bus lines outside the MCU. This allows the MCU to operate as a microprocessor by interfacing with external memory and/or other peripheral devices. The expansion bus waveforms and timing are described in Section 5.

3.16 TEST MODE

The Test mode is selected by applying a low voltage to the TSTP pin.

In the Test mode, the internal ROM is deactivated and the expansion bus activated when ROM addresses are selected. Reads from addresses $000C-$000F are mapped to the expansion port.

Internal reads from Page 0 through 5 are mapped externally on the expansion bus whenever the TSTP pin is active. This provides for monitoring of internal read operations.

3.17 MASK OPTIONS

3.17.1 Selectable Options

The mask options are selectable upon production part order.

3.17.2 Mask Option Register (MOR)

The read-only Mask Option Register (MOR) at location $0008 reports the selected package bond options (Table 3-31). This register is for factory use.

Table 3-31. Register Bit Assignments: Mask Option Register - 0008h

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0008	Mask Option Register (MOR) (Read Only)	Bond Option Bit 7 (MOR7)	Bond Option Bit 6 (MOR6)	Bond Option Bit 5 (MOR5)	Bond Option Bit 4 (MOR4)	Bond Option Bit 3 (MOR3)	Bond Option Bit 2 (MOR2)	Bond Option Bit 1 (MOR1)	Bond Option Bit 0 (MOR 0)

3.18 CYCLIC REDUNDANCY CHECK (CRC) HARDWARE

The MCU contains Cyclic Redundancy Check (CRC) hardware consisting of an input buffer and the standard 16-bit CRC polynomial algorithm shown below:

$$CRC16 = (x^{16} + x^{12} + x^{5} + 1)$$

Mechanization of the 16-bit CRC is shown in Figure 3-27 and Table 3-32. Writing to address $05FF initializes the CRC polynomial to all ones. Data is loaded into the CRC Input Buffer by writing to address $05FE. The parallel-in serial-out shift register is loaded with data by a CPU write to address $05FE. Input buffer data is shifted serially LSB first into the CRC polynomial circuit. A minimum of eight clock cycles must be allowed between CPU writes. The resulting polynomial can be read at any time after allowing for the shifting operation to complete. The LSB (CRC Low Byte) of the polynomial is read at address $05FE and the MSB (CRC High Byte) is read at address $05FF.

Details of the polynomial implementation are shown in Figure 3-28.

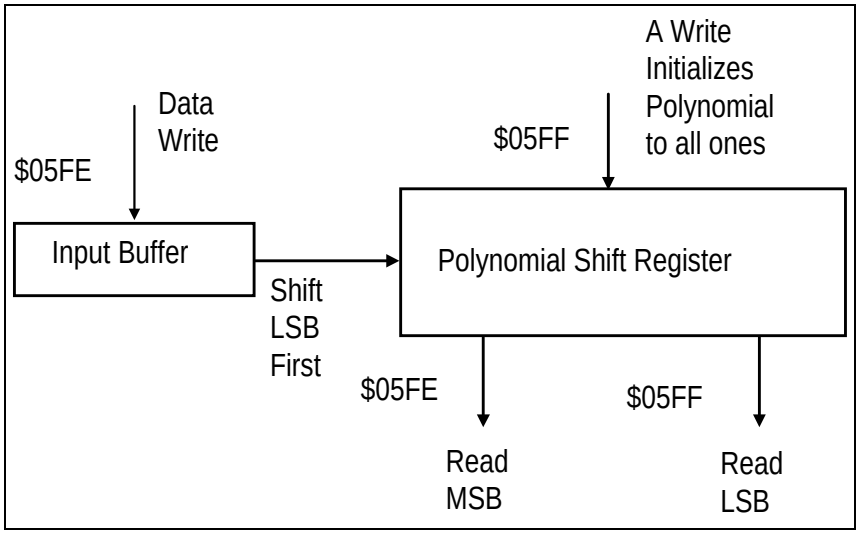


Figure 3-27. 16-bit CRC

Table 3-32. Register Bit Assignments: 05FEh - 05FFh

Addr.	Function	Bit 7	6	5	4	3	2	1	0
05FE	CRC Low Byte (Read only)	CRC 7	CRC 6	CRC 5	CRC 4	CRC 3	CRC 2	CRC 1	CRC 0
05FE	CRC Data In (Write only)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
05FF	CRC High Byte (Read only)	CRC 15	CRC 14	CRC 13	CRC 12	CRC 11	CRC 10	CRC 9	CRC 8
05FF	Initialize CRC (Write only)	A write to $05FF initializes the CRC to $FFFF.							

When repetitive data is presented to the CRC, the polynomial will generate a unique signature which can be used to verify the correctness of the received data. If the sending party attaches an inverted copy of the unique signature generated by an identical 16 bit polynomial to the end of its message, the receiving polynomial shift register will always contain $F0B8 unless an error occurred. Assuming that the correct signature is known and that the correct sequence of bytes has been sent to the polynomial shift register, the accuracy of the received data CRC signature can be checked as follows:

1. Initialize polynomial shift register to all 1's by writing to $05FF.
2. Write all message bytes in sequence to input buffer, address $05FE.
3. Invert the correct LSB signature and write to the input buffer.
4. Invert the correct MSB signature and write to the input buffer.
5. The polynomial shift register will contain $F0B8 if the received data was received correctly.

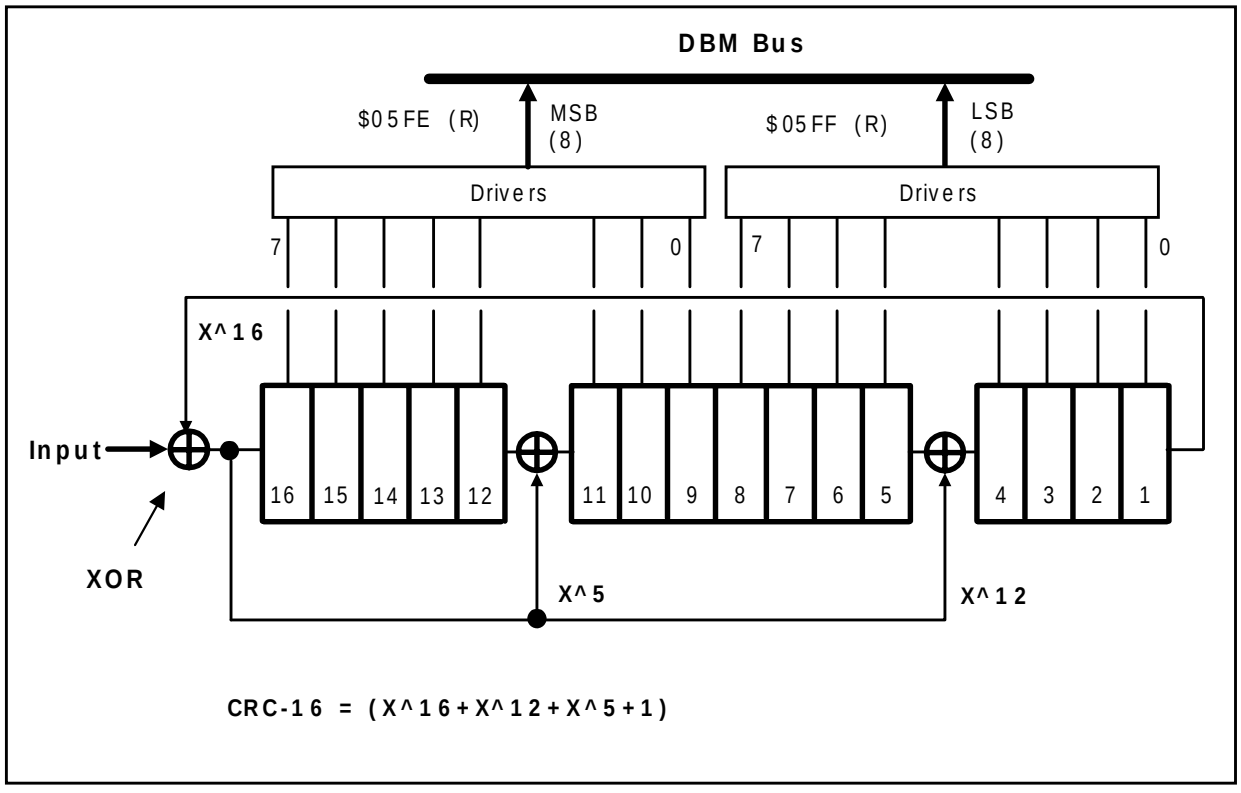


Figure 3-28. CRC-16 Polynomial

This page is intentionally blank.

4 GENERAL SPECIFICATIONS

4.1 OPERATING CONDITIONS AND MAXIMUM RATINGS

4.1.1 Operating Conditions

Parameter	Symbol	Limits	Units
Supply Voltage	VDD	+4.75 to +5.25	V
Operating Temperature	TA	-0 to + 70	°C

4.1.2 Maximum Ratings

Parameter	Symbol	Limits	Units
Supply Voltage	VDD	-0.5 to + 6.0	V
Input Voltage	VIN	-0.5 to VDD + 0.5	V
Storage Temperature	TS	-55 to +125	°C
Voltage Applied to Outputs in High Z State	VHZ	-0.5 to VDD + 0.5	V
DC Input Clamp Current	IIK	±20	mA
DC Output Clamp Current	IOK	±20	mA
Static Discharge Voltage (25°C)	ESD	±2500	V
Latch-up Current (25°C)	ITRIG	±400	mA

4.1.3 Handling CMOS Devices

a. The device contains circuitry to protect the inputs against damage due to high static voltages. However, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltage.

b. An unterminated input can acquire unpredictable voltages through coupling with stray capacitance and internal crosstalk. Both power dissipation and device noise immunity degrades. Therefore, all inputs should be connected to an appropriate supply voltage.

c. Input signals should never exceed the voltage range from 0.5V or more negative than GND to 0.5V or more positive than $V_{DD.}$ This prevents forward biasing the input protection diodes and possibly causing a latch up mode due to high current transients.

4.2 ELECTRICAL CHARACTERISTICS

Characteristics	Symbol	Min	Typ	Max	Units	Test Conditions*
Input High Voltage	VIH					
TTL		2.0	--	VDD + 0.3	V	
RESP, NMIP, TSTP		0.7 VDD	--	VDD + 0.3	V	
Input Low Voltage	VIL					
TTL		-0.3	--	0.8	V	
RESP, NMI, TSTP		-0.3	--	0.8	V	
Output High Voltage	VOH					
Except PC0-PC7		2.4	--	--	V	I_{load} = -100 µA
PC0-PC7		2.4	--	--	V	I_{load} = -6 mA
Output Low Voltage	VOL					
Except PC0-PC7		--	--	0.4	V	I_{load} = 1.6 mA
PC0-PC7		--	--	0.4	V	I_{load} = 6 mA
3-state Output Hi-Z Current	IOZ	--	--	± 10	µA	V_{in} = 0 V to V_{DD}
Input Leakage Current	II					
RESP, PD5 - PD7, and XTLI		--	--	± 10	µA	V_{in} = 0 V to V_{DD}
NMIP and TSTP		--	--	- 20	µA	V_{in} = V_{DD}
Other		15	--	100	µA	V_{in} = 0 V

* Test Conditions: VDD = 5 V ±5%, TA = -0°C to +70°C, unless otherwise noted.

4.3 POWER DISSIPATION

Mode	Current (I_D)			Power (P_D)			Notes*
	Typical	Maximum	Units	Typical	Maximum	Units	
Normal mode	2.4	2.55	mA/MHz	12.0	13.5	mW/MHz	
Stop mode	0.28	0.30	mA/MHz	1.4	1.6	mW/MHz	
Stop mode	0.28	0.30	mA	1.4	1.6	mW	
Normal mode	68	72	mA	340	380	mW	f = 28.224 MHz
Stop mode	8.0	8.6	mA	40	45	mW	
Stop mode	0.3	0.4	mA	1.5	2.1	mW	

Notes:
VDD = 5.0 VDC for typical values; VDD = 5.25 VDC for maximum values.

This page is intentionally blank.

5 TIMING CHARACTERISTICS

5.1 TEST CONDITIONS

The following conditions apply for all timing descriptions unless otherwise noted:

1. T_A = -0°C to +70°C.
2. VDD = 5V ± 5%.
3. Output loads = 50 pF + one TTL load.
4. Data bus, address bus, chip selects, RDP, and WTP input loads = 70 pF + one TTL load.
5. All times in nanoseconds (ns) except where noted.

5.2 OSCILLATOR TIMING

The MCU internal oscillator is designed to operate with an external crystal and external capacitors C_{out} and C_{in}. Be aware that at these frequencies component placement, lead lengths and grounding are critical and that component fine tuning may be required for each new circuit layout.

5.3 HOST BUS INTERFACE

The Host Bus read timing is listed in Table 5-1 and is shown in Figure 5-1 The Host Bus write timing is listed in Table 5-2 and is shown in Figure 5-2. Host Bus interrupt timing is described in Table 5-3 and Figure 5-3.

Table 5-1. Host Bus Read Timing

Parameter	Symbol	Min	Max	Units
Address Setup	t AS	5	-	ns
Address Hold	t AH	10	-	ns
Chip Select Setup	t CS	0	-	ns
Chip Select Hold	t CH	10	-	ns
Read Pulse Width	t RD	tCYC + 15	-	ns
HRDP to Data Delay	t DD	-	25	ns
HRDP to Data Hold	t DRH	10	-	ns
HDIS Enable	t DIS	-	15	ns
HDIS Hold	t DIH	0	-	ns

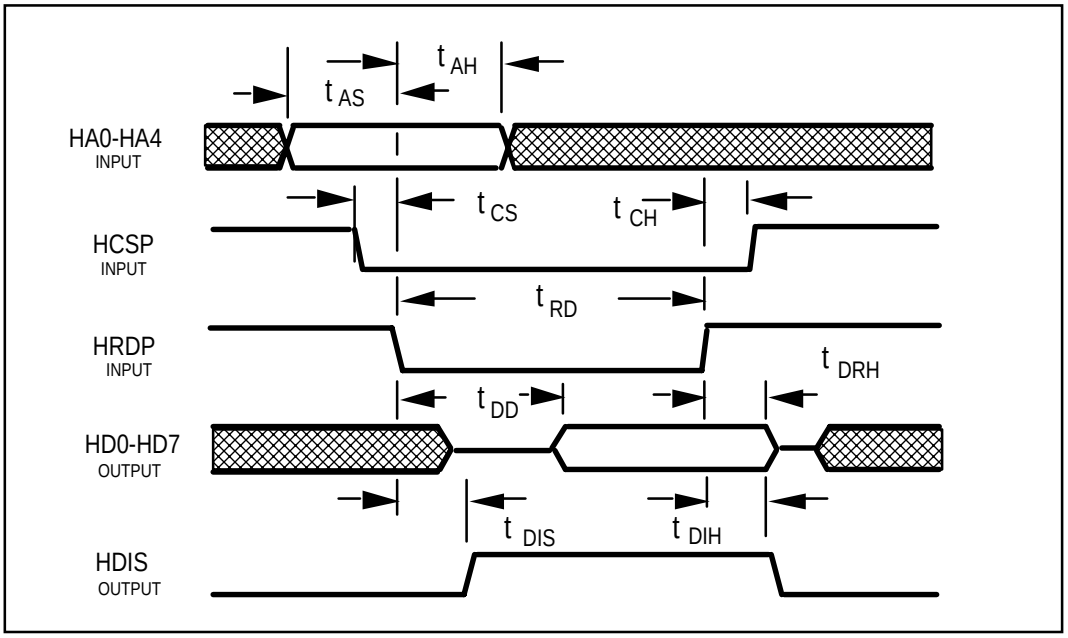


Figure 5-1. Host Bus Read Waveforms

When the Host executes consecutive Rx FIFO reads, a minimum delay of 2 times the internal MCU clock cycle plus 15 ns (215 ns at 10 MHz) is required from the falling edge of HRDP to the falling edge of the next Host Rx FIFO HRDP clock.

Table 5-2. Host Bus Write Timing

Parameter	Symbol	Min	Max	Units
Address Setup	t AS	5	-	ns
Address Hold	t AH	10	-	ns
Chip Select Setup	t CS	0	-	ns
Chip Select Hold	t CH	10	-	ns
Write Pulse Width	t WT	tCYC + 15	-	ns
Write Data Setup*	t DS	-	tCYC	ns
Write Data Hold**	t DWH	5	-	ns

* t_{DS} measured from the point at which both HCSP and HWTP are active.

