ADVANCE INFORMATION

# 82091AA
# ADVANCED INTEGRATED PERIPHERAL (AIP)

- **Single-Chip PC Compatible I/O Solution for Notebook and Desktop Platforms:**
  - **82078 Floppy Disk Controller Core**
  - **Two 16550 Compatible UARTs**
  - **One Multi-Function Parallel Port**
  - **IDE Interface**
  - **Integrated Back Power Protection**
  - **Integrated Game Port Chip Select**
  - **5V or 3.3V Supply Operation with 5V Tolerant Drive Interface**
  - **Full Power Management Support**
  - **Supports Type F DMA Transfers for Faster I/O Performance**
  - **No Wait-State Host I/O Interface**
  - **Programmable Interrupt Interfaces**
  - **Single Crystal/Oscillator Clock (24 MHz)**
  - **Software Detectable Device ID**
  - **Comprehensive Powerup Configuration**
- **The 82091AA is 100 Percent Compatible with EISA, ISA and AT**
- **Host Interface Features**
  - **8-Bit Zero Wait-State ISA Bus Interface**
  - **DMA with Type F Transfers**
  - **Five Programmable ISA Interrupt Lines**
  - **Internal Address Decoder**
- **Parallel Port Features**
  - **All IEEE Standard 1284 Protocols Supported (Compatibility, Nibble, Byte, EPP, and ECP)**
  - **Peak Bi-Directional Transfer Rate of 2 MB/sec**
  - **Provides Interface for Low-Cost Engineless Laser Printer**
  - **16-Byte FIFO for ECP**
  - **Interface Backpower Protection**
- **Floppy Disk Controller Features**
  - **100 Percent Software Compatible with Industry Standard 82077SL and 82078**
  - **Integrated Analog Data Separator 250K, 300K, 500K, and 1 MBits/sec**
  - **Programmable Powerdown Command**
  - **Auto Powerdown and Wakeup Modes**
  - **Integrated Tape Drive Support**
  - **Perpendicular Recording Support for 4 MB Drives**
  - **Programmable Write Pre-Compensation Delays**
  - **256 Track Direct Address, Unlimited Track Support**
  - **16-Byte FIFO**
  - **Supports 2 or 4 Drives**
- **16550 Compatible UART Features**
  - **Two Independent Serial Ports**
  - **Software Compatible with 8250 and 16450 UARTs**
  - **16-Byte FIFO per Serial Port**
  - **Two UART Clock Sources, Supports MIDI Baud Rate**
- **IDE Interface Features**
  - **Generates Chip Selects for IDE Drives**
  - **Integrated Buffer Control Logic**
  - **Dual IDE Interface Support**
- **Power Management Features**
  - **Transparent to Operating Systems and Applications Programs**
  - **Independent Power Control for Each Integrated Device**
- **100-Pin QFP Package**
  (See Packaging Spec. 240800)

---

The 82091AA Advanced Integrated Peripheral (AIP) is an integrated I/O solution containing a floppy disk controller, 2 serial ports, a multi-function parallel port, an IDE interface, and a game port on a single chip. The integration of these I/O devices results in a minimization of form factor, cost and power consumption. The


**December 1995** **Order Number: 290486-003**

floppy disk controller is the 82078 core. The serial ports are 16550 compatible. The parallel port supports all of the IEEE Standard 1284 protocols (ECP, EPP, Byte, Compatibility, and Nibble). The IDE interface supports 8- or 16-bit programmed I/O and 16-bit DMA. The Host Interface is an 8-bit ISA interface optimized for type ''F'' DMA and no wait-state I/O accesses. Improved throughput and performance, the 82091AA contains six 16-byte FIFOs–two for each serial port, one for the parallel port, and one for the floppy disk controller. The 82091AA also includes power management and 3.3V capability for power sensitive applications such as notebooks. The 82091AA supports both motherboard and add-in card configurations.

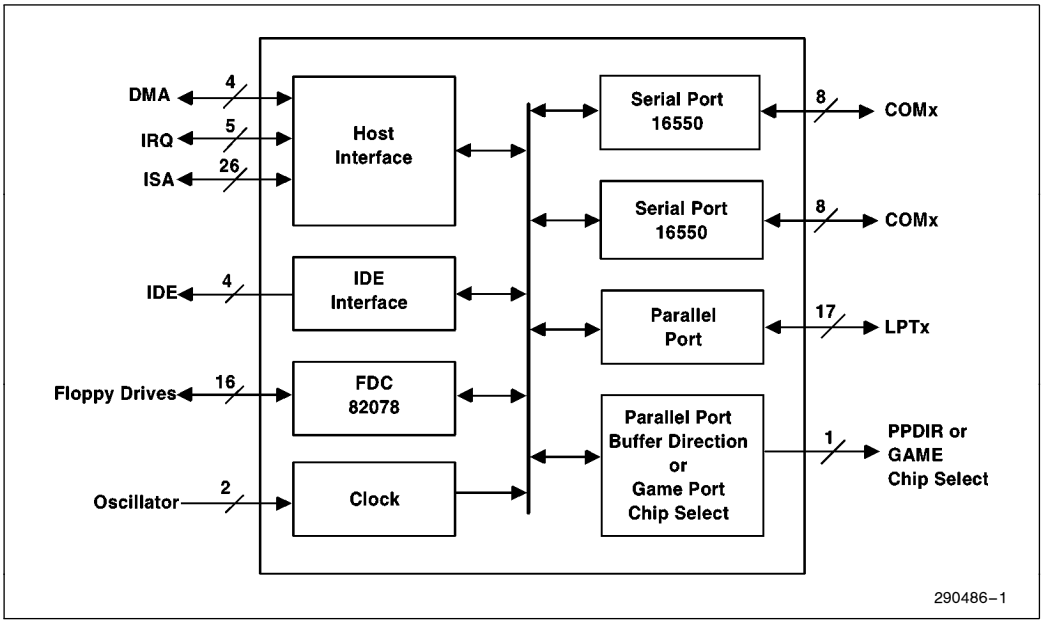



**Figure 1. 82091AA Advanced Integrated Peripheral Block Diagram**

# 82091AA
# ADVANCED INTEGRATED PERIPHERAL (AIP)

## CONTENTS

PAGE

## CONTENTS

## CONTENTS

## CONTENTS

PAGE

## CONTENTS

PAGE

## 1.0. OVERVIEW

The major functions of the 82091AA are shown in Figure 1. A brief description of each of these functions is presented in this section.

**Host Interface**

The 82091AA host interface is an 8-bit direct-drive (24 mA) ISA Bus/X-Bus interface that permits the CPU to access its registers through read/write operations in I/O space. These registers may be accessed by programmed I/O and/or DMA bus cycles. With the exception of the IDE Interface, all functions on the 82091AA require only 8-bit data accesses. The 16-bit access required for the IDE Interface is supported through the appropriate chip selects and data buffer enables from the 82091AA.

Figure 2 shows an example system implementation with the 82091AA located on an ISA Bus add-in card. This add-in card could also be used in a PCI-based system as shown in Figure 3. For motherboard implementations, the 82091AA can be located on the X-Bus as shown in Figure 4.

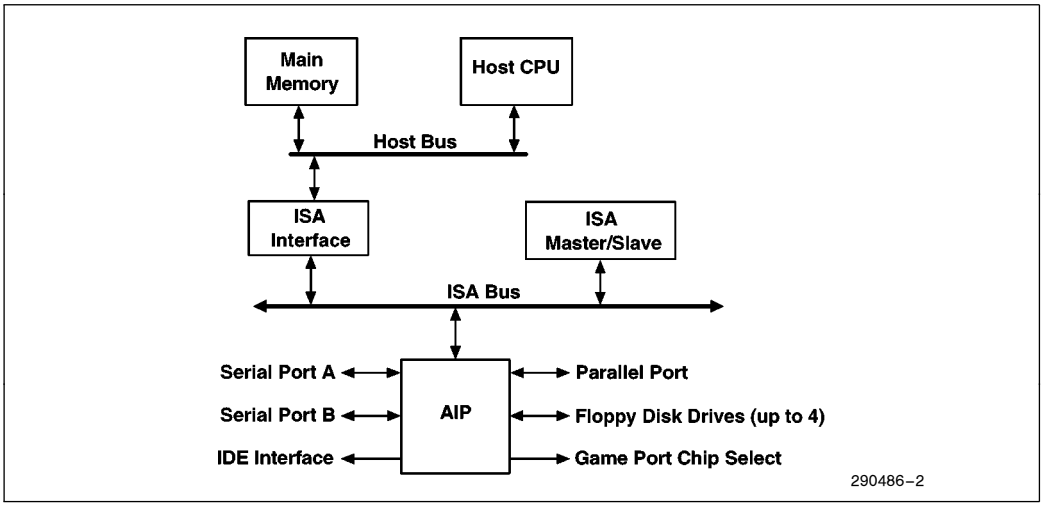



**Figure 2. Block Diagram of the 82091AA on the ISA Bus**

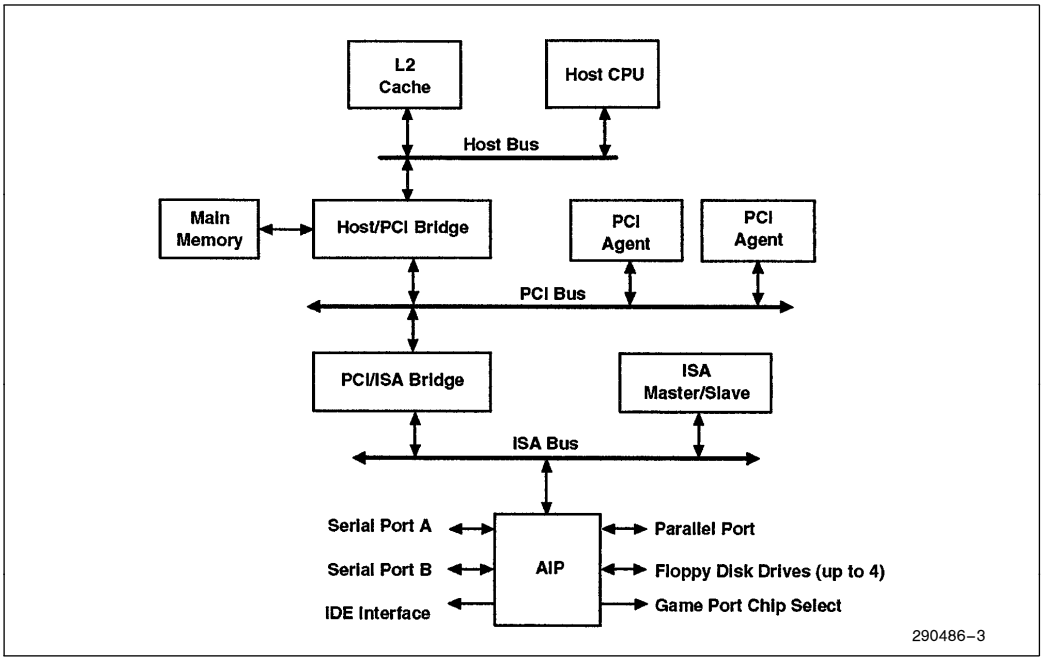



**Figure 3. Block Diagram of the 82091AA in a PCI System**

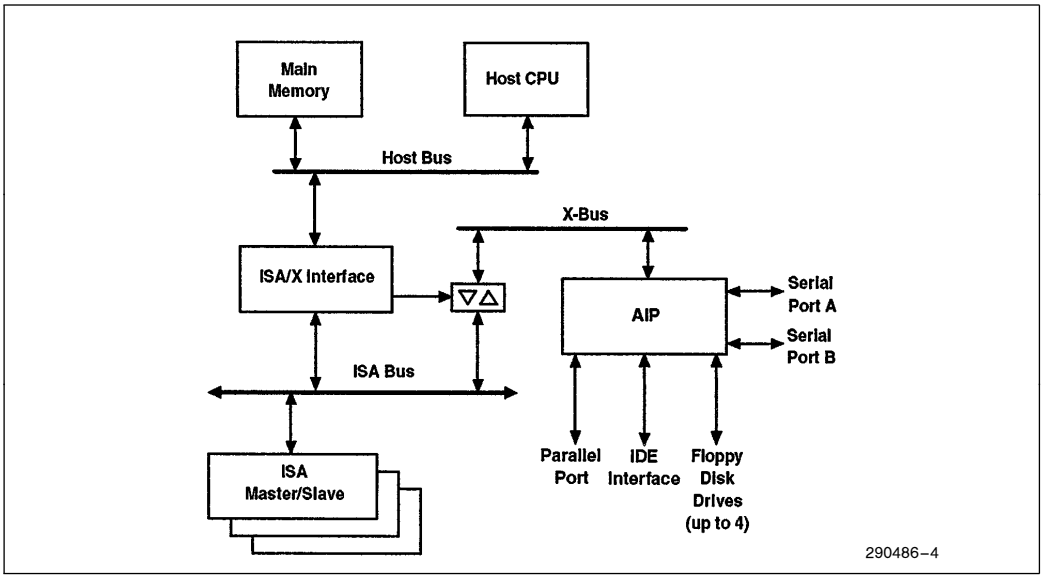



**Figure 4. Block Diagram of the 82091AA on the X-Bus**

### Floppy Disk Controller

The 82091AA's enhanced floppy disk controller (FDC) incorporates several new features allowing for easy implementation in both the portable and desktop markets. It provides a low cost, small form factor solution targeted for 5.0V and 3.3V platforms. The FDC supports up to four drives.

The 82091AA's FDC implements these new features while remaining functionally compatible with 82078/82077SL/82077AA/8272A floppy disk controllers. Together, with a 24-MHz crystal, a resistor package and a device chip select, these devices allow for the most integrated solution available. The integrated analog PLL data separator has better performance than most board level discrete PLL implementations and can be operated at 1 Mbps/500 Kbps/300 Kbps/250 Kbps. A 16-byte FIFO substantially improves system performance and is ideal for multimaster systems (e.g., EISA).

### Serial Ports

The 82091AA contains two independent serial ports that provide asynchronous communications that are equivalent to two 16550 UARTs. The serial ports have identical circuitry and provide the serial communication interface to a peripheral device or modem via Serial Port A and Serial Port B. Each serial port can be configured for one of eight address assignments. The standard PC/AT compatible logical address assignments for COM1, COM2, COM3, and COM4 are supported.

The serial ports perform serial-to-parallel conversion on data characters received from a peripheral device or modem, and parallel-to-serial conversion on data characters received from the host. The serial ports can operate in either FIFO mode or non-FIFO mode. In FIFO mode, a 16-byte transmit FIFO holds data from the host to be transmitted on the serial link and a 16-byte receive FIFO that buffers data from the serial link until read by the host.

The serial ports contain programmable baud rate generators that divide the internal reference clock by divisors of 1 to ($2^{16} - 1$), and produce a 16x clock for driving the transmitter and receiver logic. The internal reference clock can be programmed to support MIDI. The serial ports have complete modem-control capability and a prioritized interrupt system.

### Parallel Port

The 82091AA provides a multi-function parallel port that transfers information between the host and peripheral device (e.g., printer). The parallel port interface contains nine control/status lines and an 8-bit data bus. The standard PC/AT compatible logical address assignments for LPT1, LPT2, and LPT3 are supported. The parallel port can be configured for one of four modes and supports the following IEEE Standard 1284 parallel interface protocol standards:

| Parallel Port Mode | Parallel Interface Protocol |
|---|---|
| ISA-Compatible Mode | Compatibility, Nibble |
| PS/2-Compatible Mode | Byte |
| EPP Mode | EPP |
| ECP Mode | ECP |

For ISA-Compatible and PS/2-Compatible modes, software controls the handshake signals on the parallel port interface to transfer data between the host and peripheral device. Status and Control registers permit software to monitor the state of the peripheral device and generate handshake sequences.

The EPP parallel port interface protocol increases throughput by specifying an automatic handshake sequence. In EPP mode, the 82091AA parallel port automatically generates this handshake sequence in hardware to transfer data between the host and peripheral device.

In addition to a hardware handshake on the parallel port interface, the ECP protocol specification also defines DMA and FIFO capability. To minimize processor overhead data transfer to/from a peripheral device, the 82091AA parallel port, in ECP mode, provides a 16-byte FIFO with DMA capability.

**IDE Interface**

The 82091AA supports the IDE (Integrated Drive Electronics) interface by providing chip selects and lower data byte control. Two chip selects are used to access registers on the IDE device. Separate lower and upper byte data control signals are provided. With these control signals, minimal external logic is needed to implement 16-bit IDE I/O and DMA interfaces.

**Game Port**

The 82091AA provides a game port chip select signal for use when the 82091AA is in an add-in card application. This function is assigned to I/O address location 201h. Note that when the 82091AA is located on the motherboard, this feature is not available.

**Power Management**

82091AA power management provides a mechanism for saving power when the device or a portion of the device is not being used. By programming the appropriate 82091AA registers, software can invoke power management to the entire 82091AA or selected modules within the 82091AA (e.g., floppy disk controller, serial port, or parallel port). There are two methods for applying power management—direct powerdown or auto powerdown. Direct powerdown turns off the clock to a particular module immediately placing that module into a powerdown state. This method removes the clock regardless of the activity or status of the module. When auto powerdown is invoked, the module enters a powerdown state (clock is turned off) after certain conditions are met and the module is in an idle state.

## 1.1. 3.3V/5V Operating Modes

The 82091AA can operate at a power supply of 3.3V, 5V or a mix of 3.3V and 5V. The mixed power supply mode provides 5V interfaces for the floppy disk controller and parallel port while all other 82091AA interfaces and internal logic (including the floppy disk controller and parallel port internal circuitry) operate at 3.3V. The mixed mode permits 5V floppy disk drives and parallel port peripherals to be used in a 3.3V system without external buffering.

**NOTE:**
3.3V operation is available only in the 82091AA.

# 2.0. SIGNAL DESCRIPTION

This section describes the 82091AA signals. The interface signals are shown in Figure 5 and described in the following tables. Signal descriptions are organized by functional group.

Note that the ''#'' symbol at the end of a signal name indicates the active, or asserted, state occurs when the signal is at a low voltage level. When ''#'' is not present after the signal name, the signal is asserted when at the high voltage level.

The terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of ''active-low'' and ''active-high'' signals. The term **assert**, or **assertion**, indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term **negate**, or **negation**, indicates that a signal is inactive.

The following notations are used to describe pin types:

I Input Pin
O Output Pin
I/O Bi-Directional Pin

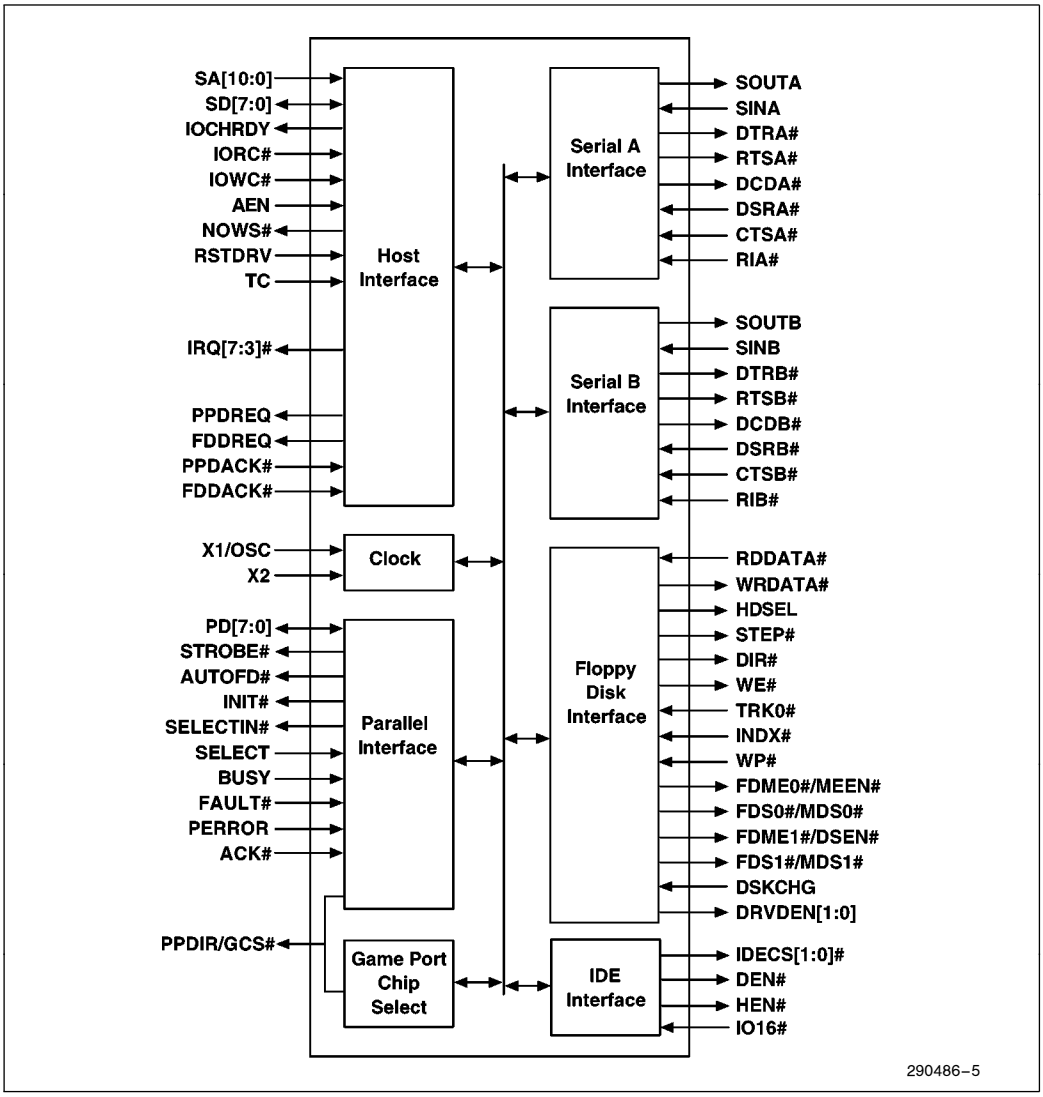



**Figure 5. 82091AA Signals**

## 2.1 Host Interface Signals

| Signal Name | Type | Description |
|---|---|---|
| **ISA SIGNALS** | | |
| SA[10:0] | I | **SYSTEM ADDRESS BUS:** The 82091AA decodes the standard ISA I/O address space using SA[9:0]. SA10 is used along with SA[9:0] to decode the extended register set of the ECP parallel port. SA[10:0] connects directly to the ISA system address bus. |
| SD[7:0] | I/O | **SYSTEM DATA BUS:** SD[7:0] is a bi-directional data bus. Data is written to and read from the 82091AA on these signal lines. SD[7:0] connect directly to the ISA system data bus. |
| IORC# | I | **I/O READ COMMAND STROBE:** IORC# is an I/O access read control signal. When a valid internal address is decoded by the 82091AA and IORC# is asserted, data at the decoded address location is driven onto the SD[7:0] signal lines. |
| IOWC# | I | **I/O WRITE COMMAND STROBE:** IOWC# is an I/O access write control signal. When a valid internal address is decoded by the 82091AA and IOWC# is asserted, data on the SD[7:0] signal lines is written into the decoded address location at the rising edge of IOWC#. |
| NOWS# | O | **NO WAIT-STATES:** End data transfer signal. The 82091AA asserts NOWS# when a valid internal address is decoded by the 82091AA and the IORC# or IOWC# signal is asserted. This reduces the total bus cycle time by eliminating the wait-states associated with the default 8-bit I/O cycles. NOWS# is not asserted for IDE accesses or DMA accesses. This is an open drain output pin. |
| IOCHRDY | O | **I/O CHANNEL READY:** The 82091AA uses this signal for parallel port data transfers when the parallel port is in EPP mode. In this case, the 82091AA negates IOCHRDY to extend the cycle to allow for completion of transfers to/from the peripheral attached to the parallel port. When the parallel port is in EPP mode, the 82091AA negates IOCHRDY to lengthen the ISA Bus cycle if the parallel port BUSY signal is asserted.<br>The 82091AA also uses IOCHRDY during hardware configuration time (see Section 4.0, AIP Configuration). If IOWC#/IORC# is asserted to the 82091AA during hardware configuration time, the 82091AA negates IOCHRDY until hardware configuration time is completed. This is an open drain output pin. |
| AEN | I | **ADDRESS ENABLE:** AEN is used during DMA cycles to prevent the 82091AA from misinterpreting DMA cycles from valid I/O cycles. When negated, AEN indicates that the 82091AA may respond to address and I/O commands addressed to the 82091AA. When asserted, AEN informs the 82091AA that a DMA transfer is occurring. When AEN is asserted and a xDACK# signal is asserted, the 82091AA responds to the cycle as a DMA cycle. |
| RSTDRV | I | **RESET DRIVE:** RSTDRV forces the 82091AA to a known state. All 82091AA registers are set to their default state. |
| X1/OSC | I | **CRYSTAL1/OSCILLATOR:** Main clock input signal can be a 24 MHz crystal connected across X1 and X2 or a 24 MHz TTL level clock input connected to X1. |
| X2 | I | **CRYSTAL2:** This signal pin is connected to one side of the crystal when a crystal oscillator is used to provide the main clock. If an external oscillator/clock is connected to X1, this pin is not used and left unconnected. |

## 2.1 Host Interface Signals (Continued)

| Signal Name | Type | Description |
|---|---|---|
| **DMA SIGNALS** | | |
| FDDREQ | O | **FLOPPY DISK CONTROLLER DMA REQUEST:** The 82091AA asserts FDDREQ to request service from a DMA controller for the FDC module. This signal is enabled/disabled by bit 3 of the Digital Output Register (DOR). When disabled, FDDREQ is tri-stated. |
| FDDACK# | I | **FLOPPY DISK CONTROLLER DMA ACKNOWLEDGE:** The DMA controller asserts this signal to acknowledge the FDC DMA request. When asserted, the IORC# and IOWC# inputs are enabled during DMA transfers. This signal is enabled/disabled by bit 3 of the DOR. |
| PPDREQ | O | **PARALLEL PORT DMA REQUEST:** Parallel port DMA service request to the system DMA controller. This signal is only used when the parallel port is in ECP hardware mode and is always negated when the parallel port is not in this mode. In ECP hardware mode DMA requests are enabled/disabled by bit 3 of the ECP Extended Control Register (ECR). When disabled, PPDREQ is tri-stated. |
| PPDACK# | I | **PARALLEL PORT DMA ACKNOWLEDGE:** The DMA controller asserts this signal to acknowledge the parallel port DMA request. When asserted the IORC# and IOWC# inputs are enabled during DMA transfers. This signal is enabled/disabled by bit 3 of the ECR Register. |
| TC | I | **TERMINAL COUNT:** The system DMA controller asserts TC to indicate it has reached the last programmed data transfer. TC is accepted only when FDDACK# or PPDACK# is asserted. |
| **INTERRUPT SIGNALS** | | |
| IRQ3, IRQ4 | O | **INTERRUPT 3 AND 4:** IRQ3 and IRQ4 are associated with the serial ports and can be programmed (via the AIPCFG2 Register) to be either active high or active low. These signals can be configured for a particular serial channel via hardware configuration (at powerup) or by software configuration.<br>**Under Hardware Configuration**<br>IRQ3 is used as a serial port interrupt if the serial port is configured at address locations 2F8h–2FFh or 2E8h–2EFh. IRQ4 is used as a serial port interrupt if the serial port is configured at address locations 3F8h–3FFh or 3E8h–3EFh.<br>**Under Software configuration**<br>IRQ3 and IRQ4 are independently configured (i.e., the IRQ does not automatically track the communication port address assignment).<br>These interrupts are enabled/disabled globally via bit 3 of the serial port Modem Control Register (MCR) and for specific conditions via the Interrupt Enable Register (IER). IRQ3 and IRQ4 are tri-stated when not enabled. |
| IRQ5, IRQ7 | O | **INTERRUPT REQUEST 5:** IRQ5 and IRQ7 are associated with the parallel port and can be programmed (via AIPCFG2 Register) to be either active high or active low. Either IRQ5 or IRQ7 is enabled/disabled via PCFG1 Register to signal a parallel port interrupt. The interrupt not selected is disabled and tri-stated.<br>During hardware configuration (see Section 4.0, AIP Configuration), IRQ5 is used if the parallel port is assigned to 278h–27Fh and IRQ7 is used if the parallel port interrupt is assigned to either 3BCh–3BFh or 378h–37Fh. |

## 2.1 Host Interface Signals (Continued)

| Signal Name | Type | Description |
|---|---|---|
| **INTERRUPT SIGNALS** (Continued) | | |
| IRQ6 | O | **INTERRUPT REQUEST 6:** IRQ6 is associated with the floppy disk controller and can be programmed (via the AIPCFG2 Register) to be either active high or active low. In non-DMA mode this signal is asserted to signal when a data transfer is ready. IRQ6 is also asserted to signal the completion of the execution phase for certain FDC commands. This signal is enabled/disabled by the DMAGATE bit in the Digital Output Register of the FDC. The signal is tri-stated when disabled. |

## 2.2 Floppy Disk Controller Interface

| Signal Name | Type | Description |
|---|---|---|
| RDDATA# | I | **READ DATA:** Serial data from the disk drive. |
| WRDATA# | O | **WRITE DATA:** MFM serial data to the disk drive. Precompensation value is selectable through software. |
| HDSEL | O | **HEAD SELECT:** Selects which side of a disk is to be accessed. When asserted (low), side 1 is selected. When negated (high), side 0 is selected. |
| STEP# | O | **STEP:** STEP# supplies step pulses (asserted) to the drive to move the head between the tracks during a seek operation. |
| DIR# | O | **DIRECTION:** Controls the direction the head moves when a step signal is present. The head moves toward the center when DIR# is asserted and away from the center when negated. |
| WE# | O | **WRITE ENABLE:** WE# is a disk drive control signal. When asserted, WE# enables the head to write to the disk. |
| TRK0# | I | **TRACK0:** The disk drive asserts this signal to indicate that the head is on track 0. |
| INDX# | I | **INDEX:** The disk drive asserts this signal to indicate the beginning of the track. |
| WP# | I | **WRITE PROTECT:** The disk drive asserts this signal to indicate that the disk drive is write-protected. |
| DSKCHG | I | **DISK CHANGE:** The disk drive asserts this signal to indicate that the drive door has been opened. The state of this signal input is available in the Digital Input Register (DIR#). |
| DRIVDEN0<br>DRIVDEN1 | O | **DRIVE DENSITY:** These signals are used by the disk drive to configure the drive for the appropriate media density. These signals are controlled by the FDC's Drive Specification Command. |

## 2.2 Floppy Disk Controller Interface (Continued)

| Signal Name | Type | Description |
|---|---|---|
| FDME1#/ DSEN#(1) | O | **FLOPPY DRIVE MOTOR ENABLE 1, IDLE, OR DRIVE SELECT ENABLE:** This signal pin has two functions(1). FDME1# is the motor enable for drive 1. FDME1# is directly controlled via the Digital Output Register (DOR) and is a function of the mapping based on the BOOTSEL bits in the Tape Drive Register (TDR).<br>The Drive Select Enable (DSEN#) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support). |
| FDS1#/ MDS1(1) | O | **FLOPPY DRIVE SELECT1, POWERDOWN, OR MOTOR DRIVE SELECT 1:** This signal pin has two functions(1). FDS1# is the floppy drive select for drive 1. FDS1# is controlled by the select bits in the DOR and is a function of the mapping based on the BOOTSEL bits in the TDR.<br>The Motor Drive Select 1 (MDS1) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support). |
| FDME0#/ MEEN#(1) | O | **FLOPPY DRIVE MOTOR ENABLE 0 OR MOTOR ENABLE ENABLE:** This signal pin has two functions(1). FDME0# is the motor enable for drive 0. FDME0# is directly controlled via the Digital Output Register (DOR) and is a function of the mapping based on the BOOTSEL bits in the Tape Drive Register (TDR).<br>The Motor Enable Enable (MEEN#) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support). |
| FDS0#/ MDS0(1) | O | **FLOPPY DRIVE SELECT 0 OR MOTOR DRIVE SELECT 0:** This signal pin has two functions(1). FDS0# is the floppy drive select for drive 0. This output is controlled by the drive select bits in the DOR and is a function of the mapping based on BOOTSEL bits in the TDR.<br>The Motor Drive Select 0 (MDS0) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support). |

**NOTE:**
1. The function selected for these pins is based on the FDDQTY bit in the FCFG1 Register as shown in the following table.

| Signal Pin | 2 Drive System (FDDQTY = 0) | 4 Drive System (FDDQTY = 1) |
|---|---|---|
| FDME1#/DSEN# | FDME1# | DSEN# |
| FDS1#/MDS1# | FDS1# | MDS1 |
| FDME0#/MEEN# | FDME0# | MEEN# |
| FDS0#/MDS0 | FDS0# | MDS0 |

When FDDQTY = 1, these signal pins are used to control an external decoder for a four floppy disk drive system as described in Appendix A, FDC Four Drive Support.

## 2.3 Serial Port Interface

Serial Port A signal names end in the letter A and Serial Port B signal names end in the letter B. Serial Port A and B signals have the same functionality.

| Signal Name | Type | Description |
|---|---|---|
| CTSA#, CTSB# | I | **CLEAR TO SEND:** When asserted, this signal indicates that the modem or data set is ready to exchange data. The CTS# signal is a modem status input whose condition the CPU can determine by reading the CTS bit in Modem Status Register (MSR) for the appropriate serial port. The CTS bit is the compliment of the CTS# signal. The DCTS bit in the MSR indicates whether the CTS# input has changed state since the previous reading of the MSR. CTS# has no effect on the transmitter. |
| DCDA#, DCDB# | I | **DATA CARRIER DETECT:** When asserted, this signal indicates that the data carrier has been detected by the modem or data set. The DCD# signal is a modem status whose condition the CPU can determine by reading the DCD bit in the MSR for the appropriate serial port. The DCD bit is the compliment of the DCD# signal. The DDCD bit in the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. DCD# has no effect on the transmitter. |
| DSRA#, DSRB# | I | **DATA SET READY:** When asserted, this signal indicates that the modem or data set is ready to establish the communications link with the serial port module. The DSR# signal is a modem status whose condition the CPU can determine by reading the DSR bit in the MSR for the appropriate serial channel. The DSR bit is the compliment of the DSR# signal. The DSR bit in the MSR indicates whether the DSR# input has changed state since the previous reading of the MSR. DSR# has no effect on the transmitter. |
| DTRA#, DTRB# | I/O | **DATA TERMINAL READY:** DTRA#/DTRB# are outputs during normal system operations. When asserted, this signal indicates to the modem or data set that the serial port module is ready to establish a communications link. The DTR# signal can be asserted via the Modem Control Register (MCR). A hard reset negates this signal.<br>**Hardware Configuration**<br>These signals are only inputs during hardware configuration time (RSTDRV asserted and for a short time after RSTDRV is negated). (See Section 4.0, AIP Configuration.) |
| RIA#, RIB# | I | **RING INDICATOR:** When asserted, this signal indicates that a telephone ringing signal has been received by the modem or data set. The RI# signal is a modem status input whose condition the CPU can determine by reading the RI bit in the MSR for the appropriate serial channel. The RI bit is the compliment of the RI# signal. The TERI bit in the MSR indicates whether the RI# input has changed from low to high since the previous reading of the MSR. |

## 2.3 Serial Port Interface (Continued)

| Signal Name | Type | Description |
|---|---|---|
| RTSA#, RTSB# | I/O | **REQUEST TO SEND:** RTSA#/RTSB# are outputs during normal system operations. When asserted, this signal informs the modem or data set that the serial port module is ready to exchange data. The RTS# signal can be asserted via the RTS bit in the Modem Control Register. A hard reset negates this signal.<br>**Hardware Configuration**<br>These signals are only inputs during hardware configuration time (RSTDRV asserted and for a short time after RSTDRV is negated). (See Section 4.0, AIP Configuration.) |
| SINA, SINB | I | **SERIAL INPUT:** Serial data input from the communications link. (Peripheral device, modem, or data set.) |
| SOUTA, SOUTB | I/O | **SERIAL OUTPUT:** SOUTA/SOUTB are serial data outputs to the communications link during normal system operations. (Peripheral device, modem, or data set.) The SOUT signal is set to a marking state (logic 1) after a hard reset.<br>**Test Mode**<br>In test mode (selected via the SACFG2 or SBCFG2 Registers), the baudout from the baud rate generator is output on SOUTx.<br>**Hardware Configuration**<br>These signals are only inputs during hardware configuration time (RSTDRV asserted and for a short time after RSTDRV is negated). (See Section 4.0, AIP Configuration.) |

## 2.4 IDE Interface

| Signal Name | Type | Description |
|---|---|---|
| IO16# | I | **16-BIT I/O:** This signal is driven by I/O devices on the ISA Bus to indicate support for 16-bit I/O bus cycles. The IDE interface asserts this signal to the 82091AA to indicate support for 16-bit transfers. For IDE transfers, the 82091AA asserts HEN# when IO16# is asserted. |
| IDECS[1:0]# | I/O | **IDE CHIP SELECT:** IDECS[1:0]# are outputs during normal system operation and are chip selects for the IDE interface. IDECS[1:0]# select the Command Block Registers of the IDE device and are decoded from SA[9:3] and AEN.<br>**Hardware Configuration**<br>These signals are only inputs during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.) |

## 2.4 IDE Interface (Continued)

| Signal Name | Type | Description |
|---|---|---|
| DEN# | I/O | **DATA ENABLE:** DEN# is an output during normal system operations and is a data enable for an external data buffer for all 82091AA and IDE accesses. The SD[7:0] signals can be connected directly to the ISA. In this case, the DEN# signal is not used. However, an external buffer can be used to isolate the SD[7:0] signals from the 240 pF loading of the ISA Bus. With an external buffer implementation, DEN# controls the external buffers for transfers to/from the ISA Bus.<br>**Hardware Configuration**<br>This signal is only an input during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.) |
| HEN# | I/O | **IDE UPPER DATA TRANSCEIVER ENABLE:** HEN# is an output during normal system operations and is a high byte data transceiver enable signal for the IDE hard disk drive interface. HEN# is asserted for I/O accesses to the IDE data register when the drive asserts IO16#.<br>**Hardware Configuration**<br>This signal is only an input during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.) |

## 2.5 Parallel Port External Buffer Control/Game Port

| Signal Name | Type | Description |
|---|---|---|
| PPDIR/GCS# | I/O | **PARALLEL PORT DIRECTION (PPDIR) or GAME PORT CHIP SELECT (GCS#):** This signal is an output during normal operations and provides the PPDIR and GCS# functions as follows:<br>**PPDIR**<br>This signal pin functions as a parallel port direction control output when the 82091AA is configured for software motherboard mode (SWMB). For configuration details, see Section 4.0, AIP Configuration. If external buffers are used on PD[7:0], PPDIR can be used to control the buffer direction. The 82091AA drives this signal low when PD[7:0] are outputs and the 82091AA drives this signal high when PD[7:0] are inputs. Note that if a configuration mode other than SWMB is selected, this signal pin is a game port chip select and does not track the PD[7:0] signal direction.<br>**GCS#**<br>This signal pin functions as a game port chip select output when 82091AA configuration is set for Software Add-In (SWAI), Hardware Basic (HWB), or Hardware Extended (HWE) modes. When the host accesses I/O address 201h, GCS# is asserted.<br>**Hardware Configuration**<br>This signal is only an input during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.) |

## 2.6 Parallel Port Interface

The 82091AA parallel port is a multi-function interface that can be configured for one of four hardware modes (see Section 4.0, AIP Configuration). The hardware modes are ISA-Compatible, PS/2-Compatible, EPP, and ECP modes. These parallel port modes support the compatibility, nibble, byte, EPP and ECP parallel interface protocols described in the IEEE 1284 standard. The operation and use of the interface signal pins are a function of the parallel port hardware mode selected and the protocol used.

Table 1 shows a matrix of the 82091AA parallel port signal names and corresponding signal names for each of the protocols. Sections 2.6.1–2.6.5 provide a signal description for the five interface protocols. Note that the 82091AA hardware operations are the same for Compatibility and Nibble protocols. The signals, however, are controlled and used differently via software and the peripheral device.

**Table 1. Parallel Port Signal Name Cross Reference**

| 82091AA Signal Names | Compatibility Protocol Signal Names | Nibble Protocol Signal Names | Byte Protocol Signal Names | EPP Protocol Signal Names | ECP Protocol Signal Names |
|---|---|---|---|---|---|
| STROBE# | Strobe# | — | HostCLK | Write# | HostClk |
| BUSY | Busy | PtrBusy | PtrBusy | Wait# | PeriphAck |
| ACK# | Ack# | PtrClk | PtrClk | Intr | PeriphClk# |
| SELECT | Select | Xflag | Xflag | Xflag | Xflag |
| PERROR | PError | AckDataReq | AckDataReq | AckDataReq | AckReverse# |
| FAULT# | Fault# | DataAvail# | DataAvail# | DataAvail# | PeriphRequest# |
| INIT# | Init# | — | — | Init# | ReverseRequest# |
| AUTOFD# | AutoFd# | HostBusy | HostBusy | DStrb# | HostAck |
| PD[7:0] | Data[8:1] | — | Data[8:1] | Data[8:1] | Data[8:1] |
| SELECTIN# | SelectIn# | — | — | AStrb# | ECP Mode |

**NOTE:**
Not all parallel port signal pins are used for certain parallel port interface protocols. These signals are labeled ‘‘—’’.

### 2.6.1 COMPATIBILITY PROTOCOL SIGNAL DESCRIPTION

Except for the data bus, the 82091AA and compatibility protocol signal names are the same. For the data bus, the 82091AA signal names PD[7:0] corresponds to the compatibility protocol signal names Data[8:1].

| 82091AA Signal Name | Type | Compatibility Protocol Signal Name and Description |
|---|---|---|
| STROBE# | O | **STROBE:** The host asserts STROBE# to latch data into the peripheral device's input latch. This signal is controlled via the PCON Register. |
| BUSY | I | **BUSY:** BUSY is asserted by the peripheral to indicate that the peripheral device is not ready to receive data. The status of this signal line is reported in the PSTAT Register. |
| ACK# | I | **ACKNOWLEDGE:** The printer asserts this signal to indicate that it has received the data and is ready for new data. The status of this signal line is reported in the PSTAT Register. |
| SELECT | I | **SELECT:** SELECT is asserted by the peripheral device to indicate that the device is on line. The status of this signal line is reported in the PSTAT Register. |
| PERROR | I | **PAPER ERROR:** The peripheral device asserts PERROR to indicate that it has encountered an error in the paper path. The exact meaning varies from peripheral device to peripheral device. The status of this signal line is reported in the PSTAT Register. |
| FAULT# | I | **FAULT:** FAULT# is asserted by the peripheral device to indicate that an error has occurred. The status of this signal line is reported in the PSTAT Register. |
| INIT# | O | **INITIALIZE:** The host asserts INIT# to issue a hardware reset to the peripheral device. This signal is controlled via the PCON Register. |
| AUTOFD# | O | **AUTO FEED:** AUTOFD# is asserted by the host to put the peripheral device into auto-line feed mode. This means that when software asserts this signal, the printer is instructed to advance the paper one line for each carriage return encountered. This signal is controlled via the PCON Register. |
| PD[7:0] | O | **DATA:** Forward channel data. |
| SELECTIN# | O | **SELECT INPUT:** SELECTIN# is asserted by the host to select a peripheral device. This signal is controlled via the PCON Register. |

### 2.6.2 NIBBLE PROTOCOL SIGNAL DESCRIPTION

The Nibble protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the Nibble protocol signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, AUTOFD# (HostBusy) asserted refers to AUTOFD# (HostBusy) at a low level.

| 82091AA Signal Name | Type | Nibble Protocol Signal Name and Description |
|---|---|---|
| STROBE# | O | **STROBE:** The host controls this signal via the PCON Register and STROBE# should be held negated by the host. |
| BUSY | I | **PRINTER BUSY (PtrBusy):** The peripheral drives this signal to transfer data bits 3 and 7 sequentially. The status of this signal line is reported in the PSTAT Register. |
| ACK# | I | **PRINTER CLOCK (PtrClk):** The peripheral device asserts ACK# (PtrClk) to indicate to the host that data is available. The signal is subsequently asserted to qualify data being sent to the host. The status of this signal line is reported in the PSTAT Register. If interrupts are enabled via the PCON Register, the assertion of this signal causes a host interrupt to be generated. |
| SELECT | I | **XFLAG:** The peripheral device drives this signal to transfer data bits 1 and 5 sequentially. The status of this signal line is reported in the PSTAT Register. |
| PERROR | I | **ACKNOWLEDGE DATA REQUEST (AckDataReq):** This signal is initially high. The peripheral device drives this signal low to acknowledge HostBusy assertion. PERROR is subsequently used to transfer data bits 2 and 6 sequentially. The status of this signal line is reported in the PSTAT Register. |
| FAULT# | I | **DATA AVAILABLE (DataAvail):** The peripheral device asserts FAULT# (DataAvail) to indicate data availability. Subsequently used to transfer data bits 0 and 4 sequentially. The status of this signal line is reported in the PSTAT Register. |
| INIT# | O | **INITIALIZE:** The host controls this signal via the PCON Register. |
| AUTOFD# | O | **HOST BUSY (HostBusy):** The host negates AUTOFD# (HostBusy) in response to ACK# being asserted. This signal is subsequently driven low to enable the peripheral to transfer data to the host. AUTOFD# is then driven high to acknowledge receipt of byte data. This signal is controlled via the PCON Register. |
| PD[7:0] | O | **DATA:** This 8-bit output data path to the peripheral Host data is written to the peripheral attached to the parallel port interface on these signal lines. |
| SELECTIN# | O | **SELECT INPUT:** This signal is controlled by the PCON Register. |

### 2.6.3 BYTE MODE SIGNAL DESCRIPTION

The Byte protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the Byte protocol signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, STROBE# (HostClk) asserted refers to STROBE# (HostClk) at a low level.

| 82091AA Signal Name | Type | Byte Protocol Signal Name and Description |
|---|---|---|
| STROBE# | O | **HOST CLOCK (HostClk):** This signal is strobed low by the host to acknowledge receipt of data. Note that the peripheral must not interpret this as a latch strobe for forward channel data. |
| BUSY | I | **PRINTER BUSY (PtrBusy):** The peripheral device asserts BUSY (PtrBusy) to provide forward channel peripheral busy status. The status of this signal line is reported in the PSTAT Register. |
| ACK# | I | **PRINTER CLOCK (PtrClk):** The peripheral device asserts ACK# (PtrClk) to indicate to the host that data is available. The signal is subsequently asserted to qualify data being sent to the host. The status of this signal line is reported in the PSTAT Register. If interrupts are enabled via the PCON Register, the assertion of this signal causes a host interrupt to be generated. |
| SELECT | I | **XFLAG:** SELECT (XFLAG) is asserted by the peripheral device to indicate that the device is on line. The status of this signal line is reported in the PSTAT Register. |
| PERROR | I | **ACKNOWLEDGE DATA REQUEST (AckDataReq):** This signal is initially high. The peripheral device drives this signal low to acknowledge HostBusy assertion. The status of this signal line is reported in the PSTAT Register. |
| FAULT# | I | **DATA AVAILABILITY (DataAvail):** The peripheral device asserts FAULT# (DataAvail) to indicate data availability. The status of this signal line is reported in the PSTAT Register. |
| INIT# | O | **INITIALIZE:** The host controls this signal via the PCON Register and INIT# should be held in the negated state. |
| AUTOFD# | O | **HOST BUSY (HostBusy):** The host negates AUTOFD# (HostBusy) in response to ACK# being asserted.The signal is subsequently driven low to enable the peripheral to transfer data to the host. AUTOFD# is then driven high to acknowledge receipt of nibble data. This signal is controlled via the PCON Register. |
| PD[7:0] | O | **DATA:** This 8-bit data bus is used for bi-directional data transfer. |
| SELECTIN# | I/O | **SELECT INPUT:** This signal is controlled by the PCON Register. |

### 2.6.4 ENHANCED PARALLEL PORT (EPP) PROTOCOL SIGNAL DESCRIPTION

EPP protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the EPP mode signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, BUSY (Wait#) asserted refers to BUSY (Wait#) being high.

| 82091AA Signal Name | Type | EPP Protocol Signal Name and Description |
|---|---|---|
| STROBE# | O | **WRITE (Write#):** STROBE# (Write#) indicates an address or data read/write operation to the peripheral. The 82091AA drives this signal low for a write and high for a read. |
| BUSY | I | **WAIT (Wait#):** The peripheral sets BUSY (Wait#) low to indicate that the device is not ready. When BUSY signal is low, the 82091AA negates IOCHRDY on the ISA Bus to lengthen the I/O cycles. The peripheral device sets BUSY (Wait#) high to indicate that transfer of data or address is completed. |
| ACK# | I | **INTERRUPT REQUEST (Intr):** The peripheral asserts ACK# (Intr) to generate an interrupt the host. When this signal is low and interrupts are enabled via bit 4 of the PCON Register, the 82091AA generates an interrupt request (via either IRQ5 or IRQ7) to the host. |
| SELECT | I | **SELECT:** SELECT is asserted by the peripheral device to indicate that the device is on line. The status of this signal line is reported in the PSTAT Register. |
| PERROR | I | **PAPER ERROR:** The peripheral device asserts PERROR to indicate that it has encountered an error in the paper path. The exact meaning varies from peripheral device to peripheral device. The status of this signal line is reported in the PSTAT Register. |
| FAULT# | I | **FAULT:** FAULT# is asserted by the peripheral device to indicate that an error has occurred. The status of this signal line is reported in the PSTAT Register. |
| INIT# | O | **INITIALIZE:** The host asserts INIT# to issue a hardware reset to the peripheral device. This signal is controlled via the PCON Register. |
| AUTOFD# | O | **DATA STROBE (DStrb#):** The 82091AA asserts AUTOFD# (DStrb#) to indicate that valid data is present on PD[7:0] and is used by the peripheral to latch data during write cycles. For reads, the 82091AA reads in data from PD[7:0] when this signal is asserted. |
| PD[7:0] | I/O | **DATA:** This 8-bit bi-directional bus provides addresses or data during the write cycles and supplies addresses or data to the 82091AA during the read cycles. |
| SELECTIN# | O | **ADDRESS STROBE (AStrb#):** The 82091AA asserts SELECTIN# (AStrb#) to indicate that a valid address is present on PD[7:0] and is used by the peripheral to latch addresses during write cycles. For reads, the 82091AA reads in an address from PD[7:0] when this signal is asserted. |

### 2.6.5 EXTENDED CAPABILITIES PORT (ECP) PROTOCOL SIGNAL DESCRIPTION

ECP protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the ECP protocol signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, STROBE# (HostClk) asserted refers to STROBE# (HostClk) being low.

| 82091AA Signal Name | Type | ECP Protocol Signal Name and Description |
|---|---|---|
| STROBE# | O | **HOST CLOCK (HostClk):** In the forward direction, the 82091AA asserts STROBE# (HostClk) to instruct the peripheral to latch the data on PD[7:0]. During write operations, the peripheral should latch data on the rising edge of STROBE# (HostClk). STROBE# (HostClk) handshakes with BUSY (PeriphAck) during write operations and is negated after the 82091AA detects BUSY (PeriphAck) asserted. STROBE# (HostClk) is not asserted by the 82091AA again until BUSY (PeriphAck) is detected negated. For read operations (reverse direction), STROBE# (HostClk) is not used. |
| BUSY | I | **PERIPHERAL ACKNOWLEDGE (PeriphAck):** The peripheral device asserts this signal during a host write operation to acknowledge receipt of data. The peripheral device then negates the signal after STROBE# is detected high to terminate the transfer. For host write operations (forward direction), this signal handshakes with STROBE# (HostClk). During a host read operation (reverse direction), BUSY (PeriphAck) is normally low and is driven high by the peripheral to identify Run Length Encoded (RLE) data. |
| ACK# | I | **PERIPHERAL CLOCK (PeriphClk):** During a peripheral to host transfer (reverse direction), ACK# (PeriphClk) is asserted by the peripheral to indicate data is valid on the data bus and then negated after AUTOFD# is detected high. This signal handshakes with AUTOFD# to transfer data. |
| SELECT | I | **XFLAG (Xflag):** This signal is asserted by the peripheral to indicate that it is on-line. The status of this signal line is reported in the PSTAT Register. |
| PERROR | I | **ACKNOWLEDGE REVERSE (AckReverse#):** PERROR (AckReverse#) is driven low by the peripheral to acknowledge a reverse transfer request by the host. This signal handshakes with INIT# (ReverseRequest#). The status of this signal line is reported in the PSTAT Register. |
| FAULT# | I | **PERIPHERAL REQUEST (PeriphRequest#):** The peripheral asserts FAULT# (PeriphRequest#) to request a reverse transfer. The status of this signal line is reported in the PSTAT Register. |
| INIT# | O | **REVERSE REQUEST (ReverseRequest#):** The host controls this signal via the PCON Register to indicate the transfer direction. The host asserts this signal to request a reverse transfer direction and negates the signal for a forward transfer direction. |
| AUTOFD# | O | **HOST ACKNOWLEDGE (HostAck):** The 82091AA asserts AUTOFD# (HostAck) to request data from the peripheral (reverse direction). This signal handshakes with ACK# (PeriphClk). AUTOFD# (HostAck) is negated when the peripheral indicates valid state of the data bus (i.e., ACK# is detected asserted).<br>In the forward direction, AUTOFD# (HostAck) indicates whether PD[7:0] contain an address/RLE or data. The 82091AA asserts this signal to identify an address/RLE transfer and negates it to identify a data transfer. |
| PD[7:0] | I/O | **DATA:** PD[7:0] is a bi-directional data bus that transfers data, addresses, or RLE data. |
| SELECTIN# | O | **ECP MODE (ECPmode):** The host (via the PCON Register) negates this signal during ECP mode operation. |

## 2.7 Hard Reset Signal Conditions

Table 1 shows the state of all 82091AA output and bi-directional signals during hard reset (RSTDRV asserted). The strapping options described in Section 4.0, AIP Configuration are sampled when the 82091AA is hard reset.

**Table 2. Output and I/O Signal States During a Hard Reset**

| Signal Name | State |
|---|---|
| ACK# | — |
| AEN | — |
| AUTOFD# | Tri-state |
| BUSY | — |
| CTS[A,B]# | — |
| DCD[A,B]# | High |
| DEN# | High(1) |
| DIR# | High |
| DRVDEN[1:0]0 | Low |
| DSKCHG# | — |
| DTR[A,B]# | High(1) |
| FAULT# | — |
| FDDACK# | — |
| FDDREQ | Tri-state |
| FDME0#/MEEN# | High |
| FDME1#/DSEN# | High |
| FDS0#/MDS0 | High |
| FDS1#/MDS1 | High |

| Signal Name | State |
|---|---|
| HDSEL | High |
| HEN# | High(1) |
| IDECS[1,0]# | High(1) |
| INDX# | — |
| INIT# | Low |
| IO16# | — |
| IOCHRDY | Tri-state(2) |
| IORC# | — |
| IOWC# | — |
| IRQ[7:3] | Tri-state |
| NOWS# | Tri-state |
| PD[7:0] | Low |
| PERROR | — |
| PPDACK# | — |
| PPDREQ | Tri-state |
| PPDIR/GCS# | High(1) |
| RDDATA | — |
| RI[A,B]# | — |

| Signal Name | State |
|---|---|
| RSTDRV | — |
| RTS[A,B]# | High(1) |
| SA[10,0] | — |
| SD[7:0] | Tri-state |
| SELECT | — |
| SELECTIN# | Tri-state |
| SIN[A,B] | — |
| SOUT[A,B] | High(1) |
| STEP# | High |
| STROBE# | Tri-state |
| TC | — |
| TRK0# | — |
| WE# | High |
| WP# | — |
| WRDATA# | High |
| X1/OSC | — |
| X2 | — |

**NOTES:**
1. During and immediately after a hard reset, this signal is an input for hardware configuration. After the hardware configuration time, these signals go to the state specified in the table.
2. If IORC# or IOWC# is asserted, IOCHRDY will be asserted by the IOCHRDY.
3. Dashes represent input signals.

## 2.8 Power And Ground

| Signal Name | Type | Description |
|---|---|---|
| $V_{SS}$ | I | **GROUND:** The ground reference for the 82091AA. |
| $V_{CC}$ | I | **POWER:** The 5V/3.3V(1) modes are selected via strapping options at power-up (see Section 4.2, hardware Configuration). When strapping options ($V_{SEL}$) are set to 5V, the $V_{CC}$ pins must be connected to 5V. When strapping options are set to 3.3V, the $V_{CC}$ pins must be connected to 3.3V. |
| $V_{CCF}$ | I | **POWER:** The 5V/3.3V(1) power supply for the 82091AA. In 5V or 3.3V power supply modes (non-mixed mode), the voltage applied to $V_{CCF}$ is the same voltage as applied to $V_{CC}$.<br>For mixed mode operations, 5V is applied to $V_{CCF}$. This voltage provides 5V reference for the parallel port and floppy disk controller interfaces. Note that in mixed mode, 3.3V is applied to $V_{CC}$. |

**NOTE:**
1. 3.3V operation is available only in the 82091AA.

## 3.0 I/O ADDRESS ASSIGNMENTS

The 82091AA assigns CPU I/O address locations to its game port chip select, IDE interface, serial ports, parallel port, floppy disk controller, and the 82091AA configuration registers as indicated in Table 3. Except for the game port chip select (address 201h), address assignments are configurable. For example, the serial port can be assigned to one of eight address blocks. The parallel port can be assigned to one of three address blocks, and the IDE interface and floppy disk controller can be assigned to one of two address blocks. These address assignments are made during 82091AA configuration (either hardware configuration at powerup or a hard reset, or software configuration by programming the 82091AA configuration registers). In addition, the 82091AA configuration registers can be located at one of two address blocks during hardware configuration.

All of the 82091AA address locations are located in the host I/O address space. The address block assignments are shown in Table 3. The first hex address in the Address Block column represents the base address for that particular block.

**Table 3. AIP Address Assignments**

| Address Block (ISA Bus) | Assignment |
|---|---|
| 170–177h | IDE Interface—Secondary Address Block |
| 1F0–1F7h | IDE Interface—Primary Address Block |
| 201h | Game Port Chip Select |
| 220–227h | Serial Port |
| 228–22Fh | Serial Port |
| 238–23Fh | Serial Port |
| 26E–26Fh | 82091AA Configuration Registers—Primary Address Block (022–023h on X-Bus) |
| 278–27Fh | Parallel Port |
| 2E8–2EFh | Serial Port |
| 2F8–2FFh | Serial Port |
| 338–33Fh | Serial Port |
| 370–377h | Floppy Disk Controller—Secondary Address Block (376h and 377h are Shared with the IDE Drive Interface Secondary Address) |
| 378–37Fh | Parallel Port |
| 398–399h | 82091AA Configuration Registers—Secondary Address Block (024–025h on X-Bus) |
| 3BC–3BFh | Parallel Port (All Mopes Except EPP) |
| 3E8–3EFh | Serial Port |
| 3F0–3F7h | Floppy Disk Controller—Primary Address (3F6h and 3F7h are Shared with the IDE Drive Interface Primary Address) |
| 3F8–3FFh | Serial Port |
| 678–67Ah | Parallel Port (ECP Mode Peripheral Interface Protocol) |
| 778–77Ah | Parallel Port (ECP Mode Peripheral Interface Protocol) |
| 7BC–7BEh | Parallel Port (ECP Mode Peripheral Interface Protocol) |

**NOTES:**
1. The 82091AA does not contain IDE registers. However, the 82091AA provides the address block assignments for accessing the IDE registers that are located in the IDE device.
2. The standard PC/AT* compatible logical I/O address assignments are supported. For example, COM1 (3F8–3FFh) and COM2 (2F8–2FFh) are part of the serial port assignments and LPT1 (3BC–3BFh), LPT2 (378–37Fh), and LPT3 (278–27Fh) are part of the parallel port assignments.

*Other brands and names are the property of their respective owners.

## 4.0 AIP CONFIGURATION

82091AA configuration consists of setting up overall device operations along with certain functions pertaining to the individual 82091AA modules (parallel port, serial ports, floppy disk controller, and IDE interface). Overall device operations include selecting the clock frequency, power supply voltage, and address assignment for the configuration registers. Overall device operations also enable/disable access to the configuration registers and provide interrupt signal level control. For the individual modules, 82091AA configuration includes module address assignment, interrupt control, module enable/disable, powerdown control, test mode control, module reset, and certain functions specific to each module. The remainder of the functions unique to each module are handled via the individual module registers.

Two methods are provided for configuring the 82091AA—hardware configuration via strapping options at powerup (or whenever RSTDRV is asserted) and software configuration by programming the configuration registers. (For information on hardware configuration, see Section 4.2, Hardware Configuration. For information on software configuration, see Section 4.1, Configuration Registers.)

**NOTE:**

1. There are four hardware configuration modes—SWMB (Software Motherboard), SWAI (Software Add-In), HWB (Hardware Basic), and HWE (Hardware Extended). Some of these modes can be used without the need for programming the 82091AA configuration registers. Other modes use both hardware configuration strapping options and programming the configuration registers to set up the 82091AA.
2. The 82091AA's operating power supply voltage level, 82091AA clock frequency, and address assignment for the 82091AA configuration registers can only be configured by hardware configuration.

### 4.1 Configuration Registers

82091AA Configuration Space contains 13 configuration registers. Four of the registers (Product and Revision Identification Registers and the 82091AA Configuration 1 and 2 Registers) provide control and status information for the entire chip. In addition, two registers each for the floppy disk controller, parallel port, serial port A, and serial port B and one register for the IDE interface provide certain module status and control information. The 82091AA configuration registers are indirectly addressed by first writing to the 82091AA Configuration Index Register as described in Section 4.1.1. Thus, the 13 configuration registers occupy two address locations in the host's I/O address space—one for indirectly selecting the specific configuration register and the other for transfering register data. All 82091AA configuration registers are 8-bits wide and are accessed as byte quantities.

Some of the 82091AA Configuration registers described in this section contain reserved bits. These bits are labeled ''R''. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the value of reserved bit positions must first be read, merged with the new values for other bit positions, and then written back.

In addition to reserved bits within a register, the 82091AA configuration space contains address locations that are labeled ''Reserved'' (Table 5). While the 82091AA responds to accesses to these I/O addresses by completing the host cycle, writing to a reserved I/O address can result in unintended device operations. Values read from a reserved I/O address should not be used to permit future expansion and upgrades.