** t_{DWH} measured from the point at which either HCSP or HWTP becomes inactive.

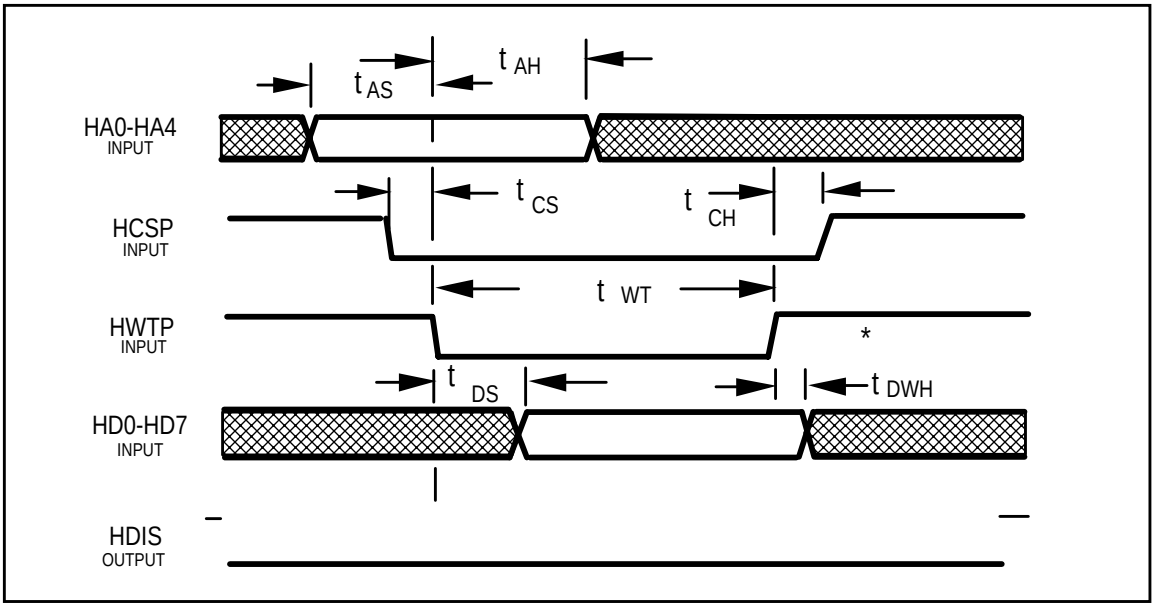


Figure 5-2. Host Bus Write Waveforms

When the Host executes consecutive Tx FIFO writes, a minimum delay of 2 times the internal MCU clock cycle plus 15 ns (215 ns at 10 MHz) is required from the falling edge of HWTP to the falling edge of the next Host Tx FIFO HWTP clock. In the GP mode, following any Host write, a minimum delay of 2 times the internal MCU clock cycle time (200 ns at 10 MHz) is required from the rising edge of HWTP to the falling edge of the next selected Host HWTP or HRDP clock.

Table 5-3. Host Bus HINT Timing

Parameter	Symbol	Typ	Units
HINT Interrupt	tINT	25	ns

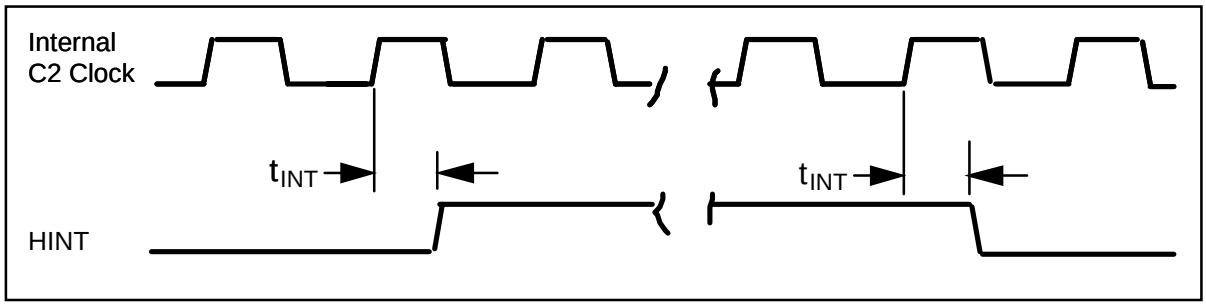


Figure 5-3. Host Bus HINT Waveforms

5.4 EXPANSION BUS TIMING

The Expansion Bus read timing is listed in Table 5-4 and is shown in Figure 5-4. The Expansion Bus write timing is listed in Table 5-5 and is shown in Figure 5-5. An 8-bit data I/O bus (D0-D7), an 18-bit address bus (A0-A17), a RDP clock, and a WTP clock provide an extended memory expansion bus interface. Addresses 13-17 are controlled by the selected Banking RAM register. Chip selects ES0, ES2, and ES3 are also controlled by the banking RAM. ES4 is memory mapped at $0600-$07FF. The Port B select register controls the selection of A16, A17, ES0, ES2, and ES3. CI2 provides half/full speed control for ES4. The ESS register provides one fourth, one third, one half, or full speed control for A17, ES0, ES2, and ES3.

Table 5-4. Expansion Bus Read Timing

Parameter	Symbol	Min	Typ	Max	Units
External Crystal Frequency	fCRY	-	-	35	MHz
Internal Clock Cycle	t CYC	28	-	10^4	ns
RDP ↑ to Address Valid	t AS	-	10.6	12.4	ns
RDP ↑ to ES Valid	t ES	-	11.6	13.5	ns
RDP ↓ to RDP ↑ Pulse Width	t RW	t CYC/2		-	ns
Read Data Valid to RDP ↑	t RDS	6.1	3.5	-	ns
RDP ↑ to Read Data Hold	t RDH	0		-	ns

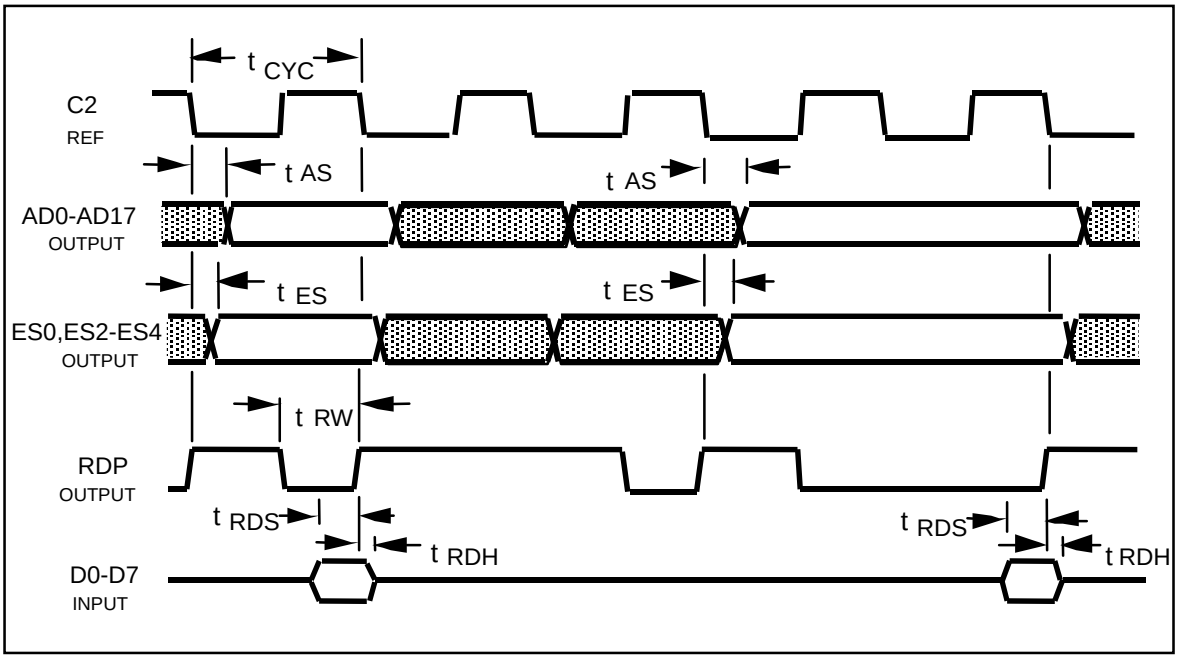


Figure 5-4. Expansion Bus Read Waveforms

Table 5-5. Expansion Bus Write Timing

Parameter	Symbol	Min	Typ	Max	Units
External Crystal Frequency	fCRY	-	-	35	MHz
Internal Clock Cycle	t CYC	28	-	10^4	ns
WTP ↑ to Address Valid	t AS	-	10.6	12.4	ns
WTP ↑ to ES Valid	t ES	-	11.6	13.5	ns
WTP ↓ to WTP ↑ Pulse Width	t WW	t CYC/2		-	ns
WTP ↓ to Write Data Valid	t WTD	-	7.3	8.2	ns
WTP ↑ to Write Data Hold	t WTH	5.0		-	ns

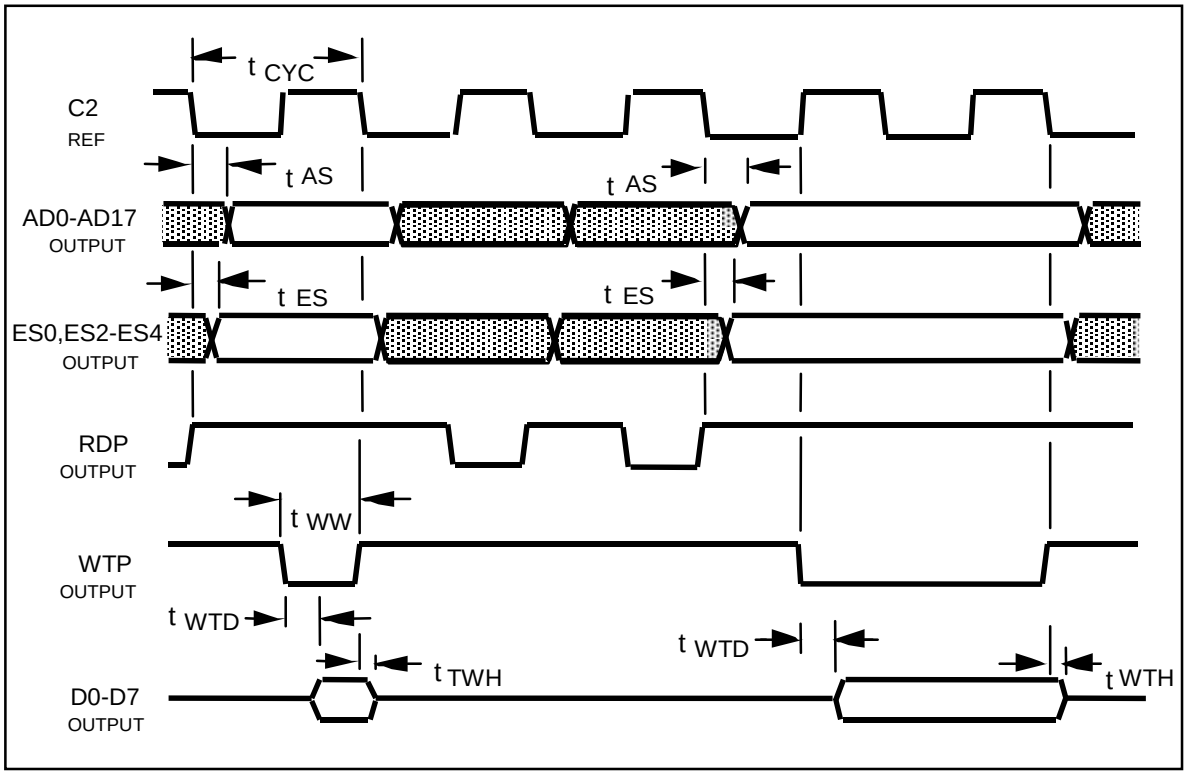


Figure 5-5. Expansion Bus Write Waveforms

5.5 RESP, NMIP, AND TSTP ASYNCHRONOUS INPUTS

The timing for the RESP, NMIP, and TSTP asynchronous inputs is shown in Figure 5-6 and is listed in Table 5-6. These inputs can have only one transition during each C2 period. The XTLI and RESP timing during MCU power turn-on is shown in Figure 5-7.

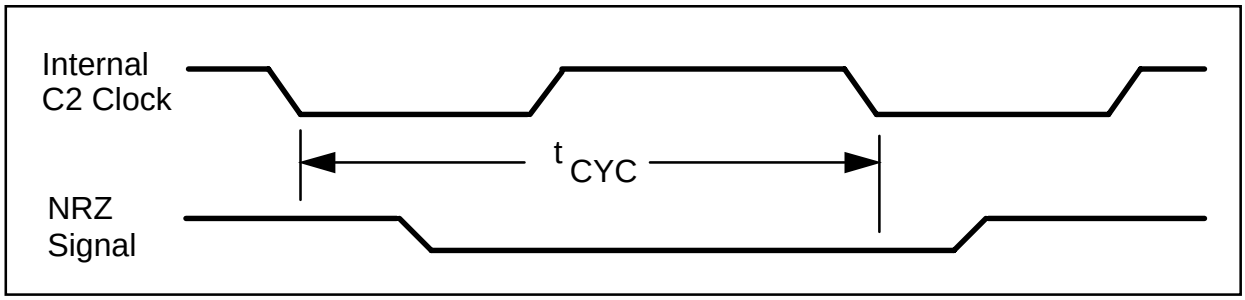


Figure 5-6. RESP, NMIP, and TSTP Input Waveforms

Table 5-6. Power On RESP, NMIP, and TSTP Timing

Term Name	Signal Active State	Minimum Active State	Remarks
RESP	low	2 tcyc*	MCU reset
NMIP	low	tcyc	Non-Maskable Interrupt
TSTP	low	tcyc	Emulate mode (factory use only)
*During power up, a minimum of 10 t_{cyc} must occur after the crystal oscillator has stabilized.			

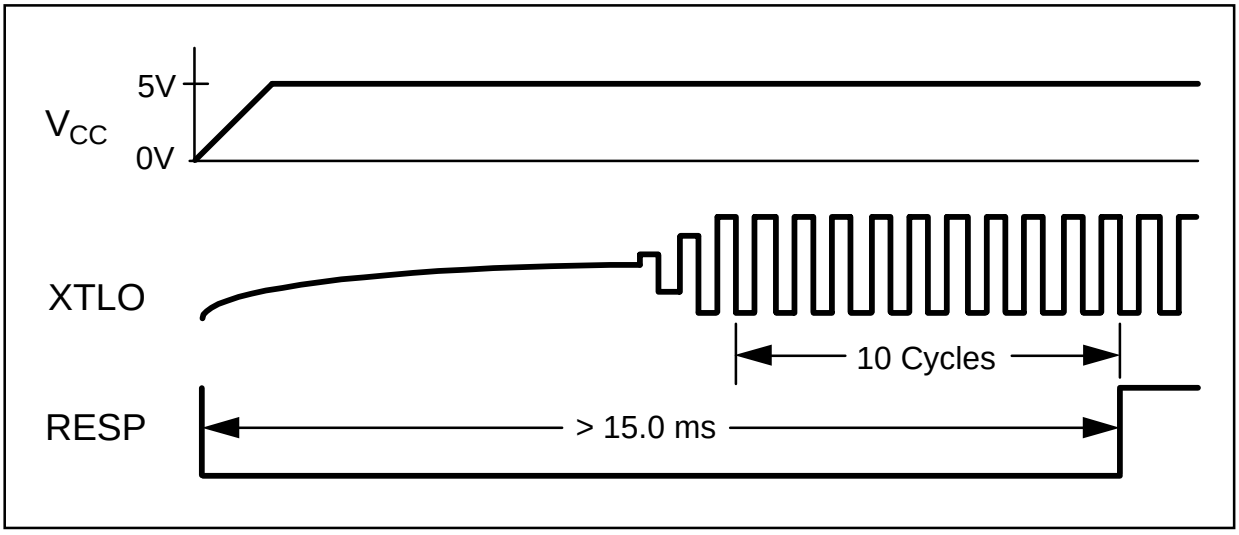
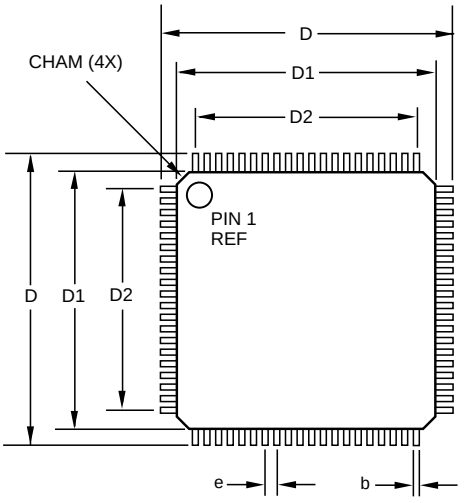


Figure 5-7. Power On RESP Timing

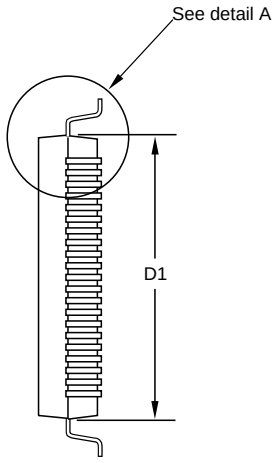
This page is intentionally blank.

6 PACKAGE DIMENSIONS

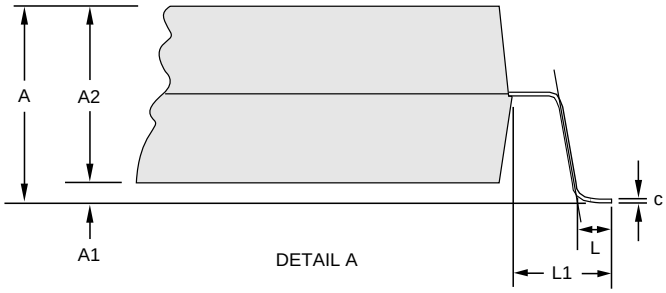
The package dimensions for the 80-pin PQFP are shown in Figure 6-1.



TOP VIEW



SIDE VIEW



DETAIL A

Dim.	Millimeters Min.	Millimeters Max.	Inches* Min.	Inches* Max.
A	2.4 MAX		0.0945 MAX	
A1	0.05	0.35	0.0020	0.0138
A2	2.0 REF		0.0787 REF	
D	16.95	17.45	0.6673	0.6870
D1	14.0 REF		0.5512 REF	
D2	12.35 REF		0.4862 REF	
L	0.73	1.03	0.0287	0.0406
L1	1.6 REF		0.0630 REF	
e	0.65 BSC		0.0256 BSC	
b	0.25	0.45	0.0098	0.0177
c	0.13	0.19	0.0051	0.0075
Coplanarity	0.1 MAX		0.004 MAX	

Ref: 80-PIN PQFP (GP00-D227)

* Metric values (millimeters) should be used for PCB layout. English values (inches) are converted from metric values and may include round-off errors.

PD-PQFP-80 (040695)

Figure 6-1. Package Dimensions - 80-Pin PQFP

A. CPU INSTRUCTION SET SUMMARY

This appendix summarizes the CPU instruction set.

Table A-1 is a matrix of CPU instructions and addressing modes arranged by operation code.

Table A-2 lists the CPU instruction mnemonics and titles by mnemonic code in Table A-1.

Table A-3 lists the CPU instruction operation codes, mnemonics and addressing modes in op code order.

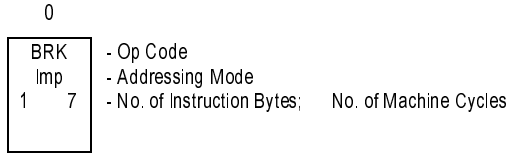
Table A-4 summarizes the operation of the CPU instructions as referenced to the R6502 CPU.

Table A-5 summarizes the differences in operation between the MCU CPU and the R6502 CPU.

Table A-6 summarizes the threaded code instructions.

Table A-1. CPU Instruction Set Operation Code Matrix

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK Imp 1 7	ORA (Ind) 2 5	MPY Imp 1 6	TIP Imp 1 2		ORA Zp 2 3	ASL Zp 2 5	RMB0 Zp 2 5	PHP Imp 1 3	ORA Imm 2 2	ASL Acc 1 2	JSB0 (FFE0) 1 6	JPI Imp 3 5	ORA Abs 3 4	ASL Abs 3 6	BBR0 Zp 3 5[b]	0
1	BPL Rel 2 2[b]	ORA (Ind),X 2 5[a]	MPA Imp 1 6	LAB Acc 1 3		ORA Zp,X 2 4	ASL Zp,X 2 6	RMB1 Zp 2 5	CLC Imp 1 2	ORA Abs,Y 3 4[a]	NEG Acc 1 2	JSB1 (FFE2) 1 6		ORA Abs,X 3 4[a]	ASL Abs,X 3 7	BBR1 Zp 3 5[b]	1
2	JSR Abs 3 5	AND (Ind) 2 5	PSH Imp 1 5	PHW Imp 1 4	BIT Zp 2 3	AND Zp 2 3	ROL Zp 2 5	RMB2 Zp 2 5	PLP Imp 1 4	AND Imm 2 2	ROL Acc 1 2	JSB2 (FFE4) 1 6	BIT Abs 3 4	AND Abs 3 4	ROL Abs 3 6	BBR2 Zp 3 5[b]	2
3	BMI Rel 2 2[b]	AND (Ind),X 2 5[a]	PUL Imp 1 6	PLW Imp 1 5		AND Zp,X 2 4	ROL Zp,X 2 6	RMB3 Zp 2 5	SEC Imp 1 2	AND Abs,Y 3 4[a]	ASR Acc 1 2	JSB3 (FFE6) 1 6		AND Abs,X 3 4[a]	ROL Abs,X 3 7	BBR3 Zp 3 5[b]	3
4	RTI Imp 1 6	EOR (Ind) 2 5	RND Imp 1 2			EOR Zp 2 3	LSR Zp 2 5	RMB4 Zp 2 5	PHA Imp 1 3	EOR Imm 2 2	LSR Acc 1 2	JSB4 (FFE8) 1 6	JMP Abs 3 3	EOR Abs 3 4	LSR Abs 3 6	BBR4 Zp 3 5[b]	4
5	BVC Rel 2 2[b]	EOR (Ind),X 2 5[a]	CLW Imp 1 2			EOR Zp,X 2 4	LSR Zp,X 2 6	RMB5 Zp 2 5	CLI Imp 1 2	EOR Abs,Y 3 4[a]	PHY Imp 1 3	JSB5 (FFEA) 1 6		EOR Abs,X 3 4[a]	LSR Abs,X 3 7	BBR5 Zp 3 5[b]	5
6	RTS Imp 1 5	ADC (Ind) 2 5[c]	TAW Imp 1 2		ADD Zp 2 3[c]	ADC Zp 2 3[c]	ROR Zp 2 5	RMB6 Zp 2 5	PLA Imp 1 4	ADC Acc 2 2[c]	ROR Acc 1 2	JSB6 (FFEC) 1 6	JMP (Abs) 3 5	ADC Abs 3 4[c]	ROR Abs 3 6	BBR6 Zp 3 5[b]	6
7	BVS Rel 2 2[b]	ADC (Ind),X 2 5[a,c]	TWA Imp 1 2		ADD Zp,X 2 4[c]	ADC Zp,X 2 4[c]	ROR Zp,X 2 6	RMB7 Zp 2 5	SEI Imp 1 2	ADC Abs,Y 3 4[a,c]	PLY Imp 1 4	JSB7 (FFEE) 1 6	JMP (Abs,X) 3 6	ADC Abs,X 3 4[a,c]	ROR Abs,X 3 7	BBR7 Zp 3 5[b]	7
8	BRA Rel 2 3[a]	STA (Ind) 2 5			STY Zp 2 3	STA Zp 2 3	STX Zp 2 3	SMB0 Zp 2 5	DEY Imp 1 2	ADD Imm 2 2[c]	TXA Imp 1 2	NXT Imp 1 4	STY Abs 3 4	STA Abs 3 4	STX Abs 3 4	BBS0 Zp 3 5[b]	8
9	BCC Rel 2 2[b]	STA (Ind),X 2 6			STY Zp,X 2 4	STA Zp,X 2 4	STX Zp,Y 2 4	SMB1 Zp 2 5	TYA Imp 1 2	STA Abs,Y 3 5	TXS Imp 1 2	LII Imp 1 5		STA Abs,X 3 5		BBS1 Zp 3 5[b]	9
A	LDY Imm 2 2	LDA (Ind) 2 5	LDX Imm 2 2		LDY Zp 2 3	LDA Zp 2 3	LDX Zp 2 3	SMB2 Zp 2 5	TAY Imp 1 2	LDA Imm 2 2	TAX Imp 1 2	LAN Imp 1 3	LDY Abs 3 4	LDA Abs 3 4	LDX Abs 3 4	BBS2 Zp 3 5[b]	A
B	BCS Rel 2 2[b]	LDA (Ind),X 2 5[a]	STI Zp 3 4		LDY Zp,X 2 4	LDA Zp,X 2 4	LDX Zp,Y 2 4	SMB3 Zp 2 5	CLV Imp 1 2	LDA Abs,Y 3 4[a]	TSX Imp 1 2	INI Imp 1 3	LDY Abs,X 3 4[a]	LDA Abs,X 3 4[a]	LDX Abs,Y 3 4[a]	BBS3 Zp 3 5[b]	B
C	CPY Imm 2 2	CMP (Ind) 2 5	RBA Abs 4 7		CPY Zp 2 3	CMP Zp 2 3	DEC Zp 2 5	SMB4 Zp 2 5	INY Imp 1 2	CMP Imm 2 2	DEX Imp 1 2	PHI Imp 1 4	CPY Abs 3 4	CMP Abs 3 4	DEC Abs 3 6	BBS4 Zp 3 5[b]	C
D	BNE Rel 2 2[b]	CMP (Ind),X 2 5[a]	SBA Abs 4 7		EXC Zp,X 2 5	CMP Zp,X 2 4	DEC Zp,X 2 6	SMB5 Zp 2 5	CLD Imp 1 2	CMP Abs,Y 3 4[a]	PHX Imp 1 3	PLI Imp 1 6		CMP Abs,X 3 4[a]	DEC Abs,X 3 7	BBS5 Zp 3 5[b]	D
E	CPX Imm 2 2	SBC (Ind) 2 5[c]	BAR Abs 5 7[b]		CPX Zp 2 3	SBC Zp 2 3[c]	INC Zp 2 5	SMB6 Zp 2 5	INX Imp 1 2	SBC Imm 2 2[c]	NOP Imp 1 2	LAI Imp 1 3	CPX Abs 3 4	SBC Abs 3 4[c]	INC Abs 3 6	BBS6 Zp 3 5[b]	E
F	BEQ Rel 2 2[b]	SBC (Ind),X 2 5[a,c]	BAS Abs 5 7[b]			SBC Zp,X 2 4[c]	INC Zp,X 2 6	SMB7 Zp 2 5	SED Imp 1 2	SBC Abs,Y 3 4[a,c]	PLX Imp 1 4	PIA Imp 1 6		SBC Abs,X 3 4[a,c]	INC Abs,X 3 7	BBS7 Zp 3 5[b]	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	



a. Add 1 to N if page boundary is crossed.

b. Add 1 to N if branch occurs to same page.
Add 2 to N if branch occurs to different page.

c . Add 1 to N if in decimal mode.

Table A-2. CPU Instruction Set by Mnemonic

Mnemonic	Instruction	Mnemonic	Instruction
ADC	Add Memory to Accumulator with Carry	MPA	Multiply and Accumulate
ADD	Add Memory to Accumulator without Carry	MPY	Multiply
AND	"AND" Memory with Accumulator		
ASL	Shift Left One Bit (Memory or Accumulator)	NEG	Negate Accumulator
ASR	Accumulator Shift Right One Bit, Sign Extend	NOP	No Operation
		NXT	Next Instruction
BAR	Branch On Bit(s) Reset		
BAS	Branch On Bit(s) Set	ORA	"OR" Memory with Accumulator
BBR	Branch On Bit Reset (8)		
BBS	Branch On Bit Set (8)	PHA	Push Accumulator on Stack
BCC	Branch on Carry Clear	PHI	Push I on Stack
BCS	Branch on Carry Set	PHP	Push Processor Status on Stack
BEQ	Branch on Equal	PHW	Push W on Stack
BIT	Test Bits in Memory with Accumulator	PHX	Push Index X on Stack
BMI	Branch on Minus	PHY	Push Index Y on Stack
BNE	Branch on Not Zero	PIA	Pull I from Stack, Load Accumulator
BPL	Branch on Plus	PLA	Pull Accumulator from Stack
BRA	Branch Always	PLI	Pull I from Stack
BRK	Break Command	PLP	Pull Processor Status from Stack
BVC	Branch on Overflow Clear	PLW	Pull W from Stack
BVS	Branch on Overflow Set	PLX	Pull Index X from Stack
		PLY	Pull Index Y from Stack
CLC	Clear Carry Flag	PSH	Push A, X and Y on Stack
CLD	Clear Decimal Mode	PUL	Pull Y, X and A from Stack
CLI	Clear Interrupt Disable Bit		
CLV	Clear Overflow Flag	RBA	Reset Bit(s) in Memory
CLW	Clear W Register and Overflow Flag	RMB	Reset Memory Bit (8)
CMP	Compare Memory and Accumulator	RND	Round
CPX	Compare Memory and Index X	ROL	Rotate Left One Bit (Memory or Accumulator)
CPY	Compare Memory and Index Y	ROR	Rotate Right One Bit (Memory or Accumulator)
		RTI	Return from Interrupt
DEC	Decrement Memory by One	RTS	Return from Subroutine
DEX	Decrement Index X by One		
DEY	Decrement Index Y by One	SBA	Set Bit(s) in Memory
		SBC	Subtract Memory from Accumulator with Borrow
EOR	"Exclusive-Or" Memory with Accumulator	SEC	Set Carry Flag
EXC	Exchange Accumulator and Memory	SED	Set Decimal Mode
		SEI	Set Interrupt Disable Status
INC	Increment Memory by One	SMB	Set Memory Bit (8)
INI	Increment I by One	STA	Store Accumulator in Memory
INX	Increment Index X by One	STI	Store Immediate to Memory
INY	Increment Index Y by One	STX	Store Index X in Memory
		STY	Store Index Y in Memory
JMP	Jump to New Location		
JPI	Jump Indirect with Return in I	TAW	Transfer Accumulator to W
JSB	Jump to Subroutine (8)	TAX	Transfer Accumulator to Index X
JSR	Jump to New Location Saving Return Address	TAY	Transfer Accumulator to Index Y
		TIP	Transfer I to Program Counter
LAB	Load Absolute to Accumulator	TSX	Transfer Stack Pointer to Index X
LAI	Load Accumlator Indirect through I	TWA	Transfer W to Accumulator
LAN	Load Accumulator Indirect and Increment I	TXA	Transfer Index X to Accumulator
LDA	Load Accumulator with Memory	TXS	Transfer Index X to Stack Pointer
LDX	Load Index X with Memory	TYA	Transfer Index Y to Accumulator
LDY	Load Index Y with Memory		
LII*	Load I Indirect through I		
LSR	Logical Shift Right One Bit (Memory or Accumulator)		

Table A-3. CPU Instruction Set by Operation Code

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
00	BRK	Implied	20	JSR	Absolute
01	ORA	(Indirect)	21	AND	(Indirect)
02	MPY	Implied	22	PSH	Implied
03	TIP	Implied	23	PHW	Implied
04	Not Used		24	BIT	Zero Page
05	ORA	Zero Page	25	AND	Zero Page
06	ASL	Zero Page	26	ROL	Zero Page
07	RMB 0	Zero Page	27	RMB 2	Zero Page
08	PHP	Implied	28	PLP	Implied
09	ORA	Immediate	29	AND	Immediate
0A	ASL	Accumulator	2A	ROL	Accumulator
0B	JSB 0	(FFE0)	2B	JSB 2	(FFE4)
0C	JPI	Implied	2C	BIT	Absolute
0D	ORA	Absolute	2D	AND	Absolute
0E	ASL	Absolute	2E	ROL	Absolute
0F	BBR 0	Zero Page	2F	BBR 2	Zero Page
10	BPL	Relative	30	BMI	Relative
11	ORA	(Indirect),X	31	AND	(Indirect),X
12	MPA	Implied	32	PUL	Implied
13	LAB	Accumulator	33	PLW	Implied
14	Not Used		34	Not Used	
15	ORA	Zero Page,X	35	AND	Zero Page,X
16	ASL	Zero Page,X	36	ROL	Zero Page,X
17	RMB 1	Zero Page	37	RMB 3	Zero Page
18	CLC	Implied	38	SEC	Implied
19	ORA	Absolute,Y	39	AND	Absolute,Y
1A	NEG	Accumulator	3A	ASR	Accumulator
1B	JSB 1	(FFE2)	3B	JSB 3	(FFE6)
1C	Not Used		3C	Not Used	
1D	ORA	Absolute,X	3D	AND	Absolute,X
1E	ASL	Absolute,X	3E	ROL	Absolute,X
1F	BBR 1	Zero Page	3F	BBR 3	Zero Page

Table A-3. CPU Instruction Set by Operation Code (Cont'd)