During a hard reset (RSTDRV asserted), the 82091AA sets its configuration registers to pre-determined **default** states. The default values are indicated in the individual register descriptions. The following nomenclature is used for register access attributes:

**RO** **Read Only.** If a register is read only, writes have no effect.

**R/W** **Read/Write.** A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

### 4.1.1 CFGINDX, CFGTRGT—CONFIGURATION INDEX REGISTER AND TARGET PORT

I/O Address: Hardware Configurable (see Table 4)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

CFGINDX and CFGTRGT are used to access 82091AA configuration space where all of the 82091AA configuration registers are located. CFGINDX and CFGTRGT are located in the host I/O address space and the address locations are hardware configurable as shown in Table 4. CFGINDX is an 8-bit register that contains the address index of the 82091AA configuration register to be accessed. CFGTRGT is a port for reading data from or writing data to the configuration register whose index address matches the address stored in the CFGINDX Register. Thus, to access a configuration register, CFGINDX must first be programmed with the index address. A software example is provided in this section demonstrating how to access the configuration registers.

**Table 4. Configuration Register Access Addresses**

| Address Selection | X-Bus Implementation | | ISA Bus Implementation | |
|---|---|---|---|---|
| | Index | Target | Index | Target |
| Primary Address | 22h | 23h | 26Eh | 26Fh |
| Secondary Address | 24h | 25h | 398h | 399h |

Table 5 summarizes the 82091AA configuration space. Following the table, is a detailed description of each register. The register descriptions are arranged in the order that they appear in Table 5.

| Bit | Description |
|---|---|
| 7:0 | **82091AA Configuration Register Address Index:** Bits[7:0] correspond to SD[7:0]. |

**Software Configuration**

Access Addresses for the two Software Configuration Modes:

| | Index | Target |
|---|---|---|
| For SWMB Mode Primary Address: | 22h | 23h |
| For SWMB Mode Secondary Address: | 24h | 25h |
| For SWAI, HWE, and HWB Modes Primary Address: | 26Eh | 26Fh |
| For SWAI, HWE, and HWB Modes Secondary Address: | 398h | 399h |

The following pseudo code sequence could be used to access the configuration registers under SWMB primary address:

Configuration register write: OUT 22h, ConfigRegAddr
OUT 23h, ConfigRegData

Configuration register read: OUT 22h, ConfigRegAddr
IN 23h

**Table 5. AIP Configuration Registers**

| 82091AA Configuration Address Index | Abbreviation | Register Name | Access |
|---|---|---|---|
| 00h | AIPID | Product Identification | RO |
| 01h | AIPREV | Revision Identification | RO |
| 02h | AIPCFG1 | 82091AA Configuration 1 | R/W |
| 03h | AIPCFG2 | 82091AA Configuration 2 | R/W |
| 04–0Fh | — | Reserved | — |
| 10h | FCFG1 | FDC Configuration | R/W |
| 11h | FCFG2 | FDC Power Management and Status | R/W |
| 12–1Fh | — | Reserved | — |
| 20h | PCFG | Parallel Port Configuration | R/W |
| 21h | PCFG2 | Parallel Port Power Management and Status | R/W |
| 22–2Fh | — | Reserved | — |
| 30h | SACFG1 | Serial Port A Configuration | R/W |
| 31h | SACFG2 | Serial Port A Power Management and Status | R/W |
| 32–3Fh | — | Reserved | — |
| 40h | SBCFG1 | Serial Port B Configuration | R/W |
| 41h | SBCFG2 | Serial Port B Power Management and Status | R/W |
| 42–4Fh | — | Reserved | — |
| 50h | ICFG | IDE Configuration | R/W |
| 51–FFh | — | Reserved | — |

**NOTE:**
Writing to a reserved I/O address should not be attempted and can result in unintended device operations.

### 4.1.2 AIPID—AIP IDENTIFICATION REGISTER

Index Address: 00h
Default Value: A0h
Attribute: Read Only
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **AIP IDENTIFICATION (AIPID):** A value of A0h is assigned to the 82091AA. This 8-bit register combined with the 82091AA Revision Identification Register uniquely identifies the device. |

### 4.1.3 AIPREV—AIP REVISION IDENTIFICATION

Index Address: 01h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register contains two fields that identify the revision of the 82091AA device. The revision number will be incremented for every stepping, even if change is invisible to software.

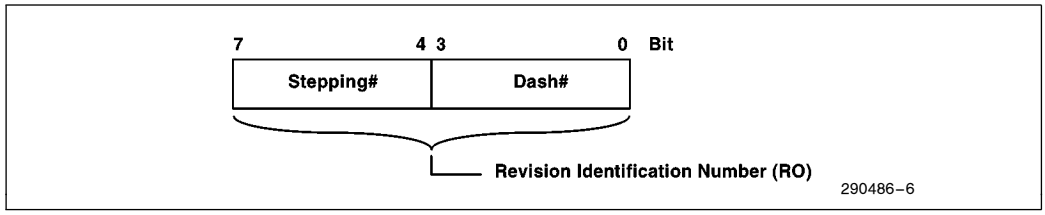


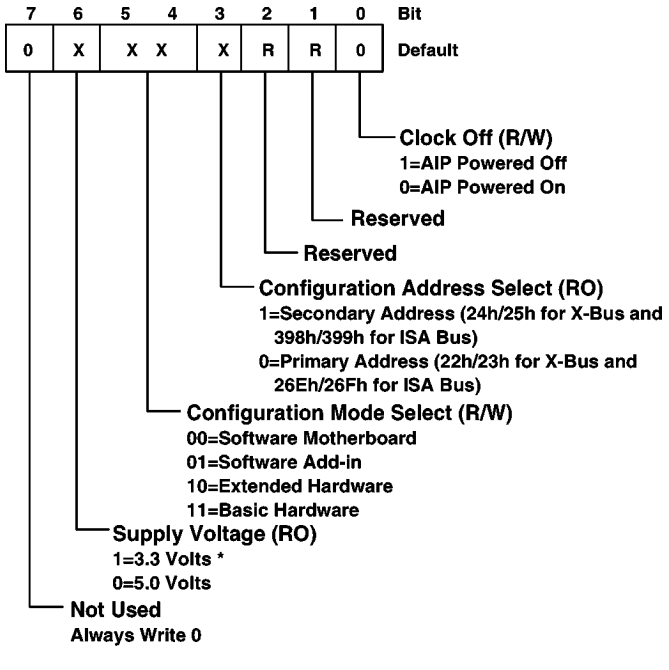

**Figure 6. AIP Revision Identification Register**

| Bit | Description |
|---|---|
| 7:4 | **STEP NUMBER:** Contains the hexadecimal representation of the device stepping. |
| 3:0 | **DASH NUMBER:** Contains the hexadecimal representation of the dash number of the device stepping. |

### 4.1.4 AIPCFG1—AIP CONFIGURATION 1 REGISTER

Index Address: 02h
Default Value: Depends upon hardware strap
Attribute: Read/Write
Size: 8 bits

The AIPCFG1 Register enables/disables master clock circuitry for power management, enables/disables access to the configuration registers, and selects the 82091AA configuration mode. This register provides status for certain hardware configuration selections—the 82091AA clock frequency, power supply voltage, and address assignment for the configuration registers (address locations of the INDEX and TARGET Registers).




**NOTES:**
*3.3V operation is available only in the 82091AA.
X = Value is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

**Figure 7. AIP Configuration 1 Register**

| Bit | Description |
|---|---|
| 7 | **NOT USED:** Always write to 0. |
| 6 | **VOLTAGE SELECT (VSEL):** This bit indicates whether 3.3V or 5V has been selected for the operating power supply voltage during hardware configuration. A 1 indicates that 3.3V is selected and a 0 indicates that 5V is selected. This bit is read only and writes have no effect.<br>**NOTE:**<br>3.3V operation is available only in the 82091AA. |
| 5:4 | **CONFIGURATION MODE SELECT (CFGMOD):** These bits indicate the configuration mode for the 82091AA. After a hard reset, these bits reflect the mode selected by hardware configuration. If configuration register access is not locked out during hardware configuration, software can change the configuration mode by writing to this field. For configuration mode details, (see Section 4.2, Hardware Configuration).<br>**Bits[5:4] Configuration Mode**<br>0 0 Software Motherboard (SWMB)<br>0 1 Software Add-in (SWAI)<br>1 0 Extended Hardware (HWE)<br>1 1 Basic Hardware (HWB) |
| 3 | **CONFIGURATION ADDRESS SELECT (CFGADS):** This read only bit indicates the address assignment for the 82091AA configuration registers as selected by hardware configuration. Hardware configuration selects between primary addresses (22h/23h and 26Eh/26Fh) and secondary addresses (24h/25h and 398h/399h) for accessing the 82091AA configuration registers. When CFGADS = 0, the primary addresses are selected and when CFGADS = 1, the secondary addresses are selected. |
| 2 | **RESERVED** |
| 1 | **RESERVED** |
| 0 | **CLOCK OFF (CLKOFF):** The CLKOFF bit is used to implement clock circuitry power management. When CLKOFF = 0, the main clock circuitry is powered on. When CLKOFF = 1, the main clock circuitry is powered off. This capability is independent of the 82091AA's powerdown state. Note that auto powerdown mode and powerdown have no effect over the power state of the clock circuitry. |

### 4.1.5 AIPCFG2—AIP CONFIGURATION 2 REGISTER

Index Address: 03h
Default Value: 0000 0RRR
Attribute: Read/Write
Size: 8 bits

This register selects the active signal level for IRQ[7:3]. The interrupt signals can be individually programmed for either active high or active low drive characteristics. The active high mode is ISA (non-share) compatible and has tri-state drive characteristic. The active low mode is EISA (sharable) compatible and has an open collector drive characteristic.

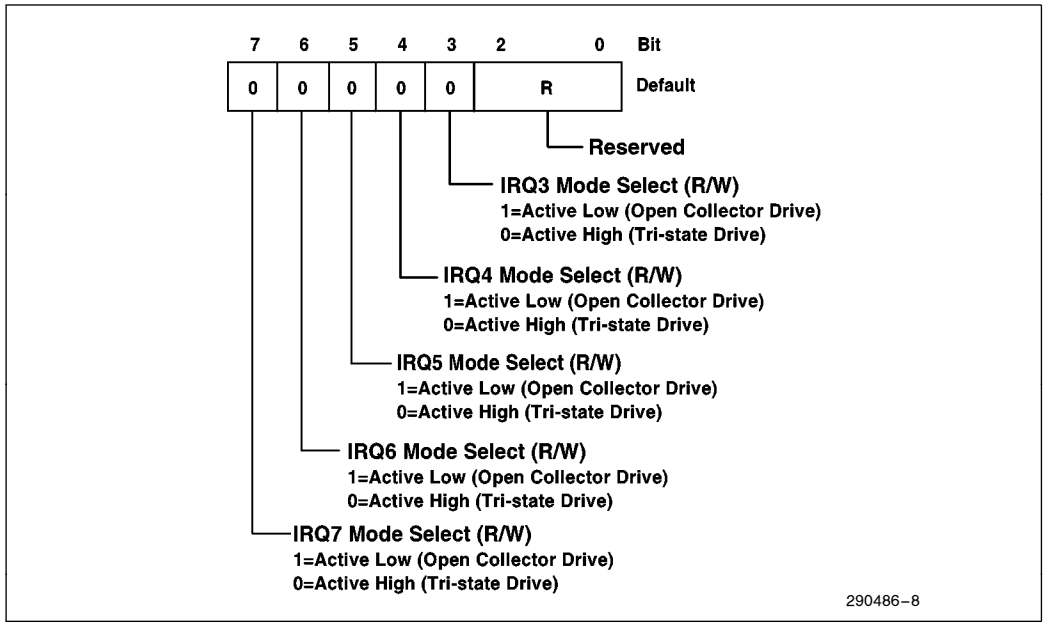


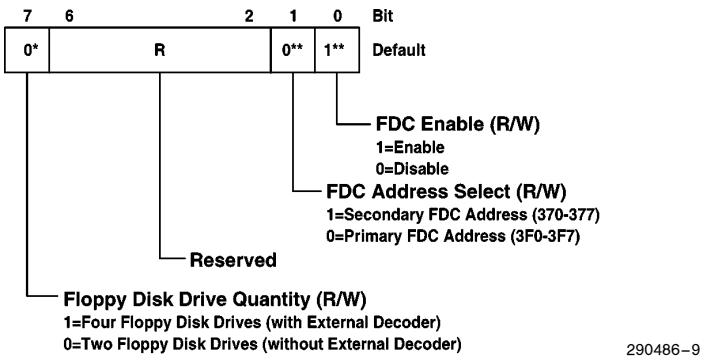

**Figure 8. AIP Configuration 2 Register**

| Bit | Description |
|---|---|
| 7 | **IRQ7 MODE SELECT (IRQ7MOD):** When IRQ7MOD = 0, IRQ7 is an active high tri-state drive signal. When IRQ7MOD = 1, IRQ7 is an active low open collector drive signal. |
| 6 | **IRQ6 MODE SELECT (IRQ6MOD):** When IRQ6MOD = 0, IRQ6 is an active high tri-state drive signal. When IRQ6MOD = 1, IRQ6 is an active low open collector drive signal. |
| 5 | **IRQ5 MODE SELECT (IRQ5MOD):** When IRQ5MOD = 0, IRQ5 is an active high tri-state drive signal. When IRQ5MOD = 1, IRQ5 is an active low open collector drive signal. |
| 4 | **IRQ4 MODE SELECT (IRQ4MOD):** When IRQ4MOD = 0, IRQ4 is an active high tri-state drive signal. When IRQ4MOD = 1, IRQ4 is an active low open collector drive signal. |
| 3 | **IRQ3 MODE SELECT (IRQ3MOD):** When IRQ3MOD = 0, IRQ3 is an active high tri-state drive signal. When IRQ3MOD = 1, IRQ3 is an active low open collector drive signal. |
| 2:0 | **RESERVED** |

### 4.1.6 FCFG1—FDC CONFIGURATION REGISTER

Index Address: 10h
Default Value: 0RRR RR01
Attribute: Read/Write
Size: 8 bits

This register selects between a 2 and 4 floppy drive system, selects primary/secondary ISA address range for the FDC, and enables/disables the FDC. All bits in this register are read/write.




**NOTES:**
*Default shown is for SWMB, SWAI, and HWB hardware configuration modes. For HWE, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.
**Default shown is for SWMB and SWAI configuration modes. For HWB and HWE configuration modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

**Figure 9. FDC Configuration Register**

| Bit | Description |
|---|---|
| 7 | **FLOPPY DISK DRIVE QUANTITY (FDDQTY):** This bit selects between two and four floppy disk drive capability. When FDDQTY = 0, the 82091AA can control two floppy disk drives directly without an external decoder. When FDDQTY = 1, the 82091AA can control four floppy disk drives with an external decoder. When FDDQTY = 1, the PDEN feature in the powerdown command is disabled. For further details, see Appendix A, FDC Four Drive Support. This bit can be configured by hardware extended configuration (HWE) at powerup. For all other hardware configuration modes (SWMB, SWAI, and HWB), the floppy disk drive quantity is not configurable by hardware strapping options and defaults to 2 drives. |
| 6:2 | **RESERVED** |
| 1 | **FLOPPY DISK CONTROLLER ADDRESS SELECT (FADS):** When FADS = 0, the primary FDC address (3F0–3F7) is selected. When FADS = 1, the secondary FDC address (370–377) is selected. For SWMB and SWAI configuration modes, the default is 0 (primary address). For HWB and HWE hardware configuration modes, the default is determined by signal pin strapping options. |
| 0 | **FLOPPY DISK CONTROLLER ENABLE (FEN):** This bit enables/disables the FDC. When FEN = 1, the FDC is enabled. When FEN = 0, the FDC module is disabled. For SWMB and SWAI configuration modes, the default is 1 (enabled). For HWB and HWE hardware configuration modes, the default is determined by signal pin strapping options. Note that, when the FDC is disabled, IRQ6 and FDDREQ are tri-stated. |

### 4.1.7 FCFG2—FDC POWER MANAGEMENT AND STATUS REGISTER

Index Address: 11h
Default Value: RRRR 0000
Attribute: Read/Write
Size: 8 bits

This register enables/disables FDC auto powerdown and can place the FDC into direct powerdown. The register also provides FDC idle status and FDC reset control.

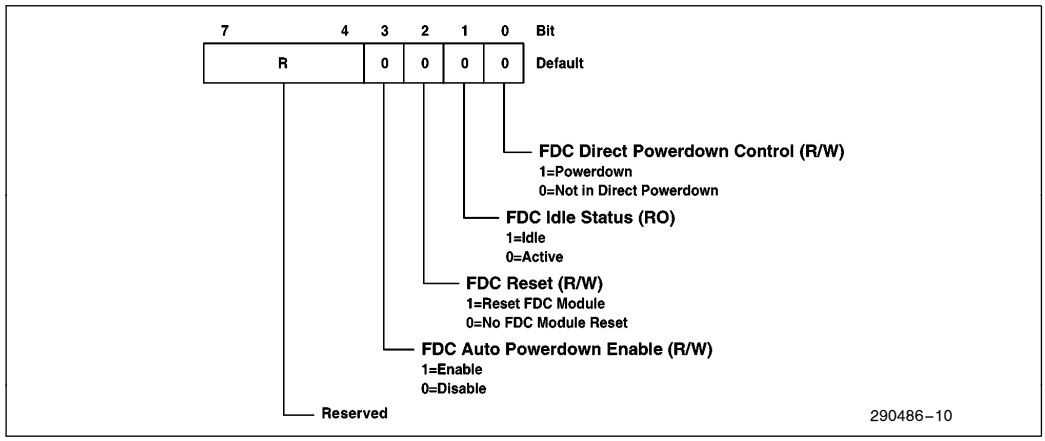


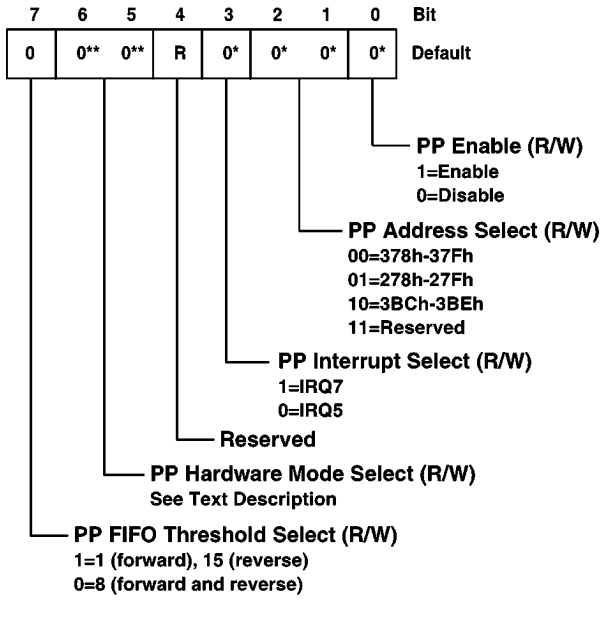

**Figure 10. FDC Power Management and Status Register**

| Bit | Description |
|---|---|
| 7:4 | **RESERVED** |
| 3 | **FLOPPY DISK AUTO POWERDOWN ENABLE (FAPDN):** This bit is used to enable/disable auto powerdown for the FDC. When FAPDN = 1, the FDC will enter auto powerdown when the required conditions are met. When FAPDN = 0, FDC auto powerdown is disabled. |
| 2 | **FLOPPY DISK CONTROLLER RESET (FRESET):** FRESET is a reset for the FDC. When FRESET = 1, the FDC is reset (i.e., all programming and current state information is lost). FRESET = 1 has the same affect on the FDC as a hard reset (asserting the RSTDRV signal). When resetting the FDC via this configuration bit, the software must toggle this bit and ensure the reset active time (FRESET = 1) of 1.13 $\mu$s minimum is met. |
| 1 | **FLOPPY DISK CONTROLLER IDLE STATUS (FIDLE):** When the FDC is in the idle state, this bit is set to 1 by the 82091AA hardware. In the idle state the FDC's Main Status Register (MSR) = 80h, IRQ6 = inactive, and the head unload timer has expired. When the FDC exits its idle state, this bit is set to 0. This bit is read only. |
| 0 | **FLOPPY DISK CONTROLLER POWERDOWN (FDPDN):** When FDPDN is set to 1, the FDC is placed in direct powerdown. Once in powerdown the following procedure should be used to bring the FDC out of powerdown:<br>• Write this bit low<br>• Apply a hardware reset (via bit 2 of this register) or a software reset (via either bit 2 of the FDC's DOR or bit 7 of the FDC's DSR).<br>**NOTE:**<br>A hard reset via the RSTDRV pin also removes the FDC powerdown. |

### 4.1.8 PCFG1—PARALLEL PORT CONFIGURATION REGISTER

Index Address: 20h
Default Value: 000R 0000
Attribute: Read/Write
Size: 8 bits

The PCFG1 Register enables/disables the parallel port, selects the parallel port address, and selects the parallel port interrupt. This register also selects the hardware operation mode for the parallel port.




**NOTES:**
*Default shown is for SWMB and SWAI configuration modes. For HWB and HWE modes, the default is determined by hardware configuration options as described in Section 4.2, Hardware Configuration.
**Default shown is for SWMB, SWAI, and HWB configuration modes. For HWE mode, the default is determined by hardware configuration options as described in Section 4.2, Hardware Configuration.

**Figure 11. Parallel Port Configuration Register**

| Bit | Description |
|---|---|
| 7 | **PARALLEL PORT FIFO THRESHOLD SELECT (PTHRSEL):** This bit controls the FIFO threshold and only affects parallel port operations when the parallel port is in ECP mode or ISA-Compatible FIFO mode. When PTHRSEL = 1, the FIFO threshhold is 1 in the forward direction and 15 in the reverse direction. When PTHRSEL = 0, the FIFO threshold is 8 in both directions. This bit can only be programmed when the parallel port is in ISA-Compatible or PS/2-Compatible mode. These modes can be selected via bits[6:5] of this register or the ECP Extended Control Register (ECR).<br>**NOTE:**<br>In the reverse direction, a threshold of 15/8 means that a request (DMA or Interrupt is enabled) is generated when 15/8 bytes are in the FIFO. In the forward direction, a threshold of 1/8 means that a request is generated when 1/8 byte locations are available. |
| 6:5 | **PARALLEL PORT HARDWARE MODE SELECT (PPHMOD):** This field selects the parallel port hardware mode. The ISA-Compatible mode is for compatibility and nibble mode peripheral interface protocols. The PS/2-Compatible mode is for the byte mode peripheral interface protocol. The EPP and ECP modes are for the EPP and ECP mode peripheral interface protocols, respectively. This field can be configured by strapping options at powerup for hardware extended configuration (HWE) mode only. For all other hardware configuration modes (SWMB, SWAI, and HWB), the default is 00 (ISA-Compatible).<br>**Bits [6:5]** — **Read** — **Write**<br>0 0 — ISA-Compatible — ISA-Compatible(1)<br>0 1 — PS/2-Compatible — PS/2-Compatible(1)<br>1 0 — EPP — EPP(1, 3)<br>1 1 — ECP(2) — Reserved; do not write(2)<br>**NOTES:**<br>1. ISA-Compatible, PS/2-Compatible, and EPP modes are selected via this field or hardware configuration. In addition, ISA-Compatible and PS/2-Compatible modes can be selected via the ECP Extended Control Register (ECR). When the ECR is programmed for one of these two modes (ECR[7:5] = 000, 001), this field is updated to match the selected mode.<br>2. ECP Mode can not be entered by programming this field. ECP Mode can only be selected through the ECR. When the ECR is programmed for ECP mode, the 82091AA sets this field to 11.<br>3. Parallel port interface signals controlled by the PCON Register (SELECTIN#, INIT#, AUTOFD#, and STROBE#) should be negated before entering EPP mode. |
| 4 | **RESERVED** |
| 3 | **PARALLEL PORT IRQ SELECT (PIRQSEL):** When PIRQSEL = 1, IRQ7 is selected as the parallel port interrupt. When PIRQSEL = 0, IRQ5 is selected as the parallel port interrupt. This field can be configured by strapping options at powerup for HWB and HWE modes only. For all other hardware configuration modes (SWMB and SWAI), the default is 0 (IRQ5). |

| Bit | Description |
|---|---|
| 2:1 | **PARALLEL PORT ADDRESS SELECT (PADS):** This field selects the address for the parallel port as follows:<br>**Bits[2:1] Address Parallel Port Hardware Mode**<br>0 0 378–37F All<br>0 1 278–27F All<br>1 0 3BC–3BE All except EPP<br>1 1 Reserved None, do not write<br>This field can be configured by strapping options at powerup for HWB and HWE modes only. For all other hardware configuration modes (SWMB and SWAI), the default is 00 (378h–37Fh). Note that the SWMB and SWAI default settings for PIRQSEL (bit 3) and PADS (bits[2,1]) do not match a standard PC/AT* combination for address assignment and interrupt setting. However, for SWMB and SWAI, the parallel port defaults to a disabled condition and this register must be programmed to enable the parallel port (i.e., bit 0 set to 1). At this time, the selections for interrupt and address assignments should be made. |
| 0 | **PARALLEL PORT ENABLE (PEN):** When PEN = 0, the parallel port is disabled. When PEN = 1, the parallel port is enabled. This bit can be configured by hardware strapping options at powerup for HWB and HWE modes only. For all other hardware configuration modes (SWMB and SWAI), the default is 0 (disabled). Note that when the parallel port is disabled, IRQ[7,5] and PPDREQ are tri-stated. |

### 4.1.9 PCFG2—PARALLEL PORT POWER MANAGEMENT AND STATUS REGISTER

Index Address: 21h
Default Value: RR0R 0000
Attribute: Read/Write
Size: 8 bits

This register enables/disables parallel port auto powerdown and can place the parallel port into a powerdown mode directly. The register also provides parallel port idle status, resets the parallel port, and reports FIFO underrun or overrun errors.

*Other brands and names are the property of their respective owners.

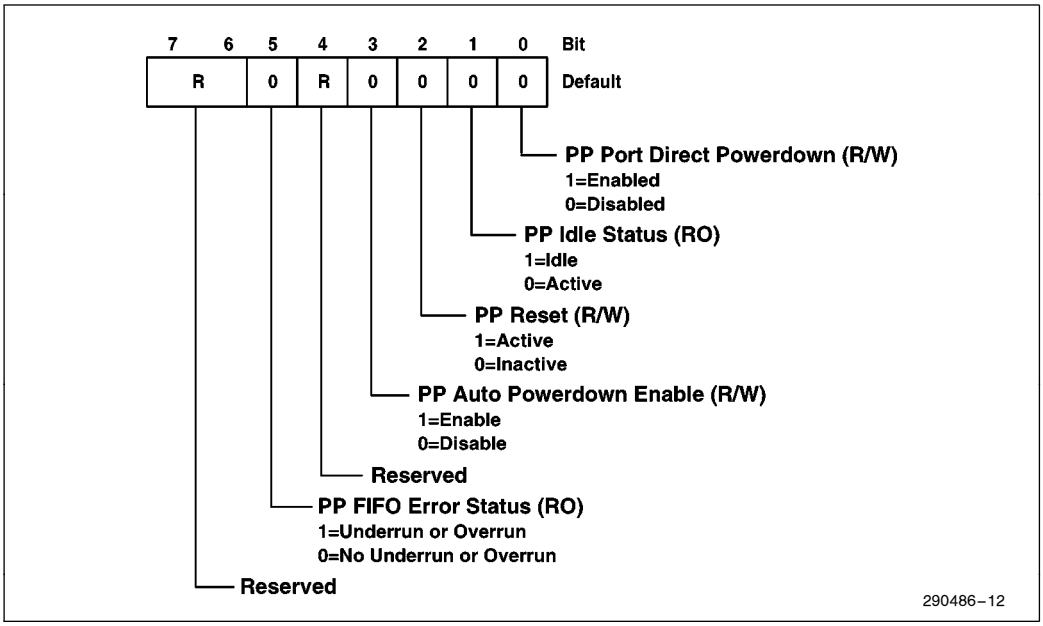



**Figure 12. Parallel Port Power Management and Status Register**

| Bit | Description |
|---|---|
| 7:6 | **RESERVED** |
| 5 | **PARALLEL PORT FIFO ERROR STATUS (PFERR):** When PFERR = 1, a FIFO underrun or overrun condition has occurred. This bit is read only. Setting PRESET to 1 clears this bit to 0. |
| 4 | **RESERVED** |
| 3 | **PARALLEL PORT AUTO POWERDOWN ENABLE (PAPDN):** When PAPDN = 1, the parallel port can enter auto powerdown if the required auto powerdown conditions are met. When PAPDN = 0, auto powerdown is disabled. |
| 2 | **PARALLEL PORT RESET (PRESET):** When PRESET is set to 1, the parallel port is reset (i.e., all programming and current state information is lost). This is the same state the module would be in after a hard reset (RSTDRV asserted) to the 82091AA. When resetting the parallel port via this configuration bit, the software must toggle this bit and ensure the reset active time (PRESET = 1) of 1.13 $\mu$s minimum is met. |
| 1 | **PARALLEL PORT IDLE STATUS (PIDLE):** This bit reflects the idle state of the parallel port. When the parallel port is in an idle state (i.e., when the same conditions are met that apply to entering auto powerdown) the 82091AA sets this bit to 1. The parallel port idle state is defined as the FIFO empty and no activity on the parallel port interface. This bit is read only. |
| 0 | **PARALLEL PORT DIRECT POWERDOWN (PDPDN):** When PDPDN is set to 1, the parallel port enters direct powerdown. When PDPDN is set to 0, the parallel port is not in direct powerdown. Note that a parallel port module reset (PRESET bit in this register) also brings the parallel port out of the direct powerdown state. |

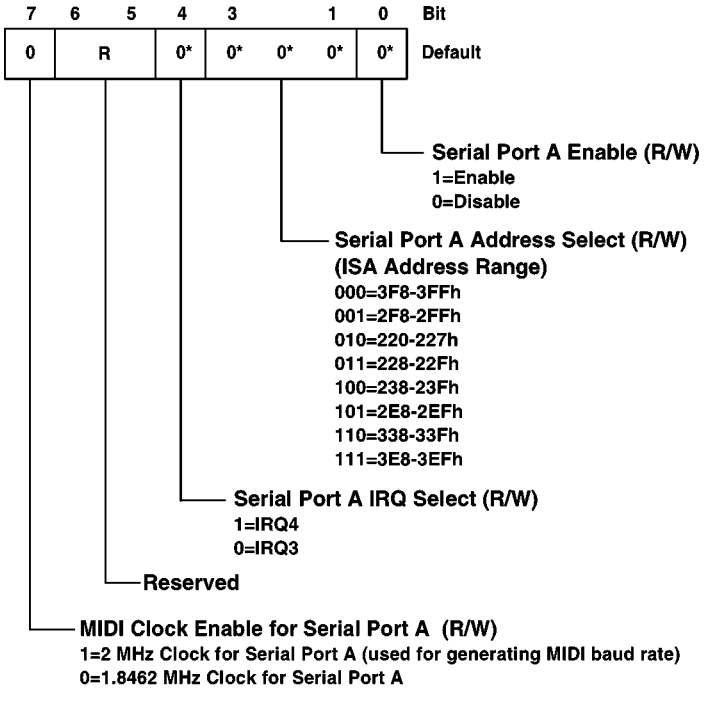
### 4.1.10 SACFG1—SERIAL PORT A CONFIGURATION REGISTER

Index Address: 30h
Default Value: 0RR0 0000
Attribute: Read/Write
Size: 8 bits

The SACFG1 register enables/disables Serial Port A, selects the Serial Port A address range, and selects between IRQ3 and IRQ4 as the Serial Port A interrupt. This register also selects the appropriate clock frequency for use with MIDI.

**NOTES:**

1. Through programming of this register and the SBCFG1 Register, the 82091AA permits serial ports A and B to be configured for the same interrupt assignment. However, software must take care in responding to interrupts correctly.
2. It is possible to enable and assign both serial ports to the same address through software. In this configuration, the 82091AA disables serial port B, but does not set serial port B into it's powerdown condition. Although this is a safe configuration for the 82091AA, it is not power conservative and is not recommended.




**NOTE:**
*Default shown is for SWMB and SWAI hardware configuration modes. For HWB and HWE modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

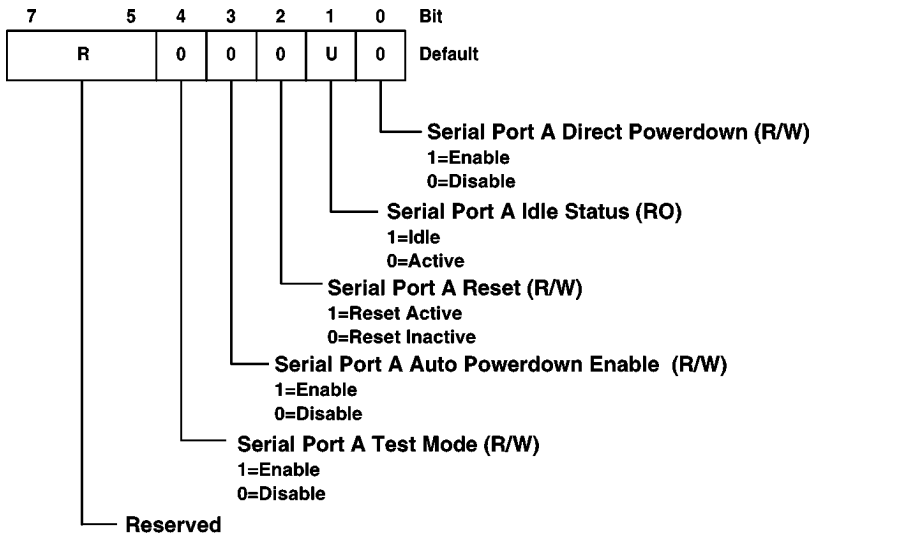
**Figure 13. Serial Port A Configuration Register**

| Bit | Description |
|---|---|
| 7 | **MIDI CLOCK FOR SERIAL PORT A ENABLE (SAMIDI):** When SAMIDI = 1, the clock into Serial Port A is changed from 1.8462 MHz–2 MHz. The 2 MHz clock is needed to generate the MIDI baud rate. When SAMIDI = 0, the clock frequency is 1.8462 MHz. |
| 6:5 | **RESERVED** |
| 4 | **SERIAL PORT A IRQ SELECT (SAIRQSEL):** When SAIRQSEL = 0, IRQ3 is selected for the Serial Port A interrupt. When SAIRQSEL = 1, IRQ4 is selected for the Serial Port A interrupt. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI hardware configuration modes, the default is 0 (IRQ3). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation. |
| 3:1 | **SERIAL PORT A ADDRESS SELECT (SAADS):** This field selects the ISA address range for Serial Port A as follows:<br>**Bits[3:1] ISA Address Range**<br>0 0 0 3F8–3FFh<br>0 0 1 2F8–2FFh<br>0 1 0 220–227h<br>0 1 1 228–22Fh<br>1 0 0 238–23Fh<br>1 0 1 2E8–2EFh<br>1 1 0 338–33Fh<br>1 1 1 3E8–3EFh<br>This field can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI hardware configuration modes, the default is 000 (3F8–3FFh). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation. |
| 0 | **SERIAL PORT A ENABLE (SAEN):** When SAEN = 1, Serial Port A is enabled. When SAEN = 0, Serial Port A is disabled. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI hardware configuration modes, the default is 0 (disabled). |

### 4.1.11 SACFG2—SERIAL PORT A POWER MANAGEMENT AND STATUS REGISTER

Index Address: 31h
Default Value: RRR0 00U0
Attribute: Read/Write
Size: 8 bits

This register enables/disables the Serial Port A module auto powerdown and can place the module into a direct powerdown mode. The register also provides Serial Port A idle status, resets the Serial Port A module, and places Serial Port A into test mode.




**NOTE:**
U = Undefined

**Figure 14. Serial Port A Power Management and Status Register**

| Bit | Description |
|---|---|
| 7:5 | **RESERVED** |
| 4 | **SERIAL PORT A TEST MODE (SATEST):** The serial port test mode provides user access to the output of the baud out generator. When SATEST = 1 (and the DLAB bit is 1 in the LCR), the Serial Port A test mode is enabled and the baud rate clock is output on the SOUTA pin (Figure 15). When SATEST = 0, the Serial Port A test mode is disabled. |

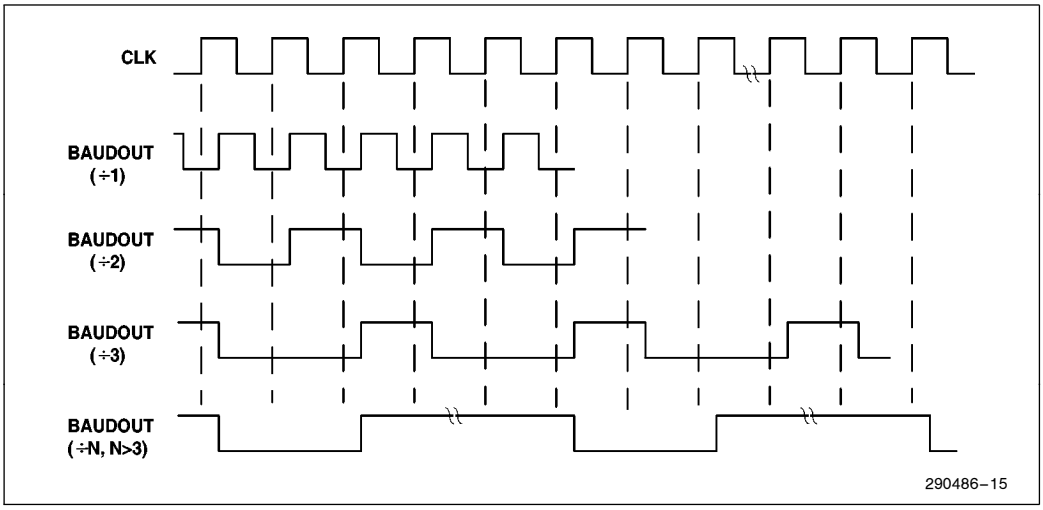



**Figure 15. Test Mode Output (SOUTA and SOUTB)**

| Bit | Description |
|---|---|
| 3 | **SERIAL PORT A AUTO POWERDOWN ENABLE (SAAPDN):** This bit enables/disables auto powerdown. When SAAPDN = 1, Serial Port A can enter auto powerdown if the required conditions are met. The required conditions are that the transmit and receive FIFOs are empty and the timeout counter has expired. When SAAPDN = 0, auto powerdown is disabled. |
| 2 | **SERIAL PORT A RESET (SARESET):** When SARESET = 1, the Serial Port A module is reset (i.e. all programming and current state information is lost). This is the same state the module would be in after a hard reset (RSTDRV asserted). When resetting the serial port via this configuration bit, the software must toggle this bit and ensure the reset active time (SARESET = 1) of 1.13 $\mu$s minimum is met. |
| 1 | **SERIAL PORT A IDLE STATUS (SAIDLE):** When Serial Port A is in an idle state the 82091AA sets this bit to 1. Serial Port A is in the idle state when the transmit and receive FIFOs are empty and the timeout counter has expired. Note that these are the same conditions that apply to entering auto powerdown. When serial port A is not in an idle state, the 82091AA sets this bit to 0. Direct powerdown does not affect this bit and in auto powerdown SAIDLE is only set to a 1 if the receive and transmit FIFOs are empty. This bit is read only.<br>During a hard reset (RSTDRV asserted), The 82091AA sets SAIDLE to 0. However, because the serial port is typically initialized by software before the idle conditions are met, the default state is shown as undefined. |
| 0 | **SERIAL PORT A DIRECT POWERDOWN (SADPDN):** When SADPDN = 1, Serial Port A is placed in direct powerdown mode. Setting this bit to 0 brings Serial Port A out of direct powerdown mode. Setting bit 2 (SARESET) of this register to 1 will also bring Serial Port A out of the direct powerdown mode.<br>**NOTE:**<br>Direct powerdown resets the receiver and transmitter portions of the serial port including the receive and transmit FIFOs. To ensure that the resetting of the FIFOs does not cause data loss, the SAIDLE bit should be 1 before placing the serial port into direct powerdown. |

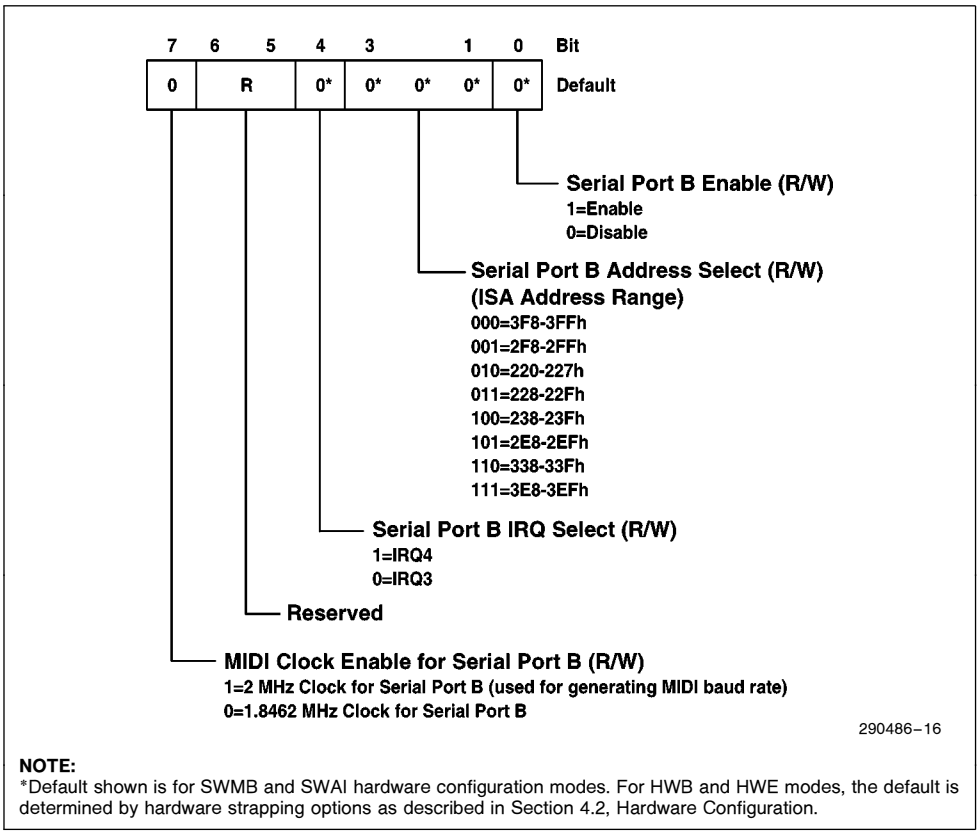
### 4.1.12 SBCFG1—SERIAL PORT B CONFIGURATION REGISTER

Index Address: 40h
Default Value: 0RR0 0000
Attribute: Read/Write
Size: 8 bits

The SBCFG1 register enables/disables Serial Port B, selects the Serial Port B address range, and selects between IRQ3 and IRQ4 as the Serial Port B interrupt. This register also selects the appropriate clock frequency for use with MIDI.

**NOTES:**

1. Through programming of this register and the SBCFG1 Register, the 82091AA permits serial ports A and B to be configured for the same interrupt assignment. However, software must take care in responding to interrupts correctly.
2. It is possible to enable and assign both serial ports to the same address through software. In this configuration, the 82091AA disables serial port B, but does not set serial port B into it's powerdown condition. Although this is a safe configuration for the 82091AA, it is not power conservative and is not recommended.




**NOTE:**
*Default shown is for SWMB and SWAI hardware configuration modes. For HWB and HWE modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

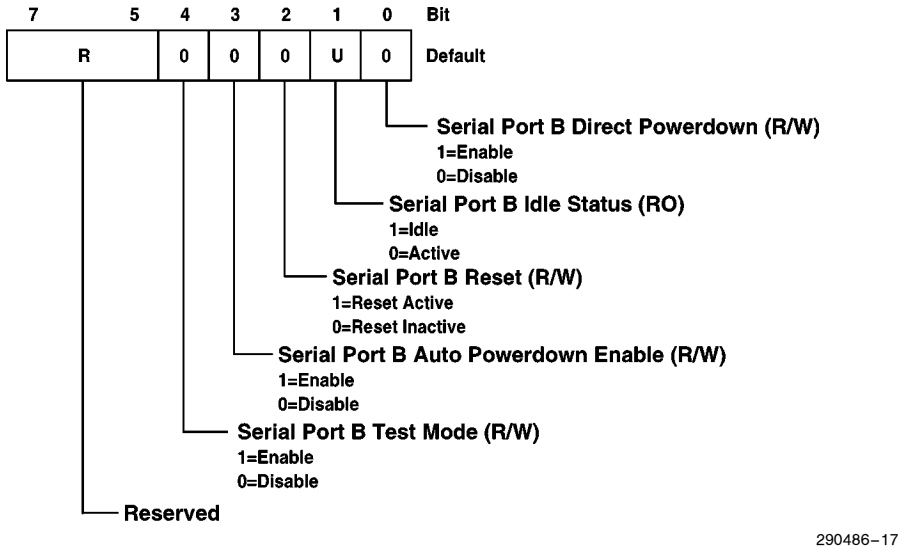
**Figure 16. Serial Port B Configuration Register**

| Bit | Description |
|---|---|
| 7 | **MIDI CLOCK FOR SERIAL PORT B ENABLE (SBMIDI):** When SBMIDI = 1, the clock into Serial Port B is changed from 1.8462 MHz to 2 MHz. The 2 MHz clock is needed to generate the MIDI baud rate. When SBMIDI = 0, the clock frequency is 1.8462 MHz. The default value is 0. |
| 6:4 | **RESERVED** |
| 4 | **SERIAL PORT B IRQ SELECT (SBIRQSEL):** When SBIRQSEL = 0, IRQ3 is selected for the Serial Port B interrupt. When SBIRQSEL = 1, IRQ4 is selected for the Serial Port B interrupt. The default value is 0. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI configuration modes, the default is 0 (IRQ3). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation. |
| 3:1 | **SERIAL PORT B ADDRESS SELECT (SBADS):** This field selects the ISA address range for Serial Port B as follows:<br>**Bits[3:1] ISA Address Range**<br>0 0 0 3F8–3FFh<br>0 0 1 2F8–2FFh<br>0 1 0 220–227h<br>0 1 1 228–22Fh<br>1 0 0 238–23Fh<br>1 0 1 2E8–2EFh<br>1 1 0 338–33Fh<br>1 1 1 3E8–3EFh<br>This field can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI configuration modes, the default is 000 (3F8–3FFh). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation. |
| 0 | **SERIAL PORT B ENABLE (SBEN):** When SBEN = 1, Serial Port B is enabled. When SAEN = 0, Serial Port B is disabled. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI configuration modes, the default is 0 (disabled). |

### 4.1.13 SBCFG2—SERIAL PORT B POWER MANAGEMENT AND STATUS REGISTER

Index Address: 41h
Default Value: RRR0 00U0
Attribute: Read/Write
Size: 8 bits

This register enables/disables the Serial Port B module auto powerdown and can place the module into a powerdown mode directly. The register also provides Serial Port B idle status, resets the Serial Port B module, and enables/disables Serial Port B test mode.




**NOTE:**
U = Undefined

**Figure 17. Serial Port B Power Management and Status Register**

| Bit | Description |
|---|---|
| 7:5 | **RESERVED** |
| 4 | **SERIAL PORT B TEST MODE (SBTEST):** The serial port test mode provides user access to the output of the baud out generator. When SBTEST = 1 (and the DLAB bit is 1 in the LCR), the Serial Port B test mode is enabled and the baud rate clock is output on the SOUTB pin (Figure 15). When SBTEST = 0, the Serial Port B test mode is disabled. |
| 3 | **SERIAL PORT B AUTO POWERDOWN ENABLE (SBAPDN):** This bit enables/disables auto powerdown. When SBAPDN = 1, Serial Port B can enter auto powerdown if the required conditions are met. The required conditions are that the transmit and receive FIFOs are empty and the timeout counter has expired. When SBAPDN = 0, auto powerdown is disabled. |
| 2 | **SERIAL PORT B RESET (SBRESET):** When SBRESET = 1, Serial Port B is reset (i.e., all programming and current state information is lost). This is the same state the module would be in after a hard reset (RSTDRV asserted). When resetting the serial port via this configuration bit, the software must toggle this bit and ensure the reset active time (SBRESET = 1) of 1.13 $\mu$s minimum is met. |
| 1 | **SERIAL PORT B IDLE STATUS (SBIDLE):** When Serial Port B is in an idle state the 82091AA sets this bit to 1. Serial Port B is in the idle state when the transmit and receive FIFOs are empty and the timeout counter has expired. Note that these are the same conditions that apply to entering auto powerdown. When serial port B is not in an idle state, the 82091AA sets this bit to 0. Direct powerdown does not affect this bit and in auto powerdown, this bit is only set to a 1 if the receive and transmit FIFOs are empty. This bit is read only.<br>During a hard reset (RSTDRV asserted), the 82091AA sets this bit to 0. However, because the serial port is typically initialized by software before the idle conditions are met, the defaullt state is shown as undefined. |
| 0 | **SERIAL PORT B DIRECT POWERDOWN (SBDPDN):** When SBDPDN = 1, Serial Port B is placed in powerdown mode. Setting this bit to 0 brings the module out of direct powerdown mode. Setting bit 2 (SBRESET) of this register to 1 will also bring Serial Port B out of the direct powerdown mode.<br>**NOTE:**<br>Direct powerdown resets the receiver and transmitter portions of the serial port including the receive and transmit FIFOs. To ensure that the resetting of the FIFOs does not cause data loss, the SBIDLE bit should be 1 before placing the serial port into direct powerdown. |

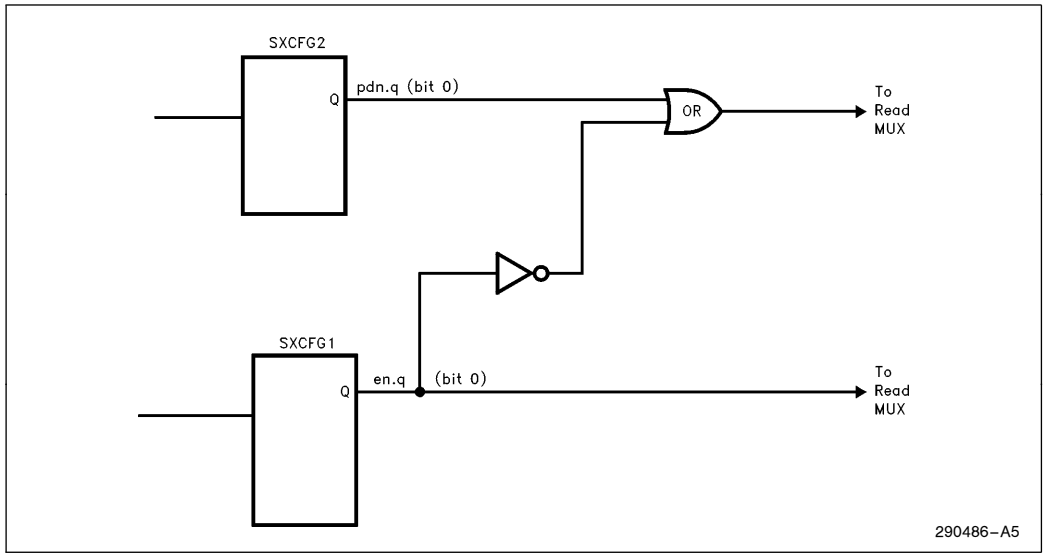
#### 4.1.13.1 Serial Port A/B Configuration Register's SxEN and SxDPDN Bits

The bits which enable the serial ports (bit 0 in both the SACFG1 and SBCFG1 registers) and the bits which provide for serial port direct powerdown (bit 0 in both the SACFG2 and SBCFG2 registers) are not mutually exclusive. The partial circuit and truth table for the two bits shows that it is possible to enable serial port A using SACFG1, for example, yet still read the serial port A SACFG2 direct powerdown bit as a ''1''.

When the SxCFG1 register bit 0 (serial port x enable) is written as a ''1'' (enable), the SxCFG2 register bit 0 (serial port x powerdown) does not change from a ''1'' (powerdown enable) to a ''0'' (powerdown disable). As can be seen in the circuit diagram, the READ activity does not see the register directly. Instead, a MUXed output is seen by the READ activity. The truth table for the two bits shows it is possible to enable a serial port yet still read the powerdown bit for that same port as a ''1'', or enabled.

**Truth Table for Reading the Enable/Powerdown Bit Status**

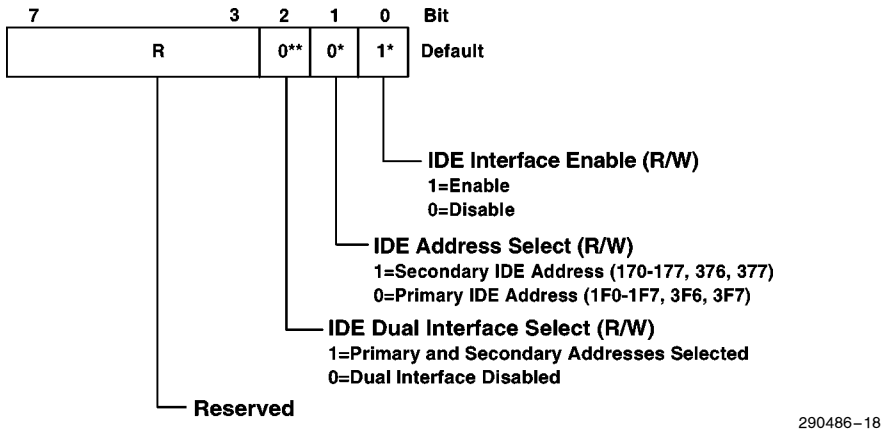
| Write Activity | | Read Activity | |
|---|---|---|---|
| **SxCFG1 Enable** | **SxCFG2 Powerdown** | **SxCFG1 Enable** | **SxCFG2 Powerdown** |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 |




### 4.1.14 IDECFG—IDE CONFIGURATION REGISTER

Index Address: 50h
Default Value: RRRR R001
Attribute: Read/Write
Size: 8 bits

The IDECFG Register sets up the 82091AA IDE interface. This register enables the IDE interface and selects the address for accessing the IDE.




**NOTES:**
* Default shown is for SWMB and SWAI configuration modes. For HWB and HWE hardware configuration modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.
** Not hardware configurable.

**Figure 18. IDE Configuration Register**

| Bit | Description |
|---|---|
| 7:3 | **RESERVED** |
| 2 | **IDE DUAL SELECT (IDUAL):** When IDUAL = 0, the IDE address selection is determined by the IADS bit. When IDUAL = 1, both the primary and secondary IDE addresses are selected and the setting of the IADS bit does not affect IDE address selection. |
| 1 | **IDE ADDRESS SELECT (IADS):** When IADS = 0, the primary IDE address is selected (1F0h–1F7h, 3F6h, 3F7h ). When IADS = 1, the secondary IDE address is selected (1F0h–1F7h, 376h, 377h). For all hardware configuration modes (SWMB, SWAI, HWB, and HWE), the default is determined by signal pin strapping options. |
| 0 | **IDE INTERFACE ENABLE (IEN):** When IEN = 0, the IDE interface is disabled (i.e., the IDE chip selects (IDECS[1:0]), DEN#, and HEN# are negated (remain inactive) for accesses to the IDE primary and secondary addresses). When IEN = 1, the IDE interface is enabled. For all hardware configuration modes (SWMB, SWAI, HWB, and HWE), the default is determined by signal pin strapping options. |

## 4.2 Hardware Configuration

Hardware configuration provides a mechanism for configuring certain 82091AA operations at powerup. Four hardware configuration modes provide different levels of configuration depending on the type of application and the degree of hardware/software configuration desired. The hardware configuration modes are:

- Software Motherboard (SWMB)
- Software Add-In (SWAI)
- Hardware Extended (HWE)
- Hardware Basic (HWB)

These modes support a variety of system implementations. For example, with Hardware Basic (HWB) and Hardware Extended (HWE) modes, an extensive set of 82091AA configuration options are available for setting up the 82091AA at powerup. This permits the 82091AA to be used in systems without 82091AA software drivers. For many of these systems, access to the 82091AA configuration registers may not be necessary. As such, access to these registers can be disabled via hardware configuration. This option could be used to prevent software from inadvertently re-configuring the 82091AA.

**NOTE:**
If the 82091AA is configured in HWB or HWE configuration mode at powerup, and reconfiguration with software is desired, the 82091AA configuration mode must first be changed to SWAI configuration mode by writing the AIPCFG1 register. The 82091AA can then remain in SWAI configuration mode to accomodate software programmable configuration changes as desired.

Software Motherboard (SWMB) and Software Add-In (SWAI) modes provide a minimum hardware configuration in systems where software/firmware drivers are used for configuration. Because access to the 82091AA configuration registers after powerup/hardware configuration is needed, the SWMB and SWAI modes do not provide disabling access to these registers (i.e., the strapping of the HEN# signal has no effect).

The desired hardware configuration mode and options within the mode are selected by strapping certain 82091AA signal pins at powerup. These signal pins are sampled when the 82091AA receives a hard reset (via RSTDRV). This section describes how to select the configuration mode and options within the mode. The section also provides example hardware connection diagrams for the different modes.

### 4.2.1 SELECTING THE HARDWARE CONFIGURATION MODE

During powerup or a hard reset, four signal pins (DEN#, PPDIR/GCS#, DTRA, and HEN#) select the hardware configuration mode, I/O address assignment for the 82091AA configuration registers, and whether software access to these configuration registers is permitted. The following mnemonics and signal pins are assigned for these functions:

CFGMOD[1,0] **Hardware Configuration Mode.** The 82091AA samples the CFGMOD0 (DEN#) and CFGMOD1 (PPDIR/GCS#) signal pins to select one of the four hardware configuration modes as shown in Table 6.

CFGADS **82091AA Configuration Register Address Assignment**. The 82091AA samples the DTRA# signal (CFGADS function) to determine the address assignment of the 82091AA configuration registers as shown in Table 6. CFGADS works in conjunction with CFGDIS. Note that the 82091AA configuration register address assignment for Hardware Basic mode is not selectable.

CFGDIS **82091AA Configuration Register Disable**. The 82091AA samples CFGDIS (HEN# signal) to enable/disable access to the 82091AA configuration registers as shown in Table 6. Note that CFGDIS only affects the HWE and HWB modes.

**NOTE:**
For Extended Hardware Configuration, the time immediately following the RSTDRV pulse is required to complete the configuration time. If IORC#/IOWC# are asserted during this time, IOCHRDY will be negated (wait-states inserted) until the 82091AA configuration time expires.

**Table 6. AIP Configuration Mode Register Address Assignment**

| CFGDIS (HEN#) | CFGMOD1 (PPDIR) | CFGMOD0 (DEN#) | CFGADS (DTRA#) | Configuration Mode | Configuration Register ISA Address (INDEX/TARGET) |
|---|---|---|---|---|---|
| X | 0 | 0 | 0 | SWMB | 22h/23h |
| X | 0 | 0 | 1 | SWMB | 24h/25h |
| X | 0 | 1 | 0 | SWAI | 26Eh/26Fh |
| X | 0 | 1 | 1 | SWAI | 398h/399h |
| 0 | 1 | 0 | 0 | HWE | 26Eh/26Fh |
| 0 | 1 | 0 | 1 | HWE | 398h/399h |
| 1 | 1 | 0 | X | HWE | Access Disabled |
| 0 | 1 | 1 | n/a | HWB | 398h/399h |
| 1 | 1 | 1 | n/a | HWB | Access Disabled |

### 4.2.2 SELECTING HARDWARE CONFIGURATION MODE OPTIONS

Within each hardware configuration mode, a number of options are available. For the HWB and HWE hardware configuration modes, the user can enable/disable the floppy disk controller and the IDE interface via the IDE chip select pins (see Table 7). If enabled, these signal pins also select the address assignment. For SWMB and SWAI configuration modes, these signal pins have no effect.

**Table 7. FDC and IDE Enable/Disable**

| DDCFG1 (IDECS1#) | DDCFG0 (IDECS0#) | Floppy Disk Controller | IDE |
|---|---|---|---|
| 0 | 0 | Disable | Disable |
| 0 | 1 | Enabled (3F6–3F7h; Primary) | Disable |
| 1 | 0 | Enabled (370–377h; Secondary) | Enabled (170–177h; Secondary) |
| 1 | 1 | Enabled (3F6–3F7h; Primary) | Enabled (1F0–1F7h; Primary) |

The 82091AA provides additional hardware configuration options through the SOUTA, SOUTB, RTSA#, RTSB#, DTRA#, and DTRB# signal pins as shown in Table 8. In the case of the Hardware Extended Mode, the 82091AA samples the signal pins at two different times (once for HWEa options and again for HWEb options). The timing for signal sampling is discussed in Section 4.2.3, Hardware Configuration Timing Relationships. The options provide configuration of the serial ports, floppy disk controller, parallel port, IDE interface, 82091AA operating power supply voltage, 82091AA clock frequency, and address assignment for the 82091AA configuration registers. Table 8 provides a matrix of the options available for each hardware configuration mode. The configuration options are selected as shown in Table 8 through Table 14.

Note that for the SWAI and SWMB modes, the selection of the operating frequency (CLKSEL), power supply voltage level (VSEL), and 82091AA configuration register address assignment (CFGADS) are the only hardware configuration options (Table 8). In these modes, software/firmware provides the remainder of the 82091AA configuration by programming the 82091AA configuration registers (see Section 4.1, Configuration Registers). For the SWAI and SWMB modes, the 82091AA modules are placed in the following states after powerup or a hard reset:

- Serial ports disabled
- Parallel port disabled
- FDC enabled for two drives (primary address)
- IDE enabled (primary address)

**Table 8. Hardware Configuration Mode Option Matrix**

| Signal Name | Basic Hardware Configuration | Extended Hardware Configuration | | Software Add-In Configuration | Software MotherBoard Configuration |
|---|---|---|---|---|---|
| | HWB | HWEa | HWEb | SWAI | SWMB |
| SOUTA | SPCFG0 | CLKSEL(3) | SPCFG0 | CLKSEL(3) | CLKSEL(3) |
| SOUTB | SPCFG1 | PPMOD0 | SPCFG1 | — | — |
| RTSA# | SPCFG2 | PPMOD1 | SPCFG2 | — | — |
| RTSB# | SPCFG3 | FDDQTY | SPCFG3 | — | — |
| DTRA# | PPCFG0 | CFGADS | PPCFG0 | CFGADS | CDGADS |
| DTRB# | PPCFG1 | VSEL | PPCFG1 | VSEL | VSEL |

**NOTES:**
1. HWEa and HWEb reference the switching banks shown in Figure 22.
2. The following mnemonics are used in the table: SPCFGx = serial port configuration, PPCFGx = parallel port configuration, CLKSEL = clock select, PPMODx = parallel port hardware mode, FDDQTY = floppy disk drive quantity, VSEL = power supply voltage select, CFGADS = 82091AA configuration register address assignment select.
3. Always tie this signal low with a 10K resistor.

**Table 9. Serial Port Address and Interrupt Assignments**

| SPCFG3 (RTSB#) | SPCFG2 (RTSA#) | SPCFG1 (SOUTB) | SPCFG0 (SOUTA) | Serial Port B | | Serial Port A | |
|---|---|---|---|---|---|---|---|
| | | | | Address Assignment | Interrupt Assignment | Address Assignment | Interrupt Assignment |
| 0 | 0 | 0 | 0 | Disable | — | Disable | — |
| 0 | 0 | 0 | 1 | Disable | — | 3F8–3FFh | IRQ4 |
| 0 | 0 | 1 | 0 | Disable | — | 2F8–2FFh | IRQ3 |
| 0 | 0 | 1 | 1 | Disable | — | 3E8–3EFh | IRQ4 |
| 0 | 1 | 0 | 0 | 3F8–3FFh | IRQ4 | Disable | — |
| 0 | 1 | 0 | 1 | 3E8–3EFh | IRQ4 | Disable | — |
| 0 | 1 | 1 | 0 | 3F8–3FFh | IRQ4 | 2F8–2FFh | IRQ3 |
| 0 | 1 | 1 | 1 | 3F8–3FFh | IRQ4(1) | 3E8–3EFh | IRQ4(1) |
| 1 | 0 | 0 | 0 | 2F8–2FFh | IRQ3 | Disable | — |
| 1 | 0 | 0 | 1 | 2F8–2FFh | IRQ3 | 3F8–3FFh | IRQ4 |
| 1 | 0 | 1 | 0 | Disable | — | 2E8–2EFh | IRQ3 |
| 1 | 0 | 1 | 1 | 2F8–2FFh | IRQ3 | 3E8–3EFh | IRQ4 |
| 1 | 1 | 0 | 0 | 2E8–2EFh | IRQ3 | Disable | — |
| 1 | 1 | 0 | 1 | 2E8–2EFh | IRQ3 | 3F8–3FFh | IRQ4 |
| 1 | 1 | 1 | 0 | 2E8–2EFh | IRQ3(1) | 2F8–2FFh | IRQ3(1) |
| 1 | 1 | 1 | 1 | 2E8–2EFh | IRQ3 | 3E8–3EFh | IRQ4 |

**NOTE:**
1. In this configuration, the two serial ports share the same interrupt line. Responding correctly to interrupts generated in this configuration is the exclusive responsibility of software.

**Table 10. Parallel Port Address and Interrupt Assignments**

| PPCFG1 (DTRB#) | PPCFG0 (DTRA#) | Parallel Port Address Assignment | Parallel Port Interrupt Assignment |
|---|---|---|---|
| 0 | 0 | Disable | — |
| 0 | 1 | 378–37Fh | IRQ7 |
| 1 | 0 | 278–27Fh | IRQ5 |
| 1 | 1 | 3BC–3BFh | IRQ7 |

**Table 11. Parallel Port Hardware Mode Select**

| PPMOD1 (RTSA#) | PPMOD0 (SOUTB) | Mode |
|---|---|---|
| 0 | 0 | ISA-Compatible |
| 0 | 1 | PS/2-Compatible |
| 1 | 0 | EPP |
| 1 | 1 | Reserved |

**NOTES:**
1. PPMODx hardware configuration is effective in HWE mode only.
2. ECP mode is not selectable via hardware configuration.
3. For EPP mode, address assignment must be either 278h or 378h.

**Table 12. AIP Clock Select**

| CLKSEL (SOUTA) | |
|---|---|
| 0 | 24 MHz |

**NOTE:**
Always tie this low.

**Table 13. AIP Power Supply Voltage**

| VSEL (DTRB#) | Power Supply Voltage |
|---|---|
| 0 | 5.0V Operation |
| 1 | 3.3V Operation |

**NOTES:**
1. VSEL hardware configuration is not available in HWB mode only.
2. To operate the 82091AA and all of the interfaces at 5V or 3.3V, both $V_{CC}$ and $V_{CCF}$ are connected to 5V or 3.3V power supplies, respectively. However, in the mixed mode, hardware configuration ($V_{SEL}$) is set to 3.3V, $V_{CC}$ is connected to 3.3V, and $V_{CCF}$ connected to 5V.
3. 3.3V operation is available only in the 82091AA.

**Table 14. Floppy Drive Quantity Select**

| FDDQTY (RTSB#) | Number of Supported Floppy Drives |
|---|---|
| 0 | 2 Floppy Drives |
| 1 | 4 Floppy Drives |

**NOTES:**
1. FDDQTY hardware configuration is effective in HWE mode only.
2. Four floppy drive support requires external logic to decode.

### 4.2.3 HARDWARE CONFIGURATION TIMING RELATIONSHIPS

The 82091AA samples all of the hardware configuration signals on the high-to-low transition of RSTDRV. For the HWB, SWMB, and SWAI modes, the 82091AA completes hardware configuration on this sampling (Figure 19). For HWE mode, the 82091AA samples some of the signals twice (Figure 20). The first sampling occurs on the high-to-low transition of RSTDRV. As Figure 22 shows (see Section 4.2.5, Extended Hardware Configuration Mode), the HC367 tri-states its outputs when RSTDRV is negated. This permits the strapping options from the HWEb block to be sampled. A short time after RSTDRV is negated (the time is specified in Section 11.0, Electrical Characteristics), the 82091AA samples the SOUTA, RTSA#, DTRA#, SOUTB, RTSB#, and DTRB# signals.

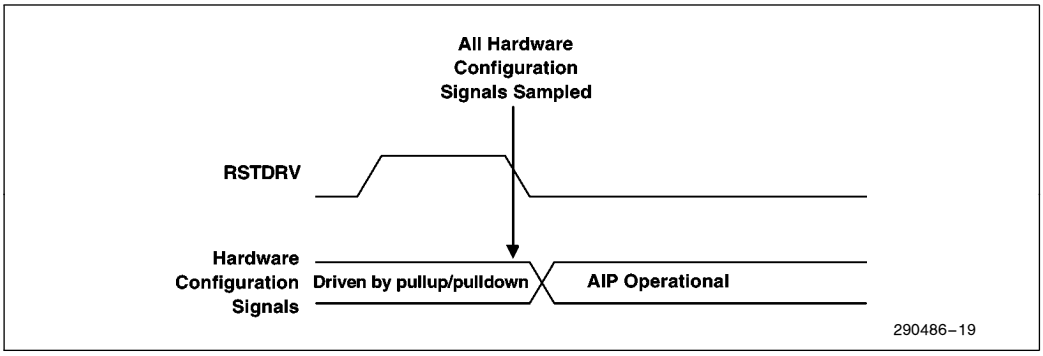



Figure 19. HWB, SWMB, and SWAI Hardware Configuration Mode Timing

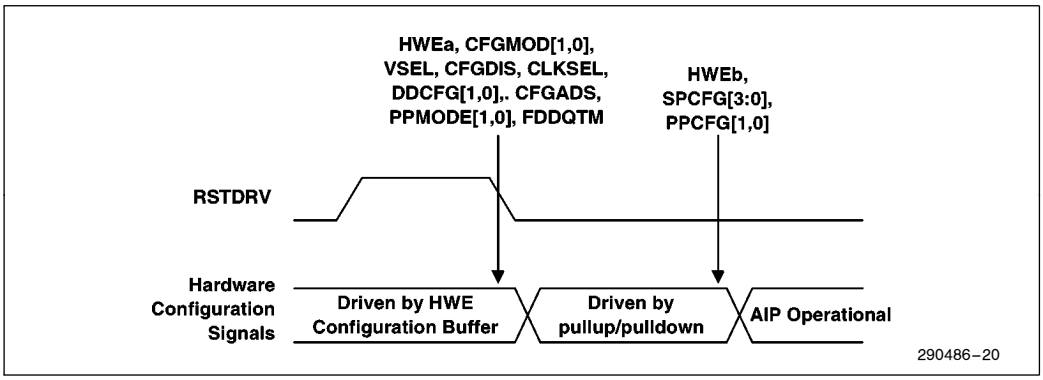



Figure 20. HWE Hardware Configuration Mode Timing

### 4.2.4 HARDWARE BASIC CONFIGURATION

The Hardware Basic configuration mode permits the user to assign addresses to the serial ports and parallel ports. This is achieved by sampling several of the serial port connections at the end of a hardware reset. The PPDIR/GCS# signal defaults to game port chip select output (GCS#). The 82091AA power supply voltage is not selectable in this mode and is fixed at 5V. The parallel port mode is set to ISA-Compatible. In addition, the FDC floppy drive support is set at two floppy drives. If configuration register access is enabled, the access address is fixed at 398h/399h. To reconfigure the 82091AA using software, the 82091AA configuration mode must be changed to SWAI mode (refer to AIPCFG1 register). Figure 21 shows the implementation of a basic hardware configuration.

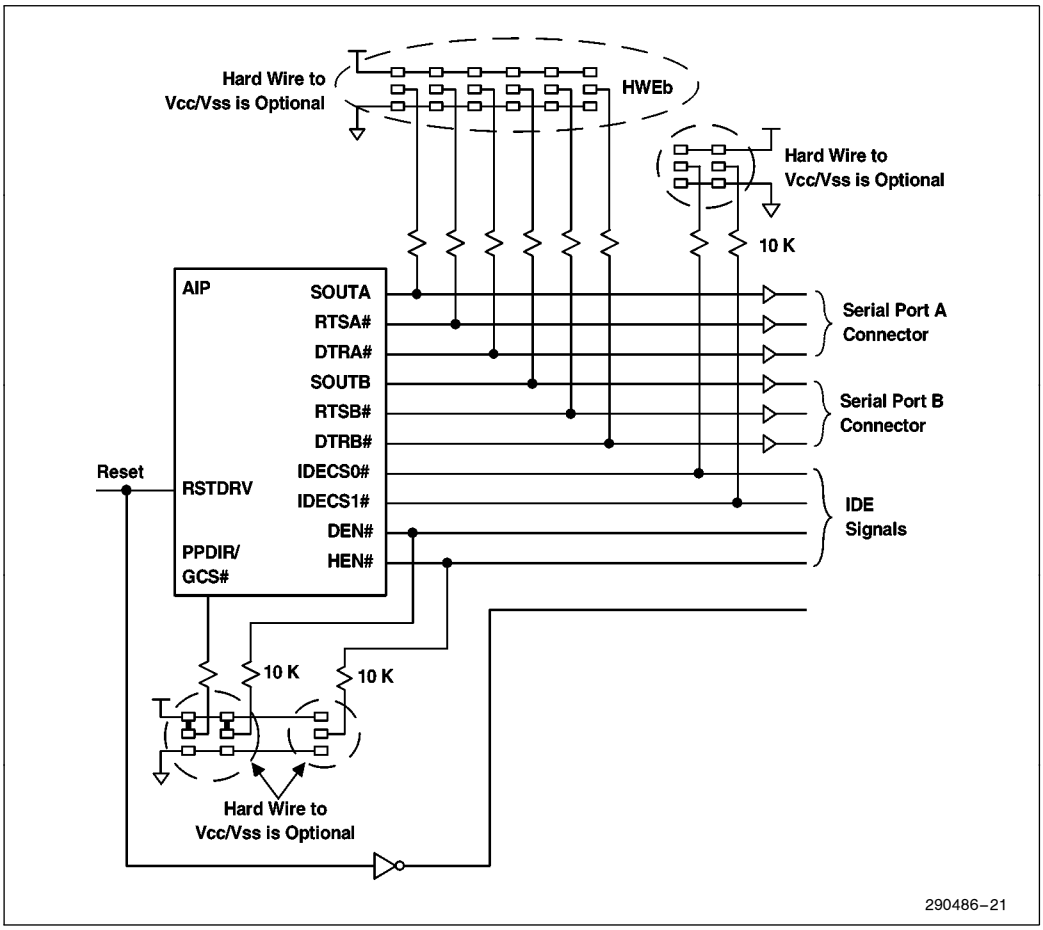



**Figure 21. Hardware Basic Configuration**

### 4.2.5 HARDWARE EXTENDED CONFIGURATION MODE

The Hardware Extended configuration mode provides all of the features of the Hardware Basic configuration mode. Additional features in Hardware Extended configuration permit the user to select quantity of floppy drives can be selected for either 2 or 4 floppy drive support. The 82091AA operating voltage is selectable between 3.3V* and 5V. In addition, the parallel port can be configured to operate in ISA-Compatible, PS/2-Compatible, or EPP modes. Hardware extended configuration provides these additional hardware configuration options by sampling the pins on the serial ports at two different times. When RSTDRV is asserted, the HC367 drives the values on SOUTA, RTSA#, DTRA#, SOUTB, RTSB#, and DTRB# (Figure 22). When RTSDRV is negated, the HC367 is disabled and these serial port signals are driven by HWEb pullup/down resistors. The PPDIR/GCS# signal defaults to a game port chip select (GCS#). To reconfigure the 82091AA using software, the 82091AA configuration mode must be changed to SWAI mode (refer to AIPCFG1 register).

**NOTE:**
*3.3V operation is only available in the 82091AA.

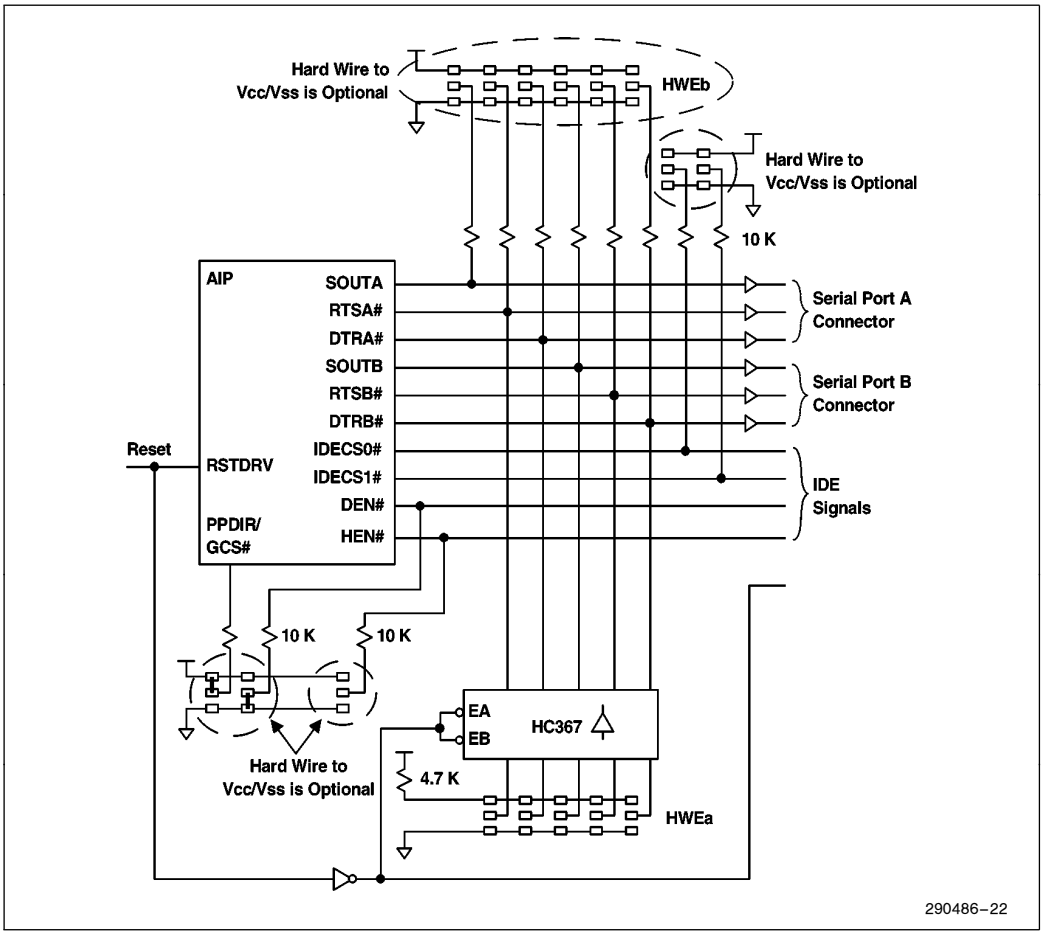



**Figure 22. Hardware Extended Configuration**

### 4.2.6 SOFTWARE ADD-IN CONFIGURATION

The Software Add-in configuration mode permits the user to assign the address for the 82091AA configuration registers, and select the power supply voltage for the 82091AA. The 82091AA configuration registers are accessible. The registers are located in the ISA Bus I/O address space and can be selected to be at either 398h/399h or 26Eh/26Fh. The PPDIR/GCS# signal defaults to a game port chip select (GCS#).

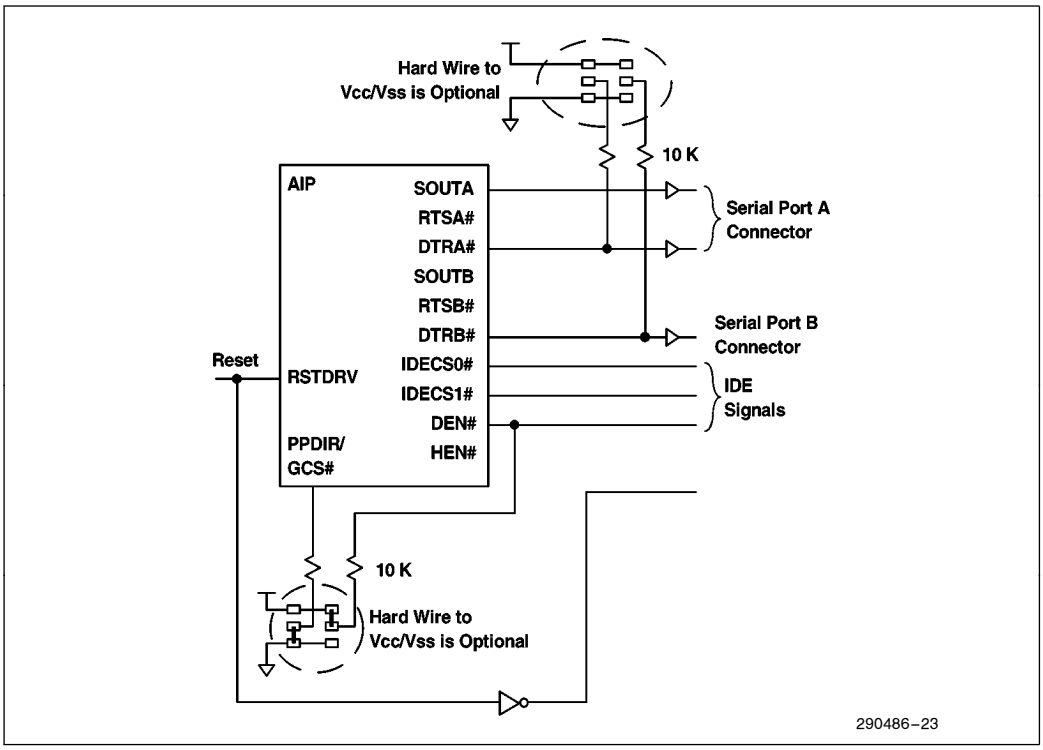



**Figure 23. Software Add-In Configuration**

### 4.2.7 SOFTWARE MOTHERBOARD CONFIGURATION

The Software Motherboard configuration mode permits the 82091AA to be located on the motherboard. In this mode, the 82091AA configuration registers are accessible via the X-Bus I/O address space and can be selected to be at either 22h/23h or 24h/25h. In addition, the user selects the power supply voltage for the 82091AA. The PPDIR/GCS# signal defaults to a Parallel Port Direction Control Output (PPDIR).

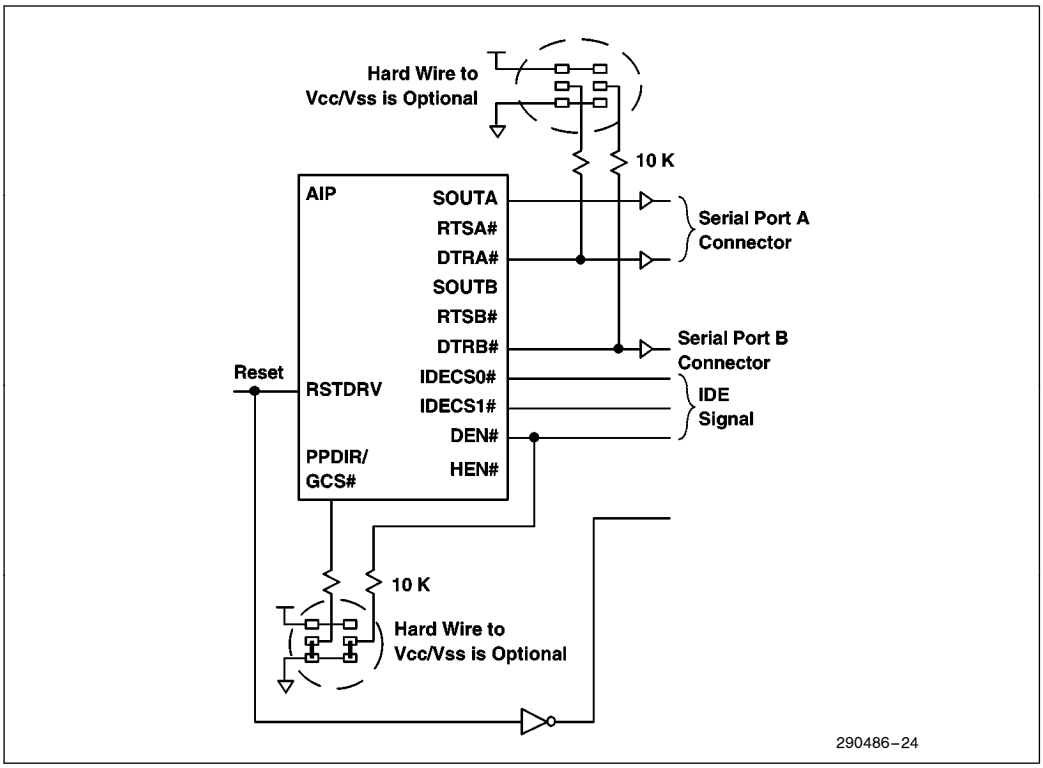



**Figure 24. Software Motherboard Hardware Configuration**

## 5.0 HOST INTERFACE

The 82091AA host interface is an 8-bit direct-drive (24 mA) ISA Bus/X-Bus interface that permits the CPU to access its registers through read/write operations in I/O space. These registers may be accessed by programmed I/O and/or DMA bus cycles. With the exception of the IDE Interface, all functions on the 82091AA require only 8-bit data accesses. The 16-bit access required for the IDE Interface is supported through the appropriate chip selects and data buffer enables from the 82091AA. The 82091AA does not participate in 16-bit IDE DMA transfers.

Although the 82091AA has an ISA/X-Bus host interface, there are a few features that differentiate it from conventional ISA/X-Bus peripherals. These features are as follows:

- **Internal, Configurable Chip Select Decode Logic.** SA[9:0] allow full decoding of the ISA I/O address space such that the functional modules contained in the 82091AA can be relocated to the desired I/O address. This feature can be used to resolve potential system configuration conflicts.
- **IOCHRDY for ISA Cycle Extension.** During certain I/O cycles to the parallel port controller in the 82091AA, it is necessary to extend the current bus cycle to match the access time of the device connected to the Parallel Port. The IOCHRDY signal is used by the 82091AA to extend ISA Bus cycles, as needed, according to the ISA protocol. IOCHRDY overrides all other strobes that attempt to shorten the bus cycle.
- **NOWS# for 3 BCLK I/O Cycles.** All programmed I/O accesses to 82091AA registers can be completed in a total of 3 BCLK cycles. This is possible because the 82091AA register access times have been minimized to allow data transfers to occur with shortened read/write control strobes. As a result, the 82091AA is well suited for use in embedded control designs that use an asynchronous microprocessor interface without any particular reference to ISA cycle timings.
- **DMA Transfers:** The 82091AA supports DMA compatible, type A, type B and type F DMA cycles. Some newer system DMA controllers are capable of generating fast DMA cycles (type F) on all DMA channels. If such a controller is used in conjunction with the 82091AA, it will be possible to accomplish a DMA transfer in 2 BCLKs.

The 82091AA ISA data lines (SD[7:0]) can be connected directly to the ISA Bus. If external buffers are used to isolate the SD[7:0] signals from the 240 pF loading of the ISA Bus, the DEN# signal can be used to control the external buffers as shown in Figure 25.

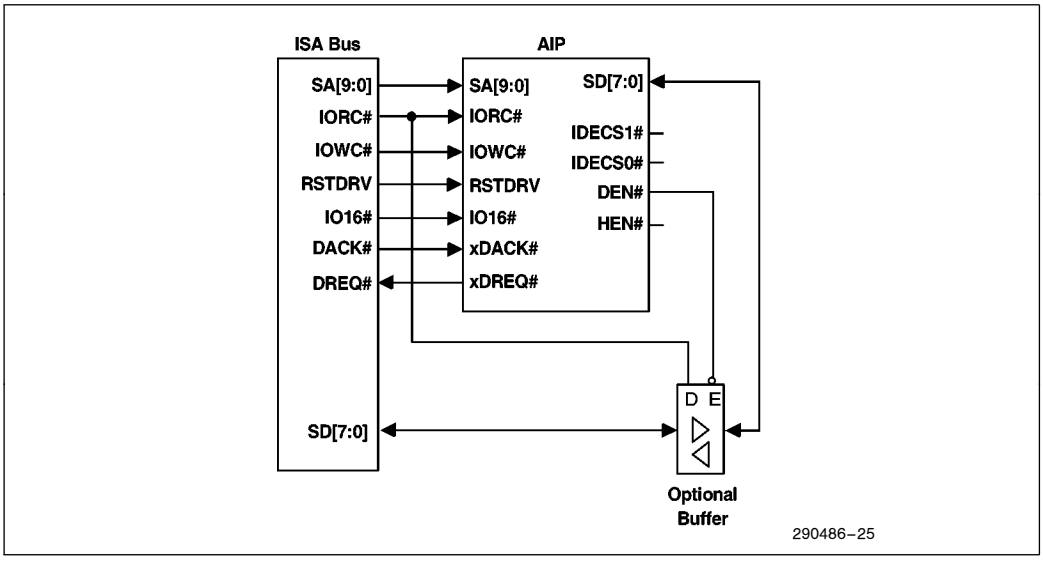



**Figure 25. ISA Interface (with Optional Data Buffer)**

## 6.0 PARALLEL PORT

The 82091AA parallel port can be configured for four parallel port modes. These parallel port modes and the associated parallel interface protocols are:

| Parallel Port Mode | Parallel Interface Protocol |
|---|---|
| ISA-Compatible Mode | Compatibility, Nibble |
| PS/2-Compatible Mode | Byte |
| EPP Mode | EPP |
| ECP | ECP |

ISA-Compatible, PS/2-Compatible, and EPP modes are selected through 82091AA configuration (see Section 4.0, AIP Configuration). ECP is selected by programming the ECP Extended Control Register (ECR).

In ISA-Compatible mode, the parallel port exactly emulates a standard ISA-style parallel port. The parallel port data bus (PD[7:0]) is uni-directional. The compatibility protocol transfers data to the peripheral device via PD[7:0] (forward direction). Note that the Nibble protocol permits data transfers from the peripheral device (reverse direction) by using four peripheral status signal lines to transfer 4 bits of data at a time.

PS/2-Compatible mode differs from ISA-Compatible mode by providing bi-directional transfers on PD[7:0]. A bit is added to the PCON Register to allow software control of the data transfer direction.

For both the ISA-Compatible and PS/2-Compatible modes, the actual data transfer over the parallel port interface is accomplished by software handshake (i.e., automatic hardware handshake is not used). Software controls data transfer by monitoring handshake signal status from the peripheral device via the PSTAT Register and controlling handshake signals to the peripheral device via the PCON Register.

EPP mode provides bi-directional transfers on PD[7:0]. The 82091AA automatically generates the address and data strobes in hardware.

ECP is a high performance peripheral interface mode. This mode uses an asynchronous automatic handshake to transfer data over the parallel port interface. In addition, the parallel port contains a FIFO for transferring data in ECP mode. The ECP register set contains an Extended Control Register (ECR) that provides a wide range of functions including the ability to operate the parallel port in either ECP, ISA-Compatible, or PS/2-Compatible modes.

**NOTE:**
In general, this document describes parallel port operations and functions in terms of how the 82091AA parallel port hardware operates. Detailed descriptions of the parallel interface protocols are beyond the scope of this document. Readers should refer to the proposed IEEE Standard 1284 for detailed descriptions of the Compatibility, Nibble, Byte, EPP, and ECP protocols.

Special circuitry on the 82091 prevents it from being powered up or being damaged while a parallel port peripheral is powered on and the 82091 is powered off.

### 6.1 Parallel Port Registers

This section is organized into three sub-sections—ISA-Compatible and PS/2-Compatible Modes, EPP Mode, and ECP Mode. Since the register sets are similar for ISA-Compatible and PS/2-Compatible modes (differing by a direction control bit in the PCON Register) the register set descriptions are combined. The EPP mode and ECP mode register sets are described separately. Each register set description contains the I/O address assignment and a complete description of the registers and register bits. Note that the PSTAT and PCON Registers are common to all modes and for completeness are repeated in each sub-section. Any difference in bit operations for a particular mode is noted in that particular register description.

The registers provide parallel port control/status information and data paths for transferring data between the parallel port interface and the 8-bit host interface. All registers are accessed as byte quantities. The base address is determined by hardware configuration at powerup (or a hard reset) or via software configuration by programming the 82091AA configuration registers as described in Section 4.0, AIP Configuration. The parallel port can be disabled or configured for a base address of 378h (all modes), 278h (all modes), or 3BCh (all modes except EPP and ECP). This provides the system designer with the option of using additional parallel ports on add-in cards that have fixed address decoding.

Some of the parallel port registers described in this section contain reserved bits. These bits are labeled ''R''. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the value of reserved bit positions must first be read, merged with the new values for other bit positions, and then written back.

During a hard reset (RSTDRV asserted), the 82091AA registers are set to pre-determined **default** states. The default values are indicated in the individual register descriptions.

The following nomenclature is used for register access attributes:

**RO** **Read Only**. Note that for registers with read only attributes, writes to the I/O address have no affect on parallel port operations.

**R/W** **Read/Write**. A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

### 6.1.1 ISA-COMPATIBLE AND PS/2-COMPATIBLE MODES

This section contains the registers used in ISA-Compatible and PS/2-Compatible modes. The I/O address assignment for this register set is shown in Table 15 and the register descriptions are presented in the order that they appear in the table.

**Table 15. Parallel Port Register (ISA-Compatible and PS/2-Compatible)**

| Parallel Port Register Address Access (AEN = 0) Base + | Abbreviation | Register Name | Access |
|---|---|---|---|
| 0h | PDATA | Data Register | R/W |
| 1h | PSTAT | Status Register | RO |
| 2h | PCON | Control Register | R/W |

**NOTE:**
Parallel port base addresses are 278h, 378h and 3BCh.

#### 6.1.1.1 PDATA—Parallel Port Data Register (ISA-Compatible and PS/2-Compatible Modes)

I/O Address: Base + 00h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

**ISA-Compatible Mode**

The PDATA Register is a uni-directional data port that transfers 8-bit data from the host to the peripheral device (forward transfer). A write to this register drives the written data onto PD[7:0]. Reads of this register should not be performed in ISA-Compatible mode. For a host read of this address location, the 82091AA completes the handshake on the ISA Bus and the value is the last value stored in the PDATA Register.

**PS/2-Compatible Mode**

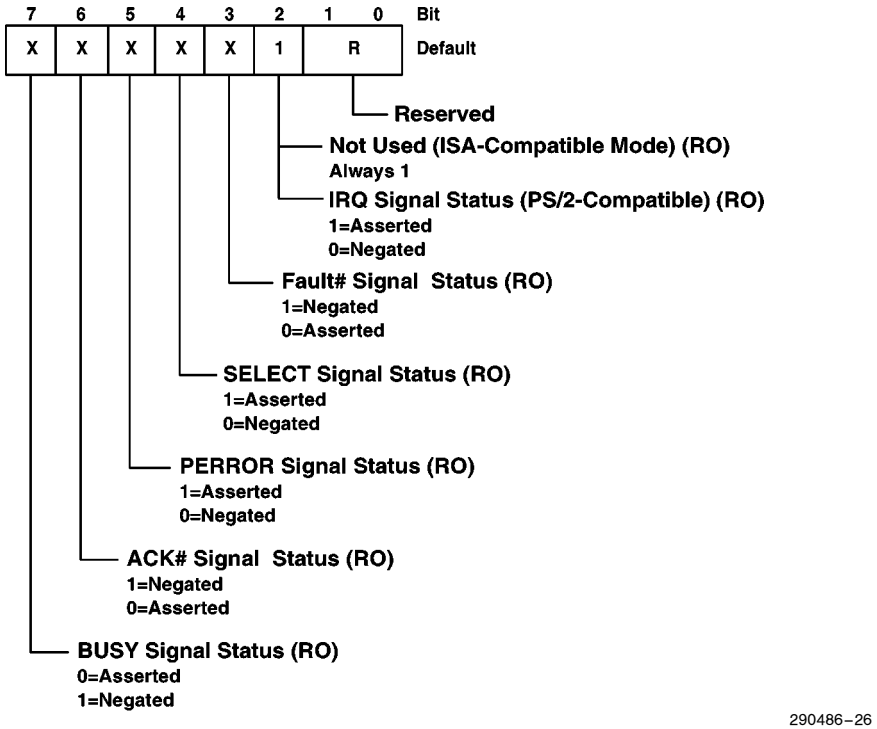
The PDATA Register is a bi-directional data port that transfers 8-bit data between the peripheral device and host. The direction of transfer is determined by the DIR# bit in the PCON Register. If DIR# = 0 (forward direction), and the host writes to this register, the data is stored in the PDATA Register and driven onto PD[7:0]. If DIR# = 1 (reverse direction), a host read of this register returns the data on PD[7:0]. Note that read data is not stored in the PDATA Register.

| Bit | Description |
|---|---|
| 7:0 | **PARALLEL PORT DATA:** Bits[7:0] correspond to parallel port data lines PD[7:0] and ISA Bus data lines SD[7:0]. |

#### 6.1.1.2 PSTAT—Status Register (ISA-Compatible and PS/2-Compatible Modes)

I/O Address: Base + 01h
Default Value: XXXX X1RR
Attribute: Read Only
Size: 8 bits

The PSTAT Register provides the status of certain parallel port signals and whether a CPU interrupt has been generated by the parallel port. This register indicates the current state of the BUSY, ACK#, PERROR, SELECT, and FAULT# signals.




**NOTE:**
X = Default value is determined by signal state at reset.

**Figure 26. Status Register (ISA-Compatible and PS/2-Compatible Modes)**

| Bit | Description |
|---|---|
| 7 | **BUSY STATUS (BUSYS):** This bit indicates the state of the parallel port interface BUSY signal. When BUSY is asserted, BUSYS = 0. When BUSY is negated, BUSYS = 1.This bit is an inverted version of the parallel port BUSY signal. |
| 6 | **ACK# STATUS (ACKS):** This bit indicates the state of the parallel port interface ACK# signal. This bit indicates when the peripheral has received a data byte and is ready for another. When ACK# is asserted, ACKS = 0. When ACK# is negated, ACKS = 1. Note that if interrupts are enabled (via bit 4 of the PCON Register), the assertion of the ACK# signal generates an interrupt to the CPU. |
| 5 | **PERROR STATUS (PERRS):** This bit indicates the state of the parallel port interface PERROR signal. This bit indicates when an error has occurred in the peripheral paper path (e.g., out of paper). When PERROR is asserted, PERRS = 1, When PERROR is negated, PERRS = 0. |
| 4 | **SELECT STATUS (SELS):** This bit indicates the state of the parallel port interface SELECT signal. When the SELECT signal is asserted, SELS = 1, When the SELECT signal is negated, SELS = 0. |
| 3 | **FAULT# STATUS (FAULTS):** This bit indicates the state of the parallel port interface FAULT# signal being driven by the peripheral device. When the FAULT# signal is asserted, FAULTS = 0. When the FAULT# signal is negated, FAULTS = 1. |
| 2 | **PARALLEL PORT INTERRUPT STATUS (PIRQ):** This bit indicates a CPU interrupt by the parallel port. PIRQ indicates that the printer has accepted the previous character and is ready for another.<br>In ISA-Compatible mode, interrupt status is not reported in this register and this bit is always 1.<br>In PS/2-Compatibile mode, if interrupts are enabled via the PCON Register and the ACK# signal is asserted (low-to-high transition), PIRQ is set to a 0 (and an IRQ generated to the CPU). The 82091AA sets PIRQ to 1 when this register is read or by a hard reset. If interrupts are disabled via the PCON Register, this bit is never set to 0. |
| 1:0 | **RESERVED** |

#### 6.1.1.3 PCON—Control Register (ISA-Compatible And PS/2-Compatible Mode)

I/O Address: Base + 02h
Default Value: RR00 0000
Attribute: Read/Write
Size: 8 bits

The PCON Register controls certain parallel port interface signals and enables/disables parallel port interrupts. This register permits software to control the STROBE#, AUTOFD#, INIT#, and SELECTIN# signals. For PS/2-Compatible mode, this register also controls the direction of transfer on PD[7:0].

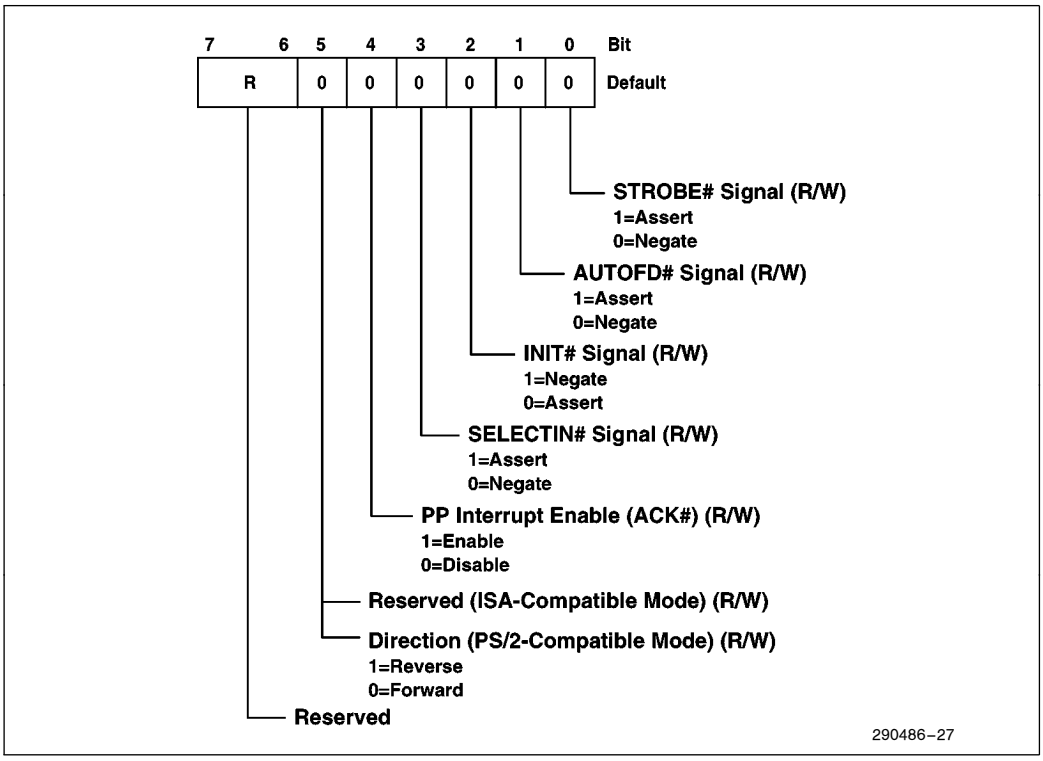



**Figure 27. Control Register (ISA-Compatible and PS/2-Compatible Modes)**

| Bit | Description |
|---|---|
| 7:6 | **RESERVED** |
| 5 | **RESERVED (ISA-COMPATIBLE MODE):** Not used and undefined when read. Writes have no affect on parallel port operations.<br>**DIRECTION (DIR#) (PS/2-COMPATIBLE MODE):** This bit is used to control the direction of data transfer on the parallel port data bus (PD[7:0]). When DIR# = 0, PD[7:0] are outputs. When DIR# = 1, PD[7:0] are inputs. |
| 4 | **ACK# INTERRUPT ENABLE (ACKINTEN):** ACKINTEN enables CPU interrupts (via either IRQ5 or IRQ7) to be generated when the ACK# signal on the parallel port interface is asserted. When ACKINTEN = 1, a CPU interrupt is generated when ACK# is asserted. When ACKINTEN = 0, the ACK# interrupt is disabled. |
| 3 | **SELECTIN# CONTROL (SELINC):** This bit controls the SELECTIN# signal. SELINC is set to 1 to select the printer. When SELINC = 1, the SELECTIN# signal is asserted, When SELINC = 0, the SELECTIN# signal is negated. |
| 2 | **INIT# CONTROL (INITC):** This bit controls the INIT# signal. When INITC = 1, the INIT# signal is negated. When INITC = 0, the INIT# signal is asserted. |
| 1 | **AUTOFD# CONTROL (AUTOFDC):** This bit controls the AUTOFD# signal. AUTOFDC is set to 1 to instruct the printer to advance the paper one line each time a carriage return is received. When AUTOFDC = 1, the AUTOFD# signal is asserted. When AUTOFDC = 0, the AUTOFD# signal is negated. |
| 0 | **STROBE# CONTROL (STROBEC):** This bit controls the STROBE# signal. The STROBE# signal is set active to instruct the printer to accept the character being presented on the data lines. When STROBEC = 1, the STROBE# signal is asserted. When STROBEC = 0, the STROBE# signal is negated. |

### 6.1.2 EPP MODE

This section contains the registers used in EPP mode. The I/O address assigment for this register set is shown in Table 16 and the register descriptions are presented in the order that they appear in the table.

**Table 16. Parallel Port Registers (EPP Mode)**

| Parallel Port Register Address Access (AEN = 0) Base + | Abbreviation | Register Name | Access |
|---|---|---|---|
| 0h | PDATA | Data Register | R/W |
| 1h | PSTAT | Status Register | RO |
| 2h | PCON | Control Register | R/W |
| 3h | ADDSTR | Address Strobe Register | R/W |
| 4h–7h | DATASTR | Data Strobe Registers | R/W |

**NOTE:**
Parallel port base addresses are 278h (LPT2) and 378h (LPT1). Base address 3BCh is not available in EPP or ECP modes.

#### 6.1.2.1 PDATA—Parallel Port Data Register (EPP Mode)

I/O Address: Base + 00h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

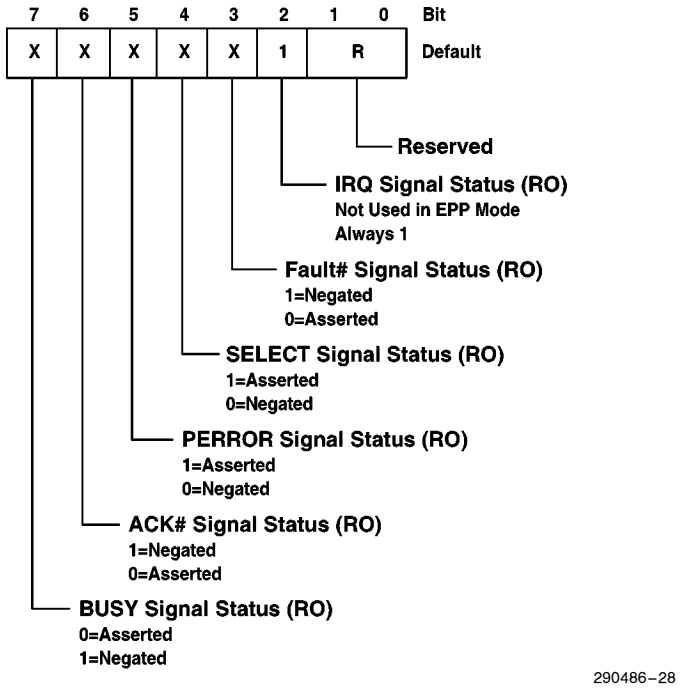
The PDATA Register is a bi-directional data port that transfers 8-bit data between the peripheral device and host. The direction of transfer is determined by the DIR# bit in the PCON Register. If DIR# = 0 (forward direction) and the host writes to this register, the data is stored in the PDATA Register and driven onto PD[7:0]. If DIR# = 1 (reverse direction), a host read of this register returns the data on PD[7:0]. However, read data is not stored in the PDATA Register.

| Bit | Description |
|---|---|
| 7:0 | **PARALLEL PORT DATA:** Bits[7:0] correspond to parallel port data lines PD[7:0] and ISA Bus data lines. |

#### 6.1.2.2 PSTAT—Status Register (EPP Mode)

I/O Address: Base +01h
Default Value: XXXX X1RR
Attribute: Read Only
Size: 8 bits

The PSTAT Register provides the status of certain parallel port signals. It also indicates whether a CPU interrupt has been generated by the parallel port. This register indicates the current state of the BUSY, ACK#, PERROR, SELECT, and FAULT# signals.




**NOTE:**
X = Default value is determined by signal state at reset.

**Figure 28. Status Register (EPP Mode)**

| Bit | Description |
|---|---|
| 7 | **BUSY STATUS (BUSYS):** This bit indicates the state of the parallel port interface BUSY signal. When BUSY is asserted, BUSYS = 0. When BUSY is negated, BUSYS = 1. This bit is an inverted version of the parallel port BUSY signal. |
| 6 | **ACK# STATUS (ACKS):** This bit indicates the state of the parallel port interface ACK# signal. This bit indicates when the peripheral has received a data byte and is ready for another. When ACK# is asserted, ACKS = 0. When ACK# is negated, ACKS = 1. Note that if interrupts are enabled (via bit 4 of the PCON Register), the assertion of the ACK# signal generates an interrupt to the CPU. |
| 5 | **PERROR STATUS (PERRS):** This bit indicates the state of the parallel port interface PERROR signal. This bit indicates when an error has occurred in the peripheral paper path (e.g., out of paper). When PERROR is asserted, PERRS = 1. When PERROR is negated, PERRS = 0. |
| 4 | **SELECT STATUS (SELS):** This bit indicates the state of the parallel port interface SELECT signal. When the SELECT signal is asserted, SELS = 1. When the SELECT signal is negated, SELS = 0. |
| 3 | **FAULT# STATUS (FAULTS):** This bit indicates the state of the parallel port interface FAULT# signal being driven by the peripheral device. When the FAULT# signal is asserted, FAULTS = 0. When the FAULT# signal is negated, FAULTS = 1. |
| 2 | **PARALLEL PORT INTERRUPT (PIRQ):** In EPP mode interrupt status is not reported in this register and this bit is always 1. |
| 1:0 | **RESERVED** |

#### 6.1.2.3 PCON—Control Register (EPP Mode)

I/O Address: Base + 02h
Default Value: RR00 0000
Attribute: Read/Write
Size: 8 bits

The PCON Register controls certain parallel port interface signals, enables/disables parallel port interrupts, and selects the direction of data transfer on PD[7:0]. This register permits software to control the INIT# signal. Note that in the EPP parallel interface protocol, the STROBE#, AUTOFD#, and SELECTIN# signals are automatically generated by the parallel port and are not controlled by software.

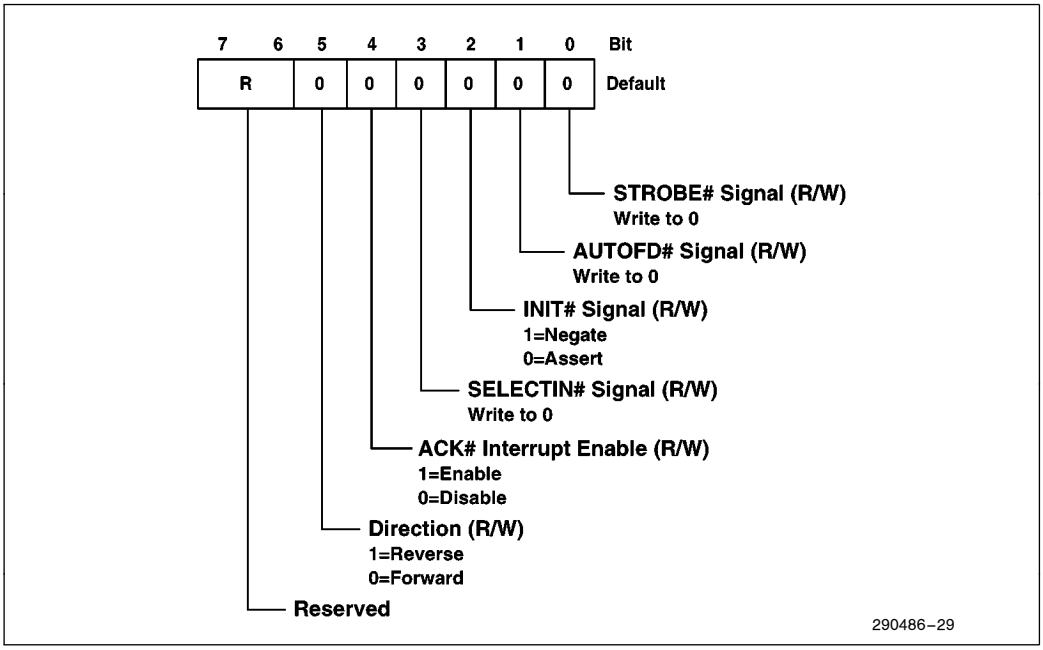



**Figure 29. Control Register (EPP Mode)**

| Bit | Description |
|---|---|
| 7:6 | **RESERVED** |
| 5 | **DIRECTION (DIR#):** This bit is used to control the direction of data transfer on the parallel port data bus (PD[7:0]). When DIR# = 0 (forward direction), PD[7:0] are outputs. When DIR# = 1 (reverse direction), PD[7:0] are inputs. |
| 4 | **ACK# INTERRUPT ENABLE (ACKINTEN):** ACKINTEN enables CPU interrupts (via IRQ5 or IRQ7) to be generated when the ACK# signal on the parallel port interface is asserted. When ACKINTEN = 1, a CPU interrupt is generated when ACK# is asserted. When ACKINTEN = 0, the ACK# interrupt is disabled. |
| 3 | **SELECTIN# CONTROL (SELINC):** Write to 0 when programming this register. This bit must be 0 for the parallel port handshake to operate properly. |
| 2 | **INIT# CONTROL (INITC):** This bit controls the INIT# signal. When INITC = 1, the INIT# signal is negated. When INITC = 0, the INIT# signal is asserted. |
| 1 | **AUTOFD# CONTROL (AUTOFDC):** Write to 0 when programming this register. |
| 0 | **STROBE# CONTROL (STROBEC):** Write to 0 when programming this register. This bit must be 0 for the parallel port handshake to operate properly. |

#### 6.1.2.4 ADDSTR—EPP Auto Address Strobe Register (EPP Mode)

I/O Address: Base + 03h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ADDSTR Register provides a peripheral address to the peripheral (via PD[7:0]) during a host address write operation and to the host (via PD[7:0]) during a host address read operation. An automatic address strobe is generated on the parallel port interface when data is read from or written to this register.

| Bit | Description |
|---|---|
| 7:0 | **EPP ADDRESS:** Bits[7:0] correspond to SD[7:0] and PD[7:0]. |

#### 6.1.2.5 DATASTR—Auto Data Strobe Register (EPP Mode)

I/O Address: Base + 04h, 05h, 06h, 07h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The DATASTR Register provides data from the host to the peripheral device (via PD[7:0]) during host write operations and from the peripheral device to the host (via PD[7:0]) during a host read operation. An automatic data strobe is generated on the parallel port interface when data is read from or written to this register. To maintain compatibility with Intel's 82360SL I/O device that has a 32-bit Host Bus interface, four consecutive byte address locations are provided for transferring data.

| Bit | Description |
|---|---|
| 7:0 | **EPP DATA:** Bits[7:0] correspond to SD[7:0] and PD[7:0]. |

### 6.1.3 ECP MODE

This section contains the registers used in ECP mode. The I/O address assignment for this register set is shown in Table 17 and the register descriptions are presented in the order that they appear in the table. The Extended Control Register (ECR) permits various modes of operation. Note that ECR[7:5] = 000 selects ISA-Compatible mode and ECR[7:5] = 001 selects PS/2-Compatibile mode. These modes are discussed in Section 6.1.1, ISA-Compatible and PS/2 Compatible modes. The other modes selected by ECR[7:5] are discussed in this section.

**Table 17. Parallel Port Registers (ECP Mode)**

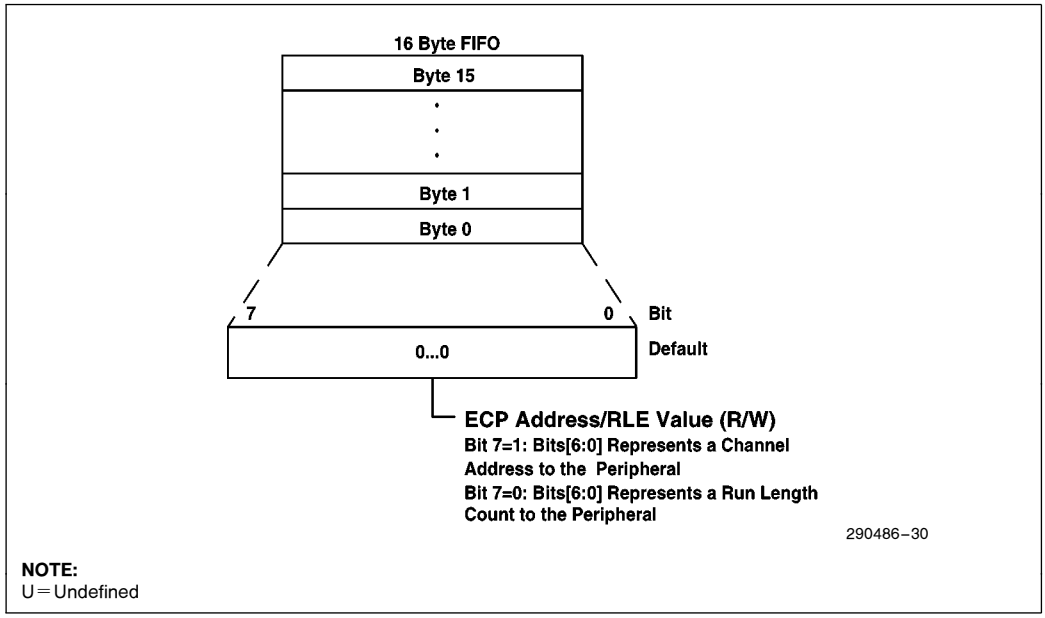
| Parallel Port Register Address Access (AEN = 0) Base + | Abbreviation | Register Name | Access | |
|---|---|---|---|---|
| | | | ECR[7:5] | Read/Write Attribute |
| 0h | ECPAFIFO | ECP Address/RLE FIFO | 011 | R/W |
| 1h | PSTAT | Status Register | All | RO |
| 2h | PCON | Control Register | All | R/W |
| 400h | SDFIFO | Standard Parallel Port Data FIFO | 010 | R/W |
| 400h | ECPDFIFO | ECP Data FIFO | 011 | R/W |
| 400h | TFIFO | Test FIFO | 110 | R/W |
| 400h | ECPCFGA | ECP Configuration A | 111 | R/W |
| 401h | ECPCFGB | ECP Configuration B | 111 | R/W |
| 402h | ECR | Extended Control Register | All | R/W |

**NOTES:**
1. Parallel port base addresses are 278h, 378h, and 3BCh.
2. A register is accessible when the ECR[7:5] field contains the value specified in the ECR[7:5] column. The register is not accessible if the ECR[7:5] field does not match the value specified in this column. The term ''All'' means that the register is accessible in all modes selected by ECR[7:5].

#### 6.1.3.1 ECPAFIFO—ECP Address/RLE FIFO Register (ECP Mode)

I/O Address: Base + 00h
Default Value: UUUU UUUU (Undefined)
Attribute: Read/Write
Size: 8 bits

The ECPAFIFO Register provides a channel address or a Run Length Count (RLE) to the peripheral, depending on the state of bit 7. This I/O address location is only used in ECP mode (ECR bits[7:5] = 011). In this mode, bytes written to this register are placed in the parallel port FIFO and transmitted over PD[7:0] using ECP protocol.




**NOTE:**
U = Undefined

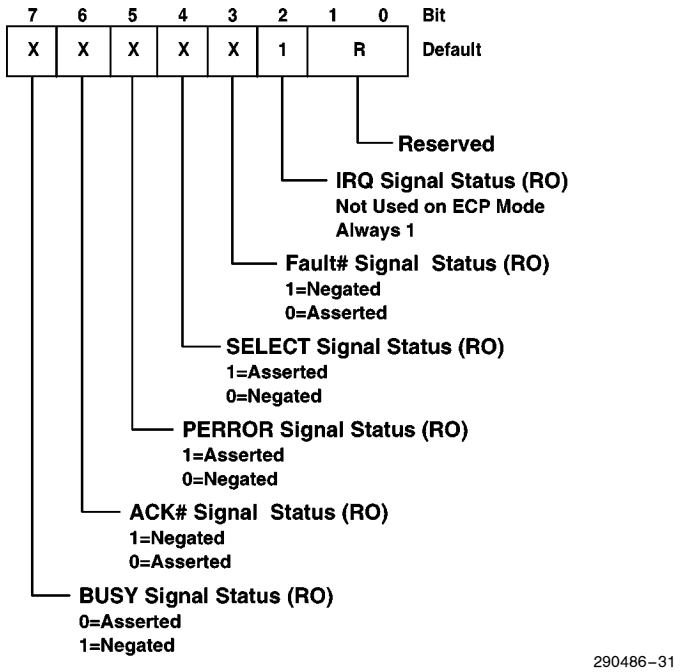
**Figure 30. ECP Address/RLE FIFO Register (ECP Mode)**

| Bit | Description |
|---|---|
| 7:0 | **ECP ADDRESS/RLE VALUE:** Bits[7:0] correspond to parallel port data lines PD[7:0] and ISA Bus data lines SD[7:0]. The peripheral device should interpret bits[6:0] as a channel address when bit 7 = 1 and as a run length count when bit 7 = 0. Note that this interpretation is performed by the peripheral device and the value of bit 7 has no affect on 82091AA operations. Note that the 82091AA asserts AUTOFD# to indicate that the information on PD[7:0] represents an ECP address/RLE count. The 82091AA negates AUTOFD# (drives high) when PD[7:0] is transferring data. |

#### 6.1.3.2 PSTAT—Status Register (ECP Mode)

I/O Address: Base +01h
Default Value: XXXX X1RR
Attribute: Read Only
Size: 8 bits

The PSTAT Register provides the status of certain parallel port signals and whether a CPU interrupt has been generated by the parallel port. This register indicates the current state of the BUSY, ACK#, PERROR, SELECT, and FAULT# signals.




**NOTE:**
X = Default value is determined by the state of the corresponding signal pin at reset.

**Figure 31. Status Register (ECP Mode)**

| Bit | Description |
|---|---|
| 7 | **BUSY STATUS (BUSYS):** This bit indicates the state of the parallel port interface BUSY signal. When BUSY is asserted, BUSYS = 0. When BUSY is negated, BUSYS = 1. This is an inverted version of the parallel port BUSY signal. Refer to Section 6.2.3 ECP Mode for more detail. |
| 6 | **ACK# STATUS (ACKS):** This bit indicates the state of the parallel port interface ACK# signal. This bit indicates when the peripheral has received a data byte and is ready for another. When ACK# is asserted, ACKS = 0. When ACK# is negated, ACKS = 1. Note that if interrupts are enabled (via bit 4 of the PCON Register), the assertion of the ACK# signal generates an interrupt to the CPU. Refer to Section 6.2.3 ECP Mode for more detail. |
| 5 | **PERROR STATUS (PERRS):** This bit indicates the state of the parallel port interface PERROR signal. This bit indicates when an error has occurred in the peripheral paper path (e.g., out of paper). When PERROR is asserted, PERRS = 1, When PERROR is negated, PERRS = 0. |
| 4 | **SELECT STATUS (SELS):** This bit is used in all parallel port modes and indicates the state of the parallel port interface SELECT signal. When the SELECT signal is asserted, SELS = 1. When the SELECT signal is negated, SELS = 0. |
| 3 | **FAULT# STATUS (FAULTS):** This bit is used in all parallel port modes and indicates the state of the parallel port interface FAULT# signal being driven by the peripheral device. When the FAULT# signal is asserted, FAULTS = 0. When the FAULT# signal is negated, FAULTS = 1. |
| 2 | **PARALLEL PORT INTERRUPT (PIRQ):** In ECP mode, interrupt status is not reported in this register and this bit is always 1. |
| 1:0 | **RESERVED** |

#### 6.1.3.3 PCON—Control Register (ECP Mode)

I/O Address: Base + 02h
Default Value: RR00 0000
Attribute: Read/Write
Size: 8 bits

The PCON Register controls certain parallel port interface signals, enables/disables parallel port interrupts, and selects the direction of data transfer on PD[7:0]. Note that the function of some bits depends on the programming of the ECR.