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
40	RTI	Implied	60	RTS	Implied
41	EOR	(Indirect)	61	ADC	(Indirect)
42	RND	Implied	62	TAW	Implied
43	Not Used		63	Not Used	
44	Not Used		64	ADD	Zero Page
45	EOR	Zero Page	65	ADC	Zero Page
46	LSR	Zero Page	66	ROR	Zero Page
47	RMB 4	Zero Page	67	RMB 6	Zero Page
48	PHA	Implied	68	PLA	Implied
49	EOR	Immediate	69	ADC	Immediate
4A	LSR	Accumulator	6A	ROR	Accumulator
4B	JSB 4	(FFE8)	6B	JSB 6	(FFEC)
4C	JMP	Absolute	6C	JMP	(Absolute)
4D	EOR	Absolute	6D	ADC	Absolute
4E	LSR	Absolute	6E	ROR	Absolute
4F	BBR 4	Zero Page	6F	BBR 6	Zero Page
50	BVC	Relative	70	BVS	Relative
51	EOR	(Indirect),X	71	ADC	(Indirect),X
51	CLW	Implied	72	TWA	Implied
53	Not Used		73	Not Used	
54	Not Used		74	ADD	Zero Page,X
55	EOR	Zero Page,X	75	ADC	Zero Page,X
56	LSR	Zero Page,X	76	ROR	Zero Page,X
57	RMB 5	Zero Page	77	RMB 7	Zero Page
58	CLI	Implied	78	SEI	Implied
59	EOR	Absolute,Y	79	ADC	Absolute,Y
5A	PHY	Implied	7A	PLY	Implied
5B	JSB 5	(FFEA)	7B	JSB 7	(FFEE)
5C	Not Used		7C	JMP	Absolute,X
5D	EOR	Absolute,X	7D	ADC	Absolute,X
5E	LSR	Absolute,X	7E	ROR	Absolute,X
5F	BBR 5	Zero Page	7F	BBR 7	Zero Page

Table A-3. CPU Instruction Set by Operation Code (Cont'd)

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
80	BRA	Relative	A0	LDY	Immediate
81	STA	(Indirect)	A1	LDA	(Indirect)
82	Not Used		A2	LDX	Immediate
83	Not Used		A3	Not Used	
84	STY	Zero Page	A4	LDY	Zero Page
85	STA	Zero Page	A5	LDA	Zero Page
86	STX	Zero Page	A6	LDX	Zero Page
87	SMB 0	Zero Page	A7	SMB 2	Zero Page
88	DEY	Implied	A8	TAY	(Indirect)
89	ADD	Immediate	A9	LDA	Immediate
8A	TXA	Implied	AA	TAX	Implied
8B	NXT	Implied	AB	LAN	Implied
8C	STY	Absolute	AC	LDY	Absolute
8D	STA	Absolute	AD	LDA	Absolute
8E	STX	Absolute	AE	LDX	Absolute
8F	BBS 0	Zero Page	AF	BBS 2	Zero Page
90	BCC	Relative	B0	BCS	Relative
91	STA	(Indirect),X	B1	LDA	(Indirect),X
92	Not Used		B2	STI	Zero Page
93	Not Used		B3	Not Used	
94	STY	Zero Page,X	B4	LDY	Zero Page,X
95	STA	Zero Page,X	B5	LDA	Zero Page,X
96	STX	Zero Page,Y	B6	LDX	Zero Page,Y
97	SMB 1	Zero Page	B7	SMB 3	Zero Page
98	TYA	Implied	B8	CLV	Implied
99	STA	Absolute,Y	B9	LDA	Absolute,Y
9A	TXS	Implied	BA	TSX	Implied
9B	LII	Implied	BB	INI	Implied
9C	Not Used		BC	LDY	Absolute,X
9D	STA	Absolute,X	BD	LDA	Absolute,X
9E	Not Used		BE	LDX	Absolute,Y
9F	BBS 1	Zero Page	BF	BBS 3	Zero Page

Table A-3. CPU Instruction Set by Operation Code (Cont'd)

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
C0	CPY	Immediate	E0	CPX	Immediate
C1	CMP	(Indirect)	E1	SBC	(Indirect)
C2	RBA	Absolute	E2	BAR	Immediate
C3	Not Used		E3	Not Used	
C4	CPY	Zero Page	E4	CPX	Zero Page
C5	CMP	Zero Page	E5	SBC	Zero Page
C6	DEC	Zero Page	E6	INC	Zero Page
C7	SMB 4	Zero Page	E7	SMB 6	Zero Page
C8	INY	Implied	E8	INX	Implied
C9	CMP	Immediate	E9	SBC	Immediate
CA	DEX	Implied	EA	NOP	Implied
CB	PHI	Implied	EB	LAI	Implied
CC	CPY	Absolute	EC	CPX	Absolute
CD	CMP	Absolute	ED	SBC	Absolute
CE	DEC	Absolute	EE	INC	Absolute
CF	BBS 4	Zero Page	EF	BBS 6	Zero Page
D0	BNE	Relative	F0	BEQ	Relative
D1	CMP	(Indirect),X	F1	SBC	(Indirect),X
D2	SBA	Absolute	F2	BAS	Absolute
D3	Not Used		F3	Not Used	
D4	EXC	Zero Page,X	F4	Not Used	
D5	CMP	Zero Page,X	F5	SBC	Zero Page,X
D6	DEC	Zero Page,X	F6	INC	Zero Page,X
D7	SMB 5	Zero Page	F7	SMB 7	Zero Page
D8	CLD	Implied	F8	SED	Implied
D9	CMP	Absolute,Y	F9	SBC	Absolute,Y
DA	PHX	Implied	FA	PLX	Implied
DB	PLI	Implied	FB	PIA	Implied
DC	Not Used		FC	Not Used	
DD	CMP	Absolute,X	FD	SBC	Absolute,X
DE	DEC	Absolute,X	FE	INC	Absolute,X
DF	BBS 5	Zero Page	FF	BBS 7	Zero Page

Table A-4. New CPU Instructions Since R6502 CPU

Fifteen Filter Enhancement Instructions

Mnemonic	Operation	Addressing Mode	No. Bytes	No. Cycles
ASR	Shift A Right, Sign Extend	Accum	1	2
CLW	Clear W and V	Implied	1	2
EXC	Swap A, M	ZP, X	1	5
JSB	Jump to Subroutine (8)	(FFE_)	1	6
LAB	Load Absolute to Accumulator	Accum	1	2
MPA	Multiply and Accumulate	Implied	1	6
MPY	Multiply	Implied	1	6
PSH	Push A, X and Y on Stack	Implied	1	5
PUL	Pull Y, X and A from Stack	Implied	1	6
RND	Round	Implied	1	2
TAW	Transfer Accumulator to W	Implied	1	2
TWA	Transfer W to Accumulator	Implied	1	2
NEG	Negate Accumulator	Accum	1	2
PHW	Push WH, WL	Implied	1	4
PLW	Pull WL, WH	Implied	1	5

Table A-4. New CPU Instructions Since R6502 CPU (Cont'd)

Ten Direct Threaded Instructions

Mnemonic	Operation	Addressing Mode	No. Bytes	No. Cycles
NXT	(I) → PC, I + 2 → I	Implied	1	4
LII	(I) → I	Implied	1	5
LAI	(I) → A	Implied	1	3
INI	I + 1 → I	Implied	1	3
PHI	Push I	Implied	1	4
PLI	Pull I	Implied	1	6
JPI	PC + 1 → IL, PC + 2 → IH, (I) → PC, I + 2 → I	Implied	3	5
TIP	I → PC	Implied	1	2
PIA	Pull I, (I) → A, (I) → X, I +1 → I	Implied	1	6
LAN	(I) → A, I + 1 → I	Implied	1	3

Seven Controller Instructions

Mnemonic	Operation	Addressing Mode	No. Bytes	No. Cycles
BAR	Branch on Bit(s) Reset	ABS	5	7, 8, 9
BAS	Branch on Bit(s) Set	ABS	5	7, 8, 9
JMP	Jump	(ABS, X)	3	6
STI	Move IMM to Memory	ZP	3	4
RBA	Reset Bit(s) in Memory	ABS	4	7
SBA	Set Bit(s) in Memory	ABS	4	7
ADD	Add without Carry	IMM	2	2
ADD	Add without Carry	ZP	2	3
ADD	Add without Carry	ZP, X	2	4

Table A-5. CPU Instruction Enhancements Over R6502 CPU

Function	NMOS R6502 CPU	CMOS MCU CPU
Jump indirect, operand = XXFF.	Page address does not increment.	Page address increments.
Read/modify/write instructions at effective address.	One read cycle and two write cycles.	Two read cycles and one write cycle.
Decimal flag.	Indeterminate after reset.	Initialized to binary mode (D = 0) after reset.
Decimal ADD/SUB execution time.	Same execution time as binary.	One additional cycle for decimal correct.
Flags after decimal ADD/SUB.	N, V and Z flags are invalid.	N, V and Z flags are valid.
Interrupt coincident with BRK instruction.	Interrupt vector is loaded, BRK vector is ignored.	Interrupt is executed, then BRK is executed.
JSR instruction.	Stacked address points to last byte of JSR instruction.	Stacked address points to next op code. Instruction is one cycle shorter.
RTS instruction.	Return address is incremented before use.	Return address is ready for use. Instruction is one cycle shorter.
Indirect Addressing Opcodes changed.	(INDIRECT, X)	(INDIRECT)
Indirect Addressing Opcodes changed.	(INDIRECT), Y	(INDIRECT), X

Table A-6. CPU Threaded Instructions

Mnemonic	Instruction	Operation	Description
NXT	Next Instruction	(I) → PC I+2 → I	The I register points to an address. The two bytes at that address are loaded into the Program Counter. The contents of the I register are incremented by 2.
LII	Load I Indirect through I	(I) → I	The I register points to an address. The two bytes at that address are loaded into the I register.
LAI	Load A Indirect through I	(I) → A	The I register points to an address. The byte at that address is loaded into the Accumulator.
INI	Increment I by One	I+1 → I	The I register is incremented by 1.
PHI	Push I on Stack	I → (stack) SP–2 → SP	The contents of the I register are pushed onto the stack, high byte first.
PLI	Pull I from Stack	(stack) → I SP+2 → SP	The two bytes pointed to by the Stack Pointer are loaded into the I register, low byte first.
JPI (Operand)	Jump Indirect with Return in I	PC+1 → I (I) → PC I+2 →I	The contents of the Program Counter (the address of the JPI instruction) +1 are loaded into the I register. I now points to the two-byte operand of the JPI instruction. This operand is used as an indirect pointer to the next execution address. I is incremented by 2 to point to the next opcode following the JPI instruction. This instruction functions as a JSR indirect with the return address in the I register.
TIP	Transfer I to Program Counter	I →PC	Transfer the contents of the I register to the Program Counter. This instruction functions as an RTS to the JPI instruction.
PIA	Pull I from Stack, Load A	(stack) → I SP+2 → SP (I) → A I+1 → I	Load the I register with the two bytes pointed to by the Stack Pointer, low byte first. Increment the Stack Pointer by 2. Load the byte pointed to by the I register into the Accumulator. Increment the I register by 1.
LAN	Load A Indirect and Increment I	(I) → A I+1 → I	Load the byte pointed to by the I register into the Accumulator. Increment the I register by 1.

This page is intentionally blank.

B. CPU INSTRUCTION SET DESCRIPTION

This appendix describes the CPU instructions. The instructions are listed in alphabetic order. The following is provided for each instruction:

- Standard mnemonic
- Title
- Symbolic operation
- Processor status bits affected
- Operation description

Also provided, in tabular form, for each available addressing mode are:

- Addressing mode
- Assembly language form
- Operation code (Op Code)
- Number of bytes in the instruction
- Number of cycles to execute the instruction

The reference number shown at the end of the description of each instruction refers to the section in Appendix C that provides additional information about the instruction.

The following notation applies to this appendix:

A	Accumulator
ADL	Effective address low
ADH	Effective address high
X,Y	Index Registers
M	Memory
M_b	Selector Zero Page Memory Bit
M_6	Memory Bit 6
M_7	Memory Bit 7
P	Processor Status Register
S	Stack Pointer
C	Change
	No Change
+	Add
^	Logical AND
-	Subtract
∨	Logical EXCLUSIVE OR
↑	Transfer from Stack
↓	Transfer to Stack
→	Transfer to
←	Transfer to
∨	Logical OR
PC	Program Counter
PCH	Program Counter High
PCL	Program Counter Low
Oper	Operand
#	Immediate Addressing Mode
M	Mask
Dest	Destination

ADC Add Memory to Accumulator with Carry

Operation: A + M + C → A, C

N	V	–	B	D	I	Z	C
C	C	–	–	–	–	C	C

ADC adds the value in memory and the C flag to the value in the accumulator and stores the result in the accumulator.

The C flag is set when the sum exceeds hexadecimal 255 (binary mode) or decimal 99 (decimal mode), otherwise C is reset. In the binary mode, the V flag is set when the sign (bit 7) changes due to the result exceeding +127 or –128, otherwise V is reset. In the decimal mode, V is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	ADC	#Oper	69	2	2[c]	C2.1
Zero Page	ADC	Oper	65	2	3[c]	C3.1
Zero Page,X	ADC	Oper,X	75	2	4[c]	C4.1
Absolute	ADC	Oper	6D	3	4[c]	C6.1
Absolute,X	ADC	Oper,X	7D	3	4[a,c]	C7.1
Absolute,Y	ADC	Oper,Y	79	3	4[a,c]	C8.1
(Indirect)	ADC	(Oper)	61	2	5[c]	C9.1
(Indirect),X	ADC	(Oper),X	71	2	5[a,c]	C10.1

a. Add 1 if a page boundary is crossed.

c. Add 1 if in decimal mode.

ADD Add Memory to Accumulator without Carry

Operation: A + M → A, C

N	V	–	B	D	I	Z	C
C	C	–	–	–	–	C	C

ADD adds the value in memory to the value in the accumulator and stores the result in the accumulator.

The C flag is set when the sum exceeds hexadecimal 255 (binary mode) or decimal 99 (decimal mode), otherwise C is reset. In the binary mode, the V flag is set when the sign (bit 7) changes due to the result exceeding +127 or –128, otherwise V is reset. In the decimal mode, V is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	ADD	#Oper	89	2	2[c]	C2.1
Zero Page	ADD	Oper	64	2	3[c]	C3.1
Zero Page,X	ADD	Oper,X	74	2	4[c]	C4.1

c. Add 1 if in decimal mode.

AND "AND" Memory with Accumulator

Operation: $A \wedge M \rightarrow A$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

AND logically ANDs the contents of the accumulator and addressed memory bit-by-bit and stores the result in the accumulator.

The Z flag is set if the result is zero, otherwise Z is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. C and V flags are unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	AND	#Oper	29	2	2	C2.1
Zero Page	AND	Oper	25	2	3	C3.1
Zero Page,X	AND	Oper,X	35	2	4	C4.1
Absolute	AND	Oper	2D	3	4	C6.1
Absolute,X	AND	Oper,X	3D	3	4[a]	C7.1
Absolute,Y	AND	Oper,Y	39	3	4[a]	C8.1
(Indirect)	AND	(Oper)	21	2	5	C9.1
(Indirect),X	AND	(Oper),X	31	2	5[a]	C10.1

a. Add 1 if page boundary is crossed.

ASL Shift Left One Bit (Memory or Accumulator)

Operation: $C \leftarrow 7\ 6\ 5\ 4\ 3\ 2\ 1 \leftarrow 0$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	C

ASL shifts the contents of either the accumulator or the addressed memory location one bit to the left, with bit 0 always being set to 0 and the bit 7 output always being shifted to the C flag. If the accumulator is addressed, memory is unaffected; if memory is addressed, the accumulator is unaffected.

The N flag is set equal to bit 7 of the result (bit 6 of the input); the Z flag is set if the result is equal to 0, otherwise Z is reset. The V flag is unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ASL	A	0A	1	2	C11.1
Zero Page	ASL	Oper	06	2	5	C12.1
Zero Page,X	ASL	Oper,X	16	2	6	C12.2
Absolute	ASL	Oper	0E	3	6	C12.3
Absolute,X	ASL	Oper,X	1E	3	7	C12.4

ASR Accumulator Shift Right One Bit, Sign Extend

Operation: $7 \rightarrow 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \rightarrow C$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	C

ASR shifts the contents of the accumulator one bit to the right, with bit 7 being set to the input bit 7 value and the input bit 0 being shifted to the C flag.