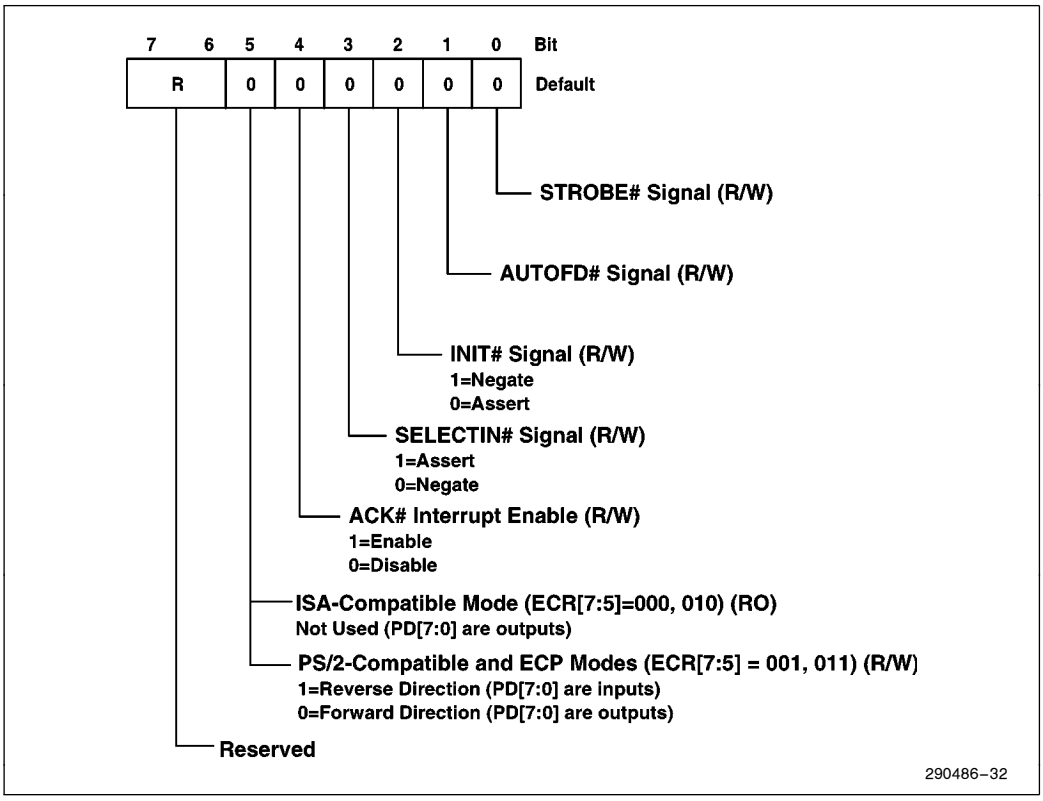


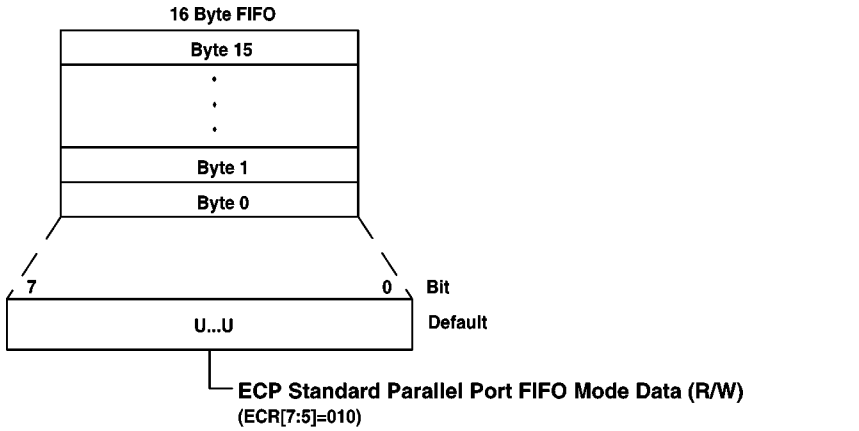

**Figure 32. Control Register (ECP Mode)**

| Bit | Description |
|---|---|
| 7:6 | **RESERVED** |
| 5 | **DIRECTION (DIR#):** This bit is used to control the direction of data transfer on the parallel port data bus (PD[7:0]). When DIR# = 0 (forward direction), PD[7:0] are outputs. When DIR# = 1 (reverse direction), PD[7:0] are inputs. |
| 4 | **INTERRUPT ENABLE (ACK#) (IRQEN):** IRQEN enables interrupts to the CPU to be generated when the ACK# signal on the parallel port interface is asserted and is used in all parallel port interface modes. When IRQEN = 1, a CPU interrupt is generated when ACK# is asserted. When IRQEN = 0, parallel port interrupts are disabled. |
| 3 | **SELECTIN# CONTROL (SELINC):** This bit controls the SELECTIN# signal. SELINC is set to 1 to select the printer. When SELINC = 1, the SELECTIN# signal is asserted, When SELINC = 0, the SELECTIN# signal is negated. |
| 2 | **INIT# CONTROL (INITC):** This bit controls the INIT# signal When INITC = 1, the INIT# signal is negated. When INITC = 0, the INIT# signal is asserted. |
| 1 | **AUTOFD# CONTROL (AUTOFDC):** In ECP mode or ISA-Compatible FIFO mode (ECR[7:5] = 011, 010), this bit has no effect. Refer to Section 6.2.3 ECP Mode for more details. |
| 0 | **STROBE# CONTROL (STROBEC):** In ECP mode or ISA-Compatible FIFO mode (ECR[7:5] = 011, 010), this bit has no effect. Refer to Section 6.2.3 ECP Mode for more details. |

#### 6.1.3.4 SDFIFO—Standard Parallel Port Data FIFO

I/O Address: Base + 400h and (ECR[7:5] = 010)
Default Value: UUUU UUUU (undefined)
Attribute: Read/Write
Size: 8 bits

SDFIFO is used to transfer data from the host to the peripheral when the ECR Register is set for ISA-Compatible FIFO mode (bits[7:5] = 010). Data bytes written or DMAed from the system to this FIFO are transmitted by a hardware handshake to the peripheral using the standard ISA-Compatible protocol. Note that bit 5 in the PCON Register must be set to 0 for a forward transfer direction.




**NOTE:**
U = Undefined

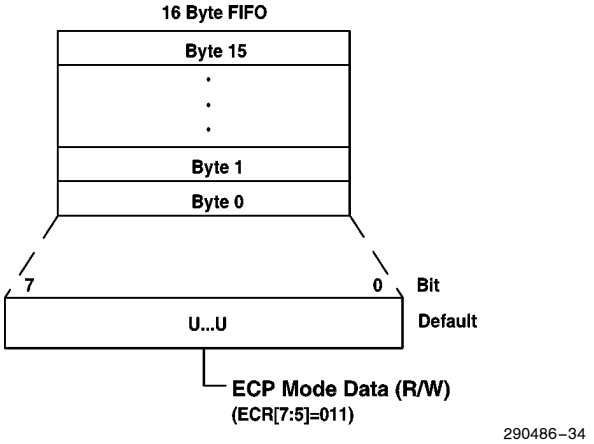
**Figure 33. ECP ISA-Compatible Data FIFO**

| Bit | Description |
|---|---|
| 7:0 | **ECP STANDARD PARALLEL PORT DATA:** Bits[7:0] correspond to SD[7:0] and PD[7:0]. |

#### 6.1.3.5 DFIFO—Data FIFO (ECP Mode)

I/O Address: Base + 400h and (ECR[7:5] = 011)
Default Value: UUUU UUUU (undefined)
Attribute: Read/Write
Size: 8 bits

This I/O address location transfers data between the host and peripheral device when the parallel port is in ECP mode (ECR Bits[7:5] = 011). Transfers use the parallel port FIFO. Data is transferred on PD[7:0] via hardware handshakes on the parallel port interface using ECP parallel port interface handshake protocol.




**NOTE:**
U = Undefined

**Figure 34. ECP Data FIFO (ECP Mode)**

| Bit | Description |
|---|---|
| 7:0 | **ECP MODE DATA:** Data bytes written or DMAed from the system to this FIFO in the forward direction (PCON bit 5 = 0) are transmitted to the peripheral by an ECP mode protocol hardware handshake. In the reverse direction (PCON bit 5 = 1) data bytes from the peripheral are transferred to the FIFO using the ECP mode protocol hardware handshake. Reads or DMAs from the FIFO return bytes of ECP data to the system. Bits[7:0] correspond to SD[7:0] and PD[7:0]. |

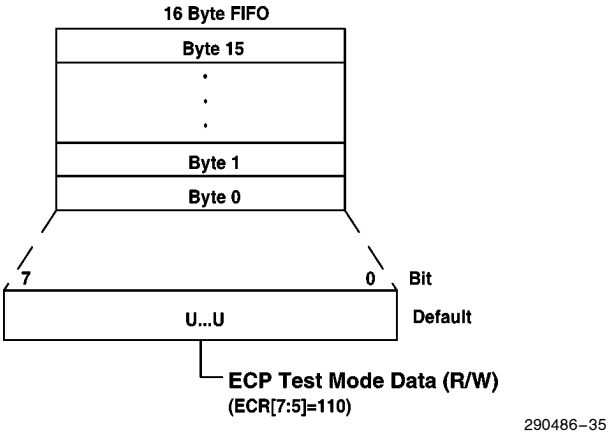
#### 6.1.3.6 TFIFO—ECP Test FIFO Register (ECP Mode)

I/O Address: Base + 400 and (ECR[7:5] = 110)
Default Value: UUUU UUUU (undefined)
Attribute: Read/Write
Size: 8 bits

The TFIFO Register provides a test mechanism for the ECP mode FIFO. Test mode is enabled via the ECR Register. In test mode (ECR[7:5] = 110), data can be read, written or DMAed to/from the FIFO by accessing this register I/O address.

Data bytes may be read, written or DMAed to or from the system to this FIFO in any direction. The parallel port interface signals are not affected by TFIFO accesses and TFIFO data is not transmitted to PD[7:0]. The test FIFO does not stall when overwritten or underrun. Data is simply re-written or over-run. The full and the empty bits in the ECR always keep track of the correct FIFO state.

The test FIFO transfers data at the maximum ISA rate so that software can generate performance metrics. The FIFO write threshold can be determined by starting with a full TFIFO and emptying it a byte at a time until a service interrupt is set to 1 in the ECR. The FIFO read threshold can be determined by setting the direction bit in the PCON Register to 1, and filling the FIFO a byte at a time until the service interrupt is set to 1 in the ECR.




**NOTE:**
U = Undefined

**Figure 35. ECP Test FIFO Register (ECP Mode)**

| Bit | Description |
|---|---|
| 7:0 | **ECP TEST FIFO** Data: Bits[7:0] correspond to SD[7:0]. |

### 6.1.3.7 ECPCFGA—ECP Configuration A Register (ECP Mode)

I/O Address: Base +400h and (ECR[7:5] = 111)
Default Value: 1001 RRRR
Attribute: Read/Write
Size: 8 bits

The ECPCFGA Register provides information about the ECP mode implementation. Access to this register is enabled by programming the ECR Register (ECR[7:5] = 111).

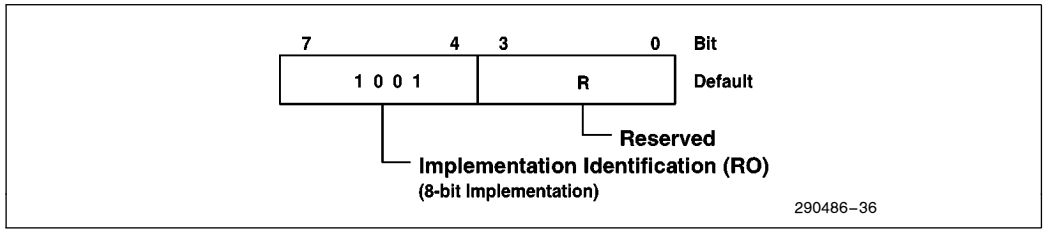



**Figure 36. ECP Configuration A Register (ECP Mode)**

| Bit | Description |
|---|---|
| 7:4 | **IMPLEMENTATION IDENTIFICATION (IMPID):** This field is hardwired to 1001 to indicate an 8-bit implementation (bit 4) and an ISA-style interrupt (bit 7). This field is read only and writes have no affect. |
| 3:0 | **RESERVED** |

#### 6.1.3.8 ECPCFGB—ECP Configuration B Register (ECP Mode)

I/O Address: Base +401h and (ECR[7:5] = 111)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ECPCFGB Register is part of the ECP specification and is implemented in the 82091AA as a scratchpad register. Software can use the fields in this register to maintain system information. Programming these bits does not affect parallel port operations. Access to the ECPCFGB Register is enabled by programming the ECR Register (ECR[7:5] = 111).

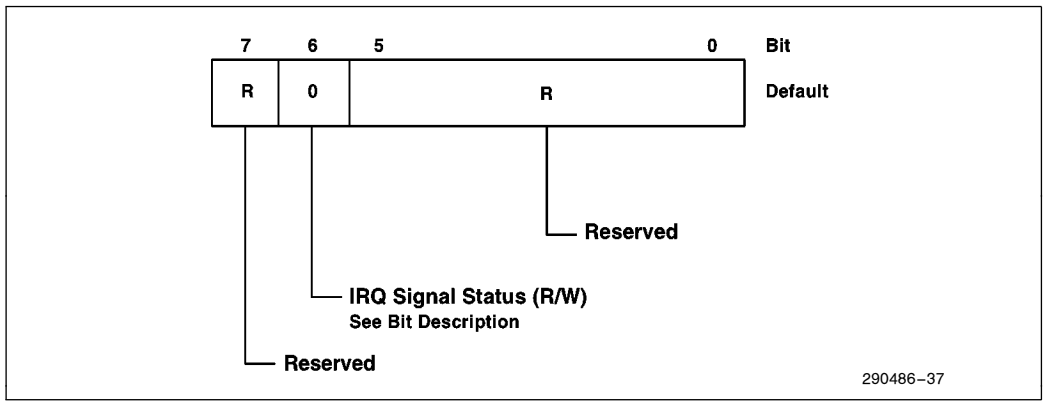



**Figure 37. ECP Extended Control Register (ECP Mode)**

| Bit | Description |
|---|---|
| 7 | **RESERVED:** This bit always reads back as 0. |
| 6 | **INTRVALUE (INTRV):** This bit returns the value on the ISA IRQ line (IRQ5/IRQ7) to determine possible conflicts. The value of either IRQ5 or IRQ7 is read back based on the parallel port interrupt selection in the 82091AA configuration registers. IRQ5/IRQ7 are tri-stated in ECP configuration mode (ECR[7:5] = 111] to allow the state of the selected parallel port interrupt line to be read back. Note that the ACKINTEN bit in the PCON register must be written to 0 before the interrupt status can be read on this bit. |
| 5:0 | **RESERVED:** These bits always read back as 0. |

#### 6.1.3.9 ECR ECP—Extended Control Register (ECP Mode)

I/O Address: Base + 402h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register selects the ECP mode, enables service and error interrupts and provides interrupt status. The ECR also enables/disables DMA operations and provides FIFO empty and FIFO full status. The FIFO empty and FIFO full status bits are also used to report FIFO overrun and underrun conditions.

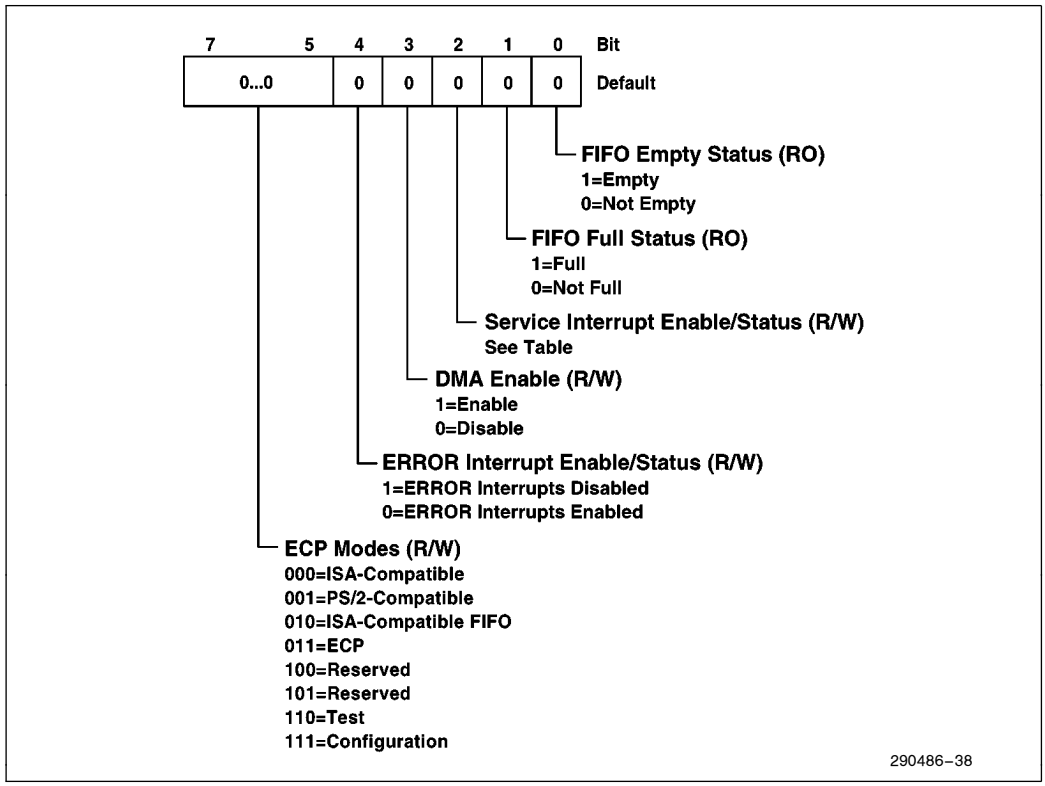



**Figure 38. ECP Extended Control Register (ECP Mode)**

| Bit | Description |
|---|---|
| 7:5 | **ECP MODE SELECT:** This field selects one of the following ECP Modes:<br>**Mode** **Operation**<br>**0 0 0** **ISA-Compatible Mode.** In this mode the parallel port operates in ISA-Compatible mode. The FIFO is reset and common collector drivers are used on the control lines (STROBE#, AUTOFD#, INIT# and SELECTIN#). Setting the direction bit to 1 in the PCON Register does not affect the parallel port interface. For register descriptions in this mode, See Section 6.1.1, ISA-Compatible and PS/2-Compatible Modes.<br>**0 0 1** **PS/2-Compatible Mode.** In this mode the parallel port operates in PS/2-Compatible mode. The FIFO is reset and common collector drivers are used on the control lines (STROBE#, AUTOFD#, INIT# and SELECTIN#). Unlike mode 000 above, setting the direction bit to 1 in the PCON Register tri-states the data lines and reading the data register returns the value on the PD[7:0]. For register descriptions in this mode, see Section 6.1.1, ISA-Compatible and PS/2-Compatible Modes.<br>**0 1 0** **ISA-Compatible FIFO Mode.** This mode is the same as mode 000 above, except that data is written or DMAed to the FIFO. FIFO data is automatically transmitted using the ISA-style protocol. For this mode, the direction control bit in the PCON register must be 0.<br>**0 1 1** **ECP Mode.** In the forward direction, bytes written to the ECP DFIFO location and bytes written to the ECP AFIFO location are placed in the ECP FIFO and transmitted automatically to the peripheral using ECP protocol. In reverse direction bytes are transferred from PD[7:0] to the ECP FIFO.<br>**1 0 0** **Reserved**<br>**1 0 1** **Reserved**<br>**1 1 0** **Test Mode.** In this mode, the FIFO may be written and read, but the data will not be transmitted on PD[7:0].<br>**1 1 1** **Configuration Mode.** In this mode, the ECP Configuration A and B Registers are accessible.<br>ECP Mode Switching Guidelines<br>Software will execute P1284 negotiations and all operation prior to a data transfer phase under programmed I/O (using mode 000 or 001). Hardware provides an automatic control line handshake, moving data between the FIFO and the ECP port only in the data transfer phase (using modes 011 or 010).<br>Setting the mode to 011 or 010 causes the hardware to initiate the data transfer.<br>If the parallel port is in mode 000 or 001, the port can be switched to any other mode. If the parallel port is not in mode 000 or 001, the port can only be switched into mode 000 or 001. The direction and the FIFO threshold can only be changed in modes 000 or 001. Note that the FIFO, FIFO Error, and TC conditions are also reset when the mode is switched to 000 or 001.<br>Once in an extended forward mode, the software should wait for the FIFO to be empty before switching back to mode 000 or 001. In this case, all control signals are negated before the mode switch. In an ECP reverse mode the software waits for all the data to be read from the FIFO before changing to mode 000 or 001. |

| Bit | Description |
|---|---|
| 4 | **ERROR INTERRUPT DISABLE (ERRINTREN):** This bit enables error interrupts to the host. In ECP Mode, When ERRINTREN = 1, interrupts are disabled. When ERRINTREN = 0, interrupts are enabled. When enabled and a high-to-low transition occurs on the FAULT# signal (FAULT# asserted), an interrupt is generated to the host. Note that if this bit is written from a 1 to a 0 while FAULT# is asserted, an interrupt is generated to the host. |
| 3 | **DMA ENABLE (DMAEN):** This bit enables/disables DMA. When DMAEN = 1, DMA is enabled and the host uses PPDREQ, PPDACK, and TC to transfer data. When DMAEN = 0, DMA is disabled and the PPDREQ output is tri-stated. In this case, programmed I/O is used to transfer data between the host and the 82091AA FIFO. The Service Interrupt (bit 2) needs to be set to 0 to allow generation of a TC interrupt. This bit must be written to 0 to reset the TC interrupt. |
| 2 | **SERVICE INTERRUPT (SERVICEINTR):** This bit enables FIFO and TC service interrupts. When the CPU writes SERVICEINTR = 1, FIFO request interrupts, FIFO error interrupts, and TC interrupts are disabled. Setting this bit to a 0 enables interrupts for one of the four cases listed below. When enabled (set to 0) and one of the four conditions below occurs, the 82091AA sets SERVICEINTR to a 1 and generates an interrupt to the host.<br>1. During DMA operations (DMAEN = 1), when terminal count is reached (TC asserted). To clear the TC interrupt, switch to ISA-Compatible or PS/2-Compatible mode (write ECR[7:5] to 000, 001) or set DMAEN to 0.<br>2. In the forward direction and DMAEN = 0, when there is a threshold number of bytes in the FIFO to be written.<br>3. In the reverse direction and DMAEN = 0, when there is a threshold number of bytes in the FIFO to be read.<br>4. In either DMA or programmed I/O mode when there is a FIFO overrun or underrun.<br>Reading the SERVICEINTR bit indicates the presence of an active interrupt when this bit has been written to a 0 prior to reading it back. To disable interrupts, the SERVICEINTR bit must be explicitly written to a 1.<br>**NOTE:**<br>The ACK# and FAULT# interrupts can be generated independent of the value of the SERVICEINTR bit. ACK# and FAULT# interrupts are enabled via the ACKINTREN bit in the PCON Register and the ERRINTREN bit in the ECR Registers, respectively. The parallel port IRQ output (IRQ5/IRQ7) is enabled when ACKINTREN = 1 in the PCON Register or when ECR[7:5] = 010, 011, or 110. Otherwise, the IRQ output is tri-stated. |
| 1 | **FIFO FULL STATUS (FIFOFS):** This bit indicates when the FIFO is full. When FIFOFS = 1 (and FIFOES = 0), the FIFO is full and cannot accept another byte of data. When FIFOFS = 0, at least one byte location is free in the FIFO. This bit is read only and writes have no affect. When a FIFO overrun or underrun occurs, the 82091AA sets both FIFOES and FIFOFS to 1. To clear the FIFO error condition interrupt, swiitch the parallel port mode from ECP (011) to either ISA-Compatible or PS/2-Compatible modes (000 or 001). |
| 0 | **FIFO EMPTY STATUS (FIFOES):** This bit indicates when the FIFO is empty. When FIFOES = 1 (and FIFOFS = 0), the FIFO is empty. When FIFOES = 0, the FIFO contains at least one byte. This bit is read only and writes have no affect. When a FIFO overrun or underrun occurs, the 82091AA sets both FIFOES and FIFOFS to 1. To clear the FIFO error condition interrupt, swiitch the parallel port mode from ECP (011) to either ISA-Compatible or PS/2-Compatible modes (000 or 001). |

## 6.2 Parallel Port Operations

The parallel port can be placed in ISA-Compatible, PS/2-Compatible, or EPP mode by hardware configuration or by writing to the 82091AA's configuration registers (PCFG1 Register). If access to the configuration registers is not disabled by hardware configuration, a hardware configured parallel port mode can be changed by programming the PCFG1 Register.

ECP mode is entered by programming the ECP Extended Control Register (ECR). Writing to this register changes any previously selected parallel port mode (via hardware configuration or writing the PCFG1 Register) to one of the ECP ECR Register modes selected via ECR[7:5]. Note that ECP mode cannot be entered by hardware configuration or programming the 82091AA configuration registers.

### 6.2.1 ISA-COMPATIBLE AND PS/2-COMPATIBLE MODES

The ISA-Compatible mode is used for standard ISA-type parallel port interfaces. Figure 39 shows the parallel port interface for ISA-Compatible mode. STROBE#, AUTOFD#, INIT#, and SELECTIN# are controlled by software via the PCON Register and the status of SELECT, PERROR, FAULT, ACK#, and BUSY are reported in the PSTAT Register. PD[7:0] are outputs only. Note that for a reverse data transfer using the Nibble protocol, the peripheral device supplies data, 4 bits at a time, using the BUSY, SELECT, PERROR, and FAULT# signals.

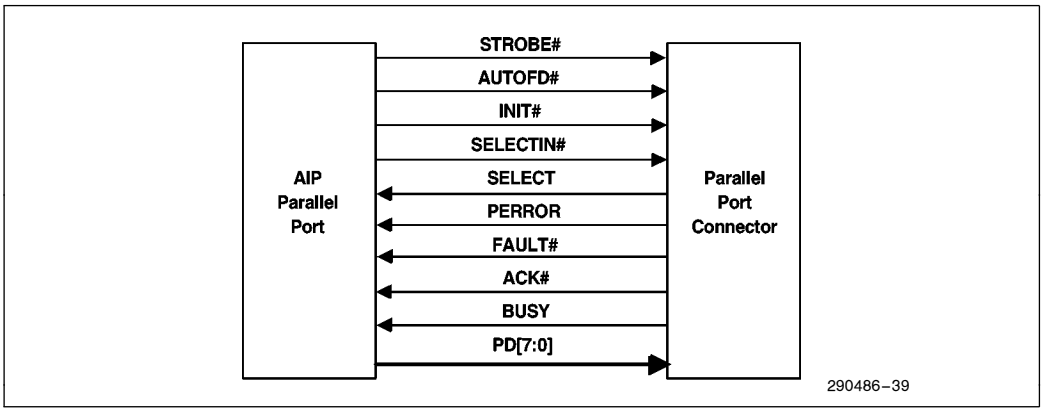



**Figure 39. ISA-Compatible Mode**

The following general protocol is used when communicating with a printer or other parallel port device.

Software selects the peripheral device by asserting the SELECTIN# signal. The peripheral device, in turn, acknowledges that it is selected by asserting the SELECT signal. The INIT# signal is used to initialize the peripheral device. If an error is encountered during initialization or normal operations, the peripheral device asserts FAULT#. If a printer (or plotter) encounters problems in the paper path, the device asserts PERROR. Other peripheral devices may not use the PERROR signal.

During normal operation, the peripheral device asserts BUSY when it is not ready to receive data. When it has finished processing the previous data, the peripheral device asserts ACK# and negates BUSY. If interrupts are enabled, a low-to-high transition on ACK# when the signal is negated generates an interrupt. If interrupts are not enabled, software must poll the PSTAT Register to determine when ACK# is pulsed.

The parallel port driver software sends data to the peripheral device by writing to the PDATA Register and asserting the STROBE# signal after an appropriate data stabilization interval. After a sufficient setup time has elapsed, software then negates STROBE#. Valid data is read by the peripheral device.

In the PS/2-Compatible mode, data can be written to or read from the parallel port. Figure 40 shows the parallel port interface for PS/2-Compatible mode. The Byte protocol signal names are shown in parenthesis. Before reading or writing the PDATA Register, the direction control bit in the PCON Register must be set to the proper transfer direction on PD[7:0]. During a write to the PDATA Register (with DIR# = 0), data is latched by the PDATA Register and driven onto PD[7:0]. During a parallel port read of the PDATA Register (with DIR# = 1), the data on PD[7:0] is driven onto SD[7:0]. The data is not latched by the PDATA Register during the read cycle.

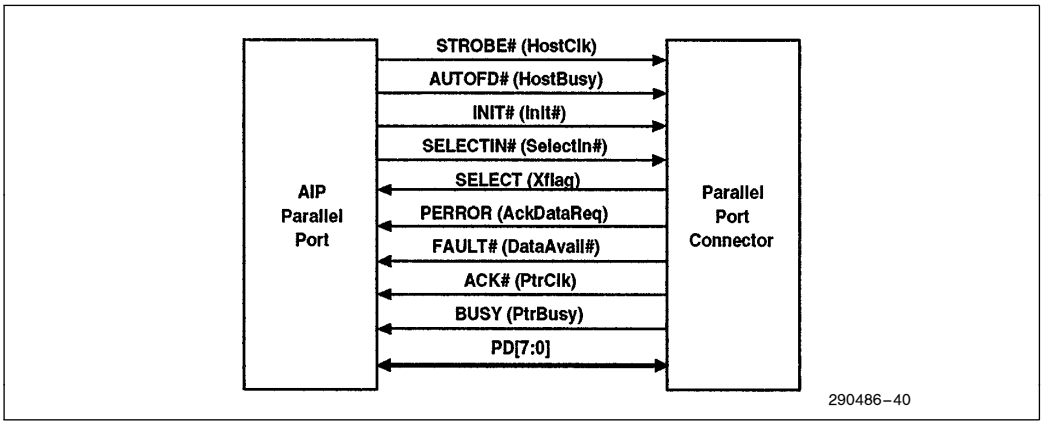



**Figure 40. PS/2-Compatible Mode**

### 6.2.2 EPP MODE

The 82091AA is EPP 1.7 compliant. This means EPP cycles will begin with WAIT (Busy) inactive, however, WAIT will still prolong the cycle when active. Figure 41 shows the parallel port interface for EPP mode. The EPP parallel port interface protocol signal names are shown in parenthesis. In EPP mode, the initialization, printer selection, and error signals (PERROR and FAULT#) work the same way as in the ISA-Compatible mode. However, in EPP mode, SELECTIN# and AUTOFD# are automatically generated and become Address Strobe (AStrb#) and Data Strobe (DStrb#), respectively, enabling direct access to parallel port devices. STROBE (Write#) is used to indicate a read or write cycle. Note that BUSY (Wait) is an active low signal in EPP mode rather than an active high signal as in ISA-Compatible mode. In addition, BUSY, in combination with IOCHRDY on the ISA Bus extends clock cycles to enable the completion of a read or write without additional wait states. EPP write and read cycles are shown in Figure 42 and Figure 43.

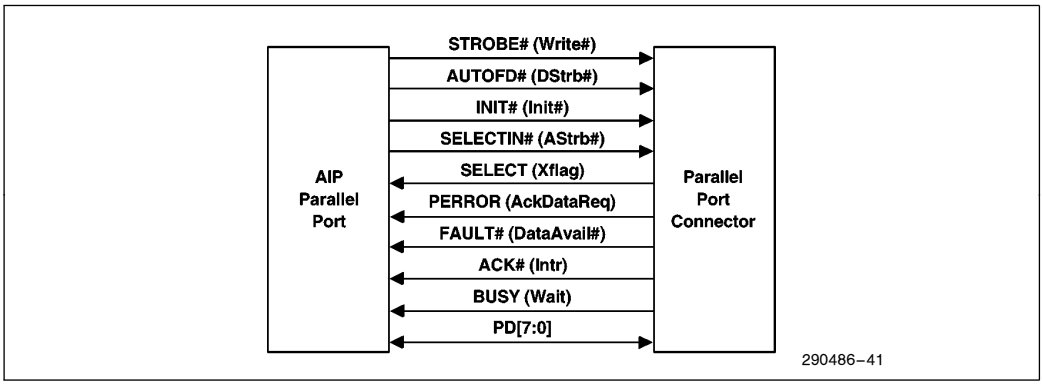



**Figure 41. EPP Mode**

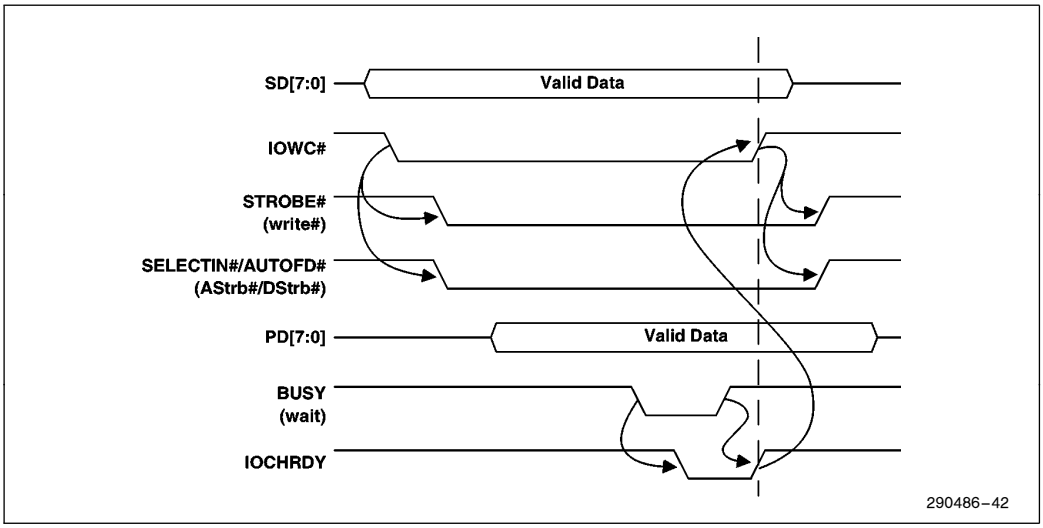



**Figure 42. EPP Mode Write Cycle**

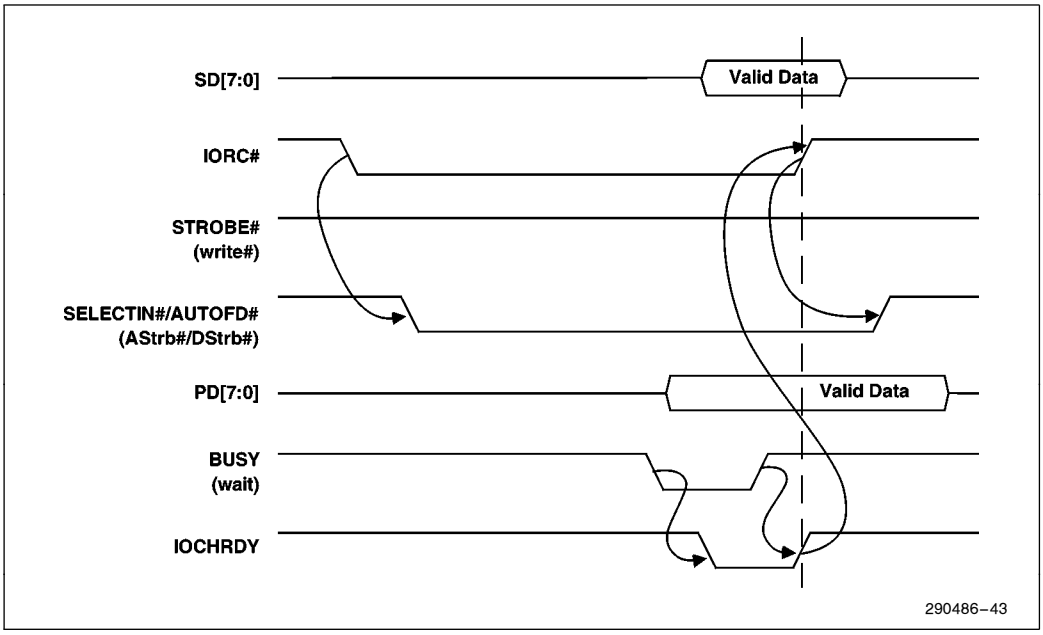



**Figure 43. EPP Mode Read Cycle**

### 6.2.3 ECP MODE

Figure 44 shows the parallel port interface for ECP mode with the ECP protocol signal names in parenthesis. The ECP modes are selected by programming the Extended Control Register (ECR bits[7:5]). Two of the modes (Test and Configuration) provide information about the 82091AA's parallel port and are not used for interfacing with a peripheral device. Four peripheral interface modes are selectable via the ECR—ISA-Compatible mode, ISA-Compatible FIFO mode, PS/2-Compatible mode, and ECP mode.

**ISA-Compatible and PS/2-Compatible Modes (ECR[7:5] = 000,001)**

The ISA-Compatible and PS/2-Compatible mode selections in the ECR are equivalent to selecting these modes via hardware configuration or programming the PCFG1 Register. For these modes the operation is the same as described in Section 6.2.1, ISA-Compatible and PS/2-Compatible Modes.

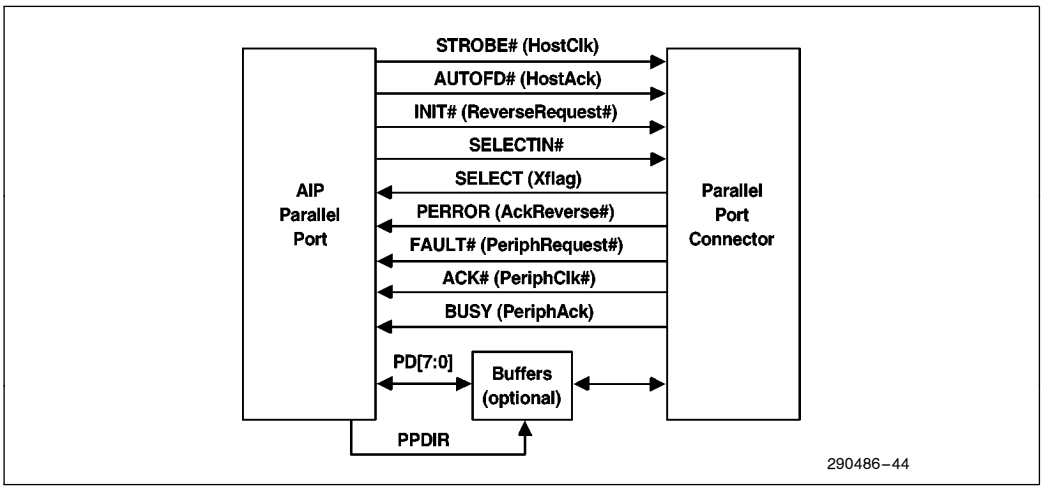



**Figure 44. ECP Mode**

### ISA-Compatible FIFO Mode (ECR[7:5] = 010)

The ISA-Compatible FIFO mode uses the same signaling protocol on the parallel port interface as the ISA-Compatible mode. However, there are two major operational differences. First, data is written to a 16-byte FIFO (via the SDFIFO address location). The FIFO empty and FIFO full bits in the ECR provide FIFO status. In addition, DMA can be used to transfer data to the FIFO by enabling this feature in the ECR.

Second, the data is transferred to the peripheral using an automatic hardware handshake. This handshake emulates the standard ISA-Compatible style software generated handshake (Figure 45). For ISA-Compatible FIFO mode, the 82091AA does not monitor the ACK# signal. Service interrupts are enabled and reported via the ECR. The generation of service interrupts is based on the state of the FIFO and not individual transfers (using ACK#) as is the case in standard ISA-Compatible mode.

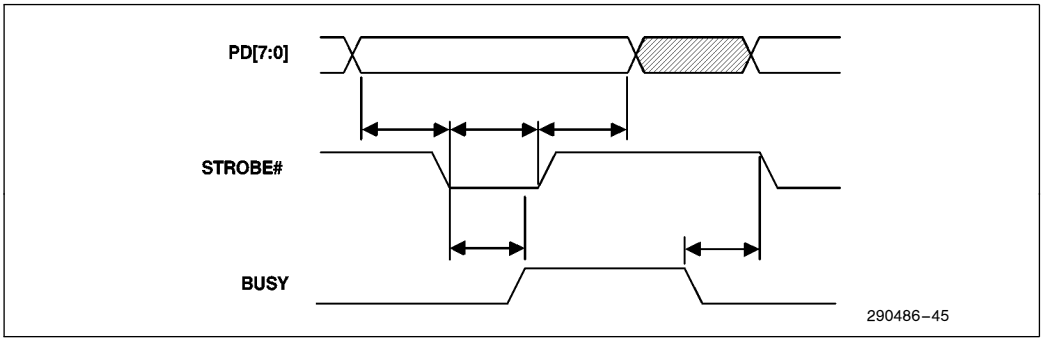



**Figure 45. ISA-Compatible Timing**

### ECP Mode (ECR[7:5] = 011)

When ECR[7:5] = 011, the parallel port operates in ECP mode. In ECP mode, both data and commands (addresses and RLE) are transferred using the parallel port 16-byte FIFO. This information can be either written to or read from the FIFO using DMA or non-DMA ISA Bus transfers. The parallel port interface transfers use an automatic handshake generated by the 82091AA. The host controls the transfer direction by programming the DIR# bit in the PCON Register.

When the host is writing to the peripheral device (forward direction), STROBE#, and BUSY provide the automatic handshake for transfer on the parallel port interface (Figure 46). The peripheral device negates BUSY when it is ready to receive data or commands. AUTOFD# indicates whether PD[7:0] contain data (AUTOFD# is high) or a command (AUTOFD# is low). For commands (address or RLE), the host writes to the ECPAFIFO Register I/O address and for data, the host writes to the DFIFO Register I/O address. The addresses and data are placed in the same 16-byte FIFO. When the FIFO is full and cannot accept more data/addresses, the FIFO Full status bit is set in the ECR.

Data/addresses written to the FIFO are transferred to the peripheral device via PD[7:0]. To begin a transfer on the peripheral interface, the 82091AA checks BUSY to make sure the peripheral is in the ready state. If BUSY is negated, the 82091AA drives PD[7:0] and AUTOFD#, and asserts STROBE# to indicate that the data/command is on PD[7:0]. The peripheral device asserts BUSY to indicate that it is receiving the data/command. BUSY asserted causes the 82091AA to negate STROBE#.

When the host is reading from the peripheral device (reverse direction), AUTOFD# and ACK# provide the automatic handshake for transfer on the parallel port interface (Figure 47). Data/commands from the peripheral device are placed in the parallel port FIFO using this handshake. In this case, BUSY indicates whether PD[7:0] contain data (BUSY is high) or a command (BUSY is low).

The peripheral device asserts ACK# to indicate that a data/command is on PD[7:0]. The 82091AA negates AUTOFD# when it is ready for a peripheral transfer and asserts AUTOFD# to indicate that it is receiving the data/command. AUTOFD# asserted causes the peripheral device to negate ACK#. The peripheral transfers are to the parallel port 16-byte FIFO.

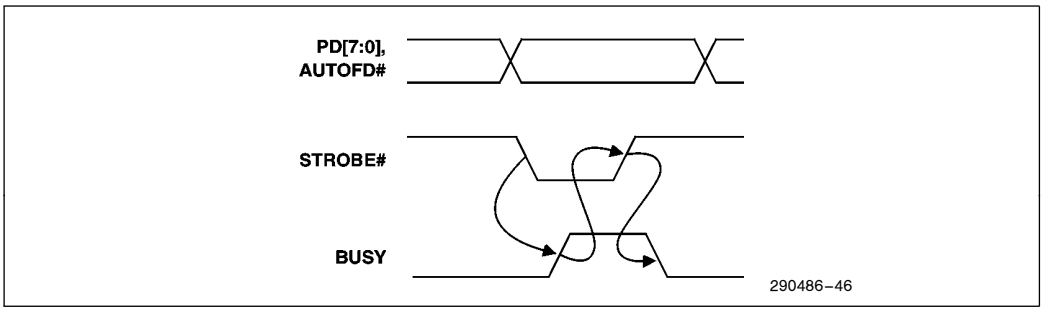



**Figure 46. ECP Mode Handshake (Forward Direction)**

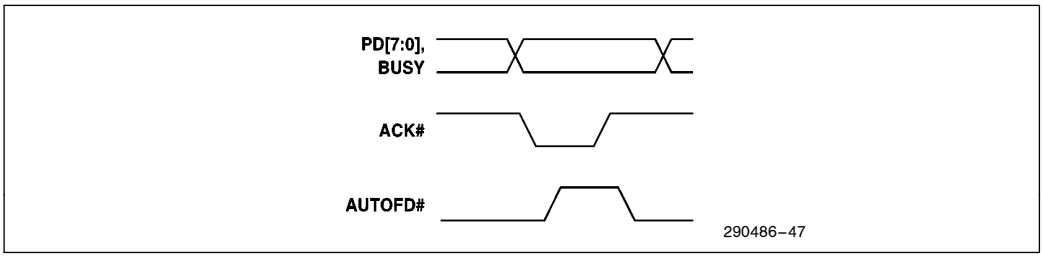



**Figure 47. ECP Mode Handshake (Reverse Direction)**

**Test Mode (ECR[7:5] = 110) and Configuration Mode (ECR7:5] = 111)**

The test mode can be used to check the FIFO read and write interrupt thresholds as described in Section 6.1.3.7, TFIFO—ECP Test FIFO Register. Note that for the 82091AA parallel port, the read and write FIFO interrupt thresholds are the same. The FIFO threshold is set by programming the PCFG1 Register in the 82091AA configuration space. The configuration mode is used to access the ECPCFGA and ECPCFGB Registers. This mode must first be set before the ECPCFGA and ECPCFGB Registers can be accessed.

#### 6.2.3.1 FIFO Operations

The parallel port FIFO is used for ECP transfers (ECR[7:5] = 011), ISA-Compatible FIFO transfers (ECR[7:5] = 010), and Test mode (ECR[7:5] = 110). Either DMA or programmed I/O can be used for transfers between the host and the parallel port.

The FIFO threshold value is selected via the 82091AA configuration registers (PCFG1 Register). The threshold is set to either 1 (forward)/15 (reverse) or 8 in both directions. A threshold setting of 1 (forward)/15 (reverse) results in longer periods of time between service request, but requires faster servicing of both read and write requests. A threshold setting of 8 results in more service requests, but tolerates slower servicing of the requests.

In modes 010 and 011, an internal temporary holding register is used in conjunction with the 16-byte FIFO to provide 17 bytes of storage for both forward and reverse transfers. Thus, in the forward direction if the peripheral asserts the BUSY signal during the filling of the FIFO, the host needs to write 17 bytes before the FIFO full flag in the ECR is set to 1. In Test mode (110) only the 16-byte FIFO is used and the temporary holding register is not used.

The FIFO is reset by a hard reset (RSTDRV asserted) or whenever the parallel port is placed in ISA-Compatible or PS/2-Compatible modes. Note that the FIFO threshold can only be changed when the parallel port is in ISA-Compatible or PS/2-Compatible mode.

#### 6.2.3.2 DMA Transfers

The 82091AA contains parallel port DMA request (PPDREQ) and acknowledge (PPDACK#) signals to communicate with a standard PC DMA controller. Before initiating a DMA transfer the direction bit in the PCON Register must be set to the proper direction. To initiate DMA transfers, software sets the DMAEN bit to 1 and the SERVICEINTR bit to 0 in the ECR. The PPDREQ and PPDACK# signals will then be used to fill (forward direction) or empty (reverse direction) the FIFO. When the DMA controller reaches terminal count and asserts the TC signal, an interrupt is generated and the SERVICEINTR bit is set to 1. To reset the TC interrupt, software can either switch the mode to 000 or 001 or write the DMAEN bit to 0.

In DMA mode, if 32 consecutive reads or writes are performed to the FIFO and PPDREQ is still asserted, the 82091AA negates PPDREQ for the length of the last PPDACK#/command pulse to force an arbitration cycle on the ISA Bus.

#### 6.2.3.3 Reset FIFO and DMA Terminal Count Interrupt

The following operations are used to reset the parallel port FIFO and TC interrupt

| Function | Reset Operations |
|---|---|
| FIFO | -Changing to modes 000 or 001<br>-Hard reset |
| FIFO Error | -Changing to modes 000 or 001<br>-Hard reset |
| TC Interrupt | -Changing to modes 000 or 001<br>-Setting DMAEN to 0 in ECR<br>-Hard reset |

#### 6.2.3.4 Programmed I/O Transfers

Programmed I/O (non-DMA) can also be used for transfers between the host and the parallel port FIFO. Software can determine the read/write FIFO thresholds and the FIFO depth by accessing the FIFO in Test mode. To use programmed I/O transfers software sets the direction bit in the PCON Register to the desired direction and sets the DMAEN bit to 0 and the SERVICEINTR bit to 0 in the ECR. The parallel port requests programmed I/O transfers from the host by asserting IRQ5/IRQ7.

In the reverse direction an interrupt occurs when SERVICEINTR = 0 either 8 or 15 bytes (depending on threshold setting) are in the FIFO. IRQ5/IRQ7 can be used in an interrupt-driven system. The host must respond to the interrupt request by reading data from the FIFO.

In the forward direction an interrupt occurs when SERVICEINTR = 0 and there are either 8 or 1 byte locations available in the FIFO (depending on threshold setting). IRQ5/IRQ7 can be used in an interrupt-driven system. The host must respond to the interrupt request by writing data to the FIFO.

#### 6.2.3.5 Data Compression

The 82091AA supports Run Length Encoded (RLE) decompression in hardware and can transfer compressed data to the peripheral. To transfer compressed data to the peripheral (forward direction), the compression count is written to the ECPAFIFO location and the data is written to the ECPDFIFO location. The most significant bit (bit 7) in the byte written to the ECPAFIFO Register informs the peripheral whether the value is a channel address (bit 7 = 1) or a run length count (bit 7 = 0). The RLE count in the ECPAFIFO (bits[6:0]) informs the peripheral of how many times the data in the ECPDFIFO is to be repeated. An RLE count of 0 indicates that only one byte of the data is present and a count of 127 indicates to the peripheral that the next byte should be expanded 128 times. An RLE count of 1 should be avoided as it will cause unnecessary expansions. Note that the 82091AA asserts AUTOFD# to indicate that PD[7:0] contains address/RLE instead of data.

In the reverse direction, the peripheral negates the BUSY signal to indicate that PD[7:0] contains address/RLE. During an address/RLE cycle, the 82091AA checks bit 7 to see if the next byte received needs to be decompressed. If bit 7 is 0, the 82091AA decompresses (replicates) the next data received by the RLE count received on bits[6:0].

### 6.2.4 PARALLEL PORT EXTERNAL BUFFER CONTROL

A multi-function signal (GCS#/PPDIR) is provided for controlling optional external parallel port data buffers. The PPDIR function is only available when the 82091AA configuration is in software motherboard (SWMB) mode. In this mode, this signal operates as a parallel port direction control signal (PPDIR). Note that, if any other configuration is used (SWAI, HWB, or HWE configuration modes), this multi-function signal operates as a game port chip select (GCS#). In SWMB, PPDIR is low when PD[7:0] are outputs and high when PD[7:0] are inputs. Figure 44 shows an example of external buffers being used when the parallel port is in ECP mode.

External buffering affects the ability of the port to read software security devices. Typically these software secutiry devices are designed to hold the data pins of the parallel port connector at either high or low logic levels when the pins are not being driven by the parallel port. The bit pattern read from the parallel port by the security software may not be correctly transferred through the external buffer.

### 6.2.5 PARALLEL PORT SUMMARY

Table 18 summarizes the parallel port interrupt, DMA, and parallel port signal drive type for the various modes of operation.

**Table 18. Parallel Port Summary**

| Parallel Port Mode | ECR[7:5] | PD[7:0] Direction | Parallel Port Control Signals Controlled By PCON | IRQ Enable | DMA Enable |
|---|---|---|---|---|---|
| ISA-Compatible | 000 | Output | Open Drain | ACKINTEN | |
| PS/2-Compatible | 001 | Bi-directional | Open Drain | ACKINTEN | |
| EPP | N/A | Bi-directional | Push Pull | ACKINTEN | |
| ISA-Compatible FIFO | 010 | Output | Push Pull | Always Enabled | DMAEN |
| ECP | 011 | Bi-directional | Push Pull | Always Enabled | DMAEN |
| ECP Test | 110 | Output | Push Pull | Always Enabled | DMAEN |
| ECP Configuration | 111 | Bi-directional | Push Pull | ACKINTEN | DMAEN |

**NOTES:**
1. The selected IRQ pin (IRQ5/IRQ7) is enabled if ACKINTEN is enabled in the PCON Register. Otherwise, the IRQ pin is tri-stated.
2. PPDREQ is enabled whenever the DMAEN bit is enabled in the ECR, independent of the parallel port mode.

## 7.0 SERIAL PORT

The two 82091AA serial ports are identical. This section describes the serial port registers and FIFO operations.

## 7.1 Register Description

The register descriptions in this section apply to both serial port A and serial port B and provide a complete operational description of the serial ports. Table 19 shows the I/O address assignments for the serial port registers. The individual register descriptions follow in the order that they appear in the table. Note that serial port interrupt assignments (IRQ3 or IRQ4) and the base address assignments are made by 82091AA configuration as described in Section 4.0, AIP Configuration.

All registers are accessed as byte quantities. The base address is determined by hardware configuration at powerup (or a hard reset) or via software configuration by programming the 82091AA configuration registers as described in Section 4.0, AIP Configuration. Note that access to certain serial port registers requires prior programming of the DLAB bit in the Line Control Register (LCR).

During a hard reset (RSTDRV asserted), the 82091AA registers are set to pre-determined **default** states. The default values are indicated in the individual register descriptions. Reserved bits in the 82091AA's serial port registers must be programmed to 0 when writing the register and these bits are 0 when read. The following bit notation is used for default settings:

**X** Default bit position value is determined by conditions on an 82091AA signal pin.

The following nomenclature is used for serial port register access attributes:

**RO** **Read Only**. Note that for all registers with read only attributes, writes to the I/O address access a different register.

**WO** **Write Only**. Note that for all registers with write only attributes, reads to the I/O address access a different register.

**R/W** **Read/Write**. A register with this attribute can be read and written. Note that some read/write registers contain bits that are read only.

**Table 19. Serial Port Registers**

| Register Address Access (AEN = 0) | | Abbreviation | Register Name | Access |
|---|---|---|---|---|
| Base + | DLAB | | | |
| 0h | 0 | THR | Transmit Holding Register | WO |
| 0h | 0 | RBR | Receiver Buffer Register | RO |
| 0h | 1 | DLL | Divisor Latch LSB | R/W |
| 1h | 1 | DLM | Divisor Latch MSB | R/W |
| 1h | 0 | IER | Interrupt Enable Register | R/W |
| 2h | — | IIR | Interrupt Identification Register | RO |
| 2h | — | FCR | FIFO Control Register | WO |
| 3h | — | LCR | Line Control Register | R/W |
| 4h | — | MCR | Modem Control Register | R/W |
| 5h | — | LSR | Line Status Register | R/W |
| 6h | — | MSR | Modem Status Register | R/W |
| 7h | | SCR | Scratch Pad Register | R/W |

**Table 20. Serial Port Register Summary**

| Bit # | Receiver Buffer Register | Transmitter Holding Register | Interrupt Enable Register | Interrupt Identification Register | FIFO Control Register | Line Control Register |
|---|---|---|---|---|---|---|
| 0 | Data Bit 0 | Data Bit 0 | Enable Received Data Available Interrupt | 0 if Interrupt Pending | FIFO Enable | Word Length Select Bit 0 |
| 1 | Data Bit 1 | Data Bit 1 | Enable XMTR Holding Register Empty Interrupt | Interrupt ID Bit | RCVR FIFO Reset | Word Length Select Bit 1 |
| 2 | Data Bit 2 | Data Bit 2 | Enable RCVR Line Status Interrupt | Interrupt ID Bit | XMIT FIFO Reset | Number of Stop Bits |
| 3 | Data Bit 3 | Data Bit 3 | Enable Modem Status Interrupt | Interrupt ID Bit (Non-FIFO = 0) | DMA Mode Select | Parity Enable |
| 4 | Data Bit 4 | Data Bit 4 | 0 | 0 | Reserved | Event Parity Select |
| 5 | Data Bit 5 | Data Bit 5 | 0 | 0 | Reserved | Stick Parity |
| 6 | Data Bit 6 | Data Bit 6 | 0 | FIFOs Enabled (Non-FIFO = 0) | RCVR Trigger (LSB) | Set Break |
| 7 | Data Bit 7 | Data Bit 7 | 0 | FIFOs Enabled (Non-FIFO = 0) | RCVR Trigger (MSB) | Divisor Latch Access Bit (DLAB) |

**Table 20. Serial Port Register Summary** (Continued)

| Bit # | Modem Control Register | Line Status Register | Modem Status Register | ScratchPad Register | Divisor Latch - MSB | Divisor Latch - LSB |
|---|---|---|---|---|---|---|
| 0 | Data Terminal Ready (DTR) | Data Ready (DR) | Delta Clear to Send | Bit 0 | Bit 0 | Bit 8 |
| 1 | Request to Send (RTS) | Overrun Error (OE) | Delta Data Set Ready | Bit 1 | Bit 1 | Bit 9 |
| 2 | Out 1 Bit | Parity Error (PE) | Trailing Edge Ring Indicator | Bit 2 | Bit 2 | Bit 10 |
| 3 | IRQ Enable | Framing Error (FE) | Delta Data Carrier Detect | Bit 3 | Bit 3 | Bit 11 |
| 4 | Loop | Break Interrupt (BI) | Clear to Send (CTS) | Bit 4 | Bit 4 | Bit 12 |
| 5 | 0 | Transmitter Holding Register (THRE) | Data Set Ready (DSR) | Bit 5 | Bit 5 | Bit 13 |
| 6 | 0 | Transmitter Empty (TEMT) | Ring Indicator (RI) | Bit 6 | Bit 6 | Bit 14 |
| 7 | 0 | Error in RCVR FIFO | Data Carrier Detect (DCD) | Bit 7 | Bit 7 | Bit 15 |

### 7.1.1 THR(A,B)—TRANSMITTER HOLDING REGISTER

I/O Address: Base + 0h (DLAB = 0)
Default Value: 00h
Attribute: Write Only
Size: 8 bits

The THR contains data to be transmitted out on the SOUT[A,B] signal line. Bit 0 is the least significant bit and is the first bit serially transmitted. If the serial word length is less than 8 bits (as selected in the LCR), the data word must be written to this register right-justified. Bit positions above the number of bits selected for the word size are discarded (not transmitted).

| Bit | Description |
|---|---|
| 7:0 | **Transmit Data:** Bits[7:0] correspond to SD[7:0]. |

### 7.1.2 RBR(A,B)—RECEIVER BUFFER REGISTER

I/O Address: Base + 0h (DLAB = 0)
Default Value: 00h
Attribute: Read Only
Size: 8 bits

The RRB contains data received from the SIN[A,B] signal line. Bit 0 is the least significant bit and is the first bit serially received. If the serial word length is less than 8 bits (as selected in the LCR), the data word in this register is right-justified. Bit positions above the number of bits selected for the word size are 0.

| Bit | Description |
|---|---|
| 7:0 | **Receiver Data:** Bits[7:0] correspond to SD[7:0]. |

### 7.1.3 DLL(A,B), DLM(A,B)—DIVISOR LATCHES (LSB AND MSB) REGISTERS

I/O Address: Base + 0h,1h (DLAB = 1)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The 82091AA contains two independently programmable baud rate generators. The 24 MHz crystal oscillator frequency input is divided by 13, resulting in a frequency of 1.8462 MHz. This frequency is the input to each baud rate generator and is divided by the divisor of the associated serial port. The output frequency of the baud rate generator (BOUT[A,B]) is 16 x the baud rate.

$$\text{divisor } \# = (\text{frequency input})/(\text{baud rate} \times 16)$$

The output of each baud rate generator drives the transmitter and receiver sections of the associated serial port. Two 8-bit latches per serial port store the divisor in a 16-bit binary format. These divisor latches must be loaded during initialization to ensure proper operation of the baud rate generator. Upon loading either of the divisor latches, a 16-bit baud counter is loaded. Table 21 provides decimal divisors to use with crystal frequencies of 24 MHz. Using a divisor of zero is not recommended.

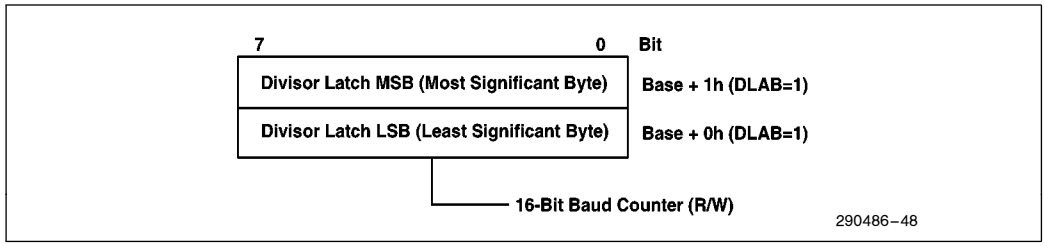



**Figure 48. Divisor Latches (LSB and MSB) Registers**

| Bit | Description |
|---|---|
| 7:0 | **Divisor Latch Data:** Bits[7:0] correspond to SD[7:0]. |

**Table 21. AIP Serial Port A and B Divisors, Baud Rates, and Clock Frequencies**

| | 24 MHz Input Divided to 1.8461 MHz | |
|---|---|---|
| **Baud Rate** | **Decimal Divisor for 16x Clock** | **Percent Error** |
| 50 | 2304 | 0.1 |
| 75 | 1536 | |
| 110 | 1047 | |
| 134.5 | 857 | 0.4 |
| 150 | 768 | — |
| 300 | 384 | — |
| 600 | 192 | — |
| 1200 | 96 | — |
| 1800 | 64 | — |
| 2000 | 58 | 0.5 |
| 2400 | 48 | — |
| 3600 | 32 | — |
| 4800 | 24 | — |
| 7200 | 16 | — |
| 9600 | 12 | — |
| 19200 | 6 | — |
| 38400 | 3 | — |
| 56000 | 2 | 3.0 |
| 115200 | 1 | — |

### 7.1.4 IER(A,B)—INTERRUPT ENABLE REGISTER

I/O Address: Base + 1h (DLAB = 0)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register enables/disables interrupts for five types of serial port conditions. If a particular condition occurs whose interrupt is disabled in this register, the corresponding interrupt status bit in the IIR will not be set and an interrupt request (IRQ3 or IRQ4) will not be generated.

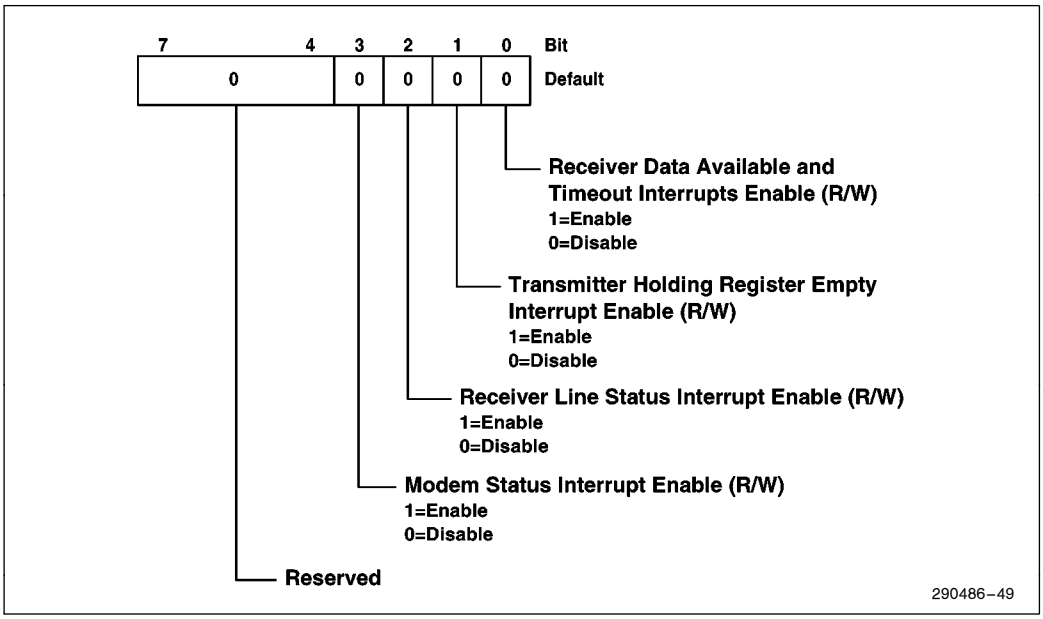



**Figure 49. Interrupt Enable Register**

| Bit | Description |
|---|---|
| 7:4 | **RESERVED** |
| 3 | **MODEM INTERRUPT ENABLE (MIE):** When MIE = 1, the Modem Status Interrupt is enabled. When MIE = 0, the Modem Status Interrupt is disabled. |
| 2 | **RECEIVER INTERRUPT ENABLE (RIE):** When RIE = 1, the Receiver Line Status interrupt is enabled. When RIE = 0, the receiver line status interrupt is disabled. |
| 1 | **TRANSMITTER HOLDING REGISTER EMPTY INTERRUPT ENABLE (THEIE):** When THREIE = 1, the Transmitter Holding Register Empty Interrupt is enabled. When THREIE = 0, the Transmitter Holding Register Empty Interrupt is disabled. |
| 0 | **RECEIVER DATA AVAILABLE INTERRUPT ENABLE AND TIMEOUT INTERRUPT ENABLE IN FIFO MODE (RAVIE):** When RAVIE = 1, the Received Data Available Interrupt is Enabled. When RAVIE = 0, the Received Data Available Interrupt is disabled. In addition, in the FIFO Mode, this bit enables the Timeout Interrupt when set to 1 and disables the Timeout Interrupt when set to 0. |

### 7.1.5 IIR(A,B)—INTERRUPT IDENTIFICATION REGISTER

I/O Address: Base + 2h
Default Value: 01h
Attribute: Read Only
Size: 8 bits

This register provides interrupt status and indicates whether the serial port receive/transmit FIFOs are enabled (FIFO mode) or disabled (non-FIFO mode). In order to provide minimum software overhead during data character transfers, the serial port prioritizes interrupts into four levels and records these in the Interrupt Identification Register. The four levels of interrupt conditions in order of priority are Receiver Line Status; Received Data Ready; Transmitter Holding Register Empty; and Modem Status. When the CPU accesses the IIR, the serial port freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this CPU access is occurring, the serial port records new interrupts, but does not change its current indication until the current access is complete.

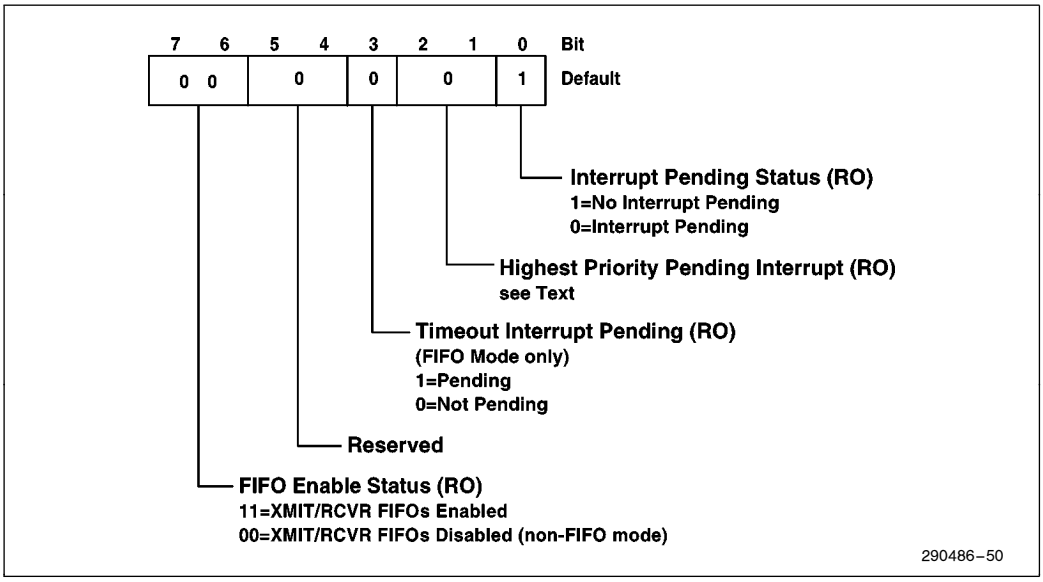



**Figure 50. Interrupt Identification Register**

| Bit | Description |
|---|---|
| 7:6 | **FIFO MODE ENABLE STATUS (FIFOES):** This status field indicates whether the serial port is in FIFO mode or non-FIFO mode (FIFO/non-FIFO mode is selected via the FCR). When FIFOES = 11, the serial port is in FIFO mode (FIFOs enabled). When FIFOES = 00, the serial port is in non-FIFO mode (FIFOs disabled). The 82091AA never sets this field to either = 01 or 10. |
| 5:4 | **RESERVED** |
| 3 | **TIMEOUT INTERRUPT PENDING (TOUTIP)—FIFO MODE ONLY:** In the non-FIFO mode, this bit is 0. In FIFO mode TOUTIP is set to 1 when no characters have been removed from or input to the receive FIFO during the last 4 character times and there is at least 1 character in the FIFO during this time. When a timeout interrupt is pending, the 82091AA sets this bit along with bit 2 of this register. |
| 2:1 | **HIGHEST PRIORITY INTERRUPT INDICATOR:** This field identifies the highest priority interrupt pending as indicated in Table 22. |
| 0 | **INTERRUPT PENDING STATUS (IPS):** This bit can be used in an interrupt environment to indicate whether an interrupt condition is pending. When IPS = 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When IPS = 1, no interrupt is pending. |

**Table 22. Interrupt Priority**

| FIFO Mode Only | Interrupt Identification Register | | | Interrupt Set and Reset Functions | | | |
|---|---|---|---|---|---|---|---|
| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset Control |
| 0 | 0 | 0 | 1 | — | None | None | — |
| 0 | 1 | 1 | 0 | Highest | Receiver Line Status | Overrun Error, Parity Error, Framing Error, or Break Interrupt | Reading the Line Status Register |
| 0 | 1 | 0 | 0 | Second | Received Data Available | Receiver Data Available | Read Receiver Buffer |
| 1 | 1 | 0 | 0 | Second | Character Timeout Indication | No Characters Have Been Removed from or Input to the RCVR FIFO during the Last 4 Char. Times and there is at least 1 Char. in it during this time | Reading the Receiver Buffer Register |
| 0 | 0 | 1 | 0 | Third | Transmitter Holding Register Empty | Transmitter Holding Register Empty | Reading the IIR Register (if Source or Interrupt) or Writing the Transmitter Holding Register |
| 0 | 0 | 0 | 0 | Fourth | Modem Status | Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect. | Reading the Modem Status Register |

### 7.1.6 FCR(A,B)—FIFO CONTROL REGISTER

I/O Address: Base + 2h
Default Value: 00h
Attribute: Write Only
Size: 8 bits

FCR is a write only register that is located at the same address as the IIR (the IIR is a read only register). FCR enables/disables the transmit/receive FIFOs, clears the transmit/receive FIFOs, and sets the receive FIFO trigger level.

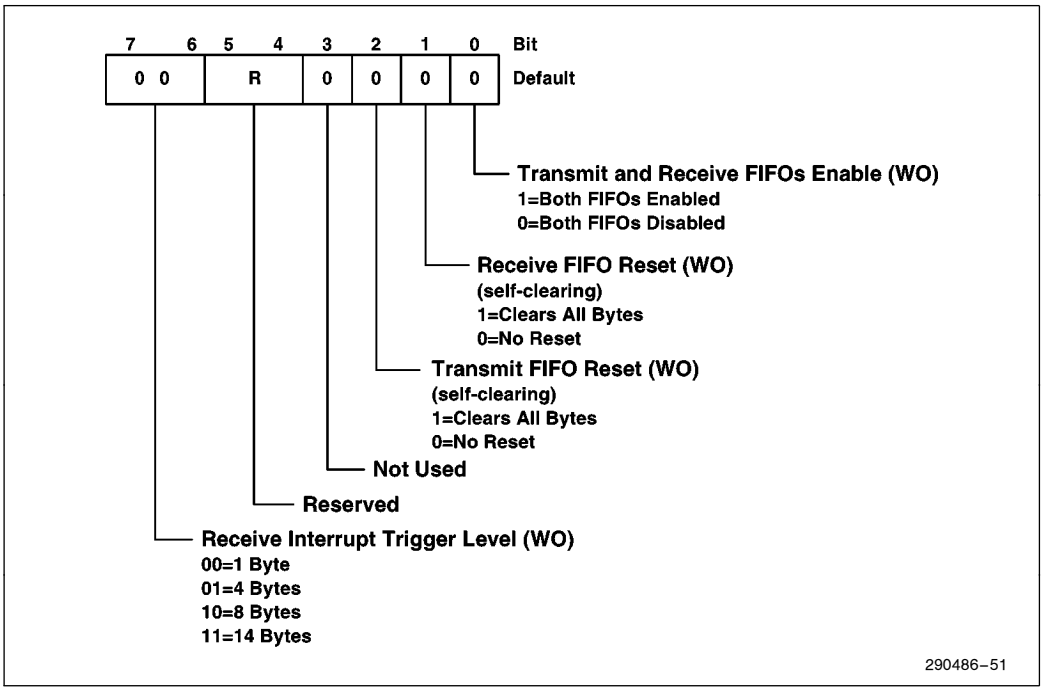



**Figure 51. FIFO Control Register**

| Bit | Description |
|---|---|
| 7:6 | **INTERRUPT TRIGGER LEVEL (ITL):** The ITL field indicates the interrupt trigger level. When the number of bytes in the receive FIFO equals the interrupt trigger level programmed into this field and the Received Data Available Interrupt enabled (via the IER), an interrupt will be generated and the appropriate bits set in the IIR.<br>**Bits [7:6] — Trigger Level (Bytes)**<br>0 0 — 01 (default)<br>0 1 — 04<br>1 0 — 08<br>1 1 — 14 |
| 5:4 | **RESERVED** |
| 3 | **NOT USED:** Writing to this bit causes no change in serial port operations. The serial port does not support DMA operations. Note that the TXRDY# and RXRDY# pins are not available in the 82091AA. |
| 2 | **RESET TRANSMITTER FIFO (RESETTF):** When RESETTF is set to a 1, the FIFO counter is set to 0. The shift register is not cleared. When the FIFO is cleared, the 82091AA sets this bit to 0. |
| 1 | **RESET RECEIVER FIFO (RESETRF):** When RESETRF is set to a 1, the FIFO counter is set to 0. The shift register is not cleared. When the FIFO is cleared, the 82091AA sets this bit to 0. |
| 0 | **TRANSMIT AND RECEIVE FIFO ENABLE (TRFIFOE):** TRFIFOE enables/disables the transmit and receive FIFOs. When TRFIFOE = 1, both FIFOs are enabled (FIFO Mode). When TRFIFOE = 0, the FIFOs are both disabled (non-FIFO MODE). Writing a 0 to this bit clears all bytes in both FIFOs. When changing from FIFO mode to non-FIFO mode and vice versa, data is automatically cleared from the FIFOs. This bit must be written with a 1 when other bits in this register are written or the other bits will not be programmed. |

### 7.1.7 LCR(A,B)—LINE CONTROL REGISTER

I/O Address: Base +3h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register specifies the format of the asynchronous data communications exchange. LCR also enables/disables access to the Baud Rate Generator Divisor latches or the Transmitter Data Holding Register, Receiver Buffer Register, and Interrupt Enable Register.

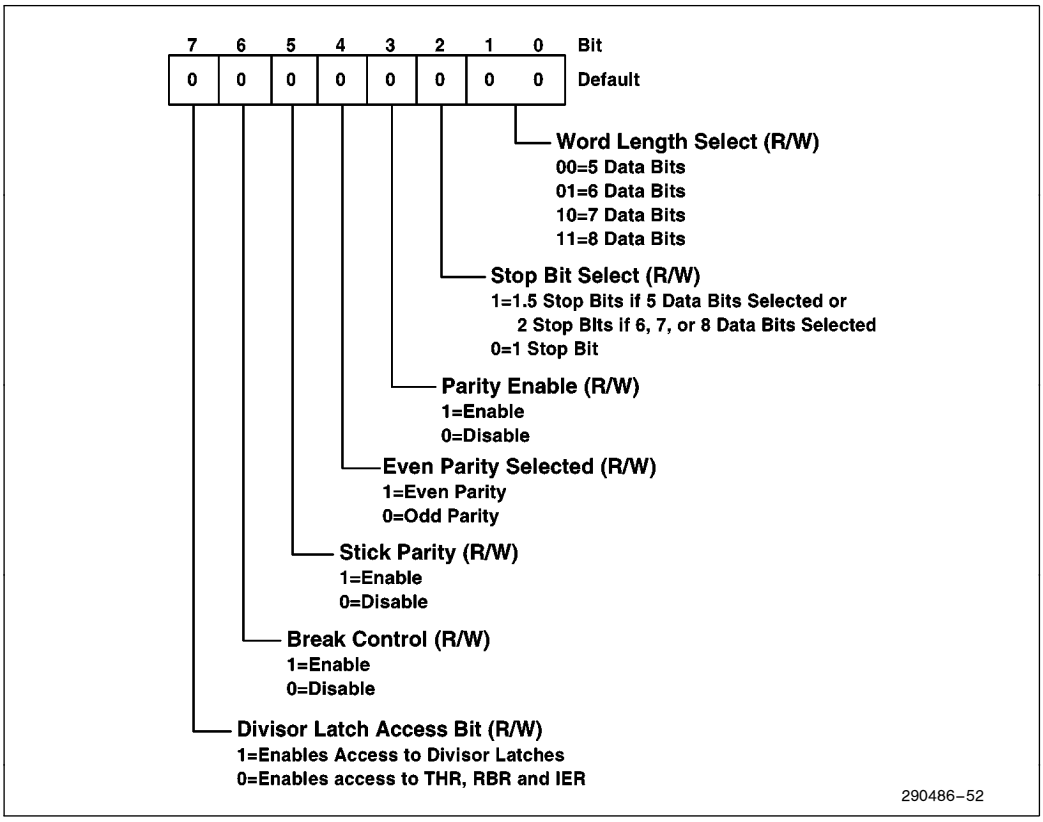



**Figure 52. Line Control Register**

| Bit | Description |
|---|---|
| 7 | **DIVISOR LATCH ACCESS BIT (DLAB):** DLAB controls access to the Baud Rate Generator Divisor Latches (and to the Transmit Holding Register, Receiver Buffer Register and Interrupt Enable Register which are located at the same I/O addresses). When DLAB = 1, access to the two Divisor Latches is selected and access to the THR, RBR, and IER is disabled. When DLAB = 0, access to the two Divisor Latches is disabled and access to the THR, RBR, and IER is selected.<br>During test mode operations, DLAB must be set to 1 for the BOUT signal to appear on the SOUT pin. |
| 6 | **BREAK CONTROL (BRCON):** When BRCON = 1, a break condition is transmitted from the 82091AA serial port to the receiving device. When BRCON = 1, the serial output (SOUT) is forced to the 'spacing' state (logical 0). BRCON only affects the SOUT signal and has no effect on the transmitter logic. Note that this feature permits the CPU to alert a terminal. If the following sequence is used, no erroneous characters will be transmitted because of the break.<br>1. Wait for the transmitter to be idle (TEMT = 1).<br>2. Set break (BRCON = 1) for the appropriate amount of time. If the transmitter will be used to time the break duration, then check that TEMT = 1 before clearing the BRCON.<br>3. Clear break (BRCON = 0) when normal transmission has to be restored.<br>During the break, the transmitter can be used as a character timer to accurately establish the break duration by sending characters and monitoring THRE and TEMT. |
| 5 | **STICKY PARITY (STICPAR):** STICPAR is the Stick Parity bit. When parity is enabled (PAREN = 1) this bit is used in conjunction with EVENPAR to select ''Mark'' or ''Space'' Parity. When bits PAREN, EVENPAR and STICPAR are 1, the parity bit is transmitted and checked as a 0 (Space Parity). If bits PAREN and STICPAR are 1 and EVENPAR is 0, the parity bit is transmitted and checked as a 1 (Mark Parity). When STICPAR = 0, stick parity is disabled. |
| 4 | **EVEN PARITY SELECT (EVENPAR):** EVENPAR selects between even and odd parity. When parity is enabled (PAREN = 1) and EVENPAR = 0, an odd number of 1s is transmitted or checked in the data word bits and parity bit. When parity is enabled and EVENPAR = 1, an even number of 1s is transmitted or checked. |
| 3 | **PARITY ENABLE (PAREN):** This bit enables/disables parity generation and checking. When PAREN = 1, a parity bit is generated (transmit data) or checked (receive data) between the last data bit and stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data bits and the Parity bit are summed.) When PAREN = 0, parity generation and checking is disabled. |
| 2 | **STOP BITS (STOPB):** This bit specifies the number of stop bits transmitted with each serial character. When STOPB = 0, one stop bit is generated in the transmitted data. When STOPB = 1 and a 5-bit data length is selected, one and a half stop bits are generated. When STOPB = 1 and either a 6-, 7-, or 8-bit data length is selected, two stop bits are generated. The receiver checks the first Stop bit only, regardless of the number of Stop bits selected. |
| 1:0 | **SERIAL DATA BITS (SERIALDB):** This field specifies the number of data bits in each transmitted or received serial character as follows:<br>**Bits[1:0] — Data Length**<br>0 0 — 5 Bits - Default<br>0 1 — 6 Bits<br>1 0 — 7 Bits<br>1 1 — 8 Bits |

### 7.1.8 MCR(A,B)—MODEM CONTROL REGISTER

I/O Address: Base + 4h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register controls the interface with the modem or data set (or a peripheral device emulating a modem).

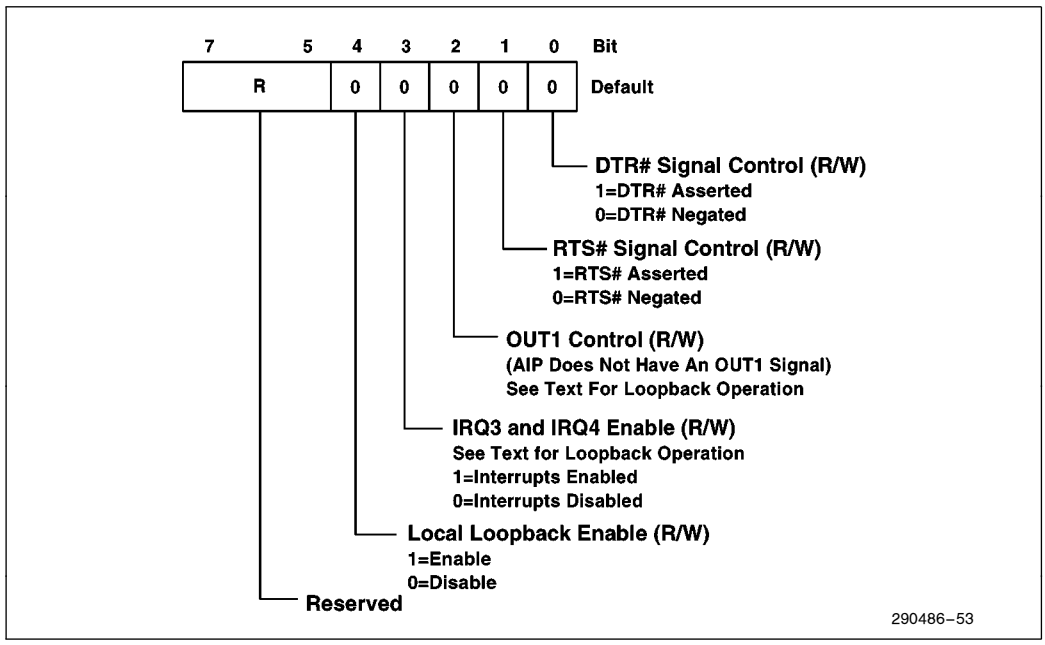



**Figure 53. Modem Control Register**

| Bit | Description |
|---|---|
| 7:5 | **RESERVED** |
| 4 | **LOOPBACK MODE ENABLE (LME):** LME provides a local loopback feature for diagnostic testing of the serial port module. When LME = 1, the following occurs:<br>1. The transmitter Serial Output (SOUT) is set to the Marking (logic 1) state.<br>2. The receiver Serial Input (SIN) is disconnected.<br>3. The output of the Transmitter Shift Register is ''looped back''(connected) to the Receiver Shift Register.<br>4. The four modem control inputs (DSR#, CTS#, RI and DCD#) are disconnected.<br>5. The DTRC, RTSC, OUT1C, IE bits in the MCR are internally connected to DSRS, CTSS, RIS, and DCDS in MSR, respectively.<br>6. The modem control output pins are forced to their high (inactive) state.<br>7. Data that is transmitted is immediately received.<br>This feature allows the CPU to verify the transmit and received data paths of the serial port. In the loopback mode, the receiver and transmitter interrupts are fully operational. The modem status interrupts are fully operational. The modem status interrupts are also operational, but the interrupt sources are the lower four bits of MCR instead of the four modem control inputs. Writing a 1 to any of these 4 MCR bits (bits[3:0]) causes an interrupt. In Loopback Mode the interrupts are still controlled by the Interrupt Enable Register. The IRQ3 and IRQ4 signal pins are tri-stated in the loopback mode. |
| 3 | **INTERRUPT ENABLE (IE):** When IE = 1, the associated interrupt is enabled (either IRQ3 or IRQ4 as selected via the associated serial port configuration register - A or B). In Local Loopback Mode, this bit controls bit 7 of the Modem Status Register. |
| 2 | **OUT1 BIT CONTROL (OUT1C):** This bit is the OUT1 bit. It does not have an output pin associated with it. It can be written to and read by the CPU. In Local Loopback Mode, this bit controls bit 6 of the Modem Status Register. |
| 1 | **REQUEST TO SEND CONTROL (RTS):** This bit controls the Request to Send (RTS#) output. When RTSC = 1, the RTS# output is asserted. When RTSC = 0, the RTS# output is negated. In Local Loopback Mode, this bit controls bit 4 of the Modem Status Register. |
| 0 | **DATA TERMINAL READY CONTROL (DTRC):** This bit controls the Data Terminal Ready (DTR#) output. When DTRC = 1, the DTR# output is asserted. When DTRC = 0, the DTR# output is negated. In Local Loopback Mode, this bit controls bit 5 of the Modem Status Register.<br>**NOTE:**<br>The DTR# and RTS# outputs of the serial port may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the modem or data set. |

### 7.1.9 LSR(A,B)—LINE STATUS REGISTER

I/O Address: Base + 5h
Default Value: 60h
Attribute: Read/Write
Size: 8 bits

This 8-bit register provides data transfer status information to the CPU. Note that the Line Status Register is intended for read operations only. Writing to this register is not recommended and could result in unintended operations. For this reason, the figure shows these bits as RO (read only).

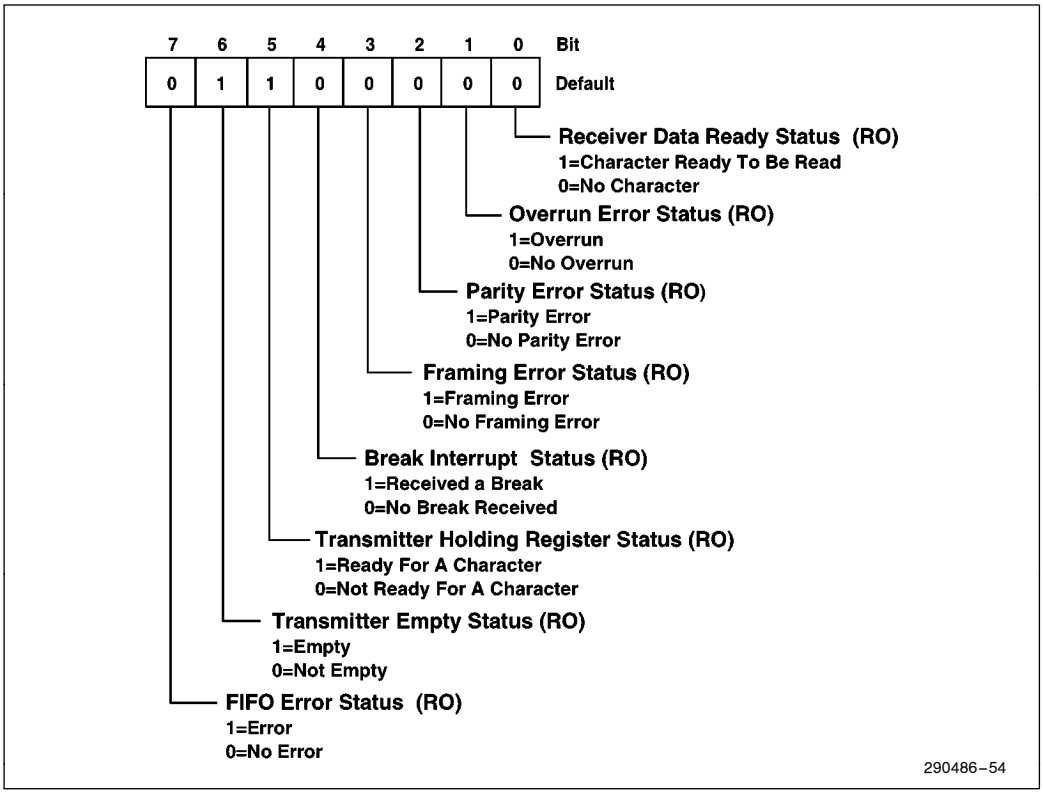



**Figure 54. Line Status Register**

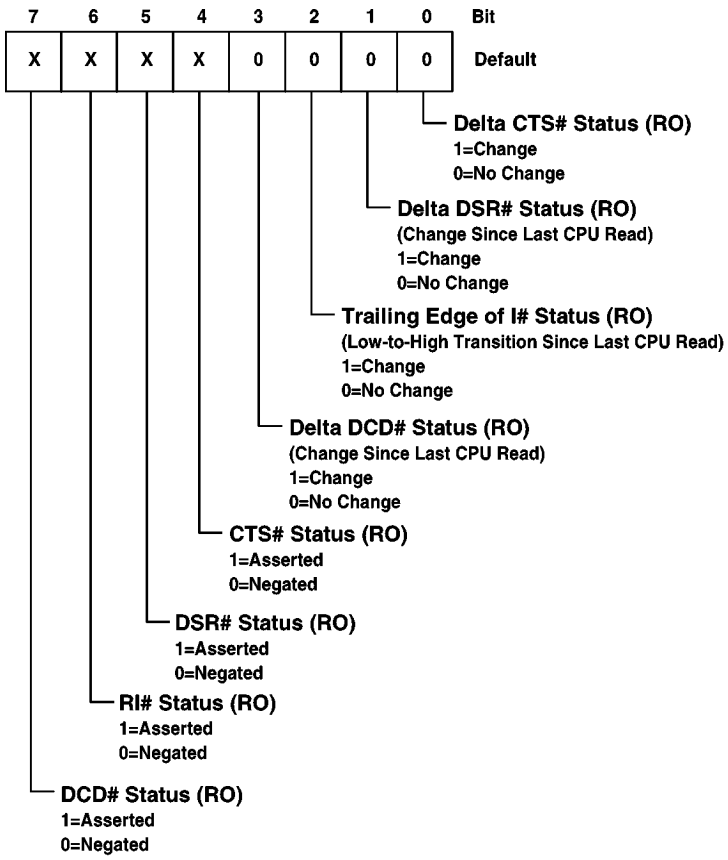
| Bit | Description |
|---|---|
| 7 | **FIFO ERROR STATUS (FIFOE):** In the non-FIFO Mode this is a 0. In the FIFO Mode, FIFOE is set to 1 when there is at least one parity error, framing error, or break indication in the FIFO. FIFOE is set to 0 when the CPU reads the LSR, if there are no subsequent errors in the FIFO. |
| 6 | **TRANSMITTER EMPTY STATUS (TEMT):** This bit is the Transmitter Empty (TEMT) indicator. When the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty, the 82091AA sets TEMT to a 1. When either the THR or TSR contains a data character, TEMT is set to a 0. The default is 0. In FIFO mode, this bit is set to 1 when the transmitter FIFO and the shift register are both empty. |

| Bit | Description |
|---|---|
| 5 | **TRANSMITTER HOLDING REGISTER STATUS (THRE):** This bit is the Transmitter Holding Register Empty (THRE) indicator. THRE indicates that the serial port module is ready to accept a new character for transmission. In addition, this bit causes the serial port module to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable is set to a 1. THRE is set to 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. THRE is set to 0 when the CPU loads the Transmitter Holding Register. In the FIFO mode, this bit is set to a 1 when the transmit FIFO is empty, and is set to 0 when at least 1 byte is written to the transmit FIFO. |
| 4 | **BREAK INTERRUPT STATUS (BI):** This bit is the Break Interrupt (BI) indicator. BI is set to a 1 when the received data input is held in the Spacing state (logic 0) for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits). When the CPU reads the contents of the Line Status Register, BI is set to 0.<br>In FIFO mode, this error is associated with the particular character in the FIFO associated with the Break. BI is indicated to the CPU when its associated character is at the top of the FIFO. When break occurs only one character is loaded into the FIFO. Restarting after a break is received requires the SIN pin to be a logical 1 for at least ½ bit times.<br>**NOTE:**<br>Bits[3:0] are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected and that interrupt is enabled. |
| 3 | **FRAMING ERROR STATUS (FE):** This bit is the Framing Error (FE) indicator. FE indicates that the received character did not have a valid stop bit. FE is set to a 1 when the stop bit following the last data bit or parity bit is 0 (spacing level). FE is set to 0 when the CPU reads the contents of the Line Status Register.<br>In FIFO mode, this error is associated with the particular character in the FIFO that it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. When a framing error is due to the next start bit, the serial port attempts to resynchronize. In this case, the serial port module samples this start bit twice and, if no FE exists, then the module takes in the rest of the bits. |
| 2 | **PARITY ERROR STATUS (PE):** This bit is the Parity Error (PE) indicator. PE indicates that the received data character does not have the correct even or odd parity, as selected by the EVENPAR bit in the Line Status Register. When a parity error is detected, PE is set to 1. PE is set to 0 when the CPU reads the contents of the Line Status Register. In the FIFO mode, this error is associated with the particular character in the FIFO that it applies to. This error is indicated to the CPU when its associated character is at the top of the FIFO. |
| 1 | **OVERRUN ERROR STATUS (OE):** OE indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register. In this case, the previous character is overwritten. When an overrun is detected, OE is set to 1. when the CPU reads the Line Status Register, OE is set to 0. This bit is read only.<br>If the FIFO mode data continues to fill the FIFO beyond the trigger level, an overrun error will occur only after the FIFO is completely full and the next character has been received in the shift register. OE is indicated to the CPU as soon as it happens. The character in the shift register is overwritten, but it is not transferred to the FIFO. |
| 0 | **RECEIVER DATA READY STATUS (DR):** DR is set to 1 when a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. When the data in the Receiver Buffer Register or FIFO is read, DR is set to 0. This bit is read only. |

### 7.1.10 MSR(A,B)—MODEM STATUS REGISTER

I/O Address: Base +6h
Default Value: XXXX 0000
Attribute: Read/Write
Size: 8 bits

The MSR provides the current state of the control lines from the Modem (or peripheral device) to the CPU. Bits[7:4] provide the status of the DCD#, RI, DSR#, and CTS# Modem signals. In addition to the current-state information of the Modem signals, bits[3:0] provide change information for these signals. Bits[3:0] are set to a 1 when the corresponding input signal changes state. Bits[3:0] are set to a 0 when the CPU reads the Modem Status Register.




**NOTE:**
X = Value determined by state of the corresponding modem control signal.

**Figure 55. Modem Status Register**

| Bit | Description |
|---|---|
| 7 | **DATA CARRIER DETECT STATUS:** This bit is the compliment of the Data Carrier Detect (DCD#) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to IRQ ENABLE in the MCR. |
| 6 | **RING INDICATOR STATUS:** This bit is the compliment of the Ring Indicator (RI) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT1 in the MCR. |
| 5 | **DATA SET READY STATUS:** This bit is the compliment of the Data Set Ready (DSR#) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR. |
| 4 | **CLEAR TO SEND STATUS:** This bit is the compliment of the Clear to Send (CTS#) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to RTS in the MCR. |
| 3 | **DELTA DATA CARRIER DETECT STATUS:** This bit is the Delta Data Carrier Detect (DDCD) indicator. Bit 3 indicates that the DCD# input to the chip has changed state.<br>**NOTE:**<br>Whenever bit 0, 1, 2, or 3 is set to logic 1, a Modem Status Interrupt is generated. |
| 2 | **TRAILING EDGE OF RING INDICATOR STATUS:** This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the RI# input to the chip has changed from a low to a high state. |
| 1 | **DELTA DATA SET READY STATUS:** This bit is the Delta Data Set Ready (DDSR) indictor. Bit 1 indicates that the DSR# input to the chip has changed state since the last time it was read by the CPU. |
| 0 | **DELTA CLEAR TO SEND STATUS:** This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the CTS# input to the chip has changed state since the last time it was read by the CPU. |

### 7.1.11 SCR(A,B)—SCRATCHPAD REGISTER

I/O Address: Base + 7h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This 8-bit read/write register does not control the serial port module in any way. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

| Bit | Description |
|---|---|
| 7:0 | **SCRATCHPAD DATA:** Bits[7:0] of this register correspond to SD[7:0]. |

## 7.2 FIFO Operations

This section describes the FIFO operations for interrupt and polled modes.

### 7.2.1 FIFO INTERRUPT MODE OPERATION

When the Receive FIFO and receiver interrupts are enabled (FCR0 = 1 and IER0 = 1), receiver interrupts occur as follows:

1. The receive data available interrupt is invoked when the FIFO has reached its programmed trigger level. The interrupt is cleared when the FIFO drops below the programmed trigger level.
2. The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, the bits are cleared when the FIFO drops below the trigger level.
3. The receiver line status interrupt (IIR-06h), as before, has higher priority than the received data available (IIR = 04h) interrupt.
4. The data ready bit (LSR0) is set as soon as a character is transferred from the shift register to the receive FIFO. This bit is set to 0 when the FIFO is empty.

When receiver FIFO and receiver interrupts are enabled, receiver FIFO timeout interrupts occur as follows:

1. A FIFO timeout interrupt occurs, if the following conditions exist:
   a. At least one character is in the FIFO.
   b. The most recent serial character received was longer than 4 continous character times ago (if 2 stop bits are programmed, the second one is included in this time delay).
   c. The most recent CPU read of the FIFO was longer than 4 continous character times ago.

   The maximum time between a received character and a timeout interrupt is 160 ms at 300 baud with a 12-bit receive character (i.e., 1 start, 8 data, 1 parity, and 2 stop bits).
2. Character times are calculated by using the RCLK input for a clock signal (this makes the delay proportional to the baud rate).
3. When a timeout interrupt occurs, it is cleared and the timer reset when the CPU reads one character from the receiver FIFO.
4. When a timeout interrupt does not occur, the timeout timer is reset after a new character is received or after the CPU reads the receiver FIFO.

When the transmit FIFO and transmitter interrupts are enabled (FCR0 = 1, IER1 = 1), transmit interrupts occur as follows:

1. The transmitter holding register interrupt occurs when the transmit FIFO is empty. The interrupt is cleared as soon as the transmitter holding register is written (1 to 16 characters may be written to the transmit FIFO while servicing the interrupt) or the IIR is read.

Character timeout and receiver FIFO trigger level interrupts have the same priority as the current received data available interrupt. Transmit FIFO empty has the same priority as the current transmitter holding register empty interrupt.

### 7.2.2 FIFO POLLED MODE OPERATION

With FIFO = 1, setting IER[3:0] to all 0s puts the serial port in the FIFO polled mode of operation. Since the receiver and transmitter are controlled separately, either one or both can be in the polled mode of operation.

In this mode, software checks receiver and transmitter status via the LSR. As stated in the register description:

- LSR0 is set as long as there is one byte in the receiver FIFO.
- LSR1 and LSR4 specify which error(s) has occurred. Character error status is handled the same way as interrupt mode. The IIR is not affected since IER2 = 0.
- LSR5 indicates when the transmitter FIFO is empty.
- LSR6 indicates that both the transmitter FIFO and shift register are empty.
- LSR7 indicates whether there are any errors in the receiver FIFO.

## 8.0 FLOPPY DISK CONTROLLER

The 82091AA's Floppy Disk Controller (FDC) is functionally compatible with 82078/82077SL/82077AA/8272A floppy disk controllers. During 82091AA configuration, the FDC can be configured for either two drive support or four drive support via the FCFG1 Register. This section provides a complete description of the FDC when it is configured for two drive support. Additional information on four drive support is provided in Appendix A, FDC Four Drive Support.

**NOTE:**
For FDC compatibility and programming guidelines, refer to the 82078 Floppy Disk Controller Data sheet.

### 8.1 Floppy Disk Controller Registers

The FDC contains seven status, control, and data registers. Table 23 shows the I/O address assignments for the FDC registers and the individual register descriptions follow in the order that they appear in the table. The registers provide control/status information and data paths for transfering data between the floppy disk controller interface and the 8-bit host interface. In some cases, two different registers occupy the same I/O address. In these cases, one register is read only and the other is write only (i.e., a read to the I/O address accesses one register and a write accesses the other register).

All registers are accessed as byte quantities. The base address is determined by hardware configuration at powerup (or a hard reset) or via software configuration by programming the 82091AA configuration registers as described in Section 4.0, AIP Configuration.

During a hard reset (RSTDRV asserted), the 82091AA registers are set to pre-determined **default** states. The default values are indicated in the individual register descriptions. Reserved bits in the FDC registers must be programmed to 0 when writing the register and these bits are 0 when read. The following bit notation is used for default settings:

**X** Default bit position value is determined by conditions on an 82091AA signal pin.

The following nomenclature is used for register access attributes:

**RO** **Read Only.** Note that for registers with read only attributes, writes to the I/O address have no affect on floppy disk operations.

**WO** **Write Only.** Note that for all FDC registers with write only attributes, reads of the I/O address access a different register.

**R/W** **Read/Write.** A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

Table 23 lists the register accesses that bring the FDC out of a powerdown state. All other registers accesses are possible without waking the part from a powerdown state and reads from these registers reflects the true status as shown in the register description. For writes that do not affect the powerdown state, the FDC retains the data and will subsequently reflect it when the FDC awakens. Note that for accesses that do not affect powerdown, the access may cause a temporary increase in FDC power consumption. The FDC reverts back to low power mode when the access has been completed. None of the extended registers effect the behavior of the powerdown mode.

**Table 23. Floppy Disk Controller Registers(1)**

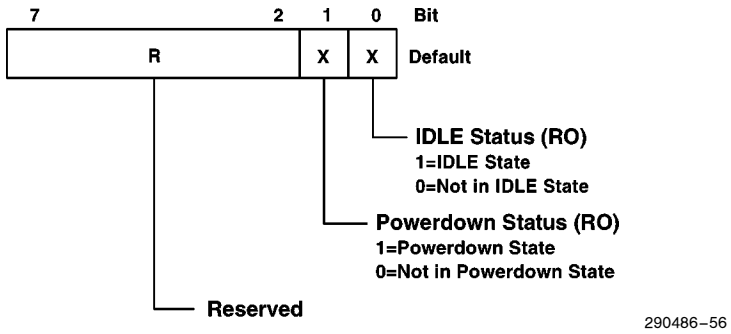
| FDC Register Address Access Base + | Abbreviation | Register Name | Access Wakes Up FDC | Access |
|---|---|---|---|---|
| 0h | — | Reserved | — | — |
| 1h | SRB | Status Register B | No | RO |
| 2h | DOR | Digital Output Register | No(2) | R/W |
| 3h | TDR | Tape Drive Register | No | R/W |
| 4h | MSR | Main Status Register | Yes | RO |
| 4h | DSR | Datarate Select Register | No(2) | WO |
| 5h | FIFO | Data FIFO | Yes | R/W |
| 6h | — | Reserved | — | — |
| 7h | DIR# | Digital Input Register | No | RO |
| 7h | CCR | Configuration Control Register | | WO |

**NOTES:**
1. The base address is 3F0h (primary address) or 370 (secondary address).
2. While writing to the DOR or DSR does not wake up the FDC, writing any of the motor enable bits in the DOR or invoking a software reset (either via DOR or DSR reset bits) will wake up the FDC.

### 8.1.1 SRB—STATUS REGISTER B (EREG EN = 1)

I/O Address: Base + 1h
Default Value: RRRR RRXX
Attribute: Read/Write
Size: 8 bits

SRB provides status and control information when auto powerdown is enabled. In the AT/EISA mode the SRB is made available whenever the EREG EN bit in the POWERDOWN MODE Command is set to 1. When EREG EN bit is set to 0, this register is not accessible. In this case, writes have no affect and reads return indeterminate values.




**NOTE:**
X = Value is determined by the state of the corresponding signal pin.

**Figure 56. Status Register B**

| Bit | Description |
|---|---|
| 7:2 | **RESERVED** |
| 1 | **POWERDOWN STATUS (PD):** This bit reflects the powerdown state of the FDC module. The 82091AA sets PD to 1 when the FDC is in the powerdown state. When PD = 0, the FDC is not in the powerdown state. |
| 0 | **IDLE STATUS (IDLE):** This bit reflects the idle state of the FDC module. The 82091AA sets IDLE to 1 when the FDC is in the idle state. When IDLE = 0, the FDC is not in the idle state. |

### 8.1.2 DOR—DIGITAL OUTPUT REGISTER

I/O Address: Base + 2h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The Digital Output Register enables/disables the floppy disk drive motors, selects the disk drives, enables/disables DMA, and provides a FDC module reset. The DOR reset bit and the motor enable bits have to be inactive when the FDC is in powerdown. The DMAGATE# and drive select bits are unchanged. During powerdown, writing to the DOR does not wake up the FDC, except for activating any of the motor enable bits. Setting the motor enable bits to 1 wakes up the FDC.

**NOTES:**

1. The descriptions in this section for DOR only apply when two-drive support is selected in the FCFG1 Register (FDDQTY = 0). For four-drive support (FDDQTY = 1), refer to Appendix A, FDC Four Drive Support.
2. The drive motor can be enabled separately without selecting the drive. This permits the motor to come up to speed before selecting the drive. Note also that only one drive can be selected at a time. However, the drive should not be selected without enabling the appropriate drive motor via bits[5:4] of this register.

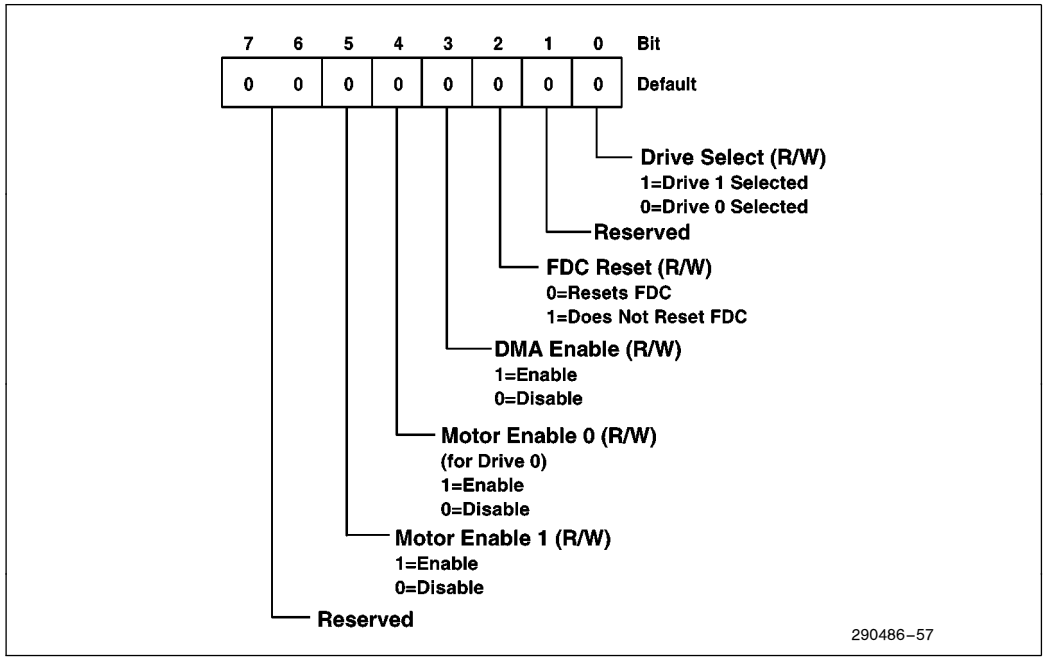



**Figure 57. Digital Output Register**

| Bit | Description |
|---|---|
| 7:6 | **RESERVED:** For a two-drive system, these bits are not used and have no affect on FDC operation. For a four drive system, see Appendix A, FDC Four Drive Support. |
| 5 | **MOTOR ENABLE 1 (ME1):** This bit controls a motor drive enable signal. ME1 directly controls either the FDME1# signal or FDME0# signal, depending on the state of the BOOTSEL bit in the TDR. When ME1 = 1, the selected motor enable signal (FDME1# or FDME0#) is asserted and when ME1 = 0, the selected motor enable signal is negated. |
| 4 | **MOTOR ENABLE 0 (ME0):** This bit controls a motor drive enable signal. ME1 directly controls either the FDME0# signal or FDME1# signal, depending on the state of the BOOTSEL bit in the TDR. When ME0 = 1, the selected motor enable signal (FDME0# or FDME1#) is asserted and when ME0 = 0, the selected motor enable signal is negated. |
| 3 | **DMA GATE (DMAGATE):** This bit enables/disables DMA for the FDC. When DMAGATE = 1, DMA for the FDC is enabled. In this mode, FDDREQ, TC, IRQ6, and FDDACK# are enabled. When DMAGATE = 0, DMA for the FDC is disabled. In this mode the IRQ6, and DRQ outputs are tri-stated and the DACK# and TC inputs are disabled to the FDC. Note that the TC input is only disabled to the FDC module. Other functional units in the 82091AA (e.g., parallel port or IDE interface) can still use the TC input signal for DMA activities. |
| 2 | **FDC RESET (DORRST):** DORRST is a software reset for the FDC module. When DORRST is set to 0, the basic core of the FDC and the FIFO circuits are cleared conditioned by the LOCK bit in the CONFIGURE Command. This bit is set to 0 by software or a hard reset (RSTDRV asserted). The FDC remains in a reset state until software sets this bit to 1. This bit does not affect the DSR, CCR and other bits of the DOR. DORRST must be held active for at least 0.5 $\mu$s at 250 Kbps. This is less than a typical ISA I/O cycle time. Thus, in most systems consecutive writes to this register to toggle this bit allows sufficient time to reset the FDC. |
| 1 | **RESERVED:** For a two-drive system, this bit is not used and must be programmed to 0. For a four drive system, see Appendix A, FDC Four Drive Support. |
| 0 | **DRIVE SELECT (DS):** This selects the floppy drive by controlling the FDS0# and FDS1# output signals. DS directly controls FDS1 and FDS0 as follows:<br>**Bit 0** — **Output Pin Status**<br>0 — FDS0# asserted (FDS1 asserted if BOOTSEL = 1)<br>1 — FDS1# asserted (FDS1 asserted if BOOTSEL = 1) |

### 8.1.3 TDR—ENHANCED TAPE DRIVE REGISTER

I/O Address: Base + 3h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. A hardware reset sets all bits in this register to 0 making drive 0 not available for tape support. A software reset via bit 2 of the DOR does not affect this register. Drive 0 is reserved for the floppy boot drive. Bits[7:2] are only available when EREG EN = 1; otherwise the bits are tri-stated. EREG EN is a bit in the POWERDOWN Command.

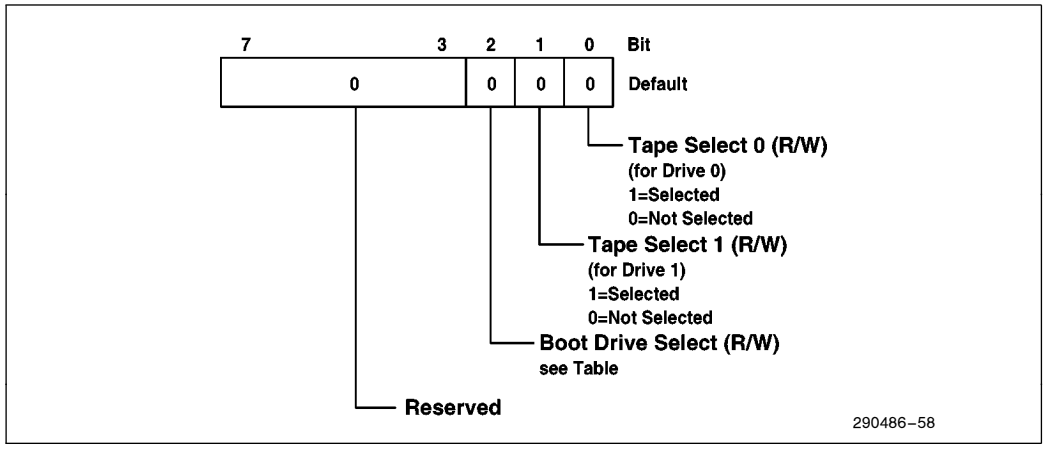



**Figure 58. Enhanced Tape Drive Register**

| Bit | Description |
|---|---|
| 7:3 | **RESERVED** |
| 2 | **BOOT DRIVE SELECT (BOOTSEL):** The BOOTSEL bit is used to remap the drive selects and motor enables. The functionality is as described below:<br>**BOOTSEL** — **Mapping**<br>0 — DS0 → FDS0, ME0 → FDME0 (default)<br>DS1 → DS1, ME1 → FDME1<br>1 — DS0 → DS1, ME0 → FDME1<br>DS1 → FDS0, ME1 → FDME0<br>Note that this mapping also applies to a four drive system (FDDQTY = 1 in the FCFG1 Register). In a four drive system, only drive 0 or drive 1 can be selected as the boot drive. |
| 1 | **RESERVED:** For a two-drive system, this bit is not used and must be programmed to 0. For a four drive system, see Appendix A, FDC Four Drive Support. |
| 0 | **TAPE SELECT (TAPESEL):** This bit is used by software to assign logical drive number 1 to be a tape drive. Other than adjusting precompensation delays for tape support, this bit does not affect the FDC hardware. The bit can be written and read by software as an indication of the tape drive assignment. Drive 0 is not available as a tape drive and is reserved as the floppy disk boot drive. The tape drive assignment is as follows:<br>**Bit 0** — **Drive Selected**<br>0 — None (all are floppy disk drives)<br>1 — Drive 1 is a tape drive. |

### 8.1.4 MSR—MAIN STATUS REGISTER

I/O Address: Base + 4h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This read only register provides FDC status information. This information is used by software to control the flow of data to and from the FIFO (accessed via the FDCFIFO Register). The MSR indicates when the FDC is ready to send or receive data through the FIFO. During non-DMA transfers, this register should be read before each byte is transferred to or from the FIFO.

After a hard or soft reset or recovery from a powerdown state, the MSR is available to be read by the host. The register value is 00h until the oscillator circuit has stabilized and the internal registers have been initialized. When the FDC is ready to receive a new command, MSR[7:0] = 80h. The worst case time allowed for the MSR to report 80h (i.e., RQM is set to 1) is 2.5 $\mu$s after a hard or soft reset.

Main Status Register is used for controlling command input and result output for all commands. Some example values of the MSR are:

- MSR = 80H; The controller is ready to receive a command.
- MSR = 90H; executing a command or waiting for the host to read status bytes (assume DMA mode).
- MSR = D0H; waiting for the host to write status bytes.

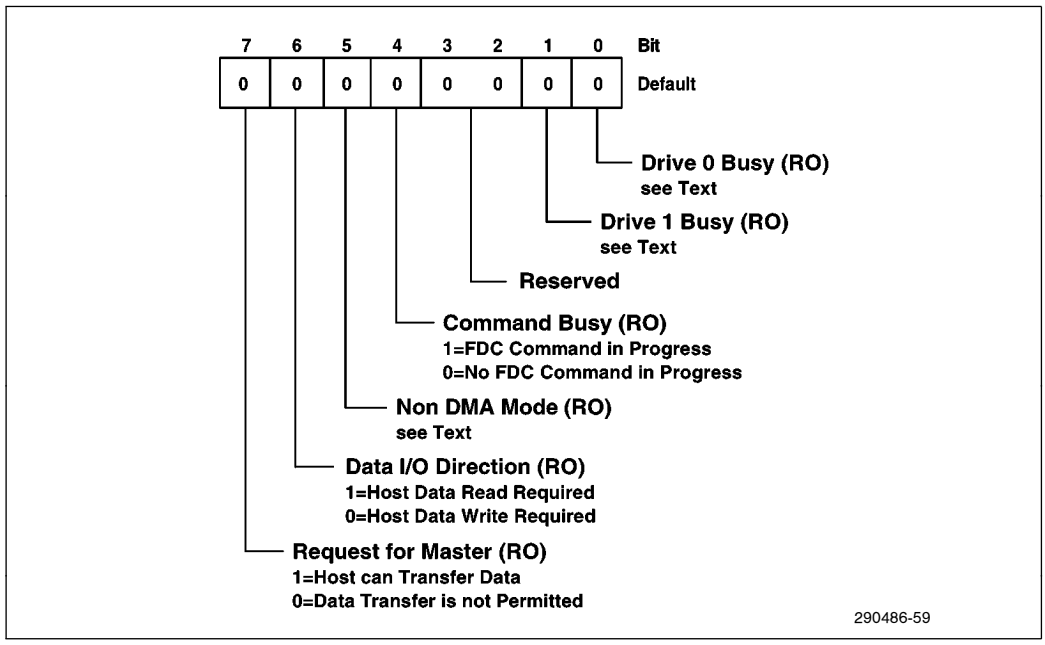



**Figure 59. Main Status Register**

| Bit | Description |
|---|---|
| 7 | **REQUEST FOR MASTER (RQM):** When RQM = 1, the FDC is ready to send/receive data through the FIFO (FDCFIFO Register). The FDC sets this bit to 0 after a byte transfer and then sets the bit to 1 when it is ready for the next byte. During non-DMA execution phase, RQM indicates the status of IRQ6. |
| 6 | **DIRECTION I/O (DIO):** When RQM = 1, DIO indicates the direction of a data transfer. When DIO = 1, the FDC is requesting a read of the FDCFIFO. When DIO = 0, the FDC is requesting a write to the FDCFIFO. |
| 5 | **NON-DMA (NONDMA):** Non-DMA mode is selected via the SPECIFY Command. In this mode, the FDC sets this bit to a 1 during the execution phase of a command. This bit is for polled data transfers and helps differentiate between the data transfer phase and the reading of result bytes. |
| 4 | **COMMAND BUSY (CMDBUSY):** CMDBUSY indicates when a command is in progress. When the first byte of the command phase is written, the FDC sets this bit to 1. CMDBUSY is set to 0 after the last byte of the result phase is read. If there is no result phase (e.g., SEEK or RECALIBRATE Commands), CMDBUSY is set to 0 after the last command byte is written. |
| 3:2 | **RESERVED:** For a two-drive system, these bits are not used and must be programmed to 0. For a four drive system, see Appendix A, FDC Four Drive Support. |
| 1 | **DRIVE 1 BUSY (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 1. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive. |
| 0 | **DRIVE 0 BUSY (DRV0BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 0. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive. |

### 8.1.5 DSR—DATA RATE SELECT REGISTER

I/O Address: Base + 4h
Default Value: 02h
Attribute: Write Only
Size: 8 bits

The DSR selects the data rate, amount of write precompenstion, invokes direct powerdown, and invokes a FDC software reset. This write only register ensures backward compatibility with the Intel series of floppy disk controllers. Changing the data rate changes the timings of the drive control signals. To ensure that drive timings are not violated when changing data rates, choose a drive timing such that the fastest data rate will not violate the timing.

In the default state, the PDOSC bit is low and the oscillator is powered up. When this bit is programmed to a 1, the oscillator is shut off. Hardware reset sets this bit to a 0. Neither of the software resets (via DOR or DSR) have any effect on this bit. Note that PDOSC should only be set to a 1 when the FDC module is in the powerdown state. Otherwise, the FDC will not function correctly and must be hardware reset once the oscillator has turned back on and stabilized. Setting the PDOSC bit has no effect on the clock input to the FDC (the X1 pin). The clock input is separately disabled when the part is powered down. The Save Command checks the status of PDOSC. However the Restore Command will not restore this bit to a 1.

Software resets do not affect the DRATE or PRECOMP bits.

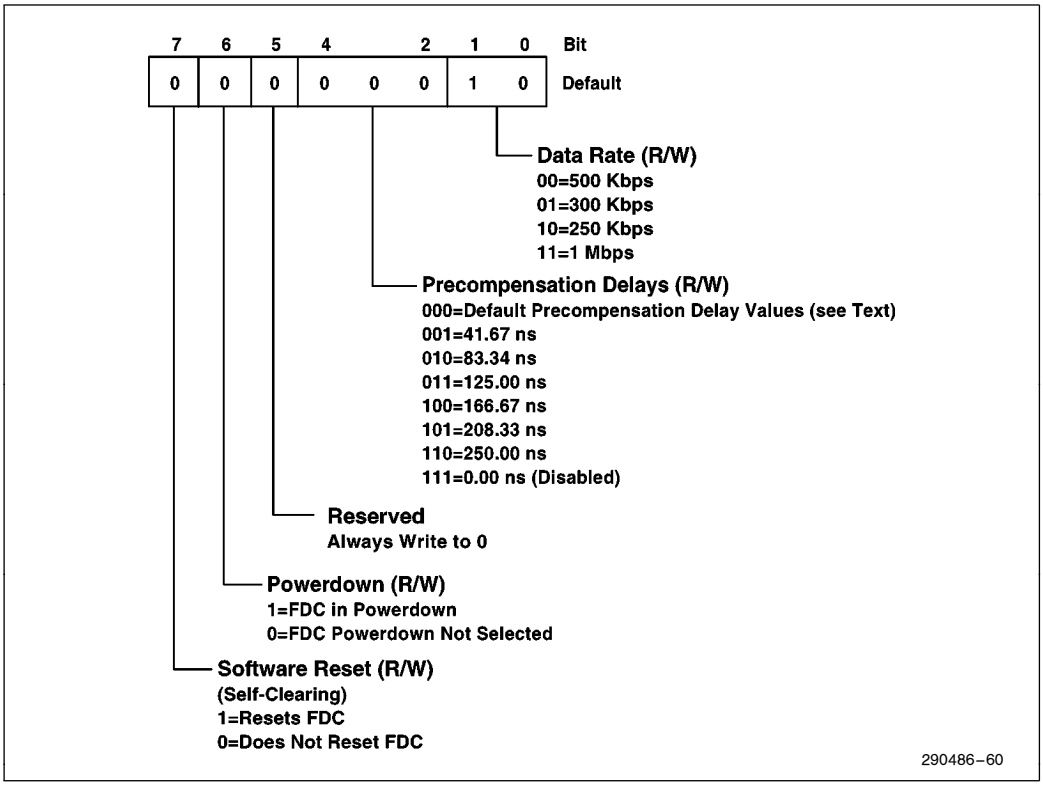



**Figure 60. Data Rate Select Register**

| Bit | Description |
|---|---|
| 7 | **SOFTWARE RESET (DSRRST):** DSRRST operates the same as the DORRST bit in the DOR, except that this bit is self clearing. |
| 6 | **POWERDOWN (FPD):** FPD provides direct powerdown for the FDC module. When FPD = 1, the FDC module enters the powerdown state, regardless of the state of the module. The FDC module is internally reset and then put into powerdown. No status is saved and any operation in progress is aborted. A hardware or software reset causes the 82091AA to exit the FDC module powerdown state. |
| 5 | **RESERVED** |
| 4:2 | **PRECOMPENSATION (PRECOMP):** Bits[4:2] adjusts the WRDATA output to the disk to compensate for magnetic media phenomena known as bit shifting. The data patterns that are susceptible to bit shifting are well understood and the FDC compensates the data pattern as it is written to the disk. The amount of precompensation depends on the drive and media but in most cases the default value is acceptable. The FDC module starts pre-compensating the data pattern starting on Track 0. The CONFIGURE Command can change the track where pre-compensating originates.<br>**Bits[4:2] — Precompensation Delays (ns)**<br>0 0 0 — Default mode<br>0 0 1 — 41.67<br>0 1 0 — 83.34<br>0 1 1 — 125.00<br>1 0 0 — 166.67<br>1 0 1 — 208.33<br>1 1 0 — 250<br>1 1 1 — 0.00 (disabled)<br>The default precompensation delay mode provides the following delays:<br>**Data Rate — Default Precompensation Delays (ns)**<br>1 Mbps — 41.67<br>0.5 Mbps — 125.00<br>0.3 Mbps — 125.00<br>0.25 Mbps — 125.00 |
| 1:0 | **DATA RATE SELECT (DRATESEL):** DRATESEL[1:0] select one of the four data rates as listed below. The default value is 250 Kbps.<br>**Bits[1:0] — Date Rate**<br>1 1 — 1 Mbps<br>0 0 — 500 Kbps<br>0 1 — 300 Kbps<br>1 0 — 250 Kbps - default |

### 8.1.6 FDCFIFO—FDC FIFO (DATA)

I/O Address: Base +5h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

All command parameter information and disk data transfers go through the 16-byte FIFO. The FIFO has programmable threshold values. Data transfers are governed by the RQM and DIO bits in the MSR. At the start of a command, the FIFO action is always disabled and command parameters must be sent based upon the RQM and DIO bit settings. At the start of the command execution phase, the FDC clears the FIFO of any data to ensure that invalid data is not transferred. An overrun or underrun will terminate the current command and the transfer of data. Disk writes complete the current sector by generating a 00 pattern and valid CRC.