The N flag is set equal to bit 7 of the result (the N flag will retain the same value as before since bit 7 copies the input bit 7); the Z flag is set if the result is equal to zero, otherwise Z is reset. The V flag is unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ASR	A	3A	1	2	C11.1

BAR Branch on Bit(s) Reset

Operation: Branch on ~Memory $\wedge$ Mask • 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BAR compares an addressed 8-bit location to an 8-bit mask in the instruction. If any bits at the addressed location contain a 0 in a bit position corresponding to a 1 in the mask, a conditional branch is taken to the relative address also contained in the instruction. Bytes 2 and 3 of the instruction contain the ADL and ADH, respectively, of the addressed memory or I/O port. Byte 4 contains the mask. Byte 5 specifies the conditional branch offset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	BAR	Oper,Mask,Dest	E2	5	7[b]	C15.4

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BAS Branch on Bit(s) Set

Operation: Branch on Memory ∧ Mask • 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BAS compares an addressed 8-bit location to an 8-bit mask in the instruction. If any bits at the addressed location contain a 1 in a bit position corresponding to a 1 in the mask, a conditional branch is taken to the relative address also contained in the instruction. Bytes 2 and 3 of the instruction contain the ADL and ADH, respectively, of the addressed memory or I/O port. Byte 4 contains the mask. Byte 5 specifies the conditional branch offset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	BAS	Oper,Mask,Dest	F2	5	7[b]	C15.4

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BBR Branch on Bit Reset (8)

Operation: Branch on Mb = 0 (b = 0 – 7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BBR tests one of 8 bits in a zero page memory location. If the tested bit is reset to a 0, a conditional branch is taken to the relative address also specified in the instruction. If the bit tested is set, the next sequential instruction is executed. The bit to test is specified by 3 bits of the op code in byte 1 of the instruction. Byte 2 of the instruction designates the zero page address of the byte to be tested. Byte 3 of the instruction specifies the 8-bit relative address for the conditional branch.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page				3	5[b]	C15.3
Bit 0	BBR	0,Oper,Dest	0F			
Bit 1	BBR	1,Oper,Dest	1F			
Bit 2	BBR	2,Oper,Dest	2F			
Bit 3	BBR	3,Oper,Dest	3F			
Bit 4	BBR	4,Oper,Dest	4F			
Bit 5	BBR	5,Oper,Dest	5F			
Bit 6	BBR	6,Oper,Dest	6F			
Bit 7	BBR	7,Oper,Dest	7F			

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BBS Branch on Bit Set (8)

Operation: Branch on M_b = 1 (b=0 – 7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BBS tests one of 8 bits in a zero page memory location. If the tested bit is set to a 1, a conditional branch is taken to the relative address also specified in the instruction. If the bit tested is not set, the next sequential instruction is executed. The bit to test is specified by 3 bits of the op code in byte 1 of the instruction. Byte 2 of the instruction designates the zero page address of the byte to be tested. Byte 3 of the instruction specifies the 8-bit relative address for the conditional branch.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page				3	5[b]	C15.3
Bit 0	BBS	0,Oper,Dest	8F			
Bit 1	BBS	1,Oper,Dest	9F			
Bit 2	BBS	2,Oper,Dest	AF			
Bit 3	BBS	3,Oper,Dest	BF			
Bit 4	BBS	4,Oper,Dest	CF			
Bit 5	BBS	5,Oper,Dest	DF			
Bit 6	BBS	6,Oper,Dest	EF			
Bit 7	BBS	7,Oper,Dest	FF			

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BCC Branch on Carry Clear

Operation: Branch on C = 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BCC tests the state of the C flag and takes a conditional branch if C is reset, otherwise the next sequential instruction is executed.

Addressing Mode	Assemble Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BCC	Dest	90	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BCS Branch on Carry Set

Operation: Branch on C = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BCS takes the conditional branch if the C flag is set (carry is on), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BCS	Dest	B0	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BEQ Branch on Equal

Operation: Branch on Z = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BEQ takes the conditional branch if the Z flag is set (result is zero), otherwise the next sequential instruction is executed. BEQ is also referred to as "Branch on Zero".

Addressing Mode	Assemble Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BEQ	Dest	F0	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BIT Test Bits in Memory with Accumulator

Operation: $A \wedge M, M_7 \rightarrow N, M_6 \rightarrow V$

N	V	–	B	D	I	Z	C
M_7	M_6	–	–	–	–	C	–

BIT performs a logical "AND" between the addressed byte and the accumulator, but does not alter the contents of the accumulator. The Z flag is set if the result of $A \wedge M$ is zero, otherwise Z is reset. In addition, bits 7 and 6 of addressed memory are copied to the processor status N and V flags, respectively.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	BIT	Oper	24	2	3	C3.1
Absolute	BIT	Oper	2C	3	4	C6.1

BMI Branch on Minus

Operation: Branch on N = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BMI takes the conditional branch if the N flag is set (result is minus), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BMI	Dest	30	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BNE Branch on Not Equal

Operation: Branch on Z = 0

N	V	–	B	I	D	Z	C
–	–	–	–	–	–	–	–

BNE takes the conditional branch if the Z flag is not set (result is non-zero), otherwise the next sequential instruction is executed. BNE is also referred to as "Branch on Not Zero".

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BNE	Dest	D0	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BPL Branch on Plus

Operation: Branch on N = 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BPL is the complementary conditional branch to BMI. BPL tests the N flag and takes the conditional branch if N is reset (result is positive), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BPL	Dest	10	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BRA Branch Always

Operation: PC + IMM → PC

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BRA is an unconditional, or forced, branch. The immediate field is added to the program counter value to form the new program counter value. This allows a short (2-byte) branch to an address +127 or –128 bytes from the current PC value (i.e., the address of next sequential instruction).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	BRA Dest	80	2	3[a]	C13.2

a. Add 1 if branch occurs to a different page.

BRK Break Command

Operation: Forced Interrupt Request: PC + 1 ↓, P ↓

N	V	–	B	D	I	Z	C
–	–	–	1	–	1	–	–

The BRK command causes the processor to go through an interrupt request sequence under program control. The address in the program counter (which points to the location of the **BRK command +1**) is pushed on the stack, along with the processor status at the beginning of the BRK instruction. The processor then transfers control to the **NMI interrupt vector (FFFC, FFFD)**. Note that the BRK command cannot be masked by setting the I flag.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	BRK	00	1	7	C14.9

BVC Branch on Overflow Clear

Operation: Branch on V = 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BVC tests the status of the V flag and takes the conditional branch if V is not set (result did not overflow). otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	BVC Dest	50	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BVS Branch on Overflow Set

Operation: Branch on V = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BVS tests the V flag and takes the conditional branch if V is set (result did overflow), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	BVS Dest	70	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

CLC Clear Carry Flag

Operation: $0 \rightarrow C$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	0

CLC initializes the C flag to a 0. This operation should normally precede an ADC loop. It is also useful when used with a ROL instruction to clear a bit in memory.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLC	18	1	2	C1.1

CLD Clear Decimal Mode

Operation: $0 \rightarrow D$

N	V	–	B	D	I	Z	C
–	–	–	–	0	–	–	–

CLD resets the D flag in the processor flag register to a 0 (binary mode). This causes all subsequent ADC, ADD and SBC instructions to operate as binary operations.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLD	D8	1	2	C1.1

CLI Clear Interrupt Disable Bit

Operation: $0 \rightarrow I$

N	V	–	B	D	I	Z	C
–	–	–	–	–	0	–	–

CLI resets the interrupt disable flag in the processor status register to a 0 (interrupt enabled). This enables the processor to act on an interrupt request (IRQ).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLI	58	1	2	C1.1

CLV Clear Overflow Flag

Operation: $0 \rightarrow V$

N	V	–	B	D	I	Z	C
–	0	–	–	–	–	–	–

This instruction clears the V flag to a 0.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	CLV	B8	1	2	C1.1

CLW Clear W Register and Overflow Flag

Operation: $0 \rightarrow W, V$

N	V	–	B	D	I	Z	C
–	0	–	–	–	–	–	–

CLW clears both halves of the W register and resets the V flag to a 0.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLW	52	1	2	C1.1

CMP Compare Memory and Accumulator

Operation: A – M

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	C

CMP subtracts the contents of addressed memory from the contents of the accumulator and set/resets the N, Z and C flags accordingly.

The Z flag is set on an equal comparison, otherwise Z is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The C flag is set when the value in memory is less than or equal to the accumulator, reset when it is greater than the accumulator. The accumulator is not affected.

Result	N Flag	C Flag	Z Flag
Accumulator < Memory	Either	Reset	Reset
Accumulator = Memory	Reset	Set	Set
Accumulator > Memory	Either	Set	Reset

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	CMP	#Oper	C9	2	2	C2.1
Zero Page	CMP	Oper	C5	2	3	C3.1
Zero Page,X	CMP	Oper,X	D5	2	4	C4.1
Absolute	CMP	Oper	CD	3	4	C6.1
Absolute,X	CMP	Oper,X	DD	3	4[a]	C7.1
Absolute,Y	CMP	Oper,Y	D9	3	4[a]	C8.1
(Indirect)	CMP	(Oper)	C1	2	5	C9.1
(Indirect),X	CMP	(Oper),X	D1	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

CPX Compare Memory and Index X

Operation: X – M

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	C

CPX subtracts the contents of the addressed memory location from the contents of index register X and sets/resets the N, Z, and C flags accordingly.

The C flag is set if the value in index register X is equal to, or greater than, the value in memory, otherwise C is cleared. The N flag is set if the result of the subtraction is negative (bit 7 is a 1), otherwise N is cleared. The Z flag is set if the value in index register X and the value in memory are equal, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	CPX	#Oper	E0	2	2	C2.1
Zero Page	CPX	Oper	E4	2	3	C3.1
Absolute	CPX	Oper	EC	3	4	C6.1

CPY Compare Memory and Index Y

Operation: Y – M

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	C

CPY subtracts the contents of the addressed memory location from the contents of index register Y and sets/resets the N, Z, and C flags accordingly.

The C flag is set if the value in index register Y is equal to, or greater than, the value in memory, otherwise C is cleared. The N flag is set if the result of the subtraction is negative (bit 7 is a 1), otherwise N is cleared. The Z flag is set if the value in index register Y and the value in memory are equal, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	CPY	#Oper	C0	2	2	C2.1
Zero Page	CPY	Oper	C4	2	3	C3.1
Absolute	CPY	Oper	CC	3	4	C6.1

DEC Decrement Memory by One

Operation: M – 1 → M

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

DEC subtracts 1 from the contents of the addressed memory location and stores the result in the addressed memory byte. A value of 0 will be decremented to $FF.

The N flag is set if bit 7 is on as a result of the decrement, otherwise N is reset. The Z flag is set if the result of the decrement is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	DEC	Oper	C6	2	5	C12.1
Zero Page,X	DEC	Oper,X	D6	2	6	C12.2
Absolute	DEC	Oper	CE	3	6	C12.3
Absolute,X	DEC	Oper,X	DE	3	7	C12.4

DEX Decrement Index X by One

Operation: $X - 1 \rightarrow X$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

DEX subtracts one from the value in the X register and stores the result in the X register. The result does not affect or consider carry so that a value of 0 in index register X will be decremented to $FF.

The N flag is set if the X register contains bit 7 on as a result of the decrement, otherwise N is reset. The Z flag is set if the X register is zero as a result of the decrement, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	DEX	CA	1	2	C1.1

DEY Decrement Index Y by One

Operation: $Y - 1 \rightarrow Y$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

DEY subtracts one from the current value in the Y register and stores the result in the Y register. The result does not affect or consider carry so that a value of 0 in index register Y will be decremented to $FF.

The N flag is set if the Y register contains bit 7 on as a result of the decrement, otherwise N is reset. The Z flag is set if the Y register is zero as a result of the decrement, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	DEY	88	1	2	C1.1

EOR "Exclusive-Or" Memory with Accumulator

Operation: $A \vee M \rightarrow A$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

EOR performs a binary "EXCLUSIVE OR" on a bit-by-bit basis between the accumulator and the addressed location and stores the result in the accumulator.

The Z flag is set if the result in the accumulator is 0, otherwise Z is reset. The N flag is set if the result in the accumulator has bit 7 on, otherwise N is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	EOR	#Oper	49	2	2	C2.1
Zero Page	EOR	Oper	45	2	3	C3.1
Zero Page,X	EOR	Oper,X	55	2	4	C4.1
Absolute	EOR	Oper	4D	3	4	C6.1
Absolute,X	EOR	Oper,X	5D	3	4[a]	C7.1
Absolute,Y	EOR	Oper,Y	59	3	4[a]	C8.1
(Indirect)	EOR	(Oper)	41	2	5	C9.1
(Indirect),X	EOR	(Oper),X	51	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

EXC Exchange Accumulator and Memory

Operation: $A \rightarrow M, M \rightarrow A$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

EXC exchanges the contents in the accumulator with the contents of addressed memory byte.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page,X	EXC	Oper,X	D4	2	5	C4.3

INC Increment Memory by One

Operation: $M + 1 \rightarrow M$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

INC adds 1 to the contents of the addressed memory location. A value of $FF is incremented to 0.

The N flag is set if bit 7 is on as the result of the increment, otherwise N is reset. If the increment causes a zero result, the Z flag is set, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	INC	Oper	E6	2	5	C12.1
Zero Page,X	INC	Oper,X	F6	2	6	C12.2
Absolute	INC	Oper	EE	3	6	C12.3
Absolute,X	INC	Oper,X	FE	3	7	C12.4

INI Increment I by One

Operation: $I + 1 \rightarrow I$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

INI increments the contents of the I register by one.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	INI	BB	1	3	C17.1

INX Increment Index X by One

Operation: $X + 1 \rightarrow X$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

INX adds 1 to the value in the X register, storing the result in the X register. A value of $FF is incremented to 0.

The N flag is set if the result of the increment has a one in bit 7, otherwise N is reset. The Z flag is set if the result of the increment is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	INX	E8	1	2	C1.1

INY Increment Index Y by One

Operation: Y + 1 → Y

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

INY adds 1 to the value in the Y register, storing the result in the Y register. A value of $FF is incremented to 0.

The N flag is set if the result has a 1 in bit 7, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	INY	C8	1	2	C1.1

JMP Jump to New Location

Operation: See below.

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

The JMP instruction establishes a new value for the program counter. It affects only the program counter in the processor and affects no flags in the status register.

In the JMP Absolute instruction, the second and third bytes of the instruction are loaded into the PCL and PCH, respectively, to specify the location of the next instruction. The symbolic operation is (PC + 1) → PCL, (PC + 2) → PCH.

In the JMP (Indirect) instruction, the second and third bytes of the instruction represent the low and high bytes, respectively, of the memory location containing the effective ADL. Once the ADL is fetched, the program counter is incremented with the next memory location containing the ADH.

The JMP (Indirect,X) instruction operates like the JMP (Indirect) instruction except the contents of the X register are added to the indirect low byte of the effective memory location containing the effective ADL (a carry, generated if a page boundary is crossed, is added to the indirect high byte).

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Absolute	JMP	Oper	4C	3	3	C13.3
(Indirect)	JMP	(Oper)	6C	3	5	C13.4
(Indirect,X)	JMP	(Oper,X)	7C	3	6	C13.5

JPI Jump Indirect with Return in I

Operation: PC + 1 → IL, PC + 2 → IH, (I) ⟶ PC, I + 2 → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

JPI executes an indirect jump to subroutine, saving the return address, PC+3, in the I register. Bytes 2 and 3 of the instruction are the indirect jump address IAL, IAH. The effective jump address bytes (ADL, ADH) are fetched from memory at locations [IAH, IAL] and [IAH, IAL] + 1, respectively.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
(Indirect)	JPI	0C	3	5	C17.6

JSB Jump to Subroutine (8)

Operation: PC + 1 ↓ , (FFEn) → PCL, (FFEn + 1) → PCH, SP – 2 ⟶ SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

JSB is a 1-byte instruction that loads the program counter with a subroutine starting address and leaves a return pointer on the stack to allow the program to return to the next instruction in the main program upon subroutine completion. The JSB instruction stores the program counter address + 1, which points to the instruction following the JSB, onto the stack. The stack byte contains the program count high first, followed by program count low. Three bits of the op code specify one of eight 2-byte vectors in ROM located from $FFE0–$FFEE. The specified vector points to the subroutine starting address. The starting address is transferred to the program counter to direct program execution to continue at that address

JSB affects no processor status flags, causes the stack pointer to be decremented by 2 and substitutes new values into program counter low and program counter high.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/FF Page				1	6	C14.8
(FFE0)	JSB	0	0B			
(FFE2)	JSB	1	1B			
(FFE4)	JSB	2	2B			
(FFE6)	JSB	3	3B			
(FFE8)	JSB	4	4B			
(FFEA)	JSB	5	5B			
(FFEC)	JSB	6	6B			
(FFEE)	JSB	7	7B			

JSR Jump to New Location Saving Return Address

Operation: PC + 3↓, (PC + 1)→ PCL, (PC + 2) → PCH, SP – 2→ SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

JSR loads the program counter with the a subroutine starting address and leaves a return pointer on the stack to allow the program to return to the next instruction in the main program upon subroutine completion. To accomplish this, the JSR instruction stores the program counter address + 3, which points to the instruction following the JSR, onto the stack. The stack byte contains the program count high first, followed by program count low. The JSR then transfers the address following the jump instruction to the program counter, thereby directing the program to begin at that new address.