The FIFO defaults to an 8272A compatible mode after a hardware reset (via RSTDRV pin). Software resets (via DOR or DSR) can also place the FDC into 8272A compatible mode, if the LOCK bit is set to 0 (see the definition of the LOCK bit) maintaining PC-AT hardware compatibility. The default values can be changed through the CONFIGURE Command (enable full FIFO operation with threshold control). The FIFO provides the system a larger DMA latency without causing a disk error. The following table gives several examples of the delays with a FIFO. The data is based upon the formula: Threshold# $\times$ 1/DATA RATE $\times$ 8 $-$ 1.5 $\mu$s = DELAY.

| FIFO Threshold | Maximum Service Delay (1 Mbps Data Rate) | Maximum Delay to Servicing at 500 Kbps Data Rate |
|---|---|---|
| 1 byte | 1 $\times$ 8 $\mu$s $-$ 1.5 $\mu$s = 6.5 $\mu$s | 1 $\times$ 16 $\mu$s $-$ 1.5 $\mu$s = 14.5 $\mu$s |
| 2 bytes | 2 $\times$ 8 $\mu$s $-$ 1.5 $\mu$s = 14.5 $\mu$s | 2 $\times$ 16 $\mu$s $-$ 1.5 $\mu$s = 30.5 $\mu$s |
| 8 bytes | 8 $\times$ 8 $\mu$s $-$ 1.5 $\mu$s = 62.5 $\mu$s | 8 $\times$ 16 $\mu$s $-$ 1.5 $\mu$s = 126.5 $\mu$s |
| 15 bytes | 15 $\times$ 8 $\mu$s $-$ 1.5 $\mu$s = 118.5 $\mu$s | 15 $\times$ 16 $\mu$s $-$ 1.5 $\mu$s = 238.5 $\mu$s |

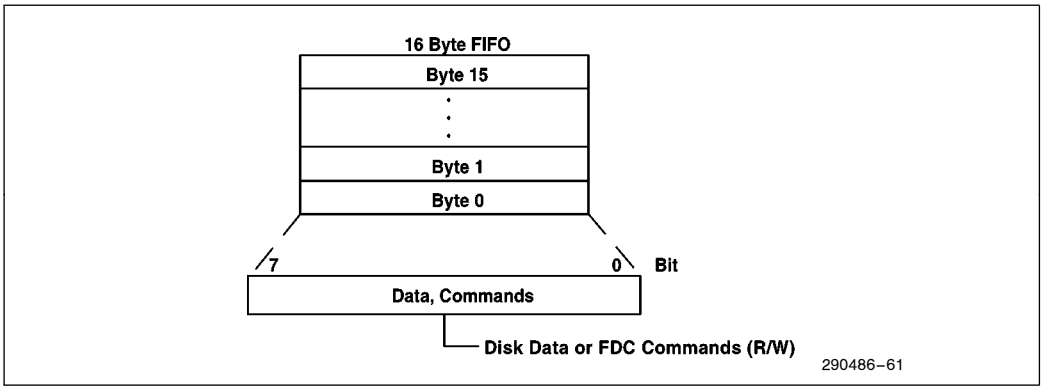



**Figure 61. FDC FIFO**

| Bit | Description |
|---|---|
| 7:0 | **FIFO DATA:** Bits[7:0] correspond to SD[7:0]. |

### 8.1.7 DIR—DIGITAL INPUT REGISTER

I/O Address: Base + 7h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register is read only in all modes. In PC-AT mode only bit 7 is driven and all other bits remain tri-stated.

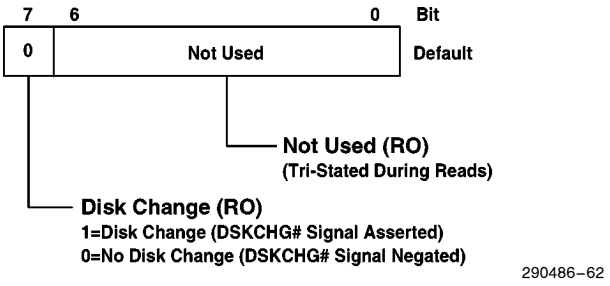



**Figure 62. Digital Input Register**

| Bit | Description |
|---|---|
| 7 | **DISK CHANGE (DSKCHG):** This bit monitors a disk change in the floppy disk drive. DSKCHG is set to a 1 when the DSKCHG# signal on the floppy interface is asserted. DSKCHG is set to a 0 when the DSKCHG# signal on the floppy interface is negated. During powerdown, this bit is invalid. |
| 6:0 | **NOT USED:** These bits are tri-stated during a read. |

### 8.1.8 CCR—CONFIGURATION CONTROL REGISTER

I/O Address: Base +7h
Default Value: 02h
Attribute: Write Only
Size: 8 bits

This register sets the data rate.

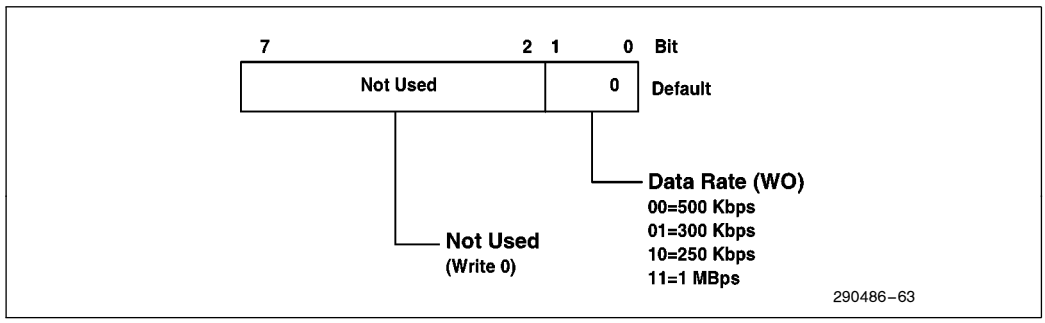



**Figure 63. Configuration Control Register**

## 8.2 Reset

There are four sources of FDC reset—a hard reset via the RSTDRV signal and three software resets (via the FCFG2, DOR, and DSR Registers). At the end of the reset, the FDC comes out of the power-down state. Note that the DOR reset condition remains in effect until software programs the DORRST bit to 1 in the DOR. All operations are terminated and the FDC enters an idle state. Invoking a reset while a disk write activity is in progress will corrupt the data and CRC. On exiting the reset state, various internal registers are cleared, and the FDC waits for a new command. Drive polling will start unless disabled by a new CONFIGURE Command.

### 8.2.1 HARD RESET AND CONFIGURATION REGISTER RESET

A hard reset (asserting RSTDRV) and a software reset through the FCFG2 Registers have the same affect on the FDC. These resets clear all FDC registers, except those programmed by the SPECIFY command. The DOR reset bit is enabled and must be set to 0 by the host to exit the reset state.

### 8.2.2 DOR RESET vs DSR RESET

The DOR and DSR resets are functionally the same. The DSR reset is included to maintain 82072 compatibility. Both reset the 8272 core, which affects drive status information. The FIFO circuits are also reset if the LOCK bit is a 0 (see definition of the LOCK bit). The DSR reset is self-clearing (exits the reset state automatically) while the DOR reset remains in the reset state until software writes the DOR reset bit to 0. DOR reset has precedence over the DSR reset. The DOR reset is set automatically when a hard reset or configuration reset occurs. Software must set the DOR reset bit to 0 to exit the reset state.

The AC Specifications gives the minimum amount of time that the DOR reset must be held active. This amount of time that the DOR reset must be held active is dependent upon the data rate. FDC requires that the DOR reset bit must be held active for at least 0.5 $\mu$s at 250 Kbps. This is less than a typical ISA I/O cycle time.

## 8.3 DMA Transfers

DMA transfers are enabled with the SPECIFY Command. When enabled, The FDC initiates DMA transfers by asserting the FDDREQ signal during a data transfer command. The FIFO is enabled directly by asserting FDDACK# and addresses need not be valid.

## 8.4 Controller Phases

The FDC handles commands in three phases—*command*, *execution* and *result*. Each phase is described in the following sections. When not processing a command, the FDC can be in the *idle*, *drive polling* or *powerdown state*. This section describes the command, execute and result phases.

### 8.4.1 COMMAND PHASE

After a reset, the FDC enters the command phase and is ready to accept a command from the host. For each of the commands, a defined set of command code bytes and parameter bytes must be written to the FDC (as described in Section 8.8, Command Set Description) before the command phase is complete. These bytes of data must be transferred in the order described.

Before writing to the FDC, the host must examine the RQM and DIO bits of the Main Status Register. RQM must be 1 and DIO must be 0, before command bytes may be written. The FDC sets RQM to 0 after each write cycle and keeps the bit at 0 until the received byte is processed. After processing the byte, the FDC sets RQM to 1 again to request the next parameter byte of the command, unless an illegal command condition is detected. After the last parameter byte is received, RQM remains 0, and the FDC automatically enters the next phase (execution or result phase) as defined by the command definition.

The FIFO is disabled during the command phase to retain compatibility with the 8272A, and to provide for the proper handling of the Invalid Command condition.

### 8.4.2 EXECUTION PHASE

The following paragraphs detail the operation of the FIFO flow control. In these descriptions, threshold is defined as the number of bytes available to the FDC when service is requested from the host, and ranges from 1 to 16. The FIFOTHR parameter, which the user programs, is one less and ranges from 0 to 15.

A low threshold value (e.g., 2) results in longer periods of time between service requests but requires faster servicing of the request for both read and write cases. The host reads (writes) from (to) the FIFO until empty (full), then the transfer request goes inactive. The host must be very responsive to the service request. This is the desired case for use with a ''fast'' system.

A high value of threshold (e.g., 12) is used with a ''sluggish'' system by affording a long latency period after a service request, but results in more frequent service requests.

#### 8.4.2.1 Non-DMA Mode Transfers from the FIFO to the Host

The IRQ6 pin and RQM bits in the Main Status Register are activated when the FIFO contains 16 (or set threshold) bytes, or the last bytes of a full sector transfer have been placed in the FIFO. The IRQ6 pin can be used for interrupt driven systems and RQM can be used for polled sytems. The host must respond to the request by reading data from the FIFO. This process is repeated until the last byte is transferred out of the FIFO, then FDC negates the IRQ6 pin and RQM bit.

#### 8.4.2.2 Non-DMA Mode Transfers from the Host to the FIFO

The IRQ6 pin and RQM bit in the Main Status Register are activated upon entering the execution phase of data transfer commands. The host must respond to the request by writing data into the FIFO. The IRQ6 pin and RQM bit remain true until the FIFO becomes full. They are set true again when the FIFO has (threshold) bytes remaining in the FIFO. The IRQ6 pin is also negated if TC and DACK# both go inactive. The FDC enters the result phase after the last byte is taken by the FDC from the FIFO (i.e. FIFO empty condition).

#### 8.4.2.3 DMA Mode Transfers from the FIFO to the Host

The FDC asserts the FDDREQ signal when the FIFO contains 16 (or set threshold) bytes or the last byte of a full sector transfer has been placed in the FIFO. The DMA controller must respond to the request by reading data from the FIFO. The FDC negates FDDREQ when the FIFO is empty. FDDREQ is negated after FDDACK# is asserted for the last byte of a data transfer (or on the active edge of RD#, on the last byte, if no edge is present on FDDACK#).

**NOTE:**
FDDACK# and TC must overlap for at least 50 ns for proper functionality. A data underrun may occur if FDDREQ is not removed in time to prevent an unwanted cycle.

#### 8.4.2.4 DMA Mode Transfers from the Host to the FIFO

The FDC asserts FDDREQ when entering the execution phase of data transfer commands. The DMA controller must respond by asserting FDDACK# and WR# signals and placing data in the FIFO. FDDREQ remains asserted until the FIFO becomes full. FDDREQ is again asserted when the FIFO has (threshold) bytes remaining in the FIFO. The FDC also negates the FDDREQ when the FIFO becomes empty (qualified by DACK# and TC overlapping by 50 ns) indicating that no more data is required. FDDREQ is negated after FDDACK# is asserted for the last byte of a data transfer (or on the active edge of WR# of the last byte, if no edge is present on DACK#). A data overrun may occur if FDDREQ is not removed in time to prevent an unwanted cycle.

### 8.4.3 DATA TRANSFER TERMINATION

The FDC supports terminal count explicitly through the TC signal and implicitly through the underrun/overrun and end-of-track (EOT) functions. For full sector transfers, the EOT parameter can define the last sector to be transferred in a single or multi-sector transfer. If the last sector to be transferred is a partial sector, the host can stop transferring the data in mid-sector and the FDC will continue to complete the sector as if a hardware TC was received. The only difference between these implicit functions and TC is that they return ''abnormal termination'' result status. Such status indications can be ignored if they were expected.

**NOTE:**

When the host is sending data to the FIFO, the internal sector count will be complete when the FDC reads the last byte from its side of the FIFO. There may be a delay in the removal of the transfer request signal of up to the time taken for the FDC to read the last 16 bytes from the FIFO. The host must be able to tolerate this. In a DMA system, FDDREQ is removed (negated) as soon as TC is received indicating the termination of the transfer. The reception of TC also generates an interrupt on IRQ6. However, in a non-DMA system the interrupt will not be generated until the FIFO is empty.

The generation of IRQ6 determines the beginning of the result phase. For each of the commands, a defined set of result bytes has to be read from the FDC before the result phase is complete (refer to Section 8.5, Command Set/Descriptions). These bytes of data must be read out for another command to start.

RQM and DIO must both be 1 before the result bytes may be read from the FIFO. After all the result bytes have been read, RQM = 1, DIO = 0, and CMDBUSY = 0 in the MSR. This indicates that the FDC is ready to accept the next command.

## 8.5 Command Set/Descriptions

Commands can be written whenever the FDC is in the command phase. Each command has a unique set of needed parameters and status results. The FDC checks to see that the first byte is a valid command and, if valid, proceeds with the command. If it was invalid, the next time the RQM bit in the MSR register is 1 the DIO and CB bits will also be 1, indicating the FIFO must be read. A result byte of 80h will be read out of the FIFO, indicating an invalid command was issued. After reading the result byte from the FIFO, the FDC returns to the command phase. Table 23 shows the FDC Command set.

**Table 24. FDC Command Set**

| Phase | R/W | Data Bus D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| **Read Data** | | | | | | | | | | |
| Command | W | MT | MFM | SK | 0 | 0 | 1 | 1 | 0 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | C | | ..... | | | Sector ID |
| | W | ..... | | | H | | ..... | | | Information Prior to |
| | W | ..... | | | R | | ..... | | | Command Execution |
| | W | ..... | | | N | | ..... | | | |
| | W | ..... | | | EOT | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | DTL | | ..... | | | |
| Execution | | | | | | | | | | Data Transfer Between the FDD and System |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | C | | ..... | | | |
| | R | ..... | | | H | | ..... | | | Sector ID |
| | R | ..... | | | R | | ..... | | | Information After |
| | R | ..... | | | N | | ..... | | | Command Execution |
| **Read Deleted Data** | | | | | | | | | | |
| Command | W | MT | MFM | SK | 0 | 1 | 1 | 0 | 0 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | C | | ..... | | | Sector ID |
| | W | ..... | | | H | | ..... | | | Information Prior to |
| | W | ..... | | | R | | ..... | | | Command Execution |
| | W | ..... | | | N | | ..... | | | |
| | W | ..... | | | EOT | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | DTL | | ..... | | | |
| Execution | | | | | | | | | | Data Transfer Between the FDD and System |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | C | | ..... | | | |
| | R | ..... | | | H | | ..... | | | Sector ID |
| | R | ..... | | | R | | ..... | | | Information After |
| | R | ..... | | | N | | ..... | | | Command Execution |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| **Write Data** | | | | | | | | | | |
| Command | W | MT | MFM | 0 | 0 | 0 | 1 | 0 | 1 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | C | | ..... | | | Sector ID |
| | W | ..... | | | H | | ..... | | | Information Prior to |
| | W | ..... | | | R | | | | | Command |
| | | | | | | | | | | Execution |
| | W | ..... | | | N | | ..... | | | |
| | W | ..... | | | EOT | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | DTL | | ..... | | | |
| Execution | | | | | | | | | | Data Transfer Between the FDD and System |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | C | | ..... | | | |
| | R | ..... | | | H | | ..... | | | Sector ID |
| | R | ..... | | | R | | ..... | | | Information After |
| | R | ..... | | | N | | ..... | | | Command |
| | | | | | | | | | | Execution |
| **Write Deleted Data** | | | | | | | | | | |
| Command | W | MT | MFM | 0 | 0 | 1 | 0 | 0 | 1 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | C | | ..... | | | Sector ID |
| | W | ..... | | | H | | ..... | | | Information Prior to |
| | W | ..... | | | R | | ..... | | | Command |
| | | | | | | | | | | Execution |
| | W | ..... | | | N | | ..... | | | |
| | W | ..... | | | EOT | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | DTL | | ..... | | | |
| Execution | | | | | | | | | | Data Transfer Between the FDD and System |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | C | | ..... | | | |
| | R | ..... | | | H | | ..... | | | Sector ID |
| | R | ..... | | | R | | ..... | | | Information After |
| | R | ..... | | | N | | ..... | | | Command |
| | | | | | | | | | | Execution |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus | | | | | | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | D7 | D6 | D5 | D4 | | D3 | D2 | D1 | D0 | |
| **Read Track** | | | | | | | | | | | |
| Command | W | 0 | MFM | 0 | 0 | | 0 | 0 | 1 | 0 | Command Codes |
| | W | 0 | 0 | 0 | 0 | | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | | C | | ..... | | | Sector ID |
| | W | ..... | | | | H | | ..... | | | Information Prior to |
| | W | ..... | | | | R | | ..... | | | Command |
| | | | | | | | | | | | Execution |
| | W | ..... | | | | N | | ..... | | | |
| | W | ..... | | | | EOT | | | | | |
| | W | ..... | | | | GPL | | ..... | | | |
| | W | ..... | | | | DTL | | ..... | | | |
| Execution | | | | | | | | | | | Data Transfer Between the FDD and System. FDC Reads All Sectors From Index Hole to EOT |
| Result | R | ..... | | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | | C | | ..... | | | |
| | R | ..... | | | | H | | ..... | | | Sector ID |
| | R | ..... | | | | R | | ..... | | | Information After |
| | R | ..... | | | | N | | ..... | | | Command |
| | | | | | | | | | | | Execution |
| **Verify** | | | | | | | | | | | |
| Command | W | MT | MFM | SK | 1 | | 0 | 1 | 1 | 0 | Command Codes |
| | W | EC | 0 | 0 | 0 | | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | | C | | ..... | | | Sector ID |
| | W | ..... | | | | H | | ..... | | | Information Prior to |
| | W | ..... | | | | R | | ..... | | | Command |
| | | | | | | | | | | | Execution |
| | W | ..... | | | | N | | ..... | | | |
| | W | ..... | | | | EOT | | ..... | | | |
| | W | ..... | | | | GPL | | ..... | | | |
| | W | ..... | | | | DTL/SC | | ..... | | | |
| Execution | | | | | | | | | | | Data Transfer Between the FDD and System |
| Result | R | ..... | | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | | C | | ..... | | | |
| | R | ..... | | | | H | | ..... | | | Sector ID |
| | R | ..... | | | | R | | ..... | | | Information After |
| | R | ..... | | | | N | | ..... | | | Command |
| | | | | | | | | | | | Execution |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus | | | | | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| **Version** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Command Codes |
| Result | W | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Enhanced Controller |
| **Format Track** | | | | | | | | | | |
| Command | W | 0 | MFM | 0 | 0 | 1 | 1 | 0 | 1 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | N | | ..... | | | Bytes/Sector |
| | W | ..... | | | SC | | ..... | | | Sector/Cylinder |
| | W | ..... | | | GPL | | ..... | | | Gap 3 |
| | W | ..... | | | D | | ..... | | | Filler Byte |
| Execution For Each Sector Repeat: | W | ..... | | | C | | ..... | | | |
| | W | ..... | | | H | | ..... | | | Input Sector |
| | W | ..... | | | R | | ..... | | | Parameters |
| | W | ..... | | | N | | ..... | | | |
| | | | | | | | | | | FDC Formats an Entire Cylinder |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | after Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | Undefined | | ..... | | | |
| | R | ..... | | | Undefined | | ..... | | | |
| | R | ..... | | | Undefined | | ..... | | | |
| | R | ..... | | | Undefined | | ..... | | | |
| **Scan Equal** | | | | | | | | | | |
| Command | W | MT | MFM | SK | 1 | 0 | 0 | 0 | 0 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | C | | ..... | | | Sector ID |
| | W | ..... | | | H | | ..... | | | Information Prior |
| | W | ..... | | | R | | ..... | | | to Command |
| | W | ..... | | | N | | ..... | | | Execution |
| | W | ..... | | | EOT | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | STP | | ..... | | | |
| Execution | | | | | | | | | | Data Compared Between the FDD and Main-System |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | C | | ..... | | | |
| | R | ..... | | | H | | ..... | | | Sector ID |
| | R | ..... | | | R | | ..... | | | Information After |
| | R | ..... | | | N | | ..... | | | Command Execution |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| **Scan Low or Equal** | | | | | | | | | | |
| Command | W | MT | MFM | SK | 1 | 1 | 0 | 0 | 1 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | C | | ..... | | | Sector ID |
| | W | ..... | | | H | | ..... | | | Information Prior |
| | W | ..... | | | R | | ..... | | | to Command |
| | W | ..... | | | N | | ..... | | | Execution |
| | W | ..... | | | EOT | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | STP | | ..... | | | |
| Execution | | | | | | | | | | Data Compared Between the FDD and Main-System |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | C | | ..... | | | |
| | R | ..... | | | H | | ..... | | | Sector ID |
| | R | ..... | | | R | | ..... | | | Information After |
| | R | ..... | | | N | | ..... | | | Command Execution |
| **Scan High or Equal** | | | | | | | | | | |
| Command | W | MT | MFM | SK | 1 | 1 | 1 | 0 | 1 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | C | | ..... | | | Sector ID |
| | W | ..... | | | H | | ..... | | | Information Prior |
| | W | ..... | | | R | | ..... | | | to Command |
| | W | ..... | | | N | | ..... | | | Execution |
| | W | ..... | | | EOT | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | STP | | ..... | | | |
| Execution | | | | | | | | | | Data Compared Between the FDD and Main-System |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information |
| | R | ..... | | | ST 1 | | ..... | | | After Command |
| | R | ..... | | | ST 2 | | ..... | | | Execution |
| | R | ..... | | | C | | ..... | | | |
| | R | ..... | | | H | | ..... | | | Sector ID |
| | R | ..... | | | R | | ..... | | | Information After |
| | R | ..... | | | N | | ..... | | | Command Execution |
| **Recalibrate** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | 0 | DS0 | DS1 | Enhanced Controller |
| Execution | | | | | | | | | | Head Retracted to Track 0 Interrupt |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus | | | | | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| **Sense Interrupt Status** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Command Codes |
| Result | R | ..... | | | ST 0 | | ..... | | | Status Information at the End of Each Seek Operation |
| | R | ..... | | | PCN | | ..... | | | |
| **Specify** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Command Codes |
| | W | ..... | SRT | ..... | | ..... | HUT | ..... | | |
| | W | ..... | | HLT | ..... | | ND | | | |
| **Sense Drive Status** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| Result | R | ..... | | | ST 3 | | ..... | | | Status Information About FDD |
| **Drive Specification Command** | | | | | | | | | | |
| Command | W | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | Command Code |
| | W | 0 | FD1 | FD0 | PTS | DRT1 | DRT0 | DT1 | DT0 | |
| | : | : | : | : | : | : | : | : | : | 0–4 bytes issued |
| | W | DN | NRP | 0 | 0 | 0 | 0 | 0 | 0 | |
| Result | R | 0 | 0 | 0 | PTS | DRT1 | DRT0 | DT1 | DT0 | Drive 0 |
| | R | 0 | 0 | 0 | PTS | DRT1 | DRT0 | DT1 | DT0 | Drive 1 |
| | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RSVD |
| | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RSVD |
| **Seek** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | Command Codes |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| Execution | W | ..... | | | NCN | | ..... | | | Head is Positioned Over Proper Cylinder on Diskette |
| **Configure** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Command Code |
| | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | W | 0 | EIA | EFIFO | POLL | ..... | FIFOTHR | ..... | | |
| | W | ..... | | | PRETRK | | ..... | | | |
| **Relative Seek** | | | | | | | | | | |
| Command | W | 1 | DIR # | 0 | 0 | 1 | 1 | 1 | 1 | Command Code |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | RCN | | ..... | | | |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus | | | | | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| **DUMPREG** | | | | | | | | | | |
| Command Execution | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | Note: Registers placed in FIFO |
| Result | R | ..... | | | | PCN-Drive 0 | | | | |
| | R | ..... | | | | PCN-Drive 1 | | | | |
| | R | ..... | | | | PCN-Drive 2 | | | | |
| | R | ..... | | | | PCN-Drive 3 | | | | |
| | R | ..... | SRT | | | | HUT | | | |
| | R | ..... | | HLT | | | | | ND | |
| | | ..... | | | | SC/EOT | | | | |
| | R | LOCK | 0 | 0 | 0 | D1 | D0 | GAP | WGATE | |
| | R | 0 | EIS | EFIFO | | POLL | FIFOTHR | | | |
| | R | ..... | | | | PRETRK | | | | |
| **Read ID** | | | | | | | | | | |
| Command | W | 0 | MFM | 0 | 0 | 1 | 0 | 1 | 0 | Commands |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | The First Correct ID Information on the Cylinder is Stored in Data Register |
| Result | R | ..... | | | | ST 0 | | | | Status Information |
| | R | ..... | | | | ST 1 | | | | After Command |
| | R | ..... | | | | ST 2 | | | | Execution |
| | R | ..... | | | | C | | | | |
| | R | ..... | | | | H | | | | Disk Status After the |
| | R | ..... | | | | R | | | | Command has |
| | R | ..... | | | | N | | | | Completed |
| **Perpendicular Mode** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Command Codes |
| | W | OW | 0 | 0 | 0 | D1 | D0 | GAP | WGATE | |
| **Lock** | | | | | | | | | | |
| Command | W | LOCK | 0 | 0 | 1 | 0 | 1 | 0 | 0 | Command Codes |
| Result | R | 0 | 0 | 0 | LOCK | 0 | 0 | 0 | 0 | |
| **Part ID** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Command Code |
| Result | R | 0 | 0 | 0 | ..... | Stepping | | ..... | 1 | Part ID Number |
| **Powerdown Mode** | | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | Command Code |
| | W | 0 | 0 | EREG EN | 0 | 0 | FDI TRI | MIN DLY | AUTO PD | |
| Result | R | 0 | 0 | EREG EN | 0 | 0 | FDI TRI | MIN DLY | AUTO PD | |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus | | | | | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| **Option** | | | | | | | | | | |
| Command | W | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Command Code |
| | W | . . . . . | | RSVD | | . . . . . | | | ISO | |
| **Save** | | | | | | | | | | |
| Command | W | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | Command Code |
| Result | R | RSVD | RSVD | PD OSC | PC2 | PC1 | PC0 | DRATE1 | DRATE0 | Save Information to Reprogram the FDC |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ISO | |
| | R | . . . . . | | | | PCN-Drive 0 | | | | |
| | R | . . . . . | | | | PCN-Drive 1 | | | | |
| | R | . . . . . | | | | PCN-Drive 2 | | | | |
| | R | . . . . . | | | | PCN-Drive 3 | | | | |
| | R | . . . . . | SRT | . . . . . | | . . . . . | HUT | . . . . . | | |
| | R | . . . . . | | | | HLT | . . . . . | | ND | |
| | R | . . . . . | | | | SC/EOT | . . . . . | | | |
| | R | LOCK | 0 | 0 | 0 | D1 | D0 | GAP | WGATE | |
| | R | 0 | EIS | EFIFO | POLL | — | . . . . . | FIFOTHR | . . . . . | |
| | R | . . . . . | | | | PRETRK | . . . . . | | | |
| | R | 0 | 0 | EREG EN | 0 | RSVD | FDI TRI | MIN DLY | AUTO PD | |
| | R | . . . . . | | | | DISK/STATUS | . . . . . | | | |
| | R | . . . . . | | | | RSVD | . . . . . | | | |
| | R | . . . . . | | | | RSVD | . . . . . | | | |
| **Restore** | | | | | | | | | | |
| Command | W | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | Command Code |
| | W | 0 | 0 | 0 | PC2 | PC1 | PC0 | DRATE1 | DRATE0 | Restore Original |
| | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ISO | Register Status |
| | W | . . . . . | | | | PCN-Drive 0 | | | | |
| | W | . . . . . | | | | PCN-Drive 1 | | | | |
| | W | . . . . . | | | | PCN-Drive 2 | | | | |
| | W | . . . . . | | | | PCN-Drive 3 | | | | |
| | W | . . . . . | SRT | . . . . . | | . . . . . | HUT | . . . . . | | |
| | W | . . . . . | | | HLT | . . . . . | | | ND | |
| | W | . . . . . | | | | SC/EOT | . . . . . | | | |
| | W | LOCK | 0 | 0 | 0 | D1 | D0 | GAP | WGATE | |
| | W | 0 | EIS | EFIFO | POLL | . . . . . | FIFOTHR | . . . . . | | |
| | W | . . . . . | | | | PRETRK | . . . . . | | | |
| | W | 0 | 0 | EREG EN | 0 | RSVD | FDI TRI | MIN DLY | AUTO PD | |
| | W | . . . . . | | | | DISK/STATUS | . . . . . | | | |
| | W | . . . . . | | | | RSVD | . . . . . | | | |
| | W | . . . . . | | | | RSVD | . . . . . | | | |

**Table 24. FDC Command Set** (Continued)

| Phase | R/W | Data Bus | | | | | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|
| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| **Format and Write** | | | | | | | | | | |
| Command | W | 1 | MFM | 1 | 0 | 1 | 1 | 0 | 1 | Command Code |
| | W | 0 | 0 | 0 | 0 | 0 | HDS | DS1 | DS0 | |
| | W | ..... | | | N | | ..... | | | |
| | W | ..... | | | SC | | ..... | | | |
| | W | ..... | | | GPL | | ..... | | | |
| | W | ..... | | | D | | ..... | | | |
| Execution | W | ..... | | | C | | ..... | | | Input |
| repeated | W | ..... | | | H | | ..... | | | Sector |
| for each | W | ..... | | | R | | ..... | | | Parameters |
| sector | W | ..... | | | N | | ..... | | | |
| | W | Data Transfer Of N Bytes | | | | | | | | |
| | | | | | | | | | | FDC Formats and Writes Entire Track |
| Result | R | ..... | | | ST 0 | | ..... | | | |
| | R | ..... | | | ST 1 | | ..... | | | |
| | R | ..... | | | ST 2 | | ..... | | | |
| | R | ..... | | | Undefined | | ..... | | | |
| | R | ..... | | | Undefined | | ..... | | | |
| | R | ..... | | | Undefined | | ..... | | | |
| | R | ..... | | | Undefined | | ..... | | | |
| **Invalid** | | | | | | | | | | |
| Command | W | ..... | | | Invalid Codes | | ..... | | | Invalid Command Codes (Noop—FDC goes into Standby State) |
| Result | R | ..... | | | ST 0 | | ..... | | | ST 0 = 80 |

**Parameter Abbreviations**

| Symbol | Description |
|---|---|
| AUTO PD | **AUTO POWERDOWN CONTROL:** When AUTO PD = 0, automatic powerdown is disabled. When AUTO PD = 1, automatic powerdown is enabled. |
| C | **CYLINDER ADDRESS:** The currently selected cylinder address, 0 to 255. |
| D0, D1 | **DRIVE SELECT 0-1:** Designates which drives are Perpendicular drives. A 1 indicates Perpendicular drive. |
| D | **DATA PATTERN:** The pattern to be written in each sector data field during formatting. |
| DN | **DONE:** This bit indicates that this is the last byte of the drive specification command. The FDC checks to see if this bit is 1 or 0. When DN = 0, the FDC expects more bytes.<br>DN = 0 FDC expects more subsequent bytes.<br>DN = 1 Terminates the command phase and enters the results phase. An additional benefit is that by setting this bit to 1, a direct check of the current drive specifications can be done. |
| DIR# | **DIRECTION CONTROL:** When DIR# = 0, the head steps out from the spindle during a relative seek. When DIR# = 1, the head steps in toward the spindle. |
| DS0, DS1 | **DISK DRIVE SELECT:** |

| DS1 | DS0 | Drive Slot |
|---|---|---|
| 0 | 0 | drive 0 |
| 0 | 1 | drive 1 |
| 1 | 0 | drive 2* |
| 1 | 1 | drive 3* |

*Available when FDDQTY = 1 in the FCFG1 Register (see Appendix A, FDC Four Drive Support)

| | |
|---|---|
| DTL | **SPECIAL SECTOR SIZE:** By setting N to zero (00), DTL may be used to control the number of bytes transferred in disk read/write commands. The sector size (N = 0) is set to 128. If the actual sector (on the diskette) is larger than DTL, the remainder of the actual sector is read but is not passed to the host during read commands; during write commands, the remainder of the actual sector is written with all zero bytes. The CRC check code is calculated with the actual sector. When N is not zero, DTL has no meaning and should be set to FFh. |
| DRATE[0:1] | **DATA RATE:** Data rate values from the DSR register. |

**Symbol** **Description**

DRT0, DRT1 **DATA RATE TABLE SELECT:** These two bits select between the different data rate tables. The default is the conventional table. These also provide mapping of the data rates selected in the DSR and CCR. The table below shows this.

| Bits in DSR | | | | | |
|---|---|---|---|---|---|
| **DRT1** | **DRT0** | **DRATE1** | **DRATE0** | **Data Rate** | **Operation** |
| 0 | 0 | 1 | 1 | 1 Mbps | Default |
| | | 0 | 0 | 500 Kbps | |
| | | 0 | 1 | 300 Kbps | |
| | | 1 | 0 | 250 Kbps | |
| 0 | 1 | RSVD | RSVD | RSVD | RSVD |
| 1 | 0 | RSVD | RSVD | RSVD | RSVD |
| 1 | 1 | 1 | 1 | 1 Mbps | Perpendicular mode FDDs |
| | | 0 | 0 | 500 Kbps | |
| | | 0 | 1 | Illegal | |
| | | 1 | 0 | 250 Kbps | |

DT0,DT1 **DRIVE DENSITY SELECT TYPE:** These bits select the outputs on DRVDEN0 and DRVDEN1 (see DRIVE SPECIFICATION Command).

EC **ENABLE COUNT:** When EC = 1, the DTL parameter of the Verify Command becomes SC (Number of sectors per track).

EFIFO **Enable FIFO:** When EFIFO = 0, the FIFO is enabled. EFIFO = 1 puts the FDC in the 8272A compatible mode where the FIFO is disabled.

EIS **ENABLE IMPLIED SEEK:** When EIS = 1, a seek operation is performed before executing any read or write command that requires the C parameter in the command phase. EIS = 0 disables the implied seek.

EOT **END OF TRACK:** The final sector number of the current track.

EREG EN **ENHANCED REGISTER ENABLE:** When EREG EN = 1, the TDR register is extended and SRB is made visible to the user. When EREG EN = 0, the standard registers are used.

FDI TRI **FLOPPY DRIVE INTERFACE TRI-STATE:** When FDI TRI = 0, the output pins of the floppy disk drive interface are tri-stated. This is also the default state. When FDI TRI = 1, the floppy disk drive interface remains unchanged.

| Symbol | Description |
|---|---|
| FD0, FD1 | **FLOPPY DRIVE SELECT:** These two bits select which physical drive is being specified. The FDn corresponds to FDSn and FDMEn on the floppy drive interface. The drive is selected independent of the BOOTSEL bit in the TDR. Refer to Section 8.1.3, TDR—Enhanced Tape Drive Register, which explains the distinction between physical drives and their virtual mapping as defined by the BOOTSEL bit. |

| FD1 | FD0 | Drive slot |
|---|---|---|
| 0 | 0 | drive 0 |
| 1 | 0 | drive 1 |
| 0 | 1 | drive 2* |
| 1 | 1 | drive 3* |

*Available if the four floppy drive option is selected in the FCFG1 Register.

| | |
|---|---|
| GAP | **GAP:** Alters Gap 2 length when using Perpendicular Mode. |
| GPL | **GAP LENGTH:** The gap 3 size. (Gap 3 is the space between sectors excluding the VCO synchronization field). |
| H/HDS | **HEAD ADDRESS:** Selected head: 0 or 1 (disk side 0 or 1) as encoded in the sector ID field. |
| HLT | **HEAD LOAD TIME:** The time interval that FDC waits after loading the head and before initiating a read or write operation. Refer to the SPECIFY Command for actual delays. |
| HUT | **HEAD UNLOAD TIME:** The time interval from the end of the execution phase (of a read or write command) until the head is unloaded. Refer to the SPECIFY Command for actual delays. |
| ISO | **ISO FORMAT:** When ISO = 1, the ISO format is used for all data transfer commands. When ISO = 0, the normal IBM system 34 and perpendicular is used. The default is ISO = 0. |
| LOCK | **LOCK:** Lock defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE Command can be reset to their default values by a software reset (Reset made by setting the proper bit in the DSR or DOR registers). |
| MFM | **MFM MODE:** A one selects the double density (MFM) mode. A zero is reserved. |

| Symbol | Description |
|---|---|
| MIN DLY | **MINIMUM POWERUP TIME CONTROL:** This bit is active only if AUTO PD bit is enabled. When MIN DLY = 0, a 10 ms minimum powerup time is assigned and when MIN DLY = 1, a 0.5 sec. minimum powerup time is assigned. |
| MT | **MULTI-TRACK SELECTOR:** When MT = 1, the multi-track operating mode is selected. In this mode, the FDC treats a complete cylinder, under head 0 and 1, as a single track. The FDC operates as if this expanded track started at the first sector under head 0 and ended at the last sector under head 1. With this flag set, a multitrack read or write operation will automatically continue to the first sector under head 1 when the FDC finishes operating on the last sector under head 0. |
| N | **SECTOR SIZE CODE:** This specifies the number of bytes in a sector. When N = 00h, the sector size is 128 bytes. The number of bytes transferred is determined by the DTL parameter. Otherwise the sector size is (2 raised to the ''N'th'' power) times 128. All values up to 07h are allowable. A value of 07h equals a sector size of 16 Kbytes. It is the users responsibility to not select combinations that are not possible with the drive. |

| N | Sector Size |
|---|---|
| 00 | 128 bytes |
| 01 | 256 bytes |
| 02 | 512 bytes |
| 03 | 1024 |
| . . | . . . |
| 07 | 16 Kbytes |

| | |
|---|---|
| NCN | **NEW CYLINDER NUMBER:** The desired cylinder number. |
| ND | **NON-DMA MODE FLAG:** When ND = 1, the FDC operates in the non-DMA mode. In this mode, the host is interrupted for each data transfer. When ND = 0, the FDC operates in DMA mode and interfaces to a DMA controller by means of the DRQ and DACK# signals. |
| NRP | **NO RESULTS PHASE:** When NRP = 1, the result phase is skipped. When NRP = 0, the result phase is generated. |
| OW | **OVERWRITTEN:** The bits denoted D0 and D1 of the PERPENDICULAR MODE Command can only be overwritten when OW = 1. |

| Symbol | Description |
|---|---|
| PCN | **PRESENT CYLINDER NUMBER:** The current position of the head at the completion of SENSE INTERRUPT STATUS Command. |
| PC2,PC1,PC0 | **PRECOMPENSATION VALUES:** Precompensation values from the DSR register. |
| PDOSC | **POWERDOWN OSCILLATOR:** When this bit is set, the internal oscillator is turned off. |
| PTS | **PRECOMPENSATION TABLE SELECT:** This bit selects whether to enable the precompensation value programmed in the DSR or not. In the default state, the value programmed in DSR will be used. More information regarding the precompensation is available in Section 8.1.5.<br>PTS = 0 DSR programmed precompensation delays<br>PTS = 1 No precompensation delay is selected for the corresponding drive. |
| POLL | **POLLING DISABLE:** When POLL = 1, the internal polling routine is disabled. When POLL = 0, polling is enabled. |
| PRETRK | **PRECOMPENSATION START TRACK NUMBER:** Programmable from track 00 to FFh. |
| R | **SECTOR ADDRESS:** The sector number to be read or written. In multi-sector transfers, this parameter specifies the sector number of the first sector to be read or written. |
| RCN | **RELATIVE CYLINDER NUMBER:** Relative cylinder offset from present cylinder as used by the RELATIVE SEEK Command. |
| SC | **NUMBER OF SECTORS:** The number of sectors to be initialized by the FORMAT Command. The number of sectors to be verified during a Verify Command, when EC = 1. |
| SK | **SKIP FLAG:** When SK = 1, sectors containing a deleted data address mark will automatically be skipped during the execution of a READ DATA Command. If a READ DELETED DATA Command is executed, only sectors with a deleted address mark will be accessed. When SK = 0, the sector is read or written the same as the read and write commands. |
| SRT | **STEP RATE INTERVAL:** The time interval between step pulses issued by the FDC. Programmable from 0.5 ms to 8 ms, in increments of 0.5 ms at the 1 Mbit data rate. Refer to the SPECIFY Command for actual delays. |
| ST0-3 | **STATUS REGISTERS 0-3.**Registers within the FDC that store status information after a command has been executed. This status information is available to the host during the result phase after command execution. |
| WGATE | **WRITE GATE:** Write gate alters timing of WE, to allow for pre-erase loads in perpendicular drives. |

### 8.5.1 STATUS REGISTER ENCODING

The contents of these registers are available only through a command sequence.

#### 8.5.1.1 Status Register 0

| Bit # | Symbol | Name | Description |
|---|---|---|---|
| 7,6 | IC | Interrupt Code | 00 Normal termination of command. The specified command was properly executed and completed without error.<br>01 Abnormal termination of command. Command execution was started, but was not successful completed.<br>10 Invalid command. The requested command could not be executed.<br>11 Abnormal termination caused by Polling. |
| 5 | SE | Seek End | The 82091AA completed a SEEK or RECALIBRATE command, or a READ or WRITE with implied seek command. |
| 4 | EC | Equipment Check | The TRK pin failed to become a ‘‘1’’ after:<br>1. 80 step pulses in the RECALIBRATE COMMAND.<br>2. The RELATIVE SEEK command causes the 82078 to step outward beyond Track 0. |
| 3 | — | — | Unused. This bit is always ‘‘0’’. |
| 2 | H | Head Address | The current head address. |
| 1,0 | DS1,0 | Drive Select | The current selected drive. |

#### 8.5.1.2 Status Register 1

| Bit # | Symbol | Name | Description |
|---|---|---|---|
| 7 | EN | End of Cylinder | The 82078 tried to access a section beyond the final sector of the track (255D). Will be set if TC is not issued after Read or Write Data Command. |
| 6 | — | — | Unused. This bit is always ‘‘0’’. |
| 5 | DE | Data Error | The 82078 detected a CRC error in either the ID field or the data field of a sector. |
| 4 | OR | Overrun/Underrun | Becomes set if the 82078 does not receive CPU or DMA service within the required time interval, resulting in data overrun or underrun. |
| 3 | — | — | Unused. Ths bit is always ‘‘0’’. |
| 2 | ND | No Data | Any one of the following:<br>1. READ DATA, READ DELETED DATA command, the 82091AA did not find the specified sector.<br>2. READ ID command, the 82091AA cannot read the ID field without an error.<br>3. READ TRACK command, the 82091AA cannot find the proper sector sequence. |
| 1 | NW | Not Writable | WP pin became a ‘‘1’’ while the 82091AA is executing a WRITE DATA, WRITE DELETED DATA, or FORMAT TRACK command. |
| 0 | MA | Missing Address Mark | Any one of the the following:<br>1. The 82091AA did not detect an ID address mark at the specified track after encountering the index pulse from the INDX# pin twice.<br>2. The 82091AA cannot detect a data address mark or a deleted data address mark on the specified track. |

#### 8.5.1.3 Status Register 2

| Bit # | Symbol | Name | Description |
|---|---|---|---|
| 7 | — | — | Unused. This bit is always ‘‘0’’. |
| 6 | CM | Control Mark | Any one of the following:<br>1. READ DATA command, the 82078 encounters a deleted data address mark.<br>2. READ DELETED DATA command, the 82078 encountered a data address mark. |
| 5 | DD | Data Error in Data Field | The 82091AA detected a CRC error in the data field. |
| 4 | WC | Wrong Cylinder | The track address from the sector ID field is different from the track address maintained inside the 82091AA. |
| 3 | — | — | Unused. This bit is always ‘‘0’’. |
| 2 | — | — | Unused. This bit is always ‘‘0’’. |
| 1 | BC | Bad Cylinder | The track address from the sector ID field is different from the track address maintained inside the 82091AA and is equal to FF hex which indicates a bad track with a hard error according to the IBM soft-sectored format. |
| 0 | MD | Missing Data Address Mark | The 82091AA cannot detect a data address mark or a deleted data address mark. |

#### 8.5.1.4 Status Register 3

| Bit # | Symbol | Name | Description |
|---|---|---|---|
| 7 | — | — | Unused. This bit is always ‘‘0’’. |
| 6 | WP | Write Protected | Indicates the status of the WP pin. |
| 5 | — | — | Unused. This bit is always ‘‘0’’. |
| 4 | T0 | Track 0 | Indicates the status of the TRK0 pin. |
| 3 | — | — | Unused. This bit is always ‘‘0’’. |
| 2 | HD | Head Address | Indicates the status of the HDSEL pin. |
| 1,0 | DS1,0 | Drive Select | Indicates the status of the DS1, DS0 pins. |

### 8.5.2 DATA TRANSFER COMMANDS

All of the READ DATA, WRITE DATA and VERIFY type commands use the same parameter bytes and return the same results information. The only difference being the coding of bits[4:0] in the first byte.

An implied seek will be executed if the feature was enabled by the CONFIGURE Command. This seek is completely transparent to the user. The Drive Busy bit for the drive will go active in the Main Status Register during the seek portion of the command. A seek portion failure is reflected in the results status normally returned for a READ/WRITE DATA Command. Status Register 0 (ST0) contains the error code and C contains the cylinder that the seek failed.

#### 8.5.2.1 Read Data

A set of nine bytes is required to place the FDC into the Read Data Mode. After the READ DATA Command has been issued, the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the SPECIFY Command), and begins reading ID address marks and ID fields. When the sector address read from the diskette matches with the sector address specified in the command, the FDC reads the sector's data field and transfers the data to the FIFO.

After completion of the read operation from the current sector, the sector address is incremented by one, and the data from the next logical sector is read and output via the FIFO. This continuous read function is called ''Multi-Sector Read Operation''. Upon receipt of TC or an implied TC (FIFO overrun/underrun), the FDC stops sending data. However, the FDC will continue to read data from the current sector, check the CRC bytes, and, at the end of the sector, terminate the READ DATA Command.

N determines the number of bytes per sector (Table 25). If N is set to zero, the sector size is set to 128. The DTL value determines the number of bytes to be transferred. If DTL is less than 128, the FDC transfers the specified number of bytes to the host. For reads, it continues to read the entire 128 byte sector and checks for CRC errors. For writes it completes the 128 byte sector by filling in zeroes. If N is not set to 00h, DTL should be set to FFh, and has no impact on the number of bytes transferred.

**Table 25. Sector Sizes**

| N | Sector Size |
|---|---|
| 00 | 128 Bytes |
| 01 | 256 Bytes |
| 02 | 512 Bytes |
| 03 | 1024 Bytes |
| . . . | . . . |
| 07 | 16 KBytes |

The amount of data that can be handled with a single command to the FDC depends on MT (multitrack) and N (Number of bytes/sector).

**Table 26. Effects of MT and N Bits**

| MT | N | Max. Transfer Capacity | Final Sector Read from Disk |
|---|---|---|---|
| 0 | 1 | 256 × 26 = 656 | 26 at side 0 or 1 |
| 1 | 1 | 256 × 52 = 13312 | 26 at side 1 |
| 0 | 2 | 512 × 15 = 7680 | 15 at side 0 or 1 |
| 1 | 2 | 512 × 30 = 15360 | 15 at side 1 |
| 0 | 3 | 1024 × 8 = 8192 | 8 at side 0 or 1 |
| 1 | 3 | 1024 × 16 = 16384 | 16 at side 1 |

The Multi-Track function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at sector 1, side 0 and completing at the last sector of the same track at side 1.

If the host terminates a read or write operation in the FDC, the ID information in the result phase is dependent on the state of the MT bit and EOT byte. Refer to Table 29. The termination must be normal.

At the completion of the READ DATA Command, the head is not unloaded until after the Head Unload Time Interval (specified in the SPECIFY Command) has elapsed. If the host issues another command before the head unloads, the head settling time may be saved between subsequent reads.

If the FDC detects a pulse on the INDEX# pin twice without finding the specified sector (meaning that the diskette's index hole passes through index detect logic in the drive twice), the FDC sets the IC

code in Status Register 0 to 01 (Abnormal termination), sets the ND bit in Status Register 1 to 1 indicating a sector not found and terminates the READ DATA Command.

After reading the ID and data fields in each sector, the FDC checks the CRC bytes. If a CRC error occurs in the ID or data field, the FDC sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the DE bit flag in Status Register 1 to 1, sets the DD bit in Status Register 2 to 1 if CRC is incorrect in the ID field, and terminates the READ DATA Command.

Table 27 describes the affect of the SK bit on the READ DATA command execution and results.

#### 8.5.2.2 Read Deleted Data

This command is the same as the READ DATA Command, except that it operates on sectors that contain a deleted data address mark at the beginning of a data field. Table 28 describes the affect of the SK bit on the READ DELETED DATA Command execution and results.

**Table 27. Skip Bit vs READ DATA Command**

| SK Bit Value | Data Address Mark Type Encountered | Sector Read | Results CM Bit of ST2 Set? | Description of Results |
|---|---|---|---|---|
| 0 | Normal Data | Yes | No | Normal Termination |
| 0 | Deleted Data | Yes | Yes | Address Not Incremented. Next Sector Not Searched For. |
| 1 | Normal Data | Yes | No | Normal Termination |
| 1 | Deleted Data | No | Yes | Normal Termination Sector Not Read (“Skipped”) |

Except where noted in Table 27, the C or R value of the sector address is automatically incremented (see Table 29).

**Table 28. Skip Bit vs READ DELETED DATA Command**

| SK Bit Value | Data Address Mark Type Encountered | Sector Read | Results CM Bit of ST2 Set? | Description of Results |
|---|---|---|---|---|
| 0 | Normal Data | Yes | Yes | Normal Termination |
| 0 | Deleted Data | Yes | No | Address Not Incremented. Next Sector Not Searched For. |
| 1 | Normal Data | No | Yes | Normal Termination Sector Not Read (“Skipped”) |
| 1 | Deleted Data | Yes | No | Normal Termination |

Except where noted in Table 28, the C or R value of the sector address is automatically incremented (see Table 29).

**Table 29. Result Phase**

| MT | Head | Final Sector Transferred to Host | ID Information at Result Phase | | | |
|---|---|---|---|---|---|---|
| | | | C | H | R | N |
| | 0 | Less than EOT | NC | NC | R + 1 | NC |
| 0 | | Equal to EOT | C + 1 | NC | 01 | NC |
| | 1 | Less than EOT | NC | NC | R + 1 | NC |
| | | Equal to EOT | C + 1 | NC | 01 | NC |
| | 0 | Less than EOT | NC | NC | R + 1 | NC |
| 1 | | Equal to EOT | NC | LSB | 01 | NC |
| | 1 | Less than EOT | NC | NC | R + 1 | NC |
| | | Equal to EOT | C + 1 | LSB | 01 | NC |

**NOTE:**
1. NC = no change; the same value as the one at the beginning of command execution.
2. LSB = least significant bit; the LSB of H is complemented.

#### 8.5.2.3 Read Track

This command is similar to the READ DATA Command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering a pulse on the INDEX# pin, the FDC starts to read all data fields on the track as continuous blocks of data without regard to logical sector numbers. If the FDC finds an error in the ID or DATA CRC check bytes, it continues to read data from the track and sets the appropriate error bits at the end of the command. The FDC compares the ID information read from each sector with the specified value in the command and sets the ND flag to 1 in Status Register 1 if there is no comparison. Multi-track or skip operations are not allowed with this command. The MT and SK bits (Bits D7 and D5 of the first command byte respectively) should always be set to 0.

This command terminates when the EOT specified number of sectors have been read. If the FDC does not find an ID address mark on the diskette after the second occurrence of a pulse on the INDEX# pin, then it sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the MA bit in Status Register 1 to 1, and terminates the command.

#### 8.5.2.4 Write Data

After the WRITE DATA Command has been issued, the FDC loads the head (if it is in the unloaded state), waits the specified head load time if unloaded (defined in the SPECIFY Command), and begins reading ID fields. When the sector address read from the diskette matches the sector address specified in the command, the FDC reads the data from the host via the FIFO, and writes it to the sector's data field.

After writing data into the current sector, the FDC computes the CRC value and writes it into the CRC field at the end of the sector transfer. The sector number stored in R is incremented by one, and the FDC continues writing to the next data field. The FDC continues this multi-sector write operation. If a terminal count signal is received or a FIFO over/under run occurs while a data field is being written, the remainder of the data field is filled with zeros.

The FDC reads the ID field of each sector and checks the CRC bytes. If the FDC detects a CRC error in one of the ID fields, it sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the DE bit of Status Register 1 to 1, and terminates the WRITE DATA Command.

The WRITE DATA Command operates in much the same manner as the READ DATA Command. The following items are the same. Please refer to the READ DATA Command for details:

- Transfer Capacity
- EN (End of Cylinder) bit
- ND (No Data) bit
- Head Load, Unload Time Interval
- ID information when the host terminates the command
- Definition of DTL when N = 0 and when N does not = 0

#### 8.5.2.5 Verify

The VERIFY Command is used to verify the data stored on a disk. This command acts exactly like a READ DATA Command except that no data is transferred to the host. Data is read from the disk, and CRC is computed and checked against the previously stored value.

Because no data is transferred to the host, the TC signal cannot be used to terminate this command. By setting the EC bit to 1, an implicit TC will be issued to the FDC. This implicit TC occurs when the SC value has decrement to 0 (a SC value of 0 verifies 256 sectors). This command can also be terminated by setting the EC bit to 0 and the EOT value equal to the final sector to be checked. When EC = 0, DTL/SC should be programmed to 0FFh. Refer to Table 29 and Table 30 for information concerning the values of MT and EC versus SC and EOT value.

Definitions:

# Sectors Per Side = Number of formatted sectors per each side of the disk.

# Sectors Remaining = Number of formatted sectors left that can be read, including side 1 of the disk when MT = 1.

**Table 30. Verify Command Result Phase**

| MT | EC | SC/EOT Value | Termination Result |
|---|---|---|---|
| 0 | 0 | SC = DTL<br>EOT ≤ # Sectors Per Side | Successful Termination<br>Result Phase Valid |
| 0 | 0 | SC = DTL<br>EOT > # Sectors Per Side | Unsuccessful Termination<br>Result Phase Invalid |
| 0 | 1 | SC ≤ # Sectors Remaining<br>AND<br>EOT ≤ # Sectors Per Side | Successful Termination<br>Result Phase Valid |
| 0 | 1 | SC > # Sectors Remaining<br>OR<br>EOT > # Sectors Per Side | Unsuccessful Termination<br>Result Phase Invalid |
| 1 | 0 | SC = DTL<br>EOT ≤ # Sectors Per Side | Successful Termination<br>Result Phase Valid |
| 1 | 0 | SC = DTL<br>EOT > # Sectors Per Side | Unsuccessful Termination<br>Result Phase Invalid |
| 1 | 1 | SC ≤ # Sectors Remaining<br>AND<br>EOT ≤ # Sectors Per Side | Successful Termination<br>Result Phase Valid |
| 1 | 1 | SC > # Sectors Remaining<br>OR<br>EOT > # Sectors Per Side | Unsuccessful Termination<br>Result Phase Invalid |

**NOTE:**
When MT = 1 and the SC value is greater than the number of remaining formatted sectors on Side 0, verification continues on Side 1 of the disk.

#### 8.5.2.6 Format Track

The FORMAT TRACK Command allows an entire track to be formatted. After a pulse from the INDEX# pin is detected, the FDC starts writing data on the disk including gaps, address marks, ID fields and data fields, per the IBM* System 34 (MFM). The particular values written to the gap and data field are controlled by the values programmed into N, SC, GPL, and D which are specified by the host during the command phase. The data field of the sector is filled with the data byte specified by D. The ID field for each sector is supplied by the host. That is, four data bytes per sector are needed by the FDC for C, H, R, and N (cylinder, head, sector number, and sector size, respectively).

After formatting each sector, the host must send new values for C, H, R, and N to the FDC for the next sector on the track. The R value (sector number) is the only value that must be changed by the host after each sector is formatted. This allows the disk to be formatted with nonsequential sector addresses (inter-leaving). This incrementing and formatting continues for the whole track until the FDC encounters a pulse on the INDEX# pin again and it terminates the command.

Table 31 contains typical values for gap fields that are dependent on the size of the sector and the number of sectors on each track. Actual values can vary due to drive electronics.

**Table 31. Typical PC/AT Values for Formatting**

| Drive Form | MEDIA | Sector Size | N | SC | GPL1 | GPL2 |
|---|---|---|---|---|---|---|
| 5.25″ | 1.2 MB | 512 | 02 | 0F | 2A | 50 |
| | 360 KB | 512 | 02 | 09 | 2A | 50 |
| 3.5″ | 2.88 MB | 512 | 02 | 24 | 38 | 53 |
| | 1.44 MB | 512 | 02 | 18 | 1B | 54 |
| | 720 KB | 512 | 02 | 09 | 1B | 54 |

**NOTES:**
1. All values are in hex, except sector size.
2. Gap3 is programmable during reads, writes, and formats.
3. GPL1 = suggested Gap3 values in read and write commands to avoid splice point between data field and ID field of contiguous sections.
4. GPL2 = suggested Gap3 value in FORMAT TRACK Command.

### 8.5.2.7 Format Field

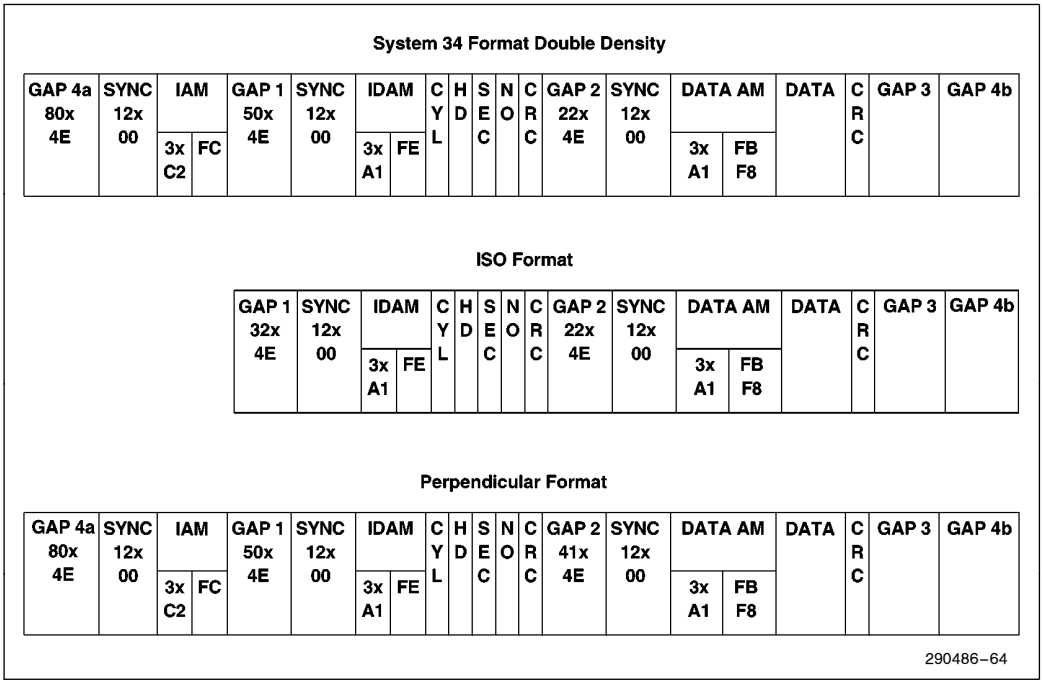



**Figure 64. System 34, ISO and Perpendicular Formats**

### 8.5.3 CONTROL COMMANDS

Control commands differ from the other commands in that no data transfer takes place. Three commands generate an interrupt when complete; READ ID, RECALIBRATE and SEEK. The other control commands do not generate an interrupt.

#### 8.5.3.1 READ ID Command

The READ ID Command is used to find the present position of the recording heads. The FDC stores the values from the first ID field it is able to read into its registers. If the FDC does not find an ID address mark on the diskette after the second occurrence of a pulse on the INDEX# pin, it then sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the MA bit in Status Register 1 to 1, and terminates the command.

The following commands will generate an interrupt upon completion. They do not return any result bytes. It is recommended that control commands be followed by the SENSE INTERRUPT STATUS Command. Otherwise, valuable interrupt status information will be lost.

#### 8.5.3.2 RECALIBRATE Command

This command causes the read/write head within the FDC to retract to the track 0 position. The FDC clears the contents of the PCN counter, and checks the status of the TRK0 pin from the FDD. As long as the TRK0 pin is low, the DIR# pin remains 0 and step pulses are issued. When the TRK0 pin goes high, the SE bit in Status Register 0 is set to 1, and the command is terminated. If the TRK0 pin is still low after 79 step pulses have been issued, the FDC sets the SE and the EC bits of Status Register 0 to 1 and terminates the command. Disks capable of handling more than 80 tracks per side may require more than one RECALIBRATE Command to return the head back to physical Track 0.

The RECALIBRATE Command does not have a result phase. The SENSE INTERRUPT STATUS Command must be issued after the RECALIBRATE Command to effectively terminate it and to provide verification of the head position (PCN). During the command phase of the recalibrate operation, the FDC is in the busy state, but during the execution phase it is in a non-busy state. At this time another RECALIBRATE Command may be issued, and in this manner, parallel RECALIBRATE operations may be done on up to 2 drives simultaneously.

After powerup, software must issue a RECALIBRATE Command to properly initialize all drives and the controller.

#### 8.5.3.3 DRIVE SPECIFICATION Command

The FDC uses two pins, DRVDEN0 and DRVDEN1 to select the density for modern drives. These signals inform the drive of the type of diskette in the drive. The DRIVE SPECIFICATION Command specifies the polarity of the DRVDEN0 and DRVDEN1 pins. It also enables/disables DSR programmed precompensation.

This command removes the need for a hardware work-around to accommodate differing specifications among drives. By programming this command during BIOS's POST routine, the floppy disk controller internally configures the correct values for DRVDEN0 and DRVDEN1 with corresponding precompensation value and data rate table enabled for the particular type of drive.

This command is protected from software resets. After executing the DRIVE SPECIFICATION Command, subsequent software resets will not clear the programmed parameters. Only another DRIVE SPECIFICATION Command or hard reset can reset it to default values. The 6 LSBs of the last byte of this command are reserved for future use.

The DRATE0 and DRATE1 are values as programmed in the DSR register. See Table 32 for pin decoding at different data rates.

Table 32 describes the drives that are supported with the DT0, DT1 bits of the DRIVE SPECIFICATION Command:

Table 32. DRVDENn Polarities

| DT1 | DT0 | Data Rate | DRVDEN1 | DRVDEN0 |
|---|---|---|---|---|
| 0* | 0* | 1 Mbps | 1 | 1 |
| | | 500 Kbps | 0 | 1 |
| | | 300 Kbps | 1 | 0 |
| | | 250 Kbps | 0 | 0 |
| 0 | 1 | 1 Mbps | 1 | 0 |
| | | 500 Kbps | 0 | 0 |
| | | 300 Kbps | 1 | 1 |
| | | 250 Kbps | 0 | 1 |
| 1 | 0 | 1 Mbps | 1 | 1 |
| | | 500 Kbps | 0 | 0 |
| | | 300 Kbps | 1 | 0 |
| | | 250 Kbps | 0 | 1 |
| 1 | 1 | 1 Mbps | 1 | 1 |
| | | 500 Kbps | 0 | 0 |
| | | 300 Kbps | 0 | 1 |
| | | 250 Kbps | 1 | 0 |

**NOTE:**
(*) Denotes the default setting

#### 8.5.3.4 SEEK Command

The read/write head within the drive is moved from track to track under the control of the SEEK Command. The FDC compares the PCN which is the current head position with the NCN and performs the following operation if there is a difference:

PCN < NCN: Direction signal to drive set to 1 (step in), and issues step pulses.

PCN > NCN: Direction signal to drive set to 0 (step out), and issues step pulses.

The rate at which step pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY Command. After each step pulse is issued, NCN is compared against PCN, and when NCN = PCN, then the SE bit in Status Register 0 is set to 1, and the command is terminated.

During the command phase of the seek or recalibrate operation, the FDC is in the busy state, but during the execution phase it is in the non-busy state.

Note that if implied seek is not enabled, the read and write commands should be preceded by:

1. SEEK Command;
   Step to the proper track
2. SENSE INTERRUPT STATUS Command;
   Terminate the SEEK Command
3. READ ID.
   Verify head is on proper track
4. Issue READ/WRITE Command.

The SEEK Command does not have a result phase. Therefore, it is highly recommended that the SENSE INTERRUPT STATUS Command be issued after the SEEK Command to terminate it and to provide verification of the head position (PCN). The H bit (Head Address) in ST0 will always return a 0. When exiting DSR Powerdown mode, the FDC clears the PCN value and the status information to zero. Prior to issuing the DSR POWERDOWN Command, it is highly recommended that the user service all pending interrupts through the SENSE INTERRUPT STATUS Command.

#### 8.5.3.5 SENSE INTERRUPT STATUS Command

An interrupt signal on the INT pin is generated by the FDC for one of the following reasons:

1. Upon entering the Result Phase of:
   a. READ DATA Command
   b. READ TRACK Command
   c. READ ID Command
   d. READ DELETED DATA Command
   e. WRITE DATA Command
   f. FORMAT TRACK Command
   g. WRITE DELETED DATA Command
   h. VERIFY Command
2. End of SEEK, RELATIVE SEEK or RECALIBRATE Command
3. FDC requires a data transfer during the execution phase in the non-DMA Mode

The SENSE INTERRUPT STATUS Command resets the interrupt signal and via the IC code and SE bit of Status Register 0, identifies the cause of the interrupt. If a SENSE INTERRUPT STATUS Command is issued when no active interrupt condition is present, the status register ST0 will return a value of 80h (invalid command).

The SEEK, RELATIVE SEEK and the RECALIBRATE Commands have no result phase. The SENSE INTERRUPT STATUS Command must be issued immediately after these commands to terminate them and to provide verification of the head position (PCN). The H (Head Address) bit in ST0 will always return a 0. If a SENSE INTERRUPT STATUS is not issued, the drive, will continue to be busy and may effect the operation of the next command.

#### 8.5.3.6 SENSE DRIVE STATUS Command

The SENSE DRIVE STATUS Command obtains drive status information. It has no execution phase and goes directly to the result phase from the command phase. STATUS REGISTER 3 contains the drive status information.

#### 8.5.3.7 SPECIFY Command

The SPECIFY Command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the execution phase of one of the read/write commands to the head unload state. The SRT (Step Rate Time) defines the time interval between adjacent step pulses. Note that the spacing between the first and second step pulses may be shorter than the remaining step pulses. The HLT (Head Load Time) defines the time between the command phase to the execution phase of a READ DATA or Write Data Command. The Head Unload Time (HUT) timer goes from the end of the execution phase to the begining of the result phase of a READ Data or Write Data Command. The values change with the data rate speed selection and are documented in Table 34.

**Table 33. Interrupt Identification**

| SE | IC | Interrupt Due To |
|---|---|---|
| 0 | 11 | Polling |
| 1 | 00 | Normal Termination of SEEK or RECALIBRATE Command |
| 1 | 01 | Abnormal Termination of SEEK or RECALIBRATE Command |

**Table 34. Drive Control Delays (ms)**

| | HUT | | | | SRT | | | |
|---|---|---|---|---|---|---|---|---|
| | **1 M** | **500K** | **300K** | **250K** | **1 M** | **500K** | **300K** | **250K** |
| 0 | 128 | 256 | 426 | 512 | 8.0 | 16 | 26.7 | 32 |
| 1 | 8 | 16 | 26.7 | 32 | 7.5 | 15 | 25 | 30 |
| .. | .. | .. | .. | .. | .. | .. | .. | .. |
| A | 80 | 160 | 267 | 320 | 3.0 | 6.0 | 10.2 | 12 |
| B | 88 | 176 | 294 | 352 | 2.5 | 5.0 | 8.3 | 10 |
| C | 96 | 192 | 320 | 384 | 2.0 | 4.0 | 6.68 | 8 |
| D | 104 | 208 | 346 | 416 | 1.5 | 3.0 | 5.01 | 6 |
| E | 112 | 224 | 373 | 448 | 1.0 | 2.0 | 3.33 | 4 |
| F | 120 | 240 | 400 | 480 | 0.5 | 1.0 | 1.67 | 2 |

**Table 35. Head Load Time (ms)**

| | HLT | | | |
|---|---|---|---|---|
| | **1M** | **500K** | **300K** | **250K** |
| 00 | 128 | 256 | 426 | 512 |
| 01 | 1 | 2 | 3.3 | 4 |
| 02 | 2 | 4 | 6.7 | 8 |
| .. | .. | .. | .. | .. |
| 7E | 126 | 252 | 420 | 504 |
| 7F | 127 | 254 | 423 | 508 |

The choice of DMA or non-DMA operations is made by the ND bit. When ND = 1, the non-DMA mode is selected, and when ND = 0, the DMA mode is selected. In DMA mode, data transfers are signalled by the DRQ pin. Non-DMA mode uses the RQM bit and the IRQ6 pin to signal data transfers.

#### 8.5.3.8 CONFIGURE Command

Issue the configure command to enable features like the programmable FIFO and set the begining track for precompensation. A CONFIGURE Command need not be issued if the default values of the FDC meets the system requirements.

**CONFIGURE DEFAULT VALUES:**

| **EIS** | **No Implied Seeks** |
|---|---|
| EFIFO | FIFO Disabled |
| POLL | Polling Enabled |
| FIFOTHR | FIFO Threshold Set to 1 Byte |
| PRETRK | Pre-Compensation Set to Track 0 |

EIS—Enable Implied Seek. When EIS = 1, the FDC will perform a SEEK operation before executing a read/write command. The default value is 0 (no implied seek).

EFIFO—Enable FIFO. When EFIFO = 1, the FIFO is disabled (8272A compatible mode). This means data transfers are asked for on a byte by byte basis. The default value is 1 (FIFO disabled). The threshold defaults to one.

POLL—Disable Polling. When POLL = 1, polling of the drives is disabled. POLL Defaults to 0 (polling enabled). When enabled, a single interrupt is generated after a reset. No polling is performed while the drive head is loaded and the head unload delay has not expired.

FIFOTHR—The FIFO threshold in the execution phase of a read/write command. This is programmable from 1 to 16 bytes. FIFOTHR defaults to one byte. A 00 selects one byte and a 0F selects 16 bytes.

PRETRK—Precompensation start track number. Programmable from track 0 to 255. PRETRK defaults to track 0. A 00h selects track 0 and a FFh selects 255.

#### 8.5.3.9 VERSION Command

The VERSION Command checks to see if the controller is an enhanced type (82077, 82077AA, 82077SL) or the older type (8272A/765A). A value of 90h is returned as the result byte, defining an enhanced FDD controller is in use. No interrupts are generated.

#### 8.5.3.10 RELATIVE SEEK Command

The RELATIVE SEEK Command is coded the same as for the SEEK Command, except for the MSB of the first byte and the DIR# bit.

DIR# Head Step Direction Control

| DIR# | ACTION |
|---|---|
| 0 | Step Head Out |
| 1 | Step Head In |

RCN Relative Cylinder Number that determines how many tracks to step the head in or out from the current track number.

The RELATIVE SEEK Command differs from the SEEK Command in that it steps the head the absolute number of tracks specified in the command instead of making a comparison against an internal register. The SEEK Command is good for drives that support a maximum of 256 tracks. RELATIVE SEEKs cannot be overlapped with other RELATIVE SEEKs. Only one RELATIVE SEEK can be active at a time. Bit 4 of Status Register 0 (EC) will be set to 1 if RELATIVE SEEK attempts to step outward beyond Track 0.

As an example, assume that a floppy drive has 300 useable tracks and that the host needs to read track 300 and the head is on any track (0–255). If a SEEK Command is issued, the head stops at track 255. If a RELATIVE SEEK Command is issued, the FDC moves the head the specified number of tracks, regardless of the internal cylinder position register (but increments the register). If the head had been on track 40 (D), the maximum track that the FDC could position the head on using RELATIVE SEEK, is 296 (D), the initial track, +256 (D). The maximum count that the head can be moved with a single RELATIVE SEEK Command is 256 (D).

The internal register, PCN, would overflow as the cylinder number crossed track 255 and would contain 40 (D). The resulting PCN value is thus (NCN + PCN) mod 256. Functionally, the FDC starts counting from 0 again as the track number goes above 255(D). It is the users responsibility to compensate FDC functions (precompensation track number) when accessing tracks greater than 255. The FDC does not keep track that it is working in an ''extended track area'' (greater than 255). Any command issued uses the current PCN value, except for the RECALIBRATE Command that only looks for the TRACK0 signal. RECALIBRATE returns an error if the head is farther than 79 due to its limitation of issuing a maximum 80 step pulses. The user simply needs to issue a second RECALIBRATE Command. The SEEK Command and implied seeks function correctly within the 44 (D) track (299–255) area of the extended track area. It is the users responsibility not to issue a new track position that exceeds the maximum track that is present in the extended area.

To return to the standard floppy range (0–255) of tracks, a RELATIVE SEEK is issued to cross the track 255 boundary.

A RELATIVE SEEK Command can be used instead of the normal SEEK Command but the host is required to calculate the difference between the current head location and the new (target) head location. This may require the host to issue a READ ID Command to ensure that the head is physically on the track that software assumes it to be. Different FDC commands return different cylinder results which may be difficult to keep track of with software without the READ ID Command.

#### 8.5.3.11 DUMPREG Command

The DUMPREG Command is designed to support system run-time diagnostics and application software development and debug. The command returns pertinent information regarding the status of many of the programmed fields in the FDC. This can be used to verify the values initialized in the FDC.

#### 8.5.3.12 PERPENDICULAR MODE Command

An added capability of the FDC is the ability to interface directly to perpendicular recording floppy drives. Perpendicular recording differs from the traditional longitudinal method by orienting the magnetic bits vertically. This scheme packs in more data bits for the same area.

The PERPENDICULAR MODE Command allows the system designers to designate specific drives as Perpendicular recording drives. Data transfers be-

tween Conventional and Perpendicular drives are allowed without having to issue PERPENDICULAR MODE Commands between the accesses of the two different drives, nor having to change write precompensation values.

With this command, the length of the Gap2 field and VCO enable timing can be altered to accommodate the unique requirements of these drives. Table 36 describes the effects of the WGATE and GAP bits for the PERPENDICULAR MODE Command.

When both GAP and WGATE equal 0 the PERPENDICULAR MODE Command will have the following effect on the FDC:

1. If any of the new bits D0 and D1 are programmed to 1, the corresponding drive is automatically programmed for Perpendicular mode (ie: GAP2 being written during a write operation, the programmed Data Rate will determine the length of GAP2), and data will be written with 0 ns write precompensation.
2. Any of the new bits (DO/D1) are programmed for 0, the designated drive is programmed for Conventional Mode and data will be written with the currently programmed write precompensation value.
3. Bits D0 and D1 can only be over-written when the OW bit is 1. The status of these bits can be determined by interpreting the eighth result byte of the DUMPREG Command. (Note: if either the GAP or WGATE bit is 1, bits D0 and D1 are ignored.)

Software and Hardware reset have the following effects on the enhanced PERPENDICULAR MODE Command:

1. A software reset (Reset via DOR or DSR registers) only sets GAP and WGATE bits to 0; D0 and D1 retain their previously programmed values.
2. A hardware reset (Reset via pin 32) sets all bits (GAP, Wgate, D0, and D1) to 0 (All Drives Conventional Mode).

#### 8.5.3.13 POWERDOWN MODE Command

The POWERDOWN MODE Command allows the automatic power management and enables the enhanced registers (EREG EN) of the FDC. The use of the command can extend the battery life in portable PC applications. To enable auto powerdown the command may be issued during the BIOS power on self test (POST).

This command includes the ability to configure the FDC into the enhanced mode extending the SRB and TDR registers. These extended registers accommodate bits that give more information about floppy drive interface, allow for boot drive selection, and identify the values of the PD and IDLE status.

As soon as the command is enabled, a 10 ms or a 0.5 sec minimum powerup timer is initiated, depending on whether the MIN DLY bit is set to 0 or 1. This timer is one of the required conditions that has to be satisfied before the FDC will enter auto powerdown.

**Table 36. Effects of WGATE and GAP Bits**

| GAP | WGATE | MODE | VCO Low Time after Index Pulse | Length of Gap2 Format Field | Portion of Gap2 Written by Write Data Operation | Gap2 VCO Low Time for Read Operations |
|---|---|---|---|---|---|---|
| 0 | 0 | Conventional Mode | 33 Bytes | 22 Bytes | 0 Bytes | 24 Bytes |
| 0 | 1 | Perpendicular Mode (500 Kbps and Lower Data Rates) | 33 Bytes | 22 Bytes | 19 Bytes | 24 Bytes |
| 1 | 0 | Reserved (Conventional) | 33 Bytes | 22 Bytes | 0 Bytes | 24 Bytes |
| 1 | 1 | Perpendicular Mode (1 Mbps Data Rate) | 18 Bytes | 41 Bytes | 38 Bytes | 43 Bytes |

**NOTE:**
When either GAP or WGATE bit is set, the current value of precompensation in the DSR is used.

Any software reset will re-initialize the timer. The timer countdown is also extended by up to 10 ms if the data rate is changed during the timer's countdown. Without this timer, the FDC would have been put to sleep immediately after FDC is idle. The minimum delay gives software a chance to interact with the FDC without incurring an additional overhead due to recovery time.

The command also allows the output pins of the floppy disk drive interface to be tri-stated or left unaltered during auto powerdown. This is done by the FDI TRI bit. In the default condition (FDI TRI = 0) the output pins of the floppy disk drive are tri-stated. Setting this bit leaves the interface unchanged from the normal state.

The results phase returns the values programmed for MIN DLY, FDI TRI and AUTO PD. The auto powerdown mode is disabled by a hardware reset. Software results have no effect on the POWERDOWN MODE Command parameters.

#### 8.5.3.14 PART ID Command

This command can be used to identify the floppy disk controller as an enhanced controller. The first stepping of the FDC (all versions) will yield 0x02 in the result phase of this command. Any future enhancements on these parts will be denoted by the 5 LSBs (0x01 to 0x1F).

#### 8.5.3.15 OPTION Command

The standard IBM format includes an index address field consisting of 80 bytes of GAP 4a, 12 bytes of the sync field, four bytes identifying the IAM and 50 bytes of GAP 1. Under the ISO format most of this preamble is not used. The ISO format allows only 32 bytes of GAP 1 after the index mark. The ISO bit in this command allows the FDC to configure the data transfer commands to recognize this format. The MSBs in this command are reserved for any other enhancements made available to the user in the future.

#### 8.5.3.16 SAVE Command

The first byte corresponds to the values programmed in the DSR with the exception of CLKSEL. The DRATE1, DRATE0 used here are unmapped. The second byte is used for configuring the bits from the OPTION Command. All future enhancements to the OPTION Command will be reflected in this byte as well. The next nine result bytes are explained in the Parameter Abbreviations section after the command summary. The 13th byte is the value associated with the POWERDOWN MODE Command. The disk status is used internally by the FDC. There are two reserved bytes at the end of this command for future use.

This command is similar to the DUMPREG Command but it additionally allows the user to read back the precompensation values as well as the programmed data rate. It also allows the user to read the values programmed in the POWERDOWN MODE Command. The precompensation values will be returned as programmed in the DSR register. This command, used in conjunction with the RESTORE Command, should prove very useful for SMM power management. This command reserves the last two bytes for future enhancements.

#### 8.5.3.17 RESTORE Command

Using the RESTORE Command with the SAVE Command, allows the SMM power management to restore the FDC to its original state after a system powerdown. It also serves as a succinct way to provide most of the initialization requirements normally handled by the system. The sequence of initializing the FDC after a reset occurred and assuming a SAVE Command was issued follows:

- Issue the DRIVE SPECIFICATION Command (if the design utilizes this command)
- Issue the RESTORE Command (pass the 16 bytes retrieved previously during SAVE)

The RESTORE Command programs the data rate and precompensation value via the DSR. It then restores the values normally programmed through the CONFIGURE, SPECIFY, and PERPENDICULAR Commands. It also enables the previously selected values for the POWERDOWN Mode Command. The PCN values are set restored to their previous values and the user is responsible for issuing the SEEK and RECALIBRATE Commands to restore the head to the proper location. There are some drives that do not recalibrate in which case the RESTORE Command restores the previous state completely. The PDOSC bit is retrievable using the SAVE Command, however, the system designer must set it correctly. The software must allow at least 20 $\mu$s to execute the RESTORE Command. When using the BOOTSEL bits in the TDR, the user must restore or reinitialize these bits to their proper values.

#### 8.5.3.18 FORMAT AND WRITE Command

The FORMAT AND WRITE Command is capable of simultaneously formatting and writing data to the diskette. It is essentially the same as the normal FORMAT Command. With the exception that included in the execution for each sector is not only the C, H, R, and N but also the data transfer of N bytes. The D value is ignored. This command formats the entire track. High speed floppy diskette duplication can be done fast and efficiently with this command. The user can format the diskette and put data on it in a single pass. This is very useful for software duplication applications by reducing the time required to format and copy diskettes.

## 9.0 IDE INTERFACE

The 82091AA supports the IDE (Integrated Drive Electronics) interface by providing two chip selects, and lower and upper data byte controls. DMA and 16-bit data transfers are supported. Minimal external logic is required to complete the optional 16-bit IDE I/O and DMA interfaces. With external logic, a fully buffered interface is also supported.

## 9.1 IDE Registers

The 82091AA does not contain IDE registers. All of the IDE device registers are located in the IDE device, except bit 7 of the Drive Address Register which is the Floppy Controller Disk Change status bit and is driven by the 82091AA.

The IDE interface contains two chip selects (IDECS0# and IDECS1#). These signals are asserted for accesses to the Command and Control Block registers located at 01Fxh and 03Fxh, respectively (Table 37).

**Table 37. IDE Register Set (Located in IDE Device)**

| Primary Address | Secondary Address | Chip Select | Registers | Access |
|---|---|---|---|---|
| 1F0h | 170h | IDECS0# | Data Register | R/W |
| 1F1h | 171h | IDECS0# | Error Register | RO |
| 1F1h | 171h | IDECS0# | Write Precomp/Features Register | WO |
| 1F2h | 172h | IDECS0# | Sector Count Register | R/W |
| 1F3h | 173h | IDECS0# | Sector Number Register | R/W |
| 1F4h | 174h | IDECS0# | Cylinder Low Register | R/W |
| 1F5h | 175h | IDECS0# | Cylinder High Register | R/W |
| 1F6h | 176h | IDECS0# | Drive/Head Register | R/W |
| 1F7h | 177h | IDECS0# | Status Register | RO |
| 1F7h | 177h | IDECS0# | Command Register | WO |
| 3F6h | 376h | IDECS1# | Alternate Status Register | RO |
| 3F6h | 376h | IDECS1# | Digital Output Register | WO |
| 3F7h | 377h | IDECS1# | Drive Address Register | RO |
| 3F7h | 377h | IDECS1# | Not Used | |

## 9.2 IDE Interface Operation

The 82091AA implements the chip select signals for the IDE interface and decodes the standard PC/AT primary and secondary I/O locations.

The 82091AA provides a data buffer enable signal (DEN#) to control the lower data byte path for buffered designs. Buffering the lower data byte path is an application option that requires an external transceiver/buffer. For buffered applications, DEN# controls an external transceiver and enables data bits IDED[7:0] onto the system data bus SD[7:0]. For non-buffered applications (typically the X-Bus configuration), IDED[7:0] are connected directly to the bus and DEN# is not used and becomes a no-connect. For 16-bit applications the upper data byte path (IDED[15:8]) is controlled by the HEN# signal.

Figure 65 shows an example IDE interface without DMA capability. In this case all IDE accesses for setting up the IDE registers and transferring data is programmed via I/O. The 82091AA generates the chip selects (IDECS0# and IDECS1#). The 82091AA also generates the DEN# and HEN# signals to enable the data buffers.

Figure 66 shows an example DMA IDE interface for type ''F'' DMA cycles. To set up the IDE interface, the host accesses the IDE registers on the IDE device. For programmed I/O accesses, the 82091AA generates the chip selects (IDECS0# and IDECS1#) to access the IDE registers and the DEN# and HEN# signals to control the data buffers. During DMA transfers the DMA handshake is between the DMA controller and IDE device via the DREQ and DACK# signals. The DACK# signal is ORed with the DEN# and HEN# signals to control the upper and lower byte buffers during DMA transfers.

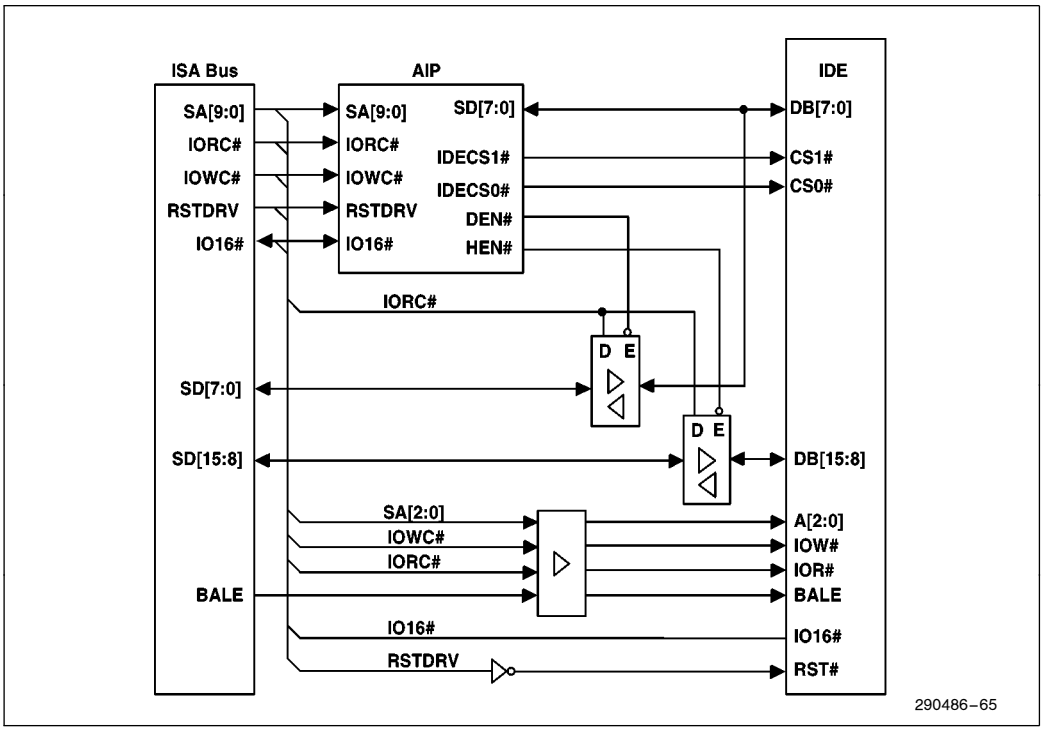



**Figure 65. IDE Interface Example (without DMA)**

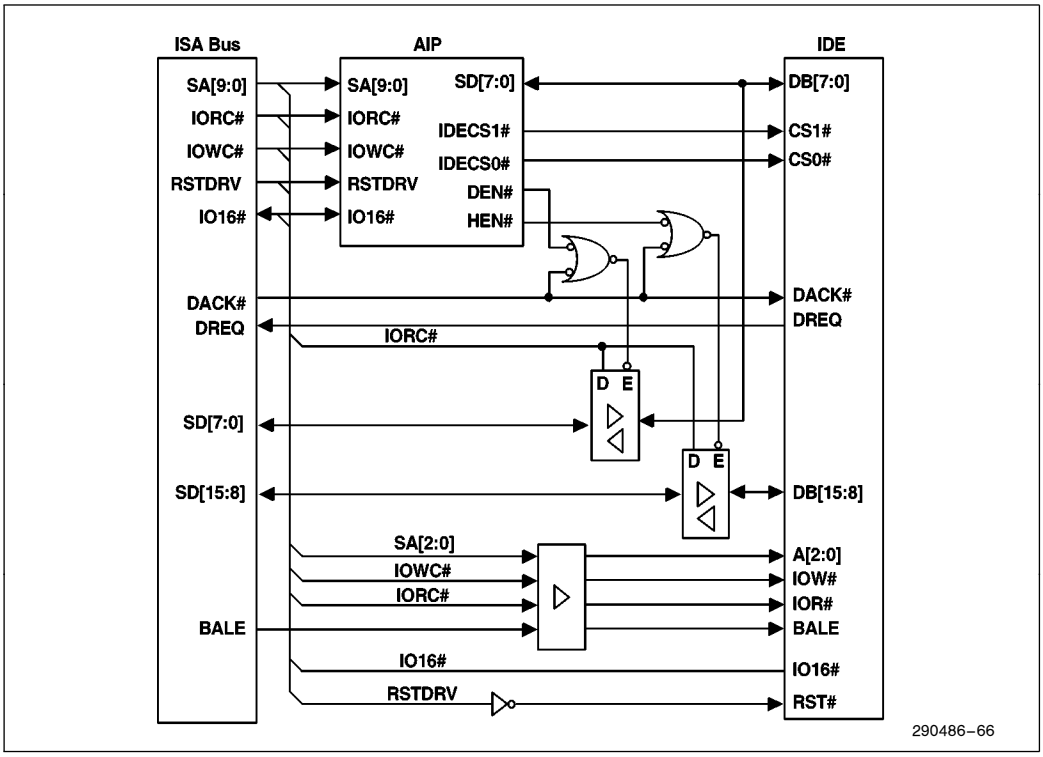



**Figure 66. IDE Interface Example (with DMA)**

# 10.0 POWER MANAGEMENT

The 82091AA provides power management capabilities for its primary functional modules (parallel port, floppy disk controller, serial port A, and serial port B). For each module, the 82091AA implements two types of power management—direct powerdown and auto powerdown. Direct powerdown, enabled via control bits in the 82091AA configuration registers, immediately places the module in a powerdown mode by turning off the clock to the associated module. Direct powerdown removes the clock regardless of the activity or status of the module. By contrast, when auto powerdown is enabled (via control bits in the 82091AA configuration registers), the associated module only enters a powerdown mode if it is in an idle state.

**NOTE:**
The entire 82091AA can be placed in direct powerdown by writing to the CLKOFF bit in the AIPCFG1 Register.

## 10.1 Power Management Registers

The floppy disk controller, parallel port, serial port A, and serial port B each have two 82091AA configuration registers. For each module, three configuration register bits control power management—xDPDN, xIDLE, and xAPDN.

- xAPDN: auto-powerdown, shuts off the oscillator to the module when the module is idle.
- xIDLE: idle status, a read only pin that indicates idle status.
- xDPDN: direct powerdown, shuts off module oscillator when active regardless of module status.

The 82091AA exits any powerdown mode after a hardware reset (RSTDRV asserted) or reset via the xRESET bit in the 82091AA configuration registers. Direct powerdown can also be exited by writing the corresponding xPDN bit in the configuration register to 0. Auto powerdown is exited by events at the module (e.g., CPU read/write or module interface activity).

**NOTE:**
The configuration registers also contain the xEN bit. This bit is used to completely disable an unused module. Enabling a disabled module takes much longer than restoring a module from powerdown. Therefore, this bit is not recommend for temporarily disabling a module as a powerdown scheme.

## 10.2 Clock Power Management

The internal clock circuitry of the 82091AA can be turned on or off as part of a power management scheme. The clock circuitry is controlled via the CLKOFF bit in the AIPCFG1 Register. If an external clock source exists, the user may want to turn off the internal oscillator to save power and provide minimum recovery time.

Auto powerdown and direct powerdown (in each module) have no effect on the state of internal oscillator.

## 10.3 FDC Power Management

This section describes the FDC direct and auto powerdown modes and recovery from the powerdown modes.

### Auto Powerdown

Automatic powerdown (APDN) has an advantage over direct powerdown (PDN) since the register contents are not lost under APDN. Automatic powerdown is invoked by either the Auto Powerdown command, or by enabling the FAPDN bit in the FDC configuration register. There are four conditions required before the FDC will enter powerdown:

1. The motor enable pins ME[3:0] must be inactive.
2. The FDC must be in an idle state. FDC idle is indicated by MSR = 80h and the IRQ6 signal is negated (IRQ6 may be asserted even if MSR = 80h due to polling interrupt).
3. The head unload timer (HUT, explained in the SPECIFY Command) must have expired.
4. The auto powerdown timer must have timed out.

An internal timer is initiated when the POWERDOWN MODE Command is executed. The amount of time can be set by the user via the MIN DLY bits in the POWERDOWN MODE Command. The module is then powered down, provided all the remaining conditions are met. A software reset reinitializes the timer. When using the FDC FAPDN bit to enable the automatic powerdown feature, the MIN DLY bit is set to the default condition.

### Recovery from Auto Powerdown

When the FDC is in auto powerdown, the module is awakened by a reset or access to the DOR, MSR or FIFO registers. The module remains in auto powerdown mode after a software reset (i.e., it will power-

down again after being idle for the time specified by MIN DLY). However, the FDC does not remain in auto powerdown mode after a hardware reset or DSR reset.

**Direct Powerdown**

Direct powerdown is invoked via the Powerdown bit in the Data Rate Select Register (bit 6), or the FDPDN bit in the FCFG2 Register. Setting FDPDN to 1 will powerdown the FDC. All status is lost when this type of powerdown mode is used. The FDC exits powerdown mode after any hardware or software reset. Direct powerdown overrides automatic powerdown.

**Recovery from Direct Powerdown**

The FDC exits the direct powerdown state by setting the FDPDN bit to 0 followed by a software or hardware reset.

After reset, the FDC goes through a normal sequence. The drive status is initialized. The FIFO mode is set to default mode on a hardware or software reset if the LOCK Command has not blocked it. Finally, after a delay, the polling interrupt is issued.

## 10.4 Serial Port Power Management

This section describes the serial port direct and auto powerdown modes and recovery from the powerdown modes.

**Auto Powerdown**

When auto powerdown is enabled in the SxCFG2 Register (SxAPDN bit is 1), the serial port enters auto powerdown based on monitoring line interface activity. During auto powerdown, the status of the serial port is maintained (the FIFO and registers are not reset). Access to any serial port register is allowed during auto powerdown. The transmitter and the receiver enter powerdown individually, depending on certain conditions. When there are no characters to transmit (TEMPTY = 1 in the LSR), the transmitter clock is shut off placing the transmitter in auto powerdown. In the case of the receiver, when serial input signal is inactive for approximately 5 character times, indicating that no character is being received, the receiver goes into auto powerdown.

**Recovery from Auto Powerdown**

The serial port recovers from auto powerdown when either the transmitter or receiver are active. If data is written to the transmitter or data is present at the receiver, the serial port exits from auto powerdown.

**Direct Powerdown**

Direct Powerdown is invoked via the SxCFG2 Register (setting the SxDPDN bit to 1). When in direct powerdown, the clock to the module is shut off. All registers are accessible while in direct powerdown. A host read of the Receiver Buffer Register or a write to the Transmitter Holding Register should not be performed during powerdown. The SINx input should remain static.

When direct powerdown is invoked, the transmit and receive sections of the serial port are reset, including the transmit and receive FIFOs. Thus, to prevent possible data loss when the FIFOs are reset, software should not invoke direct powerdown until the serial port is in the idle state as indicated by the SxIDLE bit in the SxCFG2 Register.

**Recovery from Direct Powerdown**

Recovery from direct powerdown is accomplished by writing the SxDPDN bit in the configuration register to 0 or by a module reset.

## 10.5 Parallel Port Power Management

**Auto Powerdown**

Auto powerdown is enabled via the PAPDN bit in the PCFG2 Register. When enabled, the parallel port enters auto powerdown when the module is in an idle state. If the parallel port FIFO is being used to transfer data, the parallel port is in an idle state when the FIFO is empty.

**Recovery from Auto Powerdown**

Recovery from auto powerdown occurs when the FIFO is written or as a result of parallel port interface activity.

**Direct Powerdown**

Direct powerdown is invoked via the PCFG2 Register (setting the PDPDN bit to 1). When PDPDN = 1, the clock to the printer state machine is disabled and the state machine goes into an idle state.

**Recovery from Direct Powerdown**

Recovery from direct powerdown is accomplished by setting the PDPDN bit to 0 or the PRESET bit to a 1 in the PCFG2 Register. An 82091AA hard reset (RSTDRV asserted) also brings the part out of direct powerdown.

# 11.0 ELECTRICAL CHARACTERISTICS

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

## 11.1 Absolute Maximum Ratings

Storage Temperature . . . . . −65°C to +150°C
Supply Voltage . . . . . −0.5V to +8.0V
Voltage on Any Input . . . . . GND−2V to 6.5V
Voltage on Any Output . . . GND−0.5V to $V_{CC}$ +0.5V
Power Dissipation . . . . . 1W

**WARNING: Stressing the device beyond the ''Absolute Maximum Ratings'' may cause permanent damage. These are stress ratings only. Operation beyond the ''Operating Conditions'' is not recommended and extended exposure beyond the ''Operating Conditions'' may affect device reliability.*

## 11.2 DC Characteristics

**Table 38. DC Specifications** ($V_{CC}$ = 5V ±10%, $T_{amb}$ = 0°C to 70°C)

| Symbol | Parameter | $V_{CC}$ = +5V ±10 | | | $V_{CC}$ = 3.3V ±0.3V | | |
|---|---|---|---|---|---|---|---|
| | | Min(V) | Max(V) | Notes | Min(V) | Max(V) | Notes |
| $V_{ILC}$ | Input Low Voltage, X1 | −0.5 | 0.8 | | −0.3 | 0.8 | |
| $V_{IHC}$ | Input High Voltage, X1 | 3.9 | $V_{CC}$ + 0.5 | | 2.4 | $V_{CC}$ + 0.3 | |
| $V_{IL}$ | Input Low Voltage (all pins except X1) | −0.5 | 0.8 | | −0.3 | 0.8 | |
| $V_{IH}$ | Input High Voltage (all pins except X1) | 2.0 | $V_{CC}$ + 0.5 | | 2.0 | $V_{CC}$ + 0.3 | |
| $I_{CC}$ | $V_{CC}$ Supply Current −1 Mbps FDC Data Rate $V_{IL}$ = 0.45V, $V_{IH}$ = 2.4V | | 50 mA | 1, 2 | | 40 mA | 1, 2 |
| $I_{CCSB}$ | $I_{CC}$ in Powerdown | | 100 μA | 3, 4, 5 | | 100 μA | 3, 4, 5 |
| $I_{IL}$ | Input Load Current (all input pins) | | +10μA −10 μA | 6 | | +10 μA −10 μA | 6 |
| $I_{OFL}$ | Data Bus Output Float Leakage | | +10 μA −10 μA | 7 | | +10 μA −10 μA | 8 |
| $I_{BPL}$ | Parallel Port Back-Power Leakage (All Parallel Port Signals) | | +10 μA | 9 | | +10 μA | 9 |

**NOTES:**
1. Test Conditions: Only the data bus inputs may float. All outputs are open.
2. Test Conditions: Tested while reading a sync field of ''00''. Outputs not connected to DC loads. This specification reflects the supply current when all modules within the 82091AA are active.
3. Test Conditions: $V_{IL} = V_{SS}$, $V_{IH} = V_{CC}$; Outputs not connected to DC loads.
4. Test Conditions: Typical value with the oscillator off.
5. Test Conditions: All 82091AA modules are in their powerdown state.
6. Test Conditions: 10 μA ($V_{IN} = V_{CC}$), −10μA ($V_{IN}$ = 0V)
7. Test Conditions: $0V < V_{OH} < V_{CC}$
8. Test Conditions: $0.45V < V_{OH} < V_{CC}$
9. Test Conditions: Device in Circuit $V_{CC}$ = 0V, $V_{IN}$ = 5.5V max.

**Table 39. Capacitance Specifications** ($V_{CC}$ = 5V ±10%, $T_{amb}$ = 0°C to 70°C)

| | | | | |
|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | 10 | pF | F = 1 MHz, $T_A$ = 25°C |
| $C_{IN1}$ | Clock Input Capacitance | 20 | pF | Sampled, not 100% Tested |
| $C_{I/O}$ | Input/Output Capacitance | 20 | pF | |

**NOTE:**
All pins except pins under test are tied to AC ground.

The following pin groupings are used in Table 40 and Table 41.

**DMA** FDDREQ, PPDREQ

**IRQx** IRQ3, IRQ4, IRQ5, IRQ6, IRQ7

**Serial Port** SOUTA, SOUTB, DTRA#, DTRB#, RTSA#, RTSB#

**Parallel Port** PD[7:0], STROBE#, AUTOFD#, INIT#, SELECTIN#

**FDC Interface** WRDATA, HDSEL#, STEP#, DIR#, WE#, FDME0#, FDME1#, FDS0#, FDS1#, DRVDEN[1:0]

**Table 40. $V_{OL}$ Specifications** ($V_{CC}$ = 5V ±10%, $T_{amb}$ = 0°C to 70°C)

| Symbol | Signal | $V_{CC}$ = 5V ± 10% | | | $V_{CC}$ 3.3V ± 0.3V | | |
|---|---|---|---|---|---|---|---|
| | | Min | Max | $I_{OL}$ | Min | Max | $I_{OL}$ |
| $V_{OL}$ | SD[7:0] | | 0.45V | 24 mA | | 0.45V | 12 mA |
| $V_{OL}$ | NOWS#, IOCHRDY | | 0.45V | 24 mA | | 0.45V | 12 mA |
| $V_{OL}$ | DMA,IRQx | | 0.45V | 12 mA | | 0.45V | 6 mA |
| $V_{OL}$ | Serial Port | | 0.45V | 4 mA | | 0.45V | 2 mA |
| $V_{OL}$ | Parallel Port | | 0.45V | 16 mA | | 0.45V | 8 mA |
| $V_{OL}$ | PPDIR,GCS# | | 0.45V | 4 mA | | 0.45V | 2 mA |
| $V_{OL}$ | FDC Interface | | 0.45V | 12 mA | | 0.45V | 6 mA |
| $V_{OL}$ | DEN#,HEN# | | 0.45V | 4 mA | | 0.45V | 2 mA |
| $V_{OL}$ | IDECS[1:0]# | | 0.45V | 12 mA | | 0.45V | 6 mA |

**Table 41. $V_{OH}$ Specifications** ($V_{CC}$ = 5V ±10%, $T_{amb}$ = 0°C to 70°C)

| Symbol | Signal | $V_{CC}$ = 5V ± 10% | | | $V_{CC}$ 3.3V ± 0.3V | | |
|---|---|---|---|---|---|---|---|
| | | Min | Max | $I_{OH}$ | Min | Max | $I_{OH}$ |
| $V_{OH}$ | SD[7:0] | 2.4V | | 4 mA | 2.4V | | 2 mA |
| $V_{OH}$ | DMA,IRQx | 2.4V | | 4 mA | 2.4V | | 2 mA |
| $V_{OH}$ | Serial Port | 2.4V | | 1 mA | 2.4V | | 1 mA |
| $V_{OH}$ | Parallel Port | 2.4V | | 4 mA | 2.4V | | 50 μA |
| $V_{OH}$ | PPDIR,GCS# | 2.4V | | 1 mA | 2.4V | | 1 mA |
| $V_{OH}$ | FDC Interface | 2.4V | | 4 mA | 2.4V | | 2 mA |
| $V_{OH}$ | DEN#, HEN# | 2.4V | | 1 mA | 2.4V | | 1 mA |
| $V_{OH}$ | IDECS[1:0]# | 2.4V | | 4 mA | 2.4V | | 2 mA |

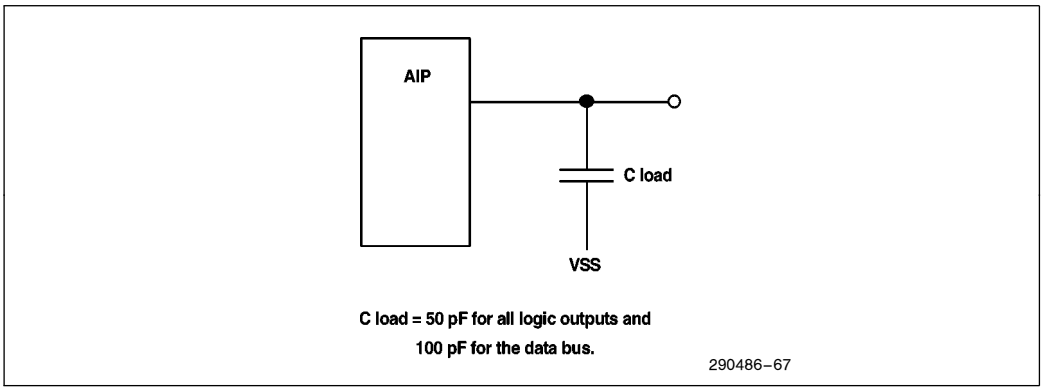



**Figure 67. Load Circuit**

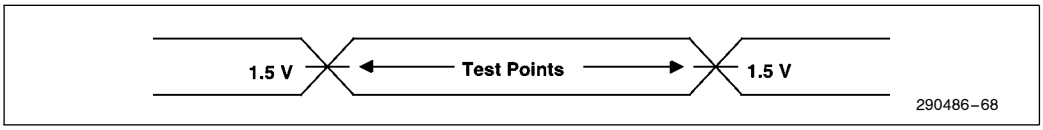



**Figure 68. AC Testing Input, Output**

## 11.3 Oscillator

The 24 MHz clock can be supplied either by a crystal (Figure 69) or a MOS level square wave. All internal timings are referenced to this clock or a scaled count that is data rate dependent. The crystal oscillator must be allowed to run for 10 ms after $V_{CC}$ has reached 4.5V or exiting the POWERDOWN mode to guarantee that it is stable.

Crystal Specifications:

| | |
|---|---|
| Freq: | 24 MHz ± 0.1% |
| Mode: | Parallel Resonant |
| | Fundamental Mode |
| Series Resistance: | < 40Ω |
| Shunt Capacitance: | < 5 pF |
| C1, C2: | 20 pF–25 pF |

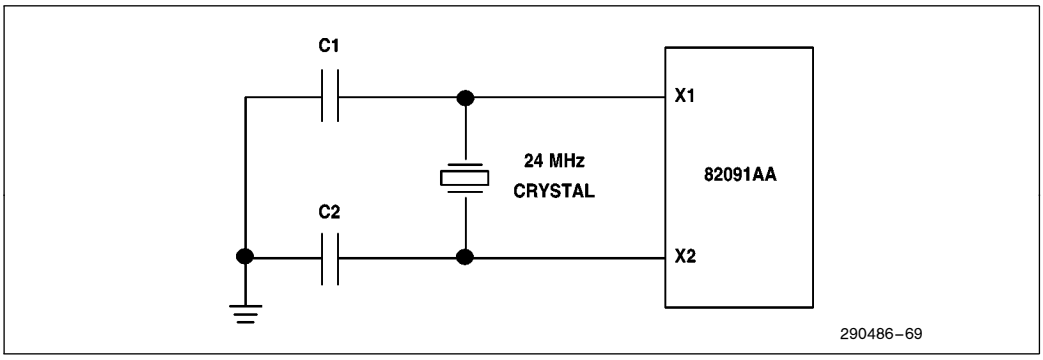



**Figure 69. Crystal Connections**

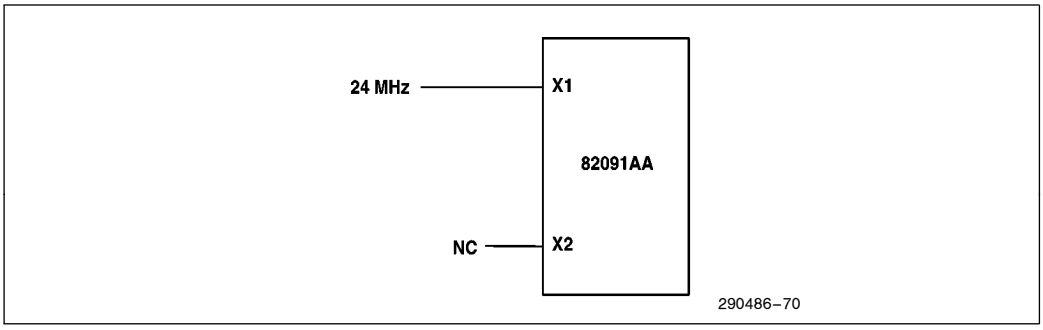



**Figure 70. Oscillator Connections**

## 11.4 AC Characteristics

**Table 42. AC Specifications** ($V_{CC}$ = 5V ± 10%, $T_{amb}$ = 0°C to 70°C)

| Symbol | Parameter | 24 MHz | | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| | | Min | Max | | | |
| t1a | Clock Rise and Fall Time | | 10 | ns | 1 | 71 |
| t1b | Clock High Time | 16 | | ns | 1 | 71 |
| t1c | Clock Low Time | 16 | | ns | 1 | 71 |
| t1d | Clock Period | 41.66 | 41.66 | ns | 2 | 71 |
| t1e | Internal Clock Period | | | | 3 | |

**NOTES:**
1. Clock input high level test points for clock high time and clock rise/fall times are 3.5V with $V_{CC}$ at 5V ±10% and 2.0V with $V_{CC}$ at 3.3V $V_{CC}$ ±10%. Clock input low level test point for clock low time and clock rise/fall time is 0.8V.
2. Clock input test point for clock period is 0.8V.
3. Certain Floppy Disk Controller module timings are a function of the selected data rate. The nominal values for the internal clock period (t1e) for the various data rates are:

| Disk Drive Disk Rate | Internal Clock Period (*nominal values) |
|---|---|
| | 24 MHz |
| 1 Mbps | 125 ns |
| 500 Kbps | 250 ns |
| 300 Kbps | 420 ns |
| 250 Kbps | 500 ns |

All information contained in ( ) in the following tables represents 3.3V specifications.

**Table 43. AC Specifications** ($V_{CC}$ = 5V ± 10%, or [3.3V ± 0.3V] $T_{amb}$ = 0°C to 70°C)

| Symbol | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| **Host** | | | | | | |
| **SA[10:0]** | | | | | | |
| t2a | SA[10:0] Setup to IORC#/IOWC# Active | 18 (25) | | ns | | 72, 73 |
| t2b | SA[10:0] Hold from IORC#/IOWC# Inactive | 0 | | ns | | 72, 73 |
| **SD[7:0]** | | | | | | |
| t3a | SD[7:0] Valid Delay from IORC# Active | | 70 (100) | ns | 1 | 72 |
| t3b | SD[7:0] Float Delay from IORC# Inactive | 5 | 35 (40) | ns | | 72 |
| t3c | SD[7:0] Setup to IOWC# Inactive | 35 | | ns | | 73 |
| t3d | SD[7:0] Hold from IOWC# Inactive | 0 | | ns | | 73 |

**Table 43. AC Specifications** ($V_{CC}$ = 5V ± 10%, or (3.3V ± 0.3V) $T_{amb}$ = 0°C to 70°C) (Continued)

| Symbol | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| **IOCHRDY** | | | | | | |
| t4a | IOCHRDY Propagation Delay from IORC# / IOWC# Active | | 55 (75) | ns | EPP | 82, 83 |
| t4b | IOCHRDY Propagation Delay from BUSY | | 34 (65) | ns | EPP | 82, 83 |
| **IORC#** | | | | | | |
| t5a | IORC# Active Pulse Width | 90 | | ns | | 72 |
| t5b | IORC# Recovery Time | 60 | | ns | | 72 |
| **IOWC#** | | | | | | |
| t6a | IOWC# Active Pulse Width | 90 | | ns | | 73 |
| t6b | IOWC# Recovery Time | 60 | | ns | | 73 |
| **AEN** | | | | | | |
| t7a | AEN Setup to IORC# / IOWC# Active | 18 | | ns | | 72, 73 |
| t7b | AEN Hold from IORC# / IOWC# Inactive | 0 | | ns | | 72, 73 |
| **NOWS#** | | | | | | |
| t8a | NOWS# Delay from IORC# / IOWC# | | 35 (50) | ns | | 72, 73 |
| **TC** | | | | | | |
| t9a | TC Active Pulse Width | 50 | | ns | 6 | 74 |
| **RESET** | | | | | | |
| **RSTDRV** | | | | | | |
| t10a | RSTDRV Active Pulse Width | 0.5 | | µs | | 75 |
| t10b | Hardware Configuration Input Setup to RSTDRV Inactive | 100 | | ns | All Configuration Modes | 76 |
| t10c | Hardware Configuration Input Hold from RSTDRV Inactive | 0 | | | All Configuration Modes | 76 |
| **INTERRUPTS** | | | | | | |
| **RQ[4,3] (Serial Ports)** | | | | | | |
| t11b | IRQ[4,3] Inactive Delay from IORC# / IOWC# Active | | 100 | ns | THR wr, RBR rd, MSR rd | 90, 91 |
| t11c | IRQ[4,3] Inactive Delay from IORC# Inactive | | 100 | ns | IIR rd, LSR rd | 90 |
| t11d | IRQ[4,3] Active Delay from DCD# / DSR# / CTS# / RI# | | 80 | ns | | 91 |

**Table 43. AC Specifications** ($V_{CC}$ = 5V ± 10%, or (3.3V ± 0.3V) $T_{amb}$ = 0°C to 70°C) (Continued)

| Symbol | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| **INTERRUPTS** | | | | | | |
| **IRQ[7,5] (Parallel Port)** | | | | | | |
| t12b | IRQ[7,5] Inactive Delay from IORC# / IOWC# Active | | 70 (90) | ns | ECP rev, fwd to FIFO | 81 |
| t12c | IRQ[7,5] Inactive Delay from IOWC# Inactive | | 70 (95) | ns | ECP fwd to ECR | 81 |
| t12d | IRQ[7,5] Delay from ACK# | | 70 (90) | ns | All Modes | 81 |
| t12e | IRQ[7,5] Delay from FAULT# | | 70 (90) | ns | ECP | 81 |
| **IRQ6 (FDC)** | | | | | | |
| t13b | IRQ6 Inactive Delay from IORC# / IOWC# Active | | t1e + 125 | ns | 2 | 80 |
| **DMA** | | | | | | |
| **FDDREQ, PPDREQ** | | | | | | |
| t14a | xDREQ Inactive Delay from xDACK# Active | | 75 (100) | ns | 4 | 74 |
| t14b | FDREQ Cycle Time (Non-Burst DMA) | 6.25 | | µs | 3 | 74 |
| t14c | xDREQ Active from IORC# / IOWC# Inactive | 100 | | ns | | 74 |
| t14d | xDREQ Setup IORC# / IOWC# | 0 | | ns | 3 | 74 |
| t14e | xDREQ Delay from IORC# / IOWC# Active | | 75 (100) | ns | 5 | 74 |
| t14f | FDREQ Inactive Delay from TC Active<br>PPDREQ Inactive Delay from TC Active | | 110<br>80 (90) | ns | | 74 |
| t14g | xDREQ to xDACK# Inactive | ⅔ t1e | | | | 74 |
| **FDDACK#, PPDACK#** | | | | | | |
| t15a | xDACK# Active Delay from xDREQ Active | 0 | | ns | | 74 |
| t15b | xDACK# Setup to IORC# / IOWC# Active | 18 | | ns | | 74 |
| t15c | xDACK# Hold from IORC# / IOWC# Inactive | 0 | | ns | | 74 |

**Table 43. AC Specifications** ($V_{CC}$ = 5V ± 10%, or (3.3V ± 0.3V) $T_{amb}$ = 0°C to 70°C) (Continued)

| Symbol | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| **PARALLEL PORT** | | | | | | |
| **PD[7:0]** | | | | | | |
| t16a | PD[7:0] Delay from IOWC# Inactive | | 60 (90) | ns | ISA,PS/2 wr | 87 |
| t16b | PD[7:0] Delay from IOWC# Active | | 70 (100) | ns | EPP wr | 82 |
| t16c | PD[7:0] Float Delay from AUTOFD#/SELECTIN# Inactive | 50 | | ns | EPP wr | 82 |
| t16d | PD[7:0] Delay from IORC# Active | | 70 (100) | ns | EPP rd | 83 |
| t16e | PD[7:0] Float Delay from AUTOFD#/SELECTIN# Inactive | 50 | | ns | EPP rd | 83 |
| t16f | PD[7:0] Setup to STROBE# Active | 450 | | | ISA FIFO | 84 |
| t16g | PD[7:0] Hold from STROBE# Inactive | 450 | | | ISA FIFO | 84 |
| t16h | PD[7:0] Hold from BUSY Inactive | 0 | | | ECP fwd | 85 |
| t16i | PD[7:0] Setup to ACK# High | 0 | | | ECP rev | 86 |
| t16j | PD[7:0] Hold from AUTOFD# Low | 0 | | | ECP rev | 86 |
| **STROBE#** | | | | | | |
| t17a | STROBE# Delay from IOWC# Inactive | | 60/ 90 | | ISA, PS/2 | 87 |
| t17b | STROBE# Delay from IORC#/IOWC# Active | | 60/ 90 | | EPP | 82, 83 |
| t17c | STROBE# Active from BUSY Inactive | 500 | | | ISA FIFO | 84 |
| t17d | STROBE Active Pulse Width | 450 | | | ISA FIFO | 84 |
| t17e | STROBE# Active from BUSY Inactive | 0 | | | ECP fwd | 85 |
| t17f | STROBE# Inactive Delay from BUSY Active | 0 | | | ECP fwd | 85 |
| **AUTOFD#** | | | | | | |
| t18a | AUTOFD# Delay from IOWC# Inactive | | 60 (90) | ns | ISA,PS/2 | 82, 87 |
| t18b | AUTOFD# Delay from IORC#/IOWC# Active | | 60 (90) | ns | EPP | 82, 83 |
| t18c | AUTOFD# Hold from BUSY Inactive | 80 | | ns | ECP fwd | 85 |
| t18d | AUTOFD# Low Delay from ACK# Inactive | 0 | | ns | ECP rev | 86 |
| t18e | AUTOFD# High Delay from ACK# Active | 0 | | ns | ECP rev | 86 |
| **INIT#** | | | | | | |
| t19a | INIT# Delay from IOWC# Inactive | | 60 (90) | ns | All Modes | 87 |

**Table 43. AC Specifications** ($V_{CC}$ = 5V ± 10%, or (3.3V ± 0.3V) $T_{amb}$ = 0°C to 70°C) (Continued)

| Symbol | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| **SELECTIN#** | | | | | | |
| t20a | SELECTIN# Delay from IOWC#/IORC# Inactive | | 60 (90) | ns | ISA, PS/2 | 82, 83, 87 |
| t20b | SELECTIN# Delay from IOWC#/IORC# Active | | 60 (90) | | EPP | 82, 83 |
| **BUSY** | | | | | | |
| t21a | BUSY Active Delay from STROBE# Active | | 500 | | ISA, PS/2 | 84 |
| t21b | BUSY Active Delay from STROBE# Active | 0 | | | | 85 |
| t21c | BUSY Inactive Delay from STROBE# Inactive | 0 | | | ECP fwd | 85 |
| t21d | BUSY Setup to ACK# Active | 0 | | | ECP rev | 86 |
| t21f | BUSY Hold from AUTOFD# Inactive | 0 | | | ECP rev | 86 |
| **ACK#** | | | | | | |
| t22a | ACK# Active Hold from AUTOFD# High | 0 | | | ECP rev | 86 |
| t22b | ACK# Inactive Hold from AUTOFD# Low | 0 | | | ECP rev | 86 |
| **PPDIR/GCS#** | | | | | | |
| t23a | GCS# Delay from SA[10:0] | | 60 (90) | | | 89 |
| t23b | PPDIR Delay from IOWC# Inactive | | 60 (90) | | ISA, PS/2, ECP | 87 |
| t23c | PPDIR Delay from IOWC# Active | | 60 (90) | | EPP | 82 |
| **IDE Interface** | | | | | | |
| **IDECS[1:0]#** | | | | | | |
| t24a | IDECSx# Delay from SA[10:0] | | 40 (70) | | | 88 |
| **DEN#** | | | | | | |
| t25a | DEN# Delay from SA[10:0] | | 40 (70) | | | 72, 73, 88 |
| t25b | DEN# Delay from xDACK# | | 40 (70) | | | 74 |
| **HEN#** | | | | | | |
| t26a | HEN# Delay from IO16# | | 35 (65) | | | 88 |

**Table 43. AC Specifications** ($V_{CC}$ = 5V ± 10%, or (3.3V ± 0.3V) $T_{amb}$ = 0°C to 70°C) (Continued)

| Symbol | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| **SERIAL PORTS** | | | | | | |
| **DTRx#, RTSx#, DCDx#** | | | | | | |
| t27a | DTRx#/RTSx#/DCDx# Active Delay from IOWC# | | 55 (70) | ns | MCR wr | 91 |
| **FLOPPY DISK CONTROLLER** | | | | | | |
| **RDDATA#** | | | | | | |
| t28a | Read Data Pulse Width | 50 | | ns | | 95 |
| t28c | PLL Data Rate | | 1M | bits/sec | | na |
| t28d | Lockup Time | | 64 | t28c | | na |
| **WRDATA#** | | | | | | |
| t29a | Data Width | see note | see note | | 7 | 77 |
| **HDSEL#** | | | | | | |
| t30a | WE# to HDSEL# Change | see note | see note | | 10 | 78 |
| **STEP#** | | | | | | |
| t31a | STEP# Active Time | 2.5 | | µs | | 78 |
| t31b | STEP# Cycle Time | see note | see note | µs | 9 | 78 |
| **DIR#** | | | | | | |
| t32a | DIR# Setup to STEP# Active | 1 | | µs | 8 | 78 |
| t32b | DIR# Hold from STEP# Inactive | 10 | | µs | | 78 |
| **WE#** | | | | | | |
| t33a | WE# Inactive Delay from RSTDRV Inactive Edge | | 2 | µs | | 75 |
| **INDEX#** | | | | | | |
| t34a | INDEX# Pulse Width | 5 | t1e | | | 78 |

**NOTES:**
1. The FDC Status Register's status bits which are not latched may be updated during a host read operation.
2. The timing t13b is specified for the FDC interrupt signal in the polling mode only. These timings in case of the result phase of the read and write commands are microcode dependent.
3. This timing is for FDC FIFO threshold = 1. When FIFO threshold is N bytes, the value should be multiplied by N and subtract 1.5 µs. The value shown is for 1 Mbps, scales linearly with data rate.
4. This timing is a function of the internal clock period (t1e) and is given as (⅔) t1e. The values of t1e are shown in Note 3.
5. If DACK# transitions before RD#, then this specification is ignored. If there is no transition on DACK#, then this becomes the DRQ inactive delay.
6. TC width is defined as the time that both TC and DACK# are active. Note that TC and DACK# must overlap at least 50 ns.