The JSR instruction affects no flags, causes the stack pointer to be decremented by 2 and substitutes new values into the program counter low and the program counter high.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Absolute	JSR Oper	20	3	5	C14.8

LAB Load Absolute to Accumulator

Operation: $|A| \rightarrow A$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

LAB loads the absolute value of the contents in the accumulator to the accumulator. If the original contents of the accumulator are negative, LAB performs a twos complement. Exception: An accumulator value of $80 will produce the result $80.

The Z flag is set if the result is zero, otherwise Z is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	LAB A	13	1	3	C11.2

LAI Load Accumulator Indirect through I

Operation: $(I) \rightarrow A$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

LAI loads the byte pointed to by the I register into the accumulator.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	LAI	EB	1	3	C17.2

LAN Load A Indirect and Increment I

Operation: $(I) \rightarrow A, I + 1 \rightarrow I$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

LAN loads the byte pointed to by the I register into the accumulator. The I register is incremented by 1.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	LAN	AB	1	3	C17.3

LDA Load Accumulator with Memory

Operation: M → A

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

LDA copies the contents of addressed memory to the accumulator.

The Z flag is set if the accumulator is zero as a result of the LDA, otherwise Z is reset. The N flag is set if bit 7 of the accumulator is a 1, otherwise N is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	LDA	#Oper	A9	2	2	C2.1
Zero Page	LDA	Oper	A5	2	3	C3.1
Zero Page,X	LDA	Oper,X	B5	2	4	C4.1
Absolute	LDA	Oper	AD	3	4	C6.1
Absolute,X	LDA	Oper,X	BD	3	4[a]	C7.1
Absolute,Y	LDA	Oper,Y	B9	3	4[a]	C8.1
(Indirect)	LDA	(Oper)	A1	2	5	C9.1
(Indirect),X	LDA	(Oper),X	B1	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

LDX Load Index X with Memory

Operation: M → X

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

LDX copies the contents of addressed memory to index register X.

The N flag is set if bit 7 of the loaded value is a 1, otherwise N is reset. The Z flag is set if the value loaded is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	LDX	#Oper	A2	2	2	C2.1
Zero Page	LDX	Oper	A6	2	3	C3.1
Zero Page,Y	LDX	Oper,Y	B6	2	4	C5.1
Absolute	LDX	Oper	AE	3	4	C6.1
Absolute,Y	LDX	Oper,Y	BE	3	4[a]	C8.1

a. Add 1 if a page boundary is crossed.

LDY Load Index Y with Memory

Operation: M → Y

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

LDY copies the contents of memory to index register Y.

The N flag is set if bit 7 of the loaded value is a 1, otherwise N is reset. The Z flag is set if the loaded value is zero, otherwise Z is reset. The other processor status flags and processor registers are unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	LDY	#Oper	A0	2	2	C2.1
Zero Page	LDY	Oper	A4	2	3	C3.1
Zero Page,X	LDY	Oper,X	B4	2	4	C4.1
Absolute	LDY	Oper	AC	3	4	C6.1
Absolute,X	LDY	Oper,X	BC	3	4[a]	C7.1

a. Add 1 if a page boundary is crossed.

LII Load I Indirect through I

Operation: (I) → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

LII loads the two bytes at the memory address pointed to by the I register into the I register.

No processor status flags or other processor registers are affected.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	LII	9B	1	5	C17.5

LSR Logical Shift Right One Bit (Memory or Accumulator)

Operation: $0 \rightarrow 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \rightarrow C$

N	V	–	B	D	I	Z	C
0	–	–	–	–	–	C	C

LSR shifts the contents of the accumulator, or of a specified memory location, one bit to the right, with the high bit of the result always being set to 0, and the shifted low bit being stored in the C flag.

The N flag is always reset to a 0. The Z flag is set if the result of the shift is 0, otherwise Z is reset. The C flag is set equal to bit 0 of the input.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	LSR	A	4A	1	2	C11.1
Zero Page	LSR	Oper	46	2	5	C12.1
Zero Page,X	LSR	Oper,X	56	2	6	C12.2
Absolute	LSR	Oper	4E	3	6	C12.3
Absolute,X	LSR	Oper,X	5E	3	7	C12.4

MPA Multiply and Accumulate

Operation: $A \times Y + W \rightarrow W$

N	V	–	B	D	I	Z	C
C	C	–	–	–	–	–	–

MPA performs a signed multiply of the value in the accumulator by the value in the Y register, adds the result of the multiplication to the value in the W register, then stores the accumulated result in the W register. The accumulator contents are not altered, however, the contents of the Y register are destroyed.

The N flag is set if the result of the accumulation in the W register is negative (e.g., bit 15 is a 1); otherwise N is reset. The V flag is set if bit 15 of the accumulated result changes due to the value exceeding +32767 ($7FFF) or –32768 ($8000), otherwise V is reset. In case of overflow, the most positive value ($7FFF) or most negative value ($8000) is left in the W register depending on overflow polarity.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	MPA	12	1	6	C1.2

MPY Multiply

Operation: A x Y → A, Y

N	V	–	B	D	I	Z	C
C	0	–	–	–	–	–	–

MPY performs a signed multiply of the value in the accumulator by the value in the Y register. The upper half of the result is stored in the accumulator and the lower half of the result is stored in Y register. The original contents of the A and Y registers are destroyed.

The V flag is reset. The N flag copies the sign of the result, accumulator bit 7.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	MPY	02	1	6	C1.2

NEG Negate Accumulator

Operation: ~A → A

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

NEG performs a twos complement of the contents in the accumulator and stores the result in the accumulator. The value $80 will produce the result $80.

The N flag is set if bit 7 of the result in the accumulator is a 1, otherwise N is reset. The Z flag is set if the result in the accumulator is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	NEG A	1A	1	2	C11.1

NOP No Operation

Operation: PC + 1 → PC

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

NOP performs no operation except incrementing the program counter by 1.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	NOP	EA	1	2	C1.1

NXT Next Instruction

Operation: (I) → PC, I + 2 → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

NXT loads the two bytes at the address pointed to by the I register into the program counter and increments the I register by 2. The low order address byte is loaded first.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	NXT	8B	1	4	C17.4

ORA "OR" Memory with Accumulator

Operation: A∨ M → A

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

The ORA instruction performs a binary "OR" on a bit-by-bit basis between the accumulator and addressed memory and stores the result in the accumulator.

The Z flag is set if the result in the accumulator is 0, otherwise Z is reset. The N flag is set if the result in the accumulator has bit 7 on, otherwise N is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	ORA	#Oper	09	2	2	C2.1
Zero Page	ORA	Oper	05	2	3	C3.1
Zero Page,X	ORA	Oper,X	15	2	4	C4.1
Absolute	ORA	Oper	0D	3	4	C6.1
Absolute,X	ORA	Oper,X	1D	3	4[a]	C7.1
Absolute,Y	ORA	Oper,Y	19	3	5[a]	C8.1
(Indirect)	ORA	(Oper)	01	2	6	C9.1
Indirect),X	ORA	(Oper),X	11	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

PHA Push Accumulator on Stack

Operation: A ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHA copies the current value of the accumulator to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHA	48	1	3	C14.1

PHI Push I on Stack

Operation: IH ↓, SP–1→ SP, IL ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHI copies the contents of the I register to the stack, high byte first, and decrements the stack pointer by 2 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHI	CB	1	4	C14.2

PHP Push Processor Status on Stack

Operation: P ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHP copies the contents of the processor status register, unchanged, to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHP	08	1	3	C14.1

PHW Push W on Stack

Operation: WH ↓, SP–1 → SP, WL ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHW copies the contents of the W register to the stack, high byte first, and decrements the stack pointer by 2 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHW	23	1	4	C14.2

PHX Push Index X on Stack

Operation: X ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHX copies the contents of index register X to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHX	DA	1	3	C14.1

PHY Push Index Y on Stack

Operation: Y ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHY copies the contents of index register Y to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHY	5A	1	3	C14.1

PIA Pull I from Stack, Load Accumulator, Increment I

Operation: SP + 1 → SP, IL ↑, SP + 1 → SP, IH ↑,
(I) → A, (I) → X, I +1 → I

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

PIA first loads the I register with two bytes from the stack, low byte first. The stack pointer is incremented by 1 before each byte is copied. Next, the byte pointed to by the I register is then loaded into the accumulator and to the X register, and then the I register is incremented by 1.

The N flag is set if bit 7 of the accumulator is 1 as a result of the instruction, otherwise N is reset. The Z flag is set if the accumulator value is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PIA	FB	1	6	C14.6

PLA Pull Accumulator from Stack

Operation: SP + 1 → SP, A ↑

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

PLA increments the stack pointer by 1 then loads the accumulator with the byte from the stack.

The N flag is set if bit 7 of the loaded value is a 1, otherwise N is reset. If the accumulator is zero as a result of the PLA, the Z flag is set, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLA	68	1	4	C14.4

PLI Pull I from Stack

Operation: SP + 1 → SP, IL ↑, SP + 1 → SP, IH ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PLI loads the I register with two bytes from the stack, low byte first. The stack pointer is incremented by 1 before each byte is copied.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLI	DB	1	6	C14.6

PLP Pull Processor Status from Stack

Operation: SP + 1 → SP, P ↑

N	V	–	B	D	I	Z	C
C	C	–	–	C	C	C	C

PLP increments the stack pointer by 1 then copies the byte in the stack to the processor status register.

Six bits in the processor status register are replaced.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLP	28	1	4	C14.4

PLW Pull W from Stack

Operation: SP + 1 → SP, WL ↑, SP + 1 → SP, WH ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PLW loads the W register with two bytes from the stack, low byte first. The stack pointer is incremented by 1 before each byte is transferred.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLW	33	1	5	C14.5

PLX Pull Index X from Stack

Operation: SP + 1 → SP, X ↑

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

PLX adds 1 to the stack pointer then copies the byte in the stack to the X register.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLX	FA	1	4	C14.4

PLY Pull Index Y from Stack

Operation: SP + 1 → SP, Y ↑

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

PLY adds 1 to the stack pointer then copies the contents of the byte in the stack pointed to by the stack pointer into the Y register.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLY	7A	1	4	C14.4

PSH Push A, X and Y on Stack

Operation: A ↓, SP–1 → SP, X ↓, SP–1 → SP, Y ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PSH copies the contents of the accumulator, X register and Y register to the stack in this order. The stack pointer is decremented by 1 after each byte is transferred.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PSH	22	1	5	C14.3

PUL Pull Y, X and A from Stack

Operation: SP + 1 → SP, Y ↑, SP + 1 → SP, X ↑, SP + 1 → SP, A ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PUL pulls the contents of three bytes from the stack to the Y register, X register and accumulator in this order. The stack pointer is incremented by one before each byte is transferred.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PUL	32	1	6	C14.7

RBA Reset Bit(s) in Memory

Operation: ~Mask $\wedge$ M $\rightarrow$ M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

RBA resets to a "0", in the addressed memory location, all bits set to a 1 in a mask. Byte 2 of the instruction contains the mask and bytes 3 and 4 of the instruction specify the absolute address of the operand.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	RBA	Mask,Addr	C2	4	7	C15.2

RMB Reset Memory Bit (8)

Operation: 0 $\rightarrow$ M_b (b = 0 –7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

RMB resets to "0" one of eight bits in an addressed memory or I/O port location. The first byte of the instruction specifies the RMB operation code and 1 of 8 bits (bit 0 to bit 7) to be reset. The second byte of the instruction designates the zero page address of the byte to be operated upon.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page				2	5	C15.1
Bit 0	RMB	0,Oper	07			
Bit 1	RMB	1,Oper	17			
Bit 2	RMB	2,Oper	27			
Bit 3	RMB	3,Oper	37			
Bit 4	RMB	4,Oper	47			
Bit 5	RMB	5,Oper	57			
Bit 6	RMB	6,Oper	67			
Bit 7	RMB	7,Oper	77			

RND Round

Operation: WH + WL7 → A

N	V	–	B	D	I	Z	C
C	C	–	–	–	–	–	–

The most significant bit of WL is added to WH and the result is stored in the accumulator. The maximum positive value stored in the accumulator is 127 ($7F) while the maximum negative value stored is –128 ($80). W remains unchanged.

The N flag is set if bit 7 in the accumulator is a 1, otherwise N is reset. If a RND is attempted when WH = $7F and WL7 = 1, then $7F is loaded in the accumulator, the V flag is set, and the contents of the Y register are destroyed; otherwise V is reset and Y remains unchanged.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	RND	42	1	2	C1.1

ROL Rotate Left One Bit (Memory or Accumulator)

Operation: C ← 7 6 5 4 3 2 1 0 ← C

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	C

ROL shifts either the accumulator or addressed memory left one bit, with the input C flag being stored in bit 0 and with the input bit 7 being stored in the C flag.

The C flag is set equal to the input bit 7. The N flag is set equal to the result bit 7. The Z flag is set if the result in memory or the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ROL A 2A	1	2	C11.1	
Zero Page	ROL Oper	26	2	5	C12.1
Zero Page,X	ROL Oper,X	36	2	6	C12.2
Absolute	ROL Oper	2E	3	6	C12.3
Absolute,X	ROL Oper,X	3E	3	7	C12.4

ROR Rotate Right One Bit (Memory or Accumulator)

Operation: C → 7 6 5 4 3 2 1 0 → C

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	C

ROR shifts either the accumulator or addressed memory right one bit, with the input C flag shifted into bit 7 and the input bit 0 shifted into the C flag.

The C flag is set to input bit 0 and the N flag is set equal to the input C flag. The Z flag is set if the result in memory or the accumulator is 0; otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ROR	A	6A	1	2	C11.1
Zero Page	ROR	Oper	66	2	5	C12.1
Zero Page,X	ROR	Oper,X	76	2	6	C12.2
Absolute	ROR	Oper	6E	3	6	C12.3
Absolute,X	ROR	Oper,X	7E	3	7	C12.4

RTI Return from Interrupt

Operation: SP + 1 → SP, P ↑, SP + 1 → SP, PCL ↑,
SP + 1 → SP, PCH ↑

N	V	–	B	D	I	Z	C
C	C	–	–	C	C	C	C

RTI transfers from the stack the processor status and the program counter location for the instruction which was interrupted. By virtue of the interrupt having stored this data before executing the instruction, and due to the fact that the RTI reinitializes the processor to the same state as when it was interrupted, the combination of interrupt plus RTI allows truly reentrant coding.

The RTI instruction reinitializes 6 flags to the state which they were when the interrupt was taken, and sets the program counter back to its pre-interrupt state. RTI affects no other registers in the processor.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	RTI	40	1	6	C14.11

RTS Return from Subroutine

Operation: SP + 1 → SP, PCL ↑, SP + 1 → SP, PCH ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

RTS loads the program counter with two bytes from the stack, low byte first. The stack pointer is incremented by 2.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	RTS	60	1	5	C14.10

SBA Set Bit(s) in Memory

Operation: Mask ∨ M → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

SBA sets to a "1" in the addressed memory location all bits set to a 1 in a mask. Byte 2 of the instruction contains the mask and bytes 3 and 4 of the instruction specify the absolute address of the operand. SBA logically ORs the mask with the operand value to form the new operand value.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	SBA	Mask,Addr	D2	4	7	C15.2

SBC Subtract Memory from Accumulator with Borrow

Operation: A – M – ~C → A (~C = Borrow)

N	V	–	B	D	I	Z	C
C	C	–	–	–	–	C	C

SBC subtracts the value in the addressed memory location and borrow (i.e., the complement of the C flag) from the value in the accumulator and stores the result in the accumulator.

The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset. The C flag is set if the result is greater than, or equal to, zero; otherwise C is reset. In the binary mode, the V flag is set if the result exceeds +127 or –128, otherwise V is reset. In the decimal mode, V is reset.