**NOTES:** (Continued)
7. Based on the internal clock period (t1e). For various data rates, the read and write data width minimum values are:

| Disk Drive Data Rate | 24 MHz |
|---|---|
| 1 Mbps | 150 ns |
| 500 Kbps | 360 ns |
| 300 Kbps | 615 ns |
| 250 Kbps | 740 ns |

8.This timing is a function of the selected data rate as follows:

| Disk Drive Data Rate | Timing |
|---|---|
| 1 Mbps | 1.0 $\mu$s Min |
| 500 Kbps | 2.0 $\mu$s Min |
| 300 Kbps | 3.3 $\mu$s Min |
| 250 Kbps | 4.0 $\mu$s Min |

9. This value can range from 0.5 ms to 8.0 ms and is dependent upon data rate and the Specify Command value.
10. The minimum MFM values for WE# to HDSEL# change for the various data rates are:

| Disk Drive Data Rate | Min MFM Value |
|---|---|
| 1 Mbps | 0.5 ms + [8 × GPL] |
| 500 Kbps | 1.0 ms + [16 × GPL] |
| 300 Kbps | 1.6 ms + [26.66 × GPL] |
| 250 Kbps | 2.0 ms + [32 × GPL] |

**GPL** is the size of gap 3 defined in the sixth byte of a Write Command.

11. Based on internal clock period.
12. Jitter tolerance is defined as:
    (Maximum bit shift from nominal position ÷ ¼ period of nominal data rate) × 100 percent is a measure of the allowable bit jitter that may be present and still be correctly detected. The data separator jitter tolerance is measured under dynamic conditions that jitters the bit stream according to a reverse precompensation algorithm.
13. The minimum reset active period for a software reset is dependent on the data rate, after the FDC module has been properly reset using the t10a spec. The minimum software reset period then becomes:

| Disk Drive Data Rate | Minimum Software Reset Active Period |
|---|---|
| | 24 MHz |
| 1 Mbps | 125 ns |
| 500 Kbps | 250 ns |
| 300 Kbps | 420 ns |
| 250 Kbps | 500 ns |

### 11.4.1 CLOCK TIMINGS

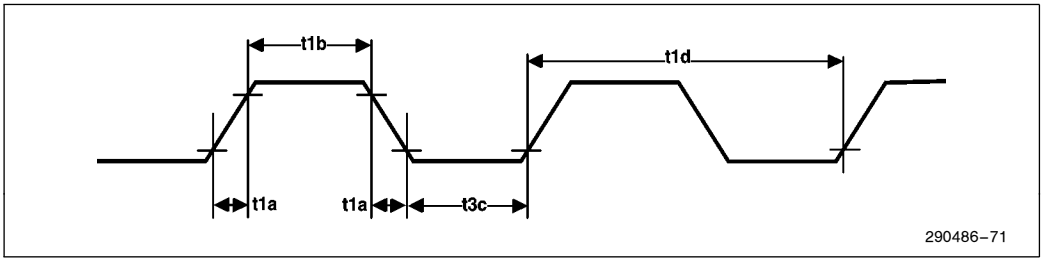



**Figure 71. Clock Timing**

### 11.4.2 HOST TIMINGS

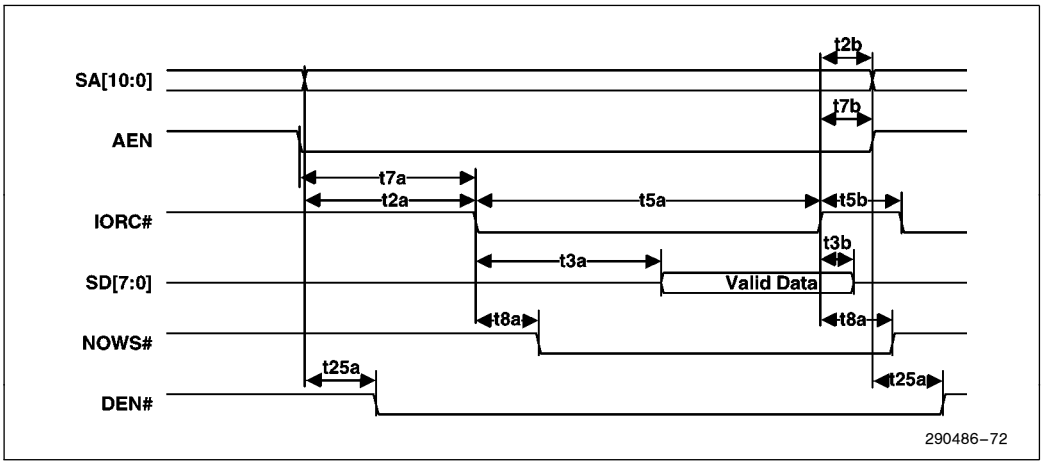



**Figure 72. Host Read**

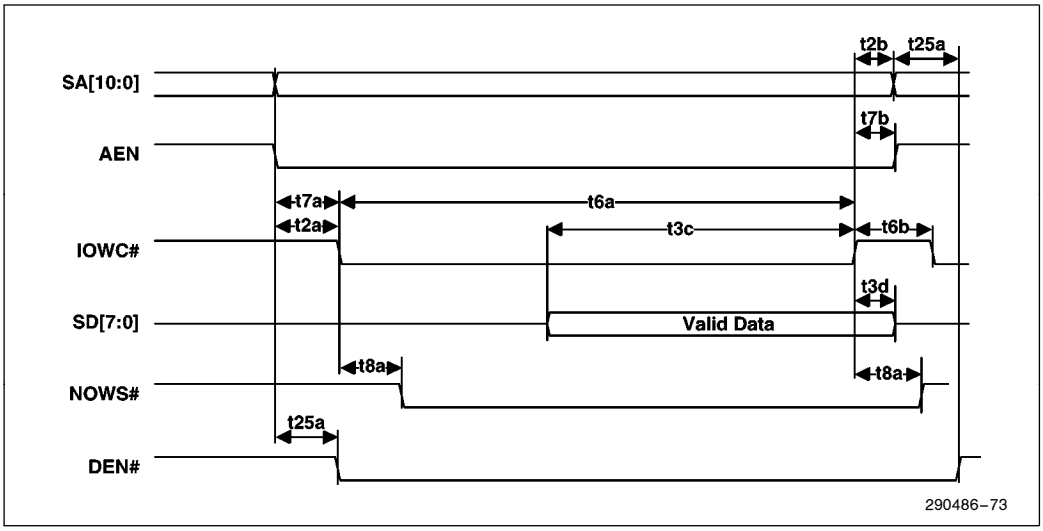



**Figure 73. Host Write**

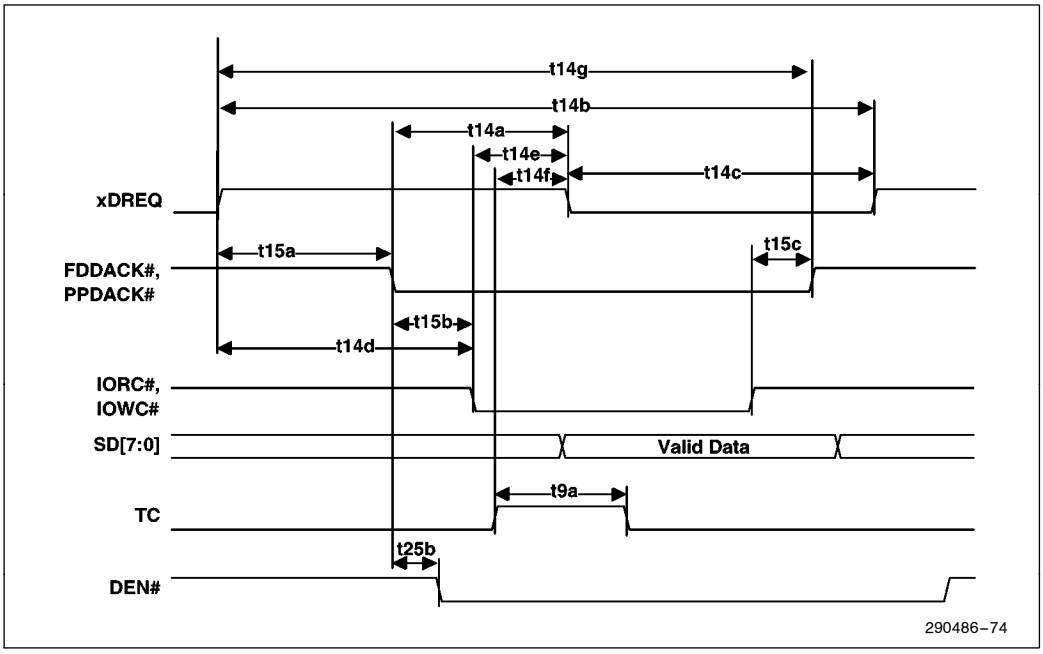
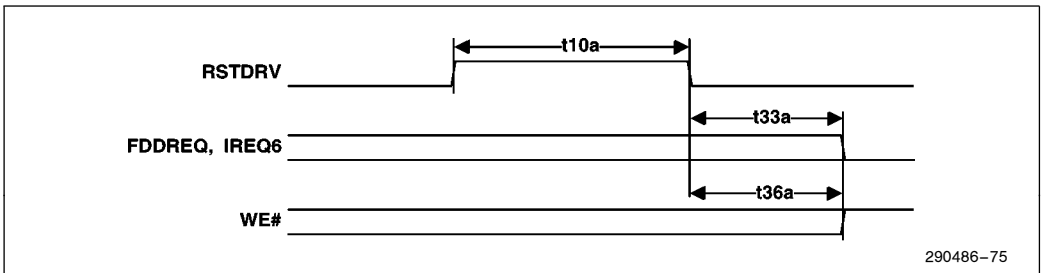



**Figure 74. DMA Timing**




**NOTE:**
FDDREQ, IRQ6 depicts the FDC enabled condition under hardware configuration. Otherwise, these signals tri-state with the same timing as IRQ[7,5,4,3].

**Figure 75. Reset Timing**

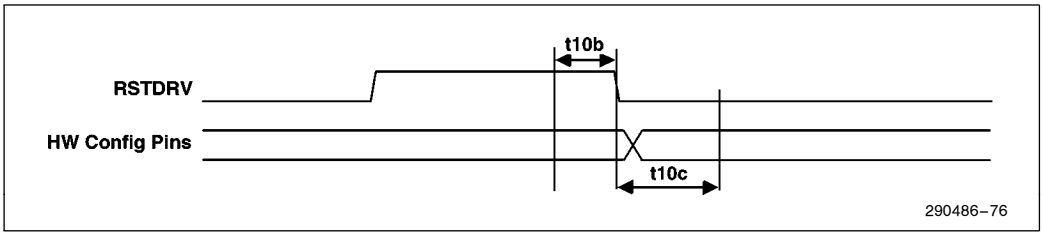



**Figure 76. Reset Timing (Hardware Extended Configuration Mode)**

### 11.4.3 FDC TIMINGS

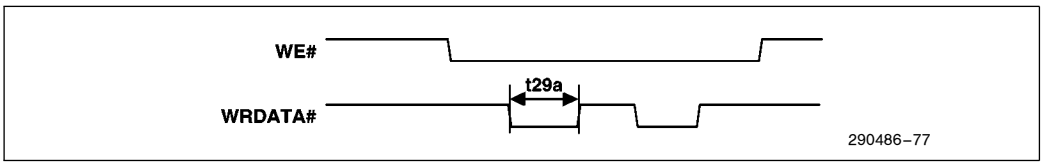
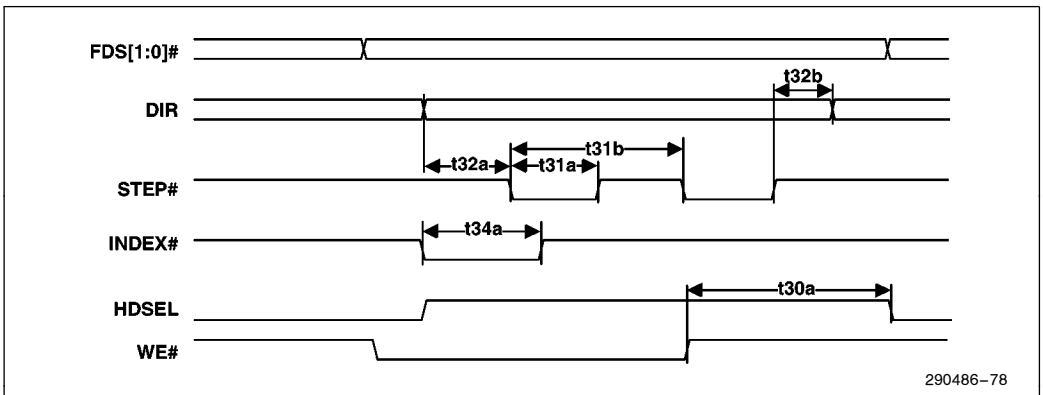



**Figure 77. Write Data Timing**




**NOTE:**
For overlapped seeks, only one step pulse per drive selection is issued. Non-overlapped seeks will issue all programmed step pulses.

**Figure 78. FDC Drive Control/Timing**

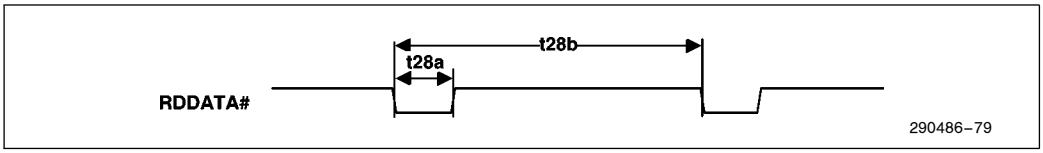



Figure 79. FDC Internal PLL Timing

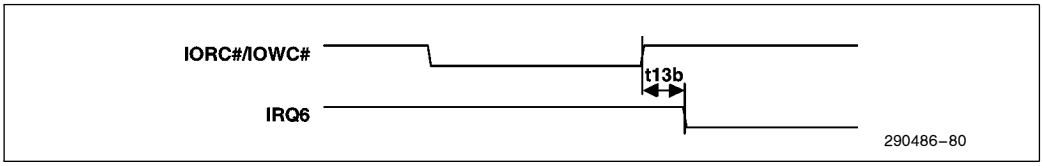



Figure 80. Floppy Disk Controller Interrupts

### 11.4.4 PARALLEL PORT TIMINGS

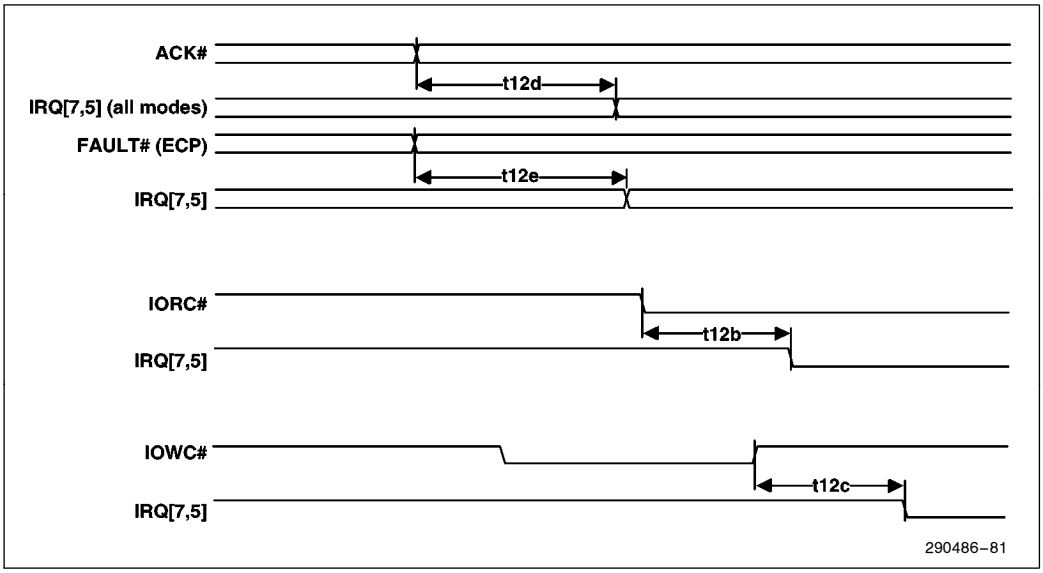



Figure 81. Parallel Port Interrupt Timing

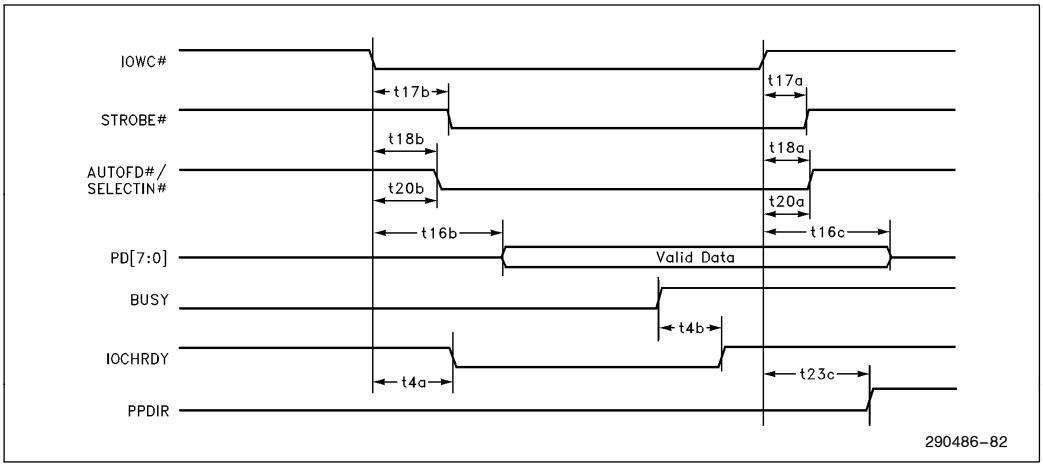



**Figure 82. EPP Write Timing**

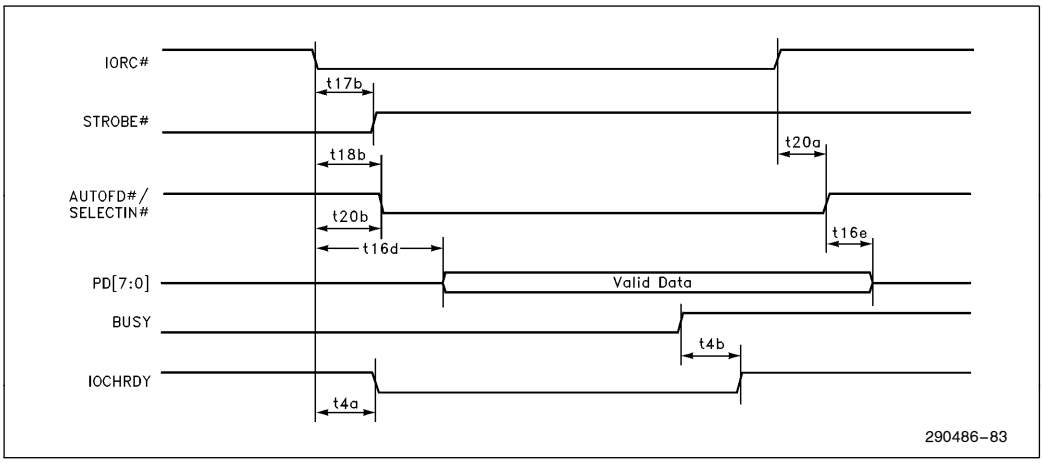



**Figure 83. EPP Read Timing**

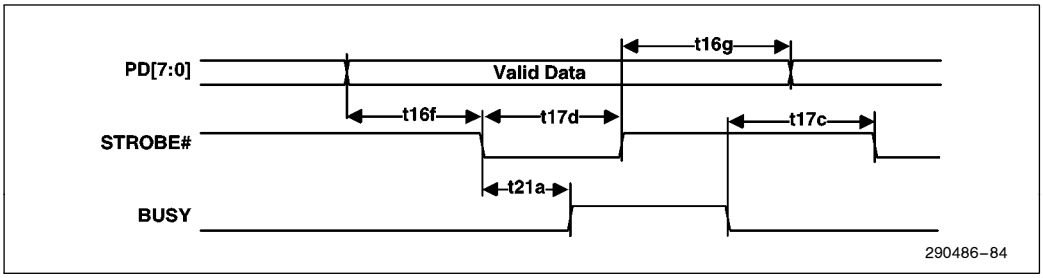



Figure 84. ISA-Compatible FIFO Timing

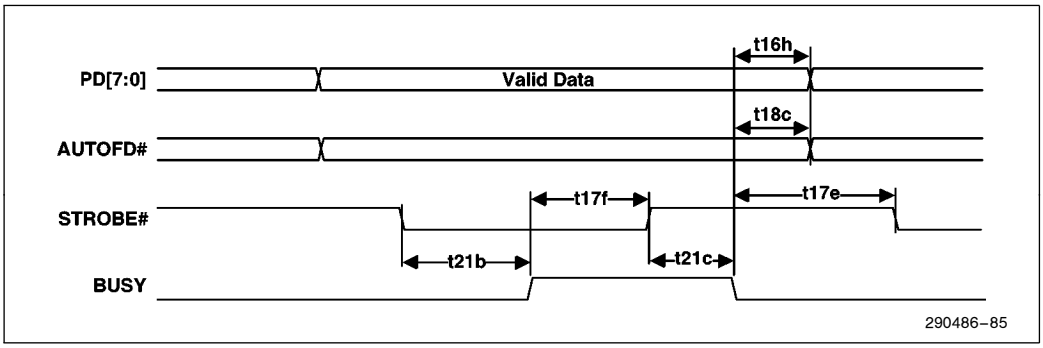



Figure 85. ECP Write Timing (Forward Direction)

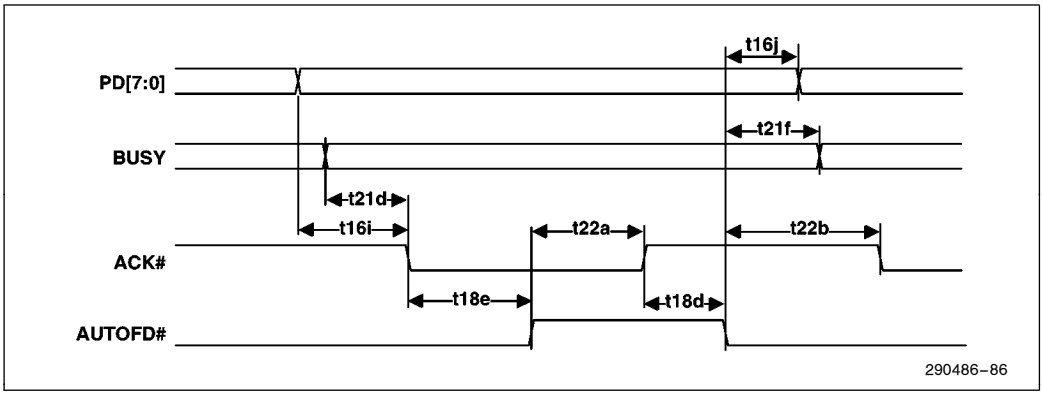



Figure 86. ECP Read Timing (Reverse Direction)

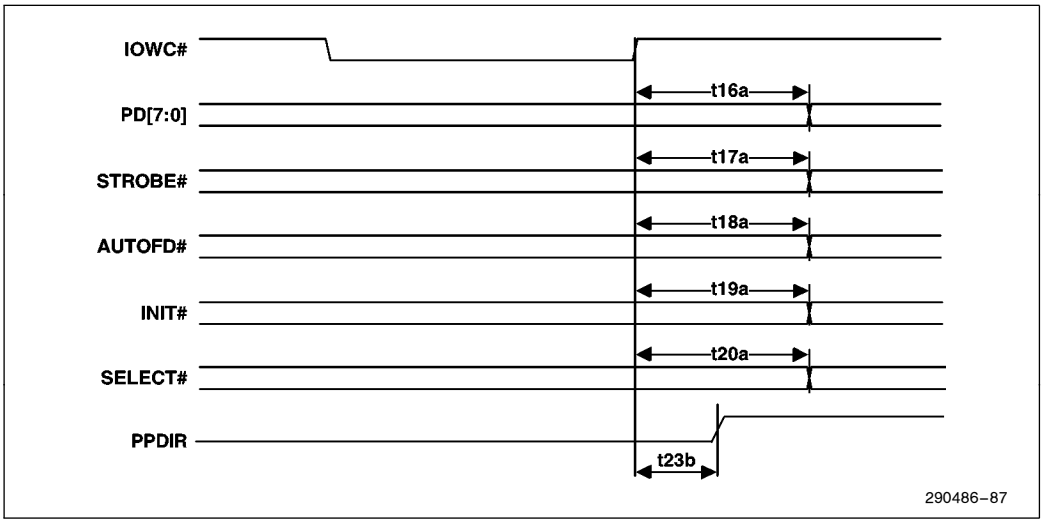



Figure 87. ISA-Compatible Write Timing

### 11.4.5 IDE TIMINGS

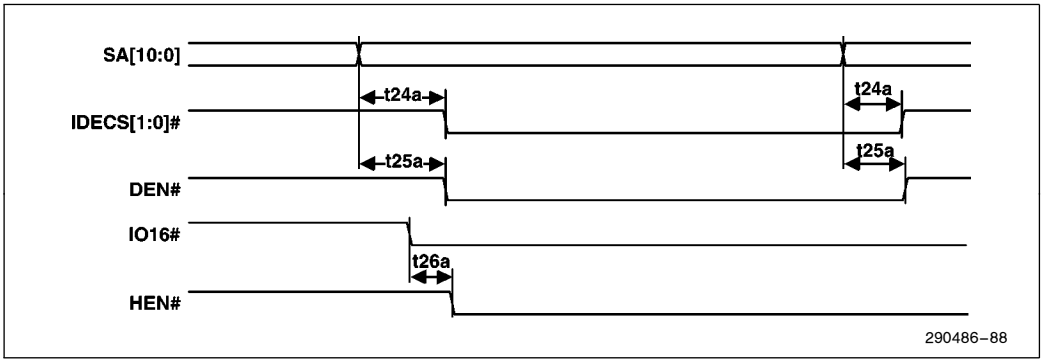



**Figure 88. IDE Timing**

### 11.4.6 GAME PORT TIMINGS

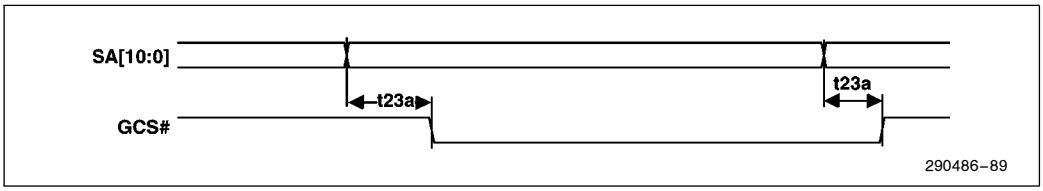



**Figure 89. Game Port Timing**

### 11.4.7 SERIAL PORT TIMINGS

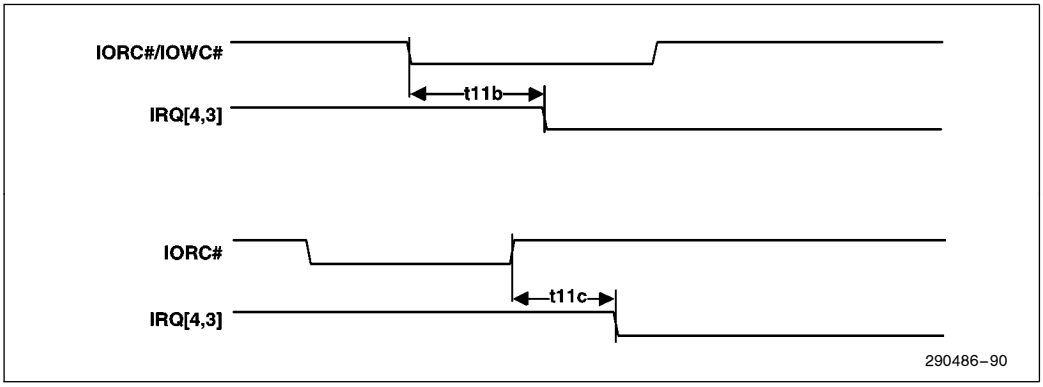



**Figure 90. Serial Port Interrupt Timing**

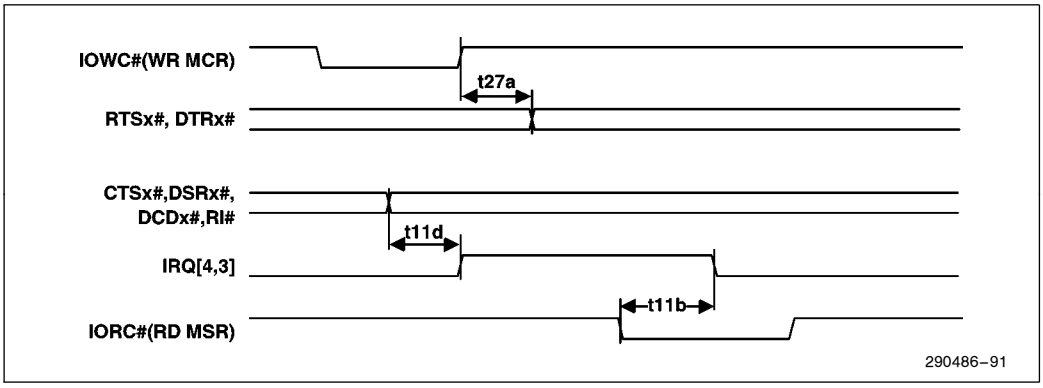



**Figure 91. Modem Control Timing**

## 12.0 PINOUT AND PACKAGE INFORMATION

### 12.1 Pin Assignment

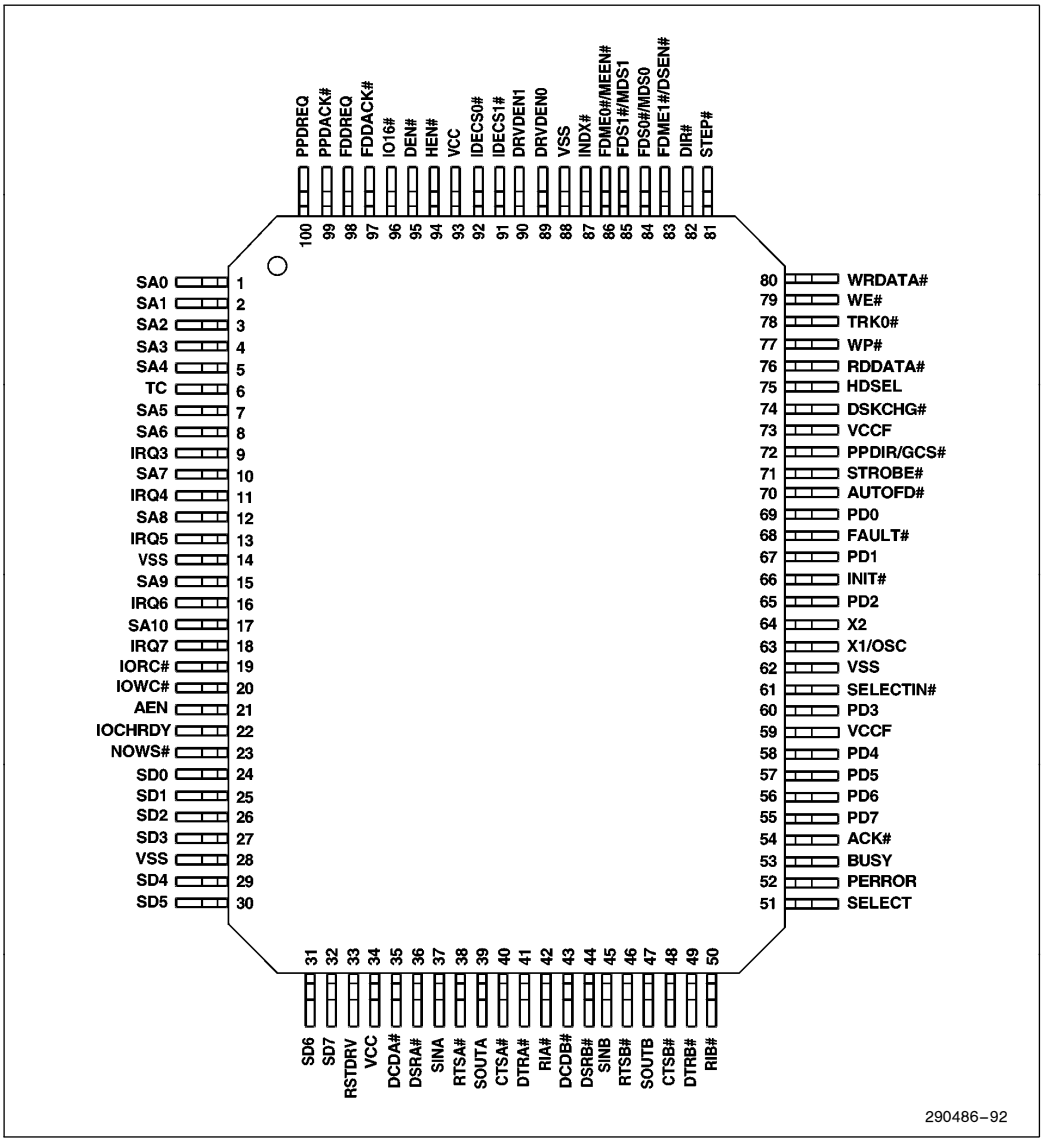



**Figure 92. 82091AA Pin Diagram**

**Table 44. Alphabetical 82091AA Pin Assignment**

| Signal Name | Pin # | Type |
|---|---|---|
| ACK# | 54 | I |
| AEN | 21 | I |
| AUTOFD# | 70 | O |
| BUSY | 53 | I |
| CTSA# | 40 | I |
| CTSB# | 48 | I |
| DCDA# | 35 | O |
| DCDB# | 43 | O |
| DEN# | 95 | I/O |
| DIR# | 82 | O |
| DRVDEN0 | 89 | O |
| DRVDEN1 | 90 | O |
| DSKCHG# | 74 | I |
| DSRA# | 36 | I |
| DSRB# | 44 | I |
| DTRA# | 41 | I/O |
| DTRB# | 49 | I/O |
| FAULT# | 68 | I |
| FDDACK# | 97 | I |
| FDDREQ | 98 | O |
| FDME0#/MEEN# | 86 | O |
| FDME1#/DSEN# | 83 | O |
| FDS0#/MDS0 | 84 | O |
| FDS1#/MDS1 | 85 | O |
| HDSEL | 75 | O |
| HEN# | 94 | I/O |
| IDECS0# | 92 | I/O |
| IDECS1# | 91 | I/O |
| INDX# | 87 | I |
| INIT# | 66 | O |
| IO16# | 96 | I |
| IOCHRDY | 22 | O |
| IORC# | 19 | I |
| IOWC# | 20 | I |

| Signal Name | Pin # | Type |
|---|---|---|
| IRQ3 | 9 | O |
| IRQ4 | 11 | O |
| IRQ5 | 13 | O |
| IRQ6 | 16 | O |
| IRQ7 | 18 | O |
| NOWS# | 23 | O |
| PD0 | 69 | I/O |
| PD1 | 67 | I/O |
| PD2 | 65 | I/O |
| PD3 | 60 | I/O |
| PD4 | 58 | I/O |
| PD5 | 57 | I/O |
| PD6 | 56 | I/O |
| PD7 | 55 | I/O |
| PERROR | 52 | I |
| PPDACK# | 99 | I |
| PPDREQ | 100 | O |
| PPDIR/GCS# | 72 | I/O |
| RDDATA# | 76 | I |
| RIA# | 42 | I |
| RIB# | 50 | I |
| RSTDRV | 33 | I |
| RTSA# | 38 | I/O |
| RTSB# | 46 | I/O |
| SA0 | 1 | I |
| SA1 | 2 | I |
| SA2 | 3 | I |
| SA3 | 4 | I |
| SA4 | 5 | I |
| SA5 | 7 | I |
| SA6 | 8 | I |
| SA7 | 10 | I |
| SA8 | 12 | I |
| SA9 | 15 | I |

**Table 44. Alphabetical 82091AA Pin Assignment** (Continued)

| Signal Name | Pin # | Type |
|---|---|---|
| SA10 | 17 | I |
| SD0 | 24 | I/O |
| SD1 | 25 | I/O |
| SD2 | 26 | I/O |
| SD3 | 27 | I/O |
| SD4 | 29 | I/O |
| SD5 | 30 | I/O |
| SD6 | 31 | I/O |
| SD7 | 32 | I/O |
| SINA | 37 | I |
| SINB | 45 | I |
| SELECT | 51 | I |
| SELECTIN# | 61 | O |
| SOUTA | 39 | I/O |
| SOUTB | 47 | I/O |
| STEP# | 81 | O |

| Signal Name | Pin # | Type |
|---|---|---|
| STROBE# | 71 | O |
| TC | 6 | I |
| TRK0# | 78 | I |
| $V_{CC}$ | 34 | V |
| $V_{CC}$ | 93 | V |
| $V_{CCF}$ | 59 | V |
| $V_{CCF}$ | 73 | V |
| $V_{SS}$ | 14 | V |
| $V_{SS}$ | 28 | V |
| $V_{SS}$ | 62 | V |
| $V_{SS}$ | 88 | V |
| WE# | 79 | O |
| WP# | 77 | I |
| WRDATA# | 80 | O |
| X1/OSC | 63 | I |
| X2 | 64 | I |

**Table 45. Numerical 82091AA Pin Assignment**

| Pin# | Signal Name | Type |
|---|---|---|
| 1 | SA0 | I |
| 2 | SA1 | I |
| 3 | SA2 | I |
| 4 | SA3 | I |
| 5 | SA4 | I |
| 6 | TC | I |
| 7 | SA5 | I |
| 8 | SA6 | I |
| 9 | IRQ3 | O |
| 10 | SA7 | I |
| 11 | IRQ4 | O |
| 12 | SA8 | I |
| 13 | IRQ5 | O |
| 14 | $V_{SS}$ | V |
| 15 | SA9 | I |

| Pin# | Signal Name | Type |
|---|---|---|
| 16 | IRQ6 | O |
| 17 | SA10 | I |
| 18 | IRQ7 | O |
| 19 | IORC# | I |
| 20 | IOWC# | I |
| 21 | AEN | I |
| 22 | IOCHRDY | O |
| 23 | NOWS# | O |
| 24 | SD0 | I/O |
| 25 | SD1 | I/O |
| 26 | SD2 | I/O |
| 27 | SD3 | I/O |
| 28 | $V_{SS}$ | V |
| 29 | SD4 | I/O |
| 30 | SD5 | I/O |

**Table 45. Numerical 82091AA Pin Assignment** (Continued)

| Pin # | Signal Name | Type |
|---|---|---|
| 31 | SD6 | I/O |
| 32 | SD7 | I/O |
| 33 | RSTDRV | I |
| 34 | $V_{CC}$ | V |
| 35 | DCDA# | O |
| 36 | DSRA# | I |
| 37 | SINA | I |
| 38 | RTSA# | I/O |
| 39 | SOUTA | I/O |
| 40 | CTSA# | I |
| 41 | DTRA# | I/O |
| 42 | RIA# | I |
| 43 | DCDB# | O |
| 44 | DSRB# | I |
| 45 | SINB | I |
| 46 | RTSB# | I/O |
| 47 | SOUTB | I/O |
| 48 | CTSB# | I |
| 49 | DTRB# | I/O |
| 50 | RIB# | I |
| 51 | SELECT | I |
| 52 | PERROR | I |
| 53 | BUSY | I |
| 54 | ACK# | I |
| 55 | PD7 | I/O |
| 56 | PD6 | I/O |
| 57 | PD5 | I/O |
| 58 | PD4 | I/O |
| 59 | $V_{CCF}$ | V |
| 60 | PD3 | I/O |
| 61 | SELECTIN# | O |
| 62 | $V_{SS}$ | V |
| 63 | X1/OSC | I |
| 64 | X2 | I |
| 65 | PD2 | I/O |

| Pin # | Signal Name | Type |
|---|---|---|
| 66 | INIT# | O |
| 67 | PD1 | I/O |
| 68 | FAULT# | I |
| 69 | PD0 | I/O |
| 70 | AUTOFD# | O |
| 71 | STROBE# | O |
| 72 | PPDIR/GCS# | I/O |
| 73 | $V_{CCF}$ | V |
| 74 | DSKCHG# | I |
| 75 | HDSEL | O |
| 76 | RDDATA# | I |
| 77 | WP# | I |
| 78 | TRK0# | I |
| 79 | WE# | O |
| 80 | WRDATA# | O |
| 81 | STEP# | O |
| 82 | DIR# | O |
| 83 | FDME1#/DSEN# | O |
| 84 | FDS0#/MDS0 | O |
| 85 | FDS1#/MDS1 | O |
| 86 | FDME0#/MEEN# | O |
| 87 | INDX# | I |
| 88 | $V_{SS}$ | V |
| 89 | DRVDEN0 | O |
| 90 | DRVDEN1 | O |
| 91 | IDECS1# | I/O |
| 92 | IDECS0# | I/O |
| 93 | $V_{CC}$ | I/O |
| 94 | HEN# | V |
| 95 | DEN# | I/O |
| 96 | IO16# | I |
| 97 | FDDACK# | I |
| 98 | FDDREQ | O |
| 99 | PPDACK# | I |
| 100 | PPDREQ | O |

## 12.2 Package Characteristics

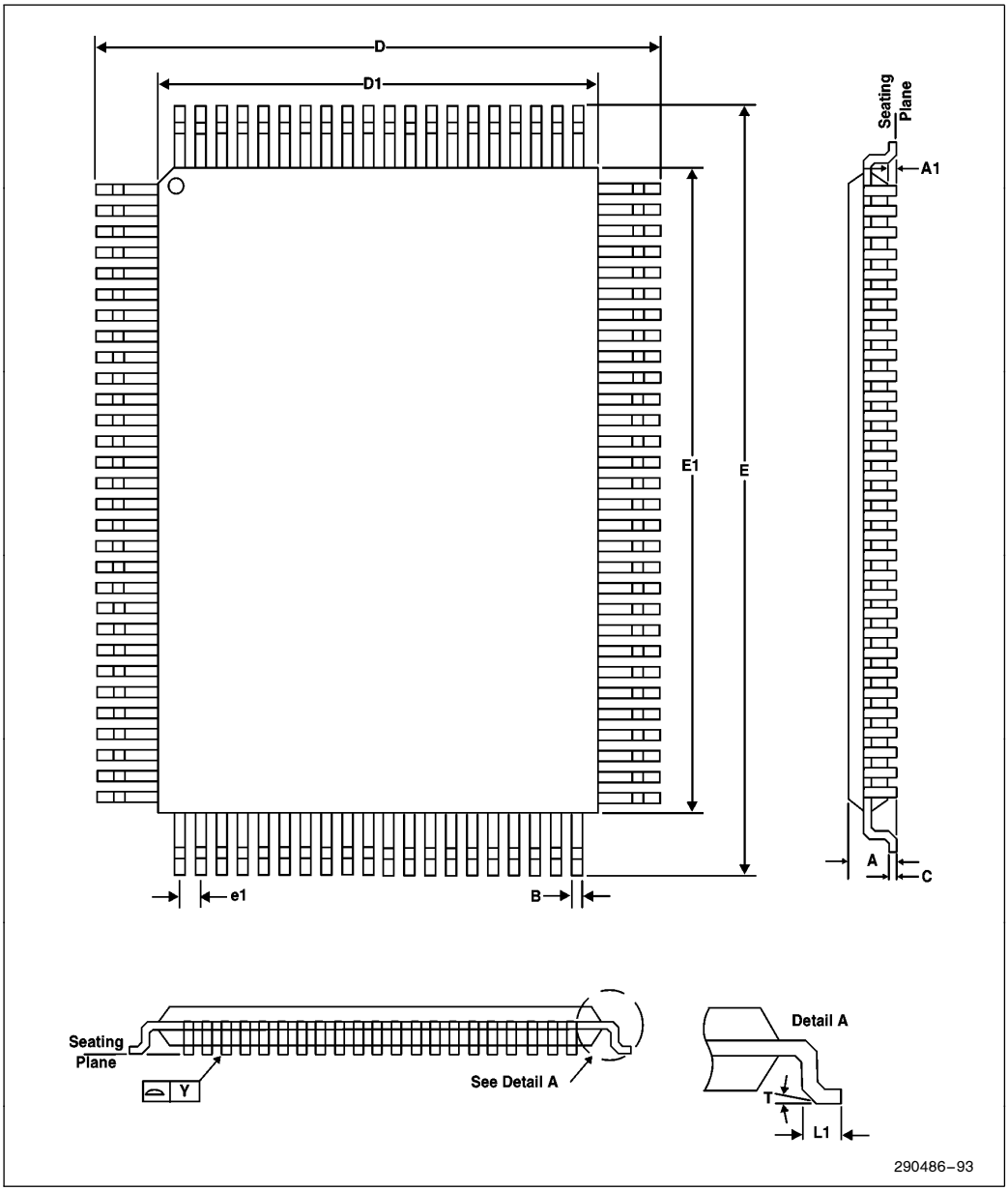



**Figure 93. 100-Pin Quad Flat Pack (QFP) Dimensions**

| Quad Flat Pack Package | | | | |
|---|---|---|---|---|
| **Symbol** | **Millimeters** | | | |
| | **Minimum** | **Nominal** | **Maximum** | **Notes** |
| A | | | 3.15 | |
| A1 | 0.0 | | | |
| B | 0.20 | 0.30 | 0.40 | |
| C | 0.10 | 0.15 | 0.20 | |
| D | 17.5 | 17.9 | 18.3 | |
| D1 | | 14.0 | | |
| E | 23.5 | 23.9 | 24.3 | |
| E1 | | 20.0 | | |
| e1 | 0.53 | 0.65 | 0.77 | |
| L1 | 0.60 | 0.80 | 1.00 | |
| N | | 100 | | Rectangle |
| T | 0.00 | | 10.0 | |
| Y | | | 0.10 | |
| ISSUE | JEDEC | | | |

## 13.0 DATA SEPARATOR CHARACTERISTICS FOR FLOPPY DISK MODE

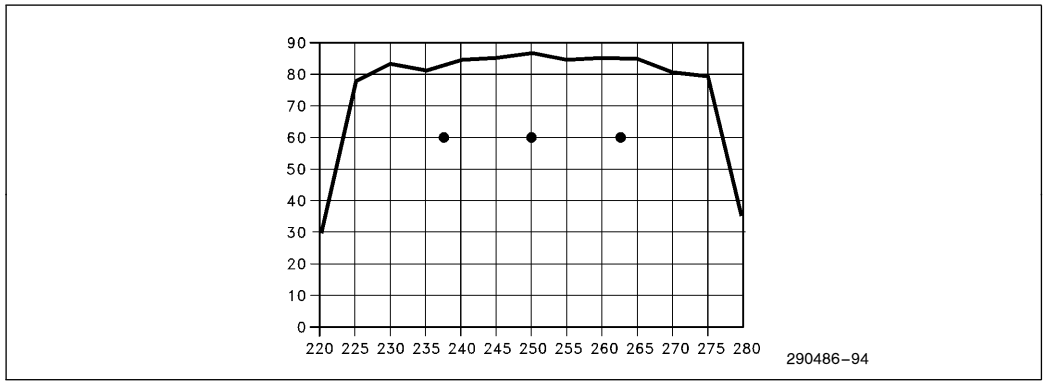



**Figure 94. Typical Jitter Tolerance vs Data Rate (Capture Range 250 Kbps)**

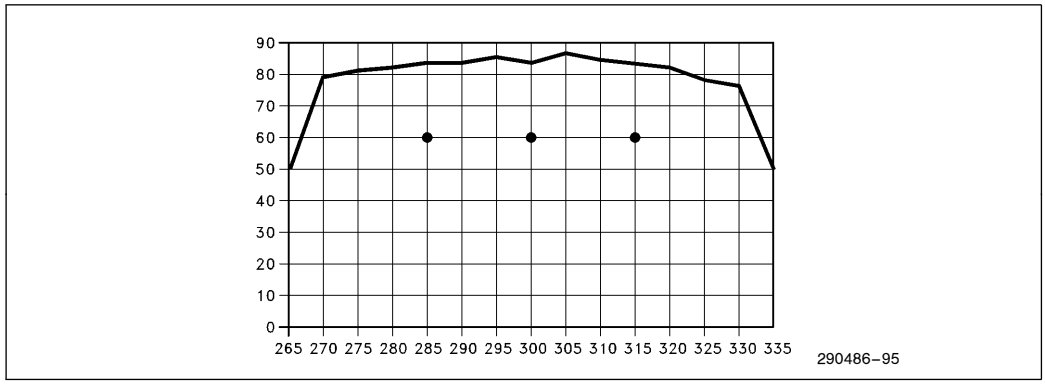



**Figure 95. Typical Jitter Tolerance vs Data Rate (Capture Range 300 Kbps)**

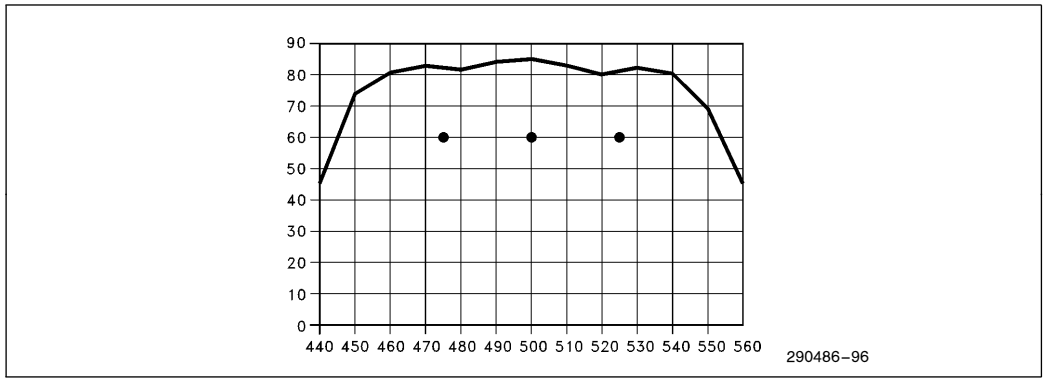



**Figure 96. Typical Jitter Tolerance vs Data Rate (Capture Range 500 Kbps)**

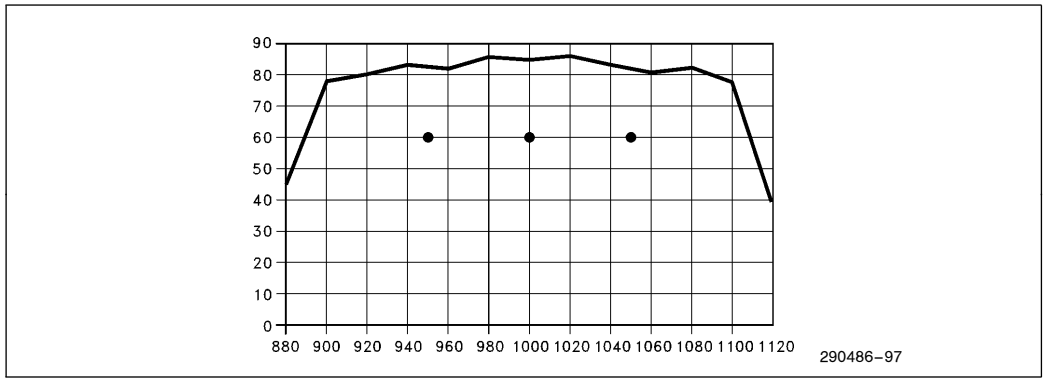


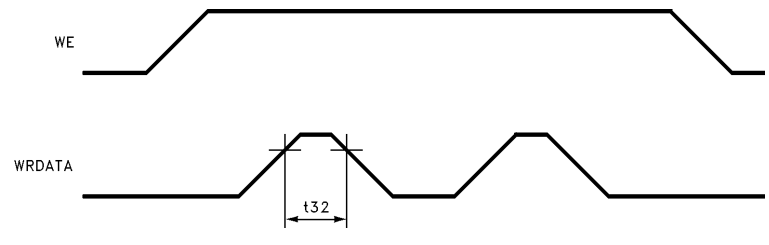

**Figure 97. Typical Jitter Tolerance vs Data Range (Capture Range 1 Mbps)**

Jitter Tolerance measured in percent. Capture range expressed as a percent of data rate, i.e., ±3% percent.

• = Test Points: 250 Kbps, 300 Kbps, 500 Kbps and 1 Mbps are center, ±5 percent @ 60 percent jitter.

Test points are tested at temparture and $V_{CC}$ limits. Refer to the datasheet. Typical conditions are: room temperature, nominal $V_{CC}$.

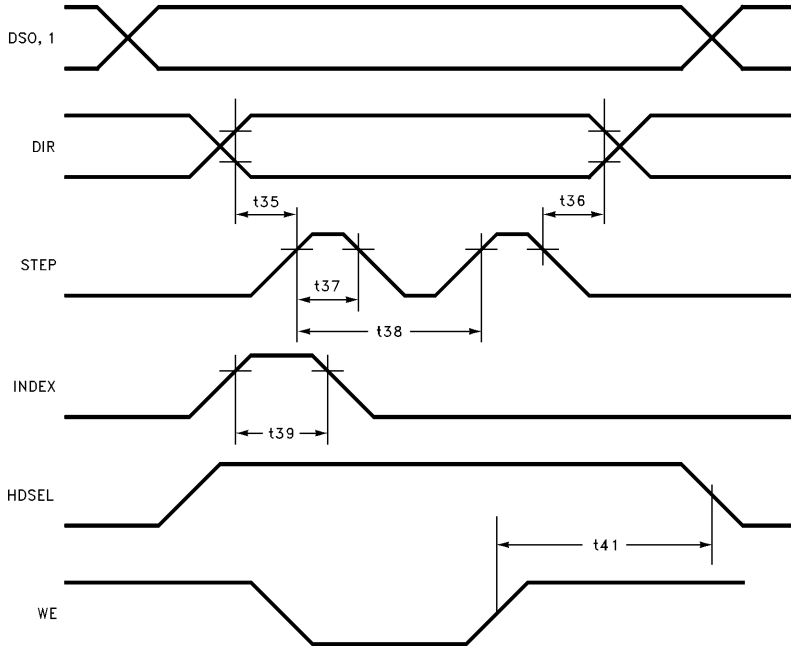
## 13.1 Write Data Timing




290486–98

**NOTE:**
Invert high.

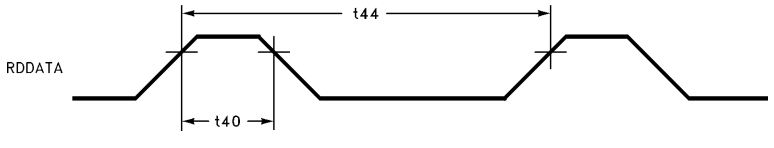
## 13.2 Drive Control




290486–99

**NOTE:**
For overlapped seeks, only one step pulse per drive selection is issued. Non-overlapped seeks will issue all programmed step pulses. Invert high.

## 13.3 Internal PLL




**NOTE:**
Invert high.

# APPENDIX A
# FDC FOUR DRIVE SUPPORT

Section 8.0 of this document completely describes the FDC when the module is configured for two drive support. In addition, the FDC commands in Section 8.0 provide four drive support information. This appendix provides additional information concerning four drive support. The signal pins that are affected by four drive support are described in Section A.1. Note that the FDC signals not discussed in this appendix operate the same for both two and four drive systems. The following registers are described in this appendix; Digital Output Register (DOR), Enhanced Tape Drive Register (TDR), and the Main Status Register (MSR). Some bits in these registers operate differently in a four drive configuration than a two drive configuration.

**NOTES:**

- The descriptions in this appendix assume that four floppy drive support has been selected by setting FDDQTY to 1 in the AIPCFG1 Register.
- Only drive 0 or drive 1 can be selected as the boot drive.

## A.1 Floppy Disk Controller Interface Signals

These signal descriptions are for a four drive system (FDDQTY = 1 in the AIPCFG1 Register). See Section 2.0 for two drive system signal descriptions.

| Signal Name | Type | Description |
|---|---|---|
| FDME1#/DSEN#(1) | O | **FLOPPY DRIVE MOTOR ENABLE 1, or DRIVE SELECT ENABLE:** In a four drive system, this signal functions as a drive select enable (DSEN#). When DSEN# is asserted, MDS1 and MDS0 reflect the selection of the drive. |
| FDS1#/MDS1(1) | O | **FLOPPY DRIVE SELECT1, or MOTOR DRIVE SELECT 1:** In a four drive system, this signal functions as a motor drive select (MDS1). MDS1, together with MDS0, indicate which of the four drives is selected, as shown in note 1. |
| FDME0#/MEEN#(1) | O | **FLOPPY DRIVE MOTOR ENABLE 0 or MOTOR ENABLE ENABLE:** In a four drive system, this signal functions as a motor enable enable (MEEN#). MEEN# is asserted to enable the external decoding of MDS1 and MDS0 for the appropriate motor enable (see note 1). |
| FDS0#/MDS0(1) | O | **FLOPPY DRIVE SELECT 0 or MOTOR DRIVE SELECT 0:** In a four drive system, this signal functions as motor drive select (MDS0). MDS0, together with MDS1, indicate which of the four drives is selected as shown in note 1. |

**NOTE:**
1. These signal pins are used to control an external decoder for four floppy disk drives as shown below. Refer to the DOR Register Description in Section A.2 for details.

| MDS1 | MDS0 | DSEN# = 0 | MEEN# = 0 |
|---|---|---|---|
| 0 | 0 | Drive 0 | ME0 |
| 0 | 1 | Drive 1 | ME1 |
| 1 | 0 | Drive 2 | ME2 |
| 1 | 1 | Drive 3 | ME3 |

## A.2 DOR—Digital Output Register

I/O Address: Base + 2h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The Digital Output Register enables/disables the floppy disk drive motors, selects the disk drives, enables/disables DMA, and provides a FDC module reset. The DOR reset bit and the Motor Enable bits have to be inactive when the 82091AA's FDC is in powerdown. The DMAGATE# and Drive Select bits are unchanged. During powerdown, writing to the DOR does not wake up the 82091AA's FDC, except for activating any of the motor enable bits. Setting the motor enable bits to 1 will wake up the module. The four internal drive select and four internal motor enable signals are encoded to a total of four output pins as described in Table 47. Figure 99 shows an example of how these four output pins can be decoded to provide four drive select and four motor enable signals. Note that only drive 0 or drive 1 can be used as the boot drive when four disk drives are enabled.

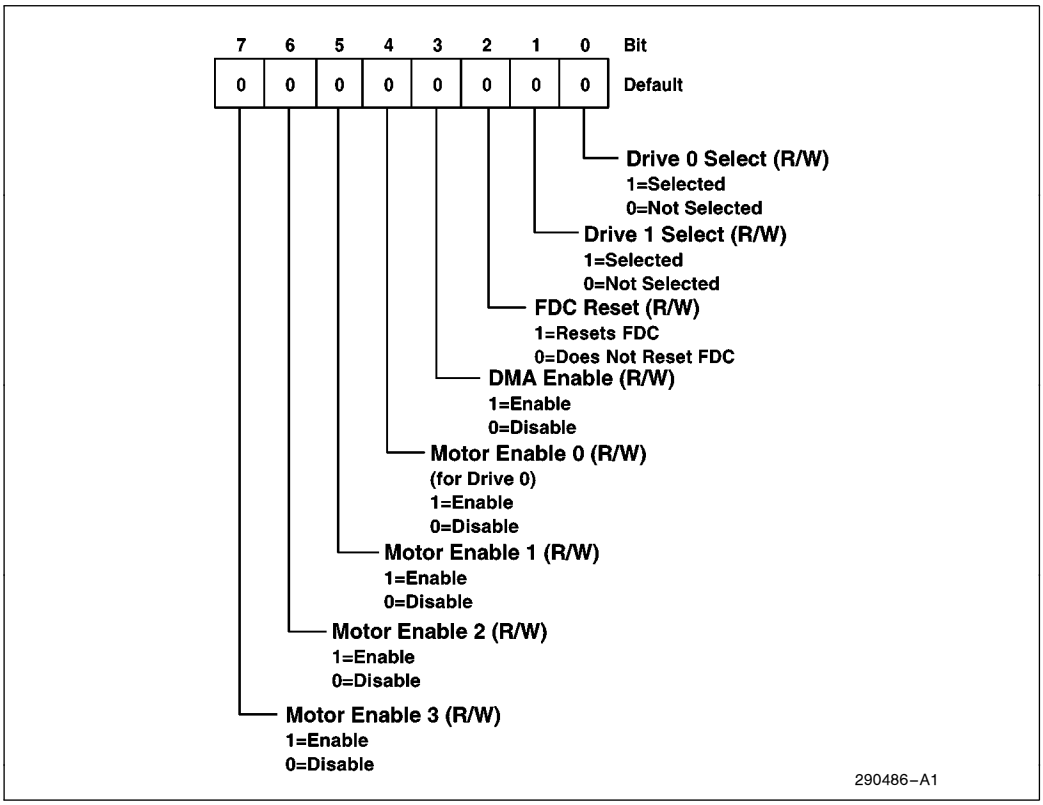



**Figure 98. Digital Output Register**

| Bit | Description |
|---|---|
| 7 | **Motor Enable 3 (ME3):** This bit controls a motor drive enable output signal and provides the signal output for the floppy drive 3 motor (via external decoding) as shown in Table 46. |
| 6 | **Motor Enable 2 (ME2):** This bit controls a motor drive enable output signal and provides the signal output for the floppy drive 2 motor (via external decoding) as shown in Table 46. |
| 5 | **Motor Enable 1 (ME1):** This bit controls a motor drive enable signal and provides the signal output for the floppy drive 1 motor (via external decoding) as shown in Table 46. |
| 4 | **Motor Enable 0 (ME0):** This bit controls a motor drive enable signal and provides the signal output for the floppy drive 0 motor (via external decoding) as shown in Table 46. |
| 3 | **DMA Gate (DMAGATE):** This bit enables/disables DMA for the FDC. When DMAGATE = 1, DMA for the FDC is enabled. In this mode FDDREQ, TC, IRQ6, and FDDACK# are enabled. When DMAGATE = 0, DMA for the FDC is disabled. In this mode, the IRQ6 and DRQ outputs are tri-stated and the DACK# and TC inputs are disabled to the FDC. Note that the TC input is only disabled to the FDC module. Other functional units in the 82091AA (e.g., parallel port or IDE interface) can still use the TC input signal for DMA activities. |
| 2 | **FDC Reset (DORRST):** DORRST is a software reset for the FDC module. When DORRST is set to 0, the basic core of the 82091AA's FDC and the FIFO circuits are cleared conditioned by the LOCK bit in the Configure Command. This bit is set to 0 by software or a hard reset (RSTDRV asserted). The FDC remains in a reset state until software sets this bit to 1. This bit does not affect the DSR, CCR and other bits of the DOR. DORRST must be held active for at least 0.5 $\mu$s at 250 Kbps. This is less than a typical ISA I/O cycle time. Thus, in most systems consecutive writes to this register to toggle this bit allows sufficient time to reset the FDC. |
| 1:0 | **Drive Select (DS[1:0]):** This field provides the output signals to select a particular