No other processor status flags or processor registers are affected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	SBC	#Oper	E9	2	2[c]	C2.1
Zero Page	SBC	Oper	E5	2	3[c]	C3.1
Zero Page,X	SBC	Oper,X	F5	2	4[c]	C4.1
Absolute	SBC	Oper	ED	3	4[c]	C6.1
Absolute,X	SBC	Oper,X	FD	3	4[a,c]	C7.1
Absolute,Y	SBC	Oper,Y	F9	3	4[a,c]	C8.1
(Indirect)	SBC	(Oper)	E1	2	5[c]	C9.1
(Indirect),X	SBC	(Oper),X	F1	3	5[a,c]	C10.1

a. Add 1 when page boundary is crossed.

c. Add 1 if in decimal mode.

SEC Set Carry Flag

Operation: 1 → C

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	1

SEC initializes the C flag to a 1. This operation should normally precede a SBC loop.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	SEC	38	1	2	C1.1

SED Set Decimal Mode

Operation: $1 \rightarrow D$

N	V	–	B	D	I	Z	C
–	–	–	–	1	–	–	–

SED sets the D flag to a 1 (decimal mode). This makes all subsequent ADC, ADD, and SBC instructions operate as a decimal arithmetic operation (until a CLD resets the D flag to a 0).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	SED	F8	1	2	C1.1

SEI Set Interrupt Disable Status

Operation: $1 \rightarrow I$

N	V	–	B	D	I	Z	C
–	–	–	–	–	1	–	–

SEI sets the I flag to a 1 (disable interrupt).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	SEI	78	1	2	C1.1

SMB Set Memory Bit (8)

Operation: $1 \rightarrow M_b$ (b = 0 - 7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

SMB sets to "1" one of the bits in an 8-bit data field specified by the zero page address (memory or I/O port). The first byte of the instruction specifies the SMB operation and 1 of 8 bits to be set. The second byte of the instruction designates the zero page address of the byte or I/O port to be operated upon.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page			2	5	C15.1
Bit 0	SMB 0,Oper	87			
Bit 1	SMB 1,Oper	97			
Bit 2	SMB 2,Oper	A7			
Bit 3	SMB 3,Oper	B7			
Bit 4	SMB 4,Oper	C7			
Bit 5	SMB 5,Oper	D7			
Bit 6	SMB 6,Oper	E7			
Bit 7	SMB 7,Oper	F7			

STA Store Accumulator in Memory

Operation: A → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STA copies the contents of the accumulator to addressed memory.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STA	Oper	85	2	3	C3.2
Zero Page,X	STA	Oper,X	95	2	4	C4.2
Absolute	STA	Oper	8D	3	4	C6.2
Absolute,X	STA	Oper,X	9D	3	5	C7.2
Absolute,Y	STA	Oper,Y	99	3	5	C8.2
(Indirect)	STA	(Oper)	81	2	5	C9.2
(Indirect),X	STA	(Oper),X	91	2	6	C10.2

STI Store Immediate to Memory

Operation: Oper → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STI stores the contents of the immediate field (byte 2 of the instruction) to the zero page address specified by byte 3 of the instruction.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STI	#Oper1, Oper2	B2	3	4	C2.2

STX Store Index X in Memory

Operation: X → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STX copies the contents of the X register to the addressed memory location.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STX	Oper	86	2	3	C3.2
Zero Page,Y	STX	Oper,Y	96	2	4	C5.2
Absolute	STX	Oper	8E	3	4	C6.2

STY Store Index Y in Memory

Operation: $Y \rightarrow M$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STY copies the contents of the Y register to the addressed memory location.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STY	Oper	84	2	3	C3.2
Zero Page,X	STY	Oper,X	94	2	4	C4.2
Absolute	STY	Oper	8C	3	4	C6.2

TAW Transfer Accumulator to W

Operation: $A \rightarrow WH, 0 \rightarrow WL$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

TAW copies the contents of the accumulator to the upper half of the W register and stores zero in the lower half of the W register.

The N flag copies bit 7 of the WH register. The Z flag is set if the WH register is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TAW	62	1	2	C1.1

TAX Transfer Accumulator to Index X

Operation: $A \rightarrow X$

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

TAX copies the contents of accumulator to the index register X without disturbing the contents of the accumulator.

The N flag copies bit 7 of the X register. The Z flag is set if the X register is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TAX	AA	1	2	C1.1

TAY Transfer Accumulator to Index Y

Operation: A → Y

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

TAY copies the contents of the accumulator to index register Y without altering the contents of the accumulator.

The N flag copies bit 7 of the Y register. The Z flag is set if the Y register is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TAY	A8	1	2	C1.1

TIP Transfer I to Program Counter

Operation: I → PC

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

TIP copies the contents of the I register to the program counter. This instruction functions as a return from subroutine (RTS) to the JPI instruction.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TIP	03	1	2	C1.1

TSX Transfer Stack Pointer to Index X

Operation: S → X

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

TSX transfers the value in the stack pointer to the index register X.

The N flag copies bit 7 of the X register. The Z flag is set if the X register is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TSX	BA	1	2	C1.1

TWA Transfer W to Accumulator

Operation: WH → A

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

TWA copies the contents of the upper half of the W register to the accumulator.

The N flag copies bit 7 of the accumulator. The Z flag is set if the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TWA	72	1	2	C1.1

TXA Transfer Index X to Accumulator

Operation: X → A

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

TXA copies the value in the index register X to the accumulator without disturbing the contents of the index register X.

The N flag copies bit 7 of the accumulator. The Z flag is set if the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TXA	8A	1	2	C1.1

TXS Transfer Index X to Stack Pointer

Operation: X → S

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

TXS copies the value in the index register X to the stack pointer. TXS changes only the stack pointer, making it equal to the contents of index register X.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TXS	9A	1	2	C1.1

TYA Transfer Index Y to Accumulator

Operation: Y → A

N	V	–	B	D	I	Z	C
C	–	–	–	–	–	C	–

TYA copies the value in index register Y to the accumulator without disturbing the contents of the Y register.

The N flag copies bit 7 of the accumulator. The Z flag is set if the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TYA	98	1	2	C1.1

C. CPU INSTRUCTION ADDRESSING MODES

This appendix describes the addressing modes for the CPU instructions. The following attributes are provided for each addressing mode:

One number of bytes and execution time in clock cycles.

The instructions using the addressing mode.

The format of the instruction.

The instruction operation by clock cycle.

The instruction formats are:

Addressing Mode	No. Bytes	Op Code	Operand	Operand Description
Absolute	3	XXX	nn	nn = 2-byte absolute address
Absolute,X	3	XXX	nn	nn = 2-byte absolute address
Absolute,Y	3	XXX	nn	nn = 2-byte absolute address
Accumulator	1	XXX		
Bit/FF Page (JSB)	1	XXX	b	b = bit 0 –7
Bit/Zero Page (BBR/BBS)	3	XXX	b,n,r	b = bit 0–7, n = 1-byte zero page address, r = 1-byte relative offset
Bit/Zero Page (RMB/SMB)	2	XXX	b,n	b = bit 0–7, n = 1-byte zero page address
Bit/Absolute (RBA/SBA)	4	XXX	bm,nn	bm = byte mask, nn = 2-byte absolute address
Bit/Absolute (BAR/BAS)	5	XXX	nn,bm,rb	bm = byte mask, nn = 2-byte absolute address, r = 1-byte relative offset
Immediate	2	XXX	#n	n = 1-byte constant
Implied	1	XXX		
Indirect Absolute	3	XXX	(nn)	nn = 2-byte absolute address
(Indirect)	2	XXX	(n)	n = 1-byte zero page address of addrLH
(Indirect),X	2	XXX	(n),X	n = 1-byte zero page address of addrLH
Relative	2	XXX	r	r = 1-byte relative offset
Zero Page	2	XXX	n	n = 1-byte zero page address
Zero Page ,X	2	XXX	n,X	n = 1-byte zero page address
Zero Page ,Y	2	XXX	n,Y	n = 1-byte zero page address

C1 IMPLIED ADDRESSING

C1.1 General – 1 Byte, 2 Cycles

Instructions: CLC, CLD, CLI, CLW, CLV, DEX, DEY, INX, INY, NOP, RND, SEC, SED, SEI, TAW, TAX, TAY, TIP, TSX, TWA, TXA, TXS, and TYA

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE
3	2201	Next OP CODE	Fetch Next OP CODE	Execute instruction, Increment PC to 2202

C1.2 Multiply – 1 Byte, 6 Cycles

Instruction: MPA and MPY

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE
3	2201	Next OP CODE	Fetch Data (Ignored)	Execute multiply steps
4	2201	Next OP CODE	Fetch Data (Ignored)	
5	2201	Next OP CODE	Fetch Data (Ignored)	
6	2201	Next OP CODE	Fetch Data (Ignored)	
7	2201	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2202

C2 IMMEDIATE ADDRESSING

C2.1 Read – 2 Bytes, 2 Cycles

Instructions: ADC, ADD, AND, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, and SBC

Format: XXX #n n= 1-byte constant

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Fetch Data	Data	Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Execute instruction, Increment PC to 2203

C2.2 Write – 2 Bytes, 4 Cycles

Instruction: STI

Format: XXX #n n= 1-byte constant

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Fetch Data	Data	Increment PC to 2202
3	2202	Fetch ADL	ADL	Increment PC to 2203
4	00, ADL	Data	Write Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C3 ZERO PAGE ADDRESSING

C3.1 Read – 2 Bytes, 3 Cycles

Instructions: ADC, ADD, AND, BIT, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, and SBC

Format: XXX n n = zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C3.2 Write – 2 Bytes, 3 Cycles

Instructions: STA, STX, and STY.

Format: XXX n n = zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Write Data	Hold PC
4	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C4 ZERO PAGE,X ADDRESSING

C4.1 Read – 2 Bytes, 4 Cycles

Instructions: ADC, ADD, AND, CMP, EOR, LDA, LDY, ORA, and SBC

Format: XXX n,X n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+X, Hold PC
4	00, BAL+X	Data	Fetch Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C4.2 Write – 2 Bytes, 4 Cycles

Instructions: STA and STY

Format: XXX n,X n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+X, Hold PC
4	00, BAL+X	Data	Write Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C4.3 Write – 2 Bytes, 5 Cycles

Instructions: EXC

Format: XXX n,X n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+X, Hold PC
4	00, BAL+X	Data	Fetch Data	Hold PC
5	00, BAL+X	Data	Write Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C5 ZERO PAGE,Y ADDRESSING

C5.1 Read – 2 Bytes, 4 Cycles

Instruction: LDX

Format: XXX n,Y n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+Y, Hold PC
4	00, BAL+Y	Data	Fetch Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C5.2 Write – 2 Bytes, 4 Cycles

Instruction: STX

Format: XXX n,Y n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+Y, Hold PC
4	00, BAL+Y	Data	Write Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish Instruction, Increment PC to 2203

C6 ABSOLUTE ADDRESSING

C6.1 Read – 3 Bytes, 4 Cycles

Instructions: ADC, AND, BIT, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, and SBC

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Fetch Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C6.2 Write – 3 Bytes, 4 Cycles

Instructions: STA, STX, and STY

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Write Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish Instruction, Increment PC to 2203

C7 ABSOLUTE,X ADDRESSING

C7.1 Read

Instructions: ADC, AND, CMP, EOR, LDA, LDY, ORA, and SBC

Format: XXX nn,X nn = 2-byte absolute address

C7.1a No Page Crossing – 3 Bytes, 4 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAH+X, Increment PC to 2203
4	BAH, BAL+X	Data	Fetch Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C7.1b Page Crossing – 3 Bytes, 5 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+X, Increment PC to 2203
4	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (1), Hold PC
5	BAH+C, BAL+X	Data	Fetch Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction Increment PC to 2204

C7.2 Write – 3 Bytes, 5 Cycles

Instruction: STA

Format: XXX nn,X nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+X, Increment PC to 2203
4	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1), Hold PC
5	BAH+C, BAL+X	Data	Write Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C8 ABSOLUTE,Y ADDRESSING

C8.1 Read

Instructions: ADC, AND, CMP, EOR, LDA, LDX, ORA, and SBC

Format: XXX nn,Y nn = 2-byte absolute address

C8.1a No Page Crossing – 3 Bytes, 4 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202,
3	2202	BAH	Fetch BAH	Add BAL+Y, Increment PC to 2203
4	BAH, BAL+Y	Data	Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C8.1b Page Crossing – 3 Bytes, 5 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+Y, Increment PC to 2203
4	BAH, BAL+Y	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (1), Hold PC
5	BAH+C, BAL+Y	Data	Fetch Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C8.2 Write – 3 Bytes, 5 Cycles

Instruction: STA

Format: XXX nn,Y nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+Y, Increment PC to 2203
4	BAH, BAL+Y	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1), Hold PC
5	BAH+C, BAL+Y	Data	Write Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C9 INDIRECT ADDRESSING

C9.1 Read - 2 Bytes, 5 Cycles

Instructions: ADC, AND, CMP, EOR, LDA, ORA, and SBC

Format: XXX (n) n = Zero page address of absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Hold BAL, Hold PC
5	BAH, BAL	Data	Fetch Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C9.2 Write – 2 Bytes, 5 Cycles

Instruction: STA

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Load OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Hold BAL, Hold PC
5	BAH, BAL	Data	Write Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C10 INDIRECT,X ADDRESSING

C10.1 Read

Instructions: ADC, AND, CMP, EOR, LDA, ORA, and SBC

Format: XXX (n),X n = Zero page address of absolute address

C10.1a No Page Crossing – 2 Bytes, 5 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Add BAL+X, Hold PC
5	BAH, BAL+X	Data	Fetch Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C10.1b Page Crossing – 2 Bytes, 6 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Load OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Add BAL to X, Hold PC
5	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (1), Hold PC
6	BAH+C, BAL+X	Data	Fetch Data	Hold PC
7	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C10.2 Write – 2 Bytes, 6 Cycles

Instruction: STA

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Load OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Add BAL to X, Hold PC
5	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1) Hold PC
6	BAH+C, BAL+X	Data	Write Data	Hold PC
7	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C11 ACCUMULATOR ADDRESSING

C11.1 1 Byte, 2 Cycles

Instructions: ASL, ASR, LSR, ROL, ROR, and NEG

Format: XXX

Operation: ROL example

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch OP CODE (Ignored)	Interpret OP CODE (ROL), Hold PC
3	2201	Next OP CODE	Fetch Next OP CODE	Shift through the Adder, Increment PC to 2202
4	2202	?	Fetch Second Byte	Store result into A, Interpret next OP CODE

C11.2 1 Byte, 3 Cycles

Instruction: LAB

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Decode OP CODE, Hold PC
3	2201	Next OP CODE	Fetch Data (Ignored)	
4	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2202
5	2202	?	Fetch Second Byte	Store result into A, Interpret next OP CODE

C12 READ/MODIFY/WRITE

C12.1 Zero Page Addressing – 2 Bytes, 5 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Decode OP CODE (ROL), Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	00, ADL	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
5	00, ADL	Shifted Data	Write Data	Set flags, Hold PC
6	2202	OP CODE	Fetch Next OP CODE	Increment PC to 2203

C12.2 Zero Page,X Addressing – 2 Bytes, 6 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Decode OP CODE (ROL), Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAH+X, Hold PC
4	00, BAL+X	Data	Fetch Data	Hold PC
5	00, BAL+X	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
6	00, BAL+X	Shifted Data	Write Data	Set flags, Hold PC
7	2202	OP CODE	Fetch Next OP CODE	Increment PC to 2203

C12.3 Absolute Addressing – 3 Bytes, 6 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Decode OP CODE (ROL), Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Fetch Data	Hold PC
5	ADH, ADL	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
6	ADH, ADL	Shifted Data	Write Data	Set flags, Hold PC
7	2203	OP CODE	Fetch Next OP CODE	Increment PC to 2204

C12.4 Absolute,X Addressing – 3 Bytes, 7 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX nn,X nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Decode OP CODE (ROL), Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+X, Increment PC to 2203
4	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1), Hold PC
5	BAH+C, BAL+X	Data	Fetch Data	Hold PC
6	BAH+C, BAL+X	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
7	BAH+C, BAL+X	Shifted Data	Write Data	Set Flags, Hold PC
8	2203	OP CODE	Fetch Next OP CODE	Increment PC to 2204

C13 BRANCH/JUMP

C13.1 Conditional Branch

Instructions: BCC, BCS, BEQ, BMI, BNE, BPL, BVC, and BVS

Format: XXX r r = 1-byte relative offset

C13.1a Branch Not Taken – 2 Bytes, 2 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset	Fetch Offset	Interpret OP CODE, Set previous instruction flags, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Check flags, Increment PC to 2203

C13.1b Branch Taken with Positive Offset, No Page Boundary Crossing – 2 Bytes, 3 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset (+50)	Fetch Offset	Interpret OP CODE, Set previous instruction flags, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Check flags, Add Offset to PCL, Increment PC to 2203
4	2252	Next OP CODE	Fetch Next OP CODE	Transfer results to PCL, Increment PC to 2253

C13.1c Branch Taken with Negative Offset, Page Boundary Crossed – 2 Bytes, 4 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset (–50)	Fetch Offset	Interpret OP CODE, Set previous instruction flags, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Check flags, Add offset to PCL
4	22B2	Data	Fetch Data (Ignored)	Store Adder result in PCL and subtract 1 from PCH
5	21B2	Next OP CODE	Fetch Next OP CODE	Put out new PCH, Increment PC to 21B3

C13.2 Unconditional Branch Crossing – 2 Bytes, 3 Cycles with Positive Offset, No Page Boundary

Instruction: BRA

Format: XXX r r = 1-byte relative offset

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset (+40)	Fetch Offset	Interpret OP CODE, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Add offset to PC, Increment PC to 2203
4	2242	Next OP CODE	Fetch Next OP CODE	Increment PC to 2243

C13.3 Jump Absolute – 3 Bytes, 3 Cycles

Instruction: JMP

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	
4	ADH, ADL	Next OP CODE	Fetch Next OP CODE	ADH, ADL to PC, Increment PC

C13.4 Jump (Indirect,X) – 3 bytes, 6 cycles

Instruction: JMP

Format: XXX (nn,X) nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	2202	IAH	Fetch IAH	Hold IAL
4	IAH, IAL+X	Data	Fetch Data (Ignored)	Add IAH + carry
5	IAH+C, IAL+X	ADL	Fetch ADL	Add 1 to [IAH, IAL]
6	[IAH+C, IAL+X]+1	ADH	Fetch ADH	
7	ADH, ADL	Next OP CODE	Fetch Next OP CODE	ADH, ADL to PC, Increment PC

C13.5 JMP (Indirect) – 3 Bytes, 5 Cycles

Instruction: JMP

Format: XXX (nn) nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	2202	IAH	Fetch IAH	Hold IAL
4	IAH, IAL	ADL	Fetch ADL	Add 1 to [IAH, IAL]
5	[IAH, IAL]+1	ADH	Fetch ADH	
6	ADH, ADL	Next OP CODE	Fetch Next OP CODE	ADH, ADL to PC,

C14 STACK

C14.1 Push 1 Byte – 1 byte, 3 Cycles

Instructions: PHA, PHP, PHX, and PHY

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (PHA)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE (Ignored)	Interpret OP CODE (PHA), Hold PC
3	01FF	(A)	Write A to Stack	Decrement SP to 01FE
4	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C14.2 Push 2 Bytes – 1 Byte, 4 Cycles

Instructions: PHI and PHW

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (PHI)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE (Ignored)	Interpret OP CODE (PHI), Hold PC
3	01FF	(IH)	Write IH to Stack	Decrement SP to 01FE
4	01FE	(IL)	Write IL to Stack	Decrement SP to 01FD
5	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C14.3 Push 3 Bytes – 1 Byte, 5 Cycles

Instruction: PSH

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (PSH)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE (Ignored)	Interpret OP CODE (PSH), Hold PC
3	01FF	(A)	Write A to Stack	Decrement SP to 01FE
4	01FE	(X)	Write X to Stack	Decrement SP to 01FD
5	01FD	(Y)	Write Y to Stack	Decrement SP to 01FC
6	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2202

C14.4 Pull 1 Byte – 1 Byte, 4 Cycles

Instructions: PLA, PLP, PLX, and PLY

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE and Discard	Interpret OP CODE, Hold PC
3	01FE	Discarded Data	Read Stack	Increment SP to 01FF
4	01FF	Data	Fetch Data	Save Stack
5	2201	Next OP CODE	Fetch Next OP CODE	Data to A

C14.5 Pull 2 Bytes – 1 Byte, 5 Cycles

Instruction: PLW

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE and Discard	Interpret OP CODE, Hold PC
3	01FD	Data	Fetch Data (Ignored)	Increment SP to 01FE
4	01FE	WL	Fetch Data from Stack	Increment SP to 01FF
5	01FF	WH	Fetch Data from Stack	Data to WL, Save Stack
6	2201	Next OP CODE	Fetch Next OP CODE	Data to WH

C14.6 Pull 2 Bytes – 1 Byte, 6 Cycles

Instructions: PLI and PIA

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish Previous Instruction
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	01FD	Data	Fetch Data (Ignored)	Increment SP to 01FE
4	01FE	IL	Fetch Data	Increment SP to 01FF
5	01FF	IH	Fetch Data	Data to IL
6	IH, IL	Data	Fetch Data	Data to IH, Save Stack, Increment I if PIA
7	2201	Next OP CODE	Fetch New OP CODE	Data to A if PIA

C14.7 Pull 3 Bytes – 1 Byte, 6 Cycles

Instructions: PUL

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE (PUL)	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE and Discard	Interpret OP CODE (PUL), Hold PC
3	01FC	Data	Fetch Data (Ignored)	Increment SP to 01FD
4	01FD	Y	Fetch Data from Stack	Increment SP to 01FE
5	01FE	X	Fetch Data from Stack	Data to Y, Increment SP to 01FF
6	01FF	A	Fetch Data from Stack	Data to X
7	2201	Next OP CODE	Fetch Next OP CODE	Data to A

C14.8 Jump to Subroutine – 3 Bytes, 5 Cycles

Instruction: JSR

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE(JSR)	Finish previous instruction, Increment PC to 2201
2	2201	New ADL	Fetch ADL	Decode OP CODE (JSR), Increment PC to 2202
3	2202	New ADH	Fetch ADH	Hold ADL
4	01FF	PCH	Write PCH to Stack	Hold ADL and ADH, Decrement SP to 01FE
5	01FE	PCL	Write PCL to Stack	Hold ADL and ADH, Decrement SP to 01FD
6	ADH, ADL	New OP CODE	Fetch New OP CODE	ADL to PCL, ADH to PCH

C14.9 Break – 1 Byte, 7 Cycles

Instruction: BRK

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Force BRK instruction, Increment PC to 2202
3	01FF	PCH	Store PCH on Stack	Decrement SP to 01FE
4	01FE	PCL	Store PCL on Stack	Decrement SP to 01FD
5	01FD	P	Store P on Stack	Decrement SP to 01FC
6	FFF0	New PCL	Fetch Vector Low	Put Away Stack
7	FFF1	New PCH	Fetch Vector High	Vector Low
8	PCH, PCL	OP CODE	Fetch Interrupt Program	Increment PC to PC + 1

C14.10 Return from Subroutine – 1 Byte, 5 Cycles

Instruction: RTS

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	3300	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 3301
2	3301	Data	Fetch Data (Ignored)	Decode OP CODE (RTS)
3	3301	Data	Fetch Data (Ignored)	Increment SP to 01FE
4	01FE	PCL (02)	Fetch PCL	Increment SP to 01FF, Hold PCL
5	01FF	PCH (22)	Fetch PCH	
6	2202	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C14.11 Return From Interrupt – 1 Byte, 6 Cycles

Instruction: RTI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	3300	OP CODE (RTI)	Fetch OP CODE	Finish previous instruction, Increment PC to 3301
2	3301	Data	Fetch Data (Ignored)	Decode OP CODE (RTI)
3	3301	Data	Fetch Data (Ignored)	Increment SP to 01FD
4	01FD	P	Fetch P from Stock	Increment SP to 01FE
5	01FE	PCL (02)	Fetch PCL from Stack	Increment SP to 01FF, Hold PCL
6	01FF	PCH (22)	Fetch PCH from Stack	M to PCL,
Store SP				
7	2202	OP CODE	Fetch OP CODE	Increment New PC

C15 BIT

C15.1 Reset/Set Memory Bit – 2 Bytes, 5 Cycles

Instructions: RMB0 – RMB7, and SMB0 – SMB7

Format: XXX n n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	00, ADL	Data	Fetch Data (Ignored)	Reset (RMB)/set (SMB) bit, Hold PC
5	00, ADL	Modified Data	Write Modified Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C15.2 Reset/Set Bits in Memory – 4 Bytes, 7 Cycles

Instructions: RBA and SBA

Format: XXX bm,nn bm = 1-byte bit mask, nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BM	Fetch BM	Interpret OP CODE, Increment PC to 2202
3	2202	ADL	Fetch ADL	Hold BM in Accumulator* Increment PC to 2203
4	2203	ADH	Fetch ADH	Hold ADL, Increment PC to 2204
5	ADH, ADL	Data	Fetch Data	Hold PC
6	ADH, ADL	Data	Fetch Data (Ignored)	Reset(RBA)/Set (SBA) bit, Hold PC
7	ADH, ADL	Modified Data	Write Modified Data	Hold PC
8	2204	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2205

* Note that Accumulator contents are altered by RBA and SBA.

C15.3 Branch on Bit Reset/Set, Branch Not Taken – 3 Bytes, 5 Cycles

Instructions: BBR0 – BBR7, and BBS0 – BBS7

Format: XXX n,r n = 1-byte zero page address, r = offset

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	00, ADL	Data	Fetch Data (Ignored)	Test the bit, Hold PC
5	2202	Offset	Fetch Branch Offset	Increment PC to 2203
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C15.4 Branch on Bit(s) Reset/Set, Branch Not Taken – 5 Bytes, 7 Cycles

Instructions: BAR and BAS

Format: XXX nn, bm, r nn = 2-byte absolute address, bm = 1-byte bit mask,
r = offset

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Fetch Data	Hold PC
5	2203	BM	Fetch BM	Hold ADL, ADH Hold PC
6	2203	BM	Fetch BM (Ignored)	Test bits, Increment PC to 2204
7	2204	Offset	Fetch Branch Offset	Increment PC to 2205
8	2205	Next OP CODE	Fetch Next OP CODE	Finish Instruction, Increment PC to 2206

C16 JSB – 1 Byte, 6 Cycles

Instructions: JSB0 – JSB7

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (JSB0)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE (JSB0), Hold PC
3	01FF	PCH	Write PCH to Stack	Decrement SP to 01FE
4	01FE	PCL	Write PCL to Stack	Decrement SP to 01FD
5	FFE0	New PCL	Fetch PCL	Save SP
6	FFE1	New PCH	Fetch PCH	Hold PCL
7	PCH, PCL	First OP CODE	Fetch First OP CODE	Increment PC to PC+1

C17 THREADED CODE

C17.1 INI – 1 Byte, 3 Cycles

Instruction: INI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	IH, IL	Data	Fetch Data (Ignored)	Increment I
4	2201	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2202

C17.2 LAI – 1 Byte, 3 Cycles

Instruction: LAI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	IH, IL	Data	Fetch Data	
4	2201	Next OP CODE	Fetch Next OP CODE	Data to A, Increment PC to 2202

C17.3 LAN – 1 Byte, 3 Cycles

Instruction: LAN

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	IH, IL	Data	Fetch Data	Increment I
4	2201	Next OP CODE	Fetch Next OP CODE	Finish instruction, Data to A Increment PC to 2202

C17.4 NXT – 1 Byte, 4 Cycles

Instruction: NXT

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE
3	IH, IL	New PCL	Fetch New PCL	
4	[IH, IL] + 1	New PCH	Fetch New PCH	Hold PCL
5	PCH, PCL	Next OP CODE	Fetch New OP CODE	Finish instruction, Increment PC to PC + 1

C17.5 LII – 1 Byte, 5 Cycles

Instruction: LII

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Ignore	Interpret OP CODE, Hold PC
3	IH, IL	New IL	Fetch IL	
4	[IH, IL] + 1	New IH	Fetch IH	Hold IL
5	New IH, IL	Data	Fetch Data (Ignored)	New IH, IL $\rightarrow$ I
6	2201	Next OP CODE	Fetch New OP CODE	Increment PC to 2202

C17.6 JPI – 3 Bytes, 5 Cycles

Instruction: JPI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	2202	IAH	Fetch IAH	Save IAL,
4	IAH, IAL	ADL	Fetch ADL	[IAH, IAL] + 1
5	[IAH, IAL] + 1	ADH	Fetch ADH	Save ADL
6	ADH, ADL	Next OP CODE	Fetch New OP CODE	Finish instruction, Increment PC to [ADH, ADL] + 1

C18 START AND INTERRUPT SEQUENCES

C18.1 Start Sequence (Power-On)

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	?	?	Fetch Data (Ignored)	Hold During Reset
2	? + 1	?	Fetch Data (Ignored)	First Start State
3	01, SP	?	Fetch Data (Ignored)	Second Start State
4	01, SP–1	?	Fetch Data (Ignored)	Third Start State
5	01, SP–2	?	Fetch Data (Ignored)	Fourth Start State
6	FFFE	PCL	Fetch PCL	
7	FFFF	PCH	Fetch PCH	Hold PCL
8	PCH, PCL	First OP CODE	Fetch First OP CODE	Increment PC to [PCH, PCL] + 1

C18.2 Interrupt Sequence (NMI) *

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Hold PC
2	2200	OP CODE	Fetch OP CODE	Force a BRK Instruction, Hold PC
3	01FF	PCH	Store PCH on Stack	Decrement SP to 01FE
4	01FE	PCL	Store PCL on Stack	Decrement SP to 01FD
5	01FD	P	Store P on Stack	Decrement SP to 01FC
6	FFFC	PCL	Fetch PCL	
7	FFFD	PCH	Fetch PCH	Hold PCL
8	PCH, PCL	Next OP CODE	Fetch Interrupt Program First OP CODE	Increment PC to PC + 1

* All other interrupts are identical except for different vector locations at cycles 6 and 7.

This page is intentionally blank.

NOTES

(Inside Back Cover)

Further Information
literature@conexant.com
1-800-854-8099 (North America)
33-14-906-3980 (International)

Web Site
www.conexant.com

World Headquarters
Conexant Systems, Inc.
4311 Jamboree Road
P. O. Box C
Newport Beach, CA
92658-8902
Phone: (949) 483-4600
Fax: (949) 483-6375

U.S. Florida/South America
Phone: (727) 799-8406
Fax: (727) 799-8306

U.S. Los Angeles
Phone: (805) 376-0559
Fax: (805) 376-8180

U.S. Mid-Atlantic
Phone: (215) 244-6784
Fax: (215) 244-9292

U.S. North Central
Phone: (630) 773-3454
Fax: (630) 773-3907

U.S. Northeast
Phone: (978) 692-7660
Fax: (978) 692-8185

U.S. Northwest/Pacific West
Phone: (408) 249-9696
Fax: (408) 249-7113

U.S. South Central
Phone: (972) 733-0723
Fax: (972) 407-0639

U.S. Southeast
Phone: (919) 858-9110
Fax: (919) 858-8669

U.S. Southwest
Phone: (949) 483-9119
Fax: (949) 483-9090

APAC Headquarters
Conexant Systems Singapore, Pte. Ltd.
1 Kim Seng Promenade
Great World City
#09-01 East Tower
SINGAPORE 237994
Phone: (65) 737 7355
Fax: (65) 737 9077

Australia
Phone: (61 2) 9869 4088
Fax: (61 2) 9869 4077

China
Phone: (86 2) 6361 2515
Fax: (86 2) 6361 2516

Hong Kong
Phone: (852) 2827 0181
Fax: (852) 2827 6488

India
Phone: (91 11) 692 4780
Fax: (91 11) 692 4712

Korea
Phone: (82 2) 565 2880
Fax: (82 2) 565 1440

Phone: (82 53) 745 2880
Fax: (82 53) 745 1440

Europe Headquarters
Conexant Systems France
Les Taissounieres B1
1681 Route des Dolines
BP 283
06905 Sophia Antipolis Cedex
FRANCE
Phone: (33 4) 93 00 33 35
Fax: (33 4) 93 00 33 03

Europe Central
Phone: (49 89) 829 1320
Fax: (49 89) 834 2734

Europe Mediterranean
Phone: (39 02) 9317 9911
Fax: (39 02) 9317 9913

Europe North
Phone: (44 1344) 486 444
Fax: (44 1344) 486 555

Europe South
Phone: (33 1) 41 44 36 50
Fax: (33 1) 41 44 36 90

Middle East Headquarters
Conexant Systems
Commercial (Israel) Ltd.
P. O. Box 12660
Herzlia 46733, ISRAEL
Phone: (972 9) 952 4064
Fax: (972 9) 951 3924

Japan Headquarters
Conexant Systems Japan Co., Ltd.
Shimomoto Building
1-46-3 Hatsudai,
Shibuya-ku, Tokyo
151-0061 JAPAN
Phone: (81 3) 5371-1567
Fax: (81 3) 5371-1501

Taiwan Headquarters
Conexant Systems, Taiwan Co., Ltd.
Room 2808
International Trade Building
333 Keelung Road, Section 1
Taipei 110, TAIWAN, ROC
Phone: (886 2) 2720 0282
Fax: (886 2) 2757 6760

SO990326(V1.2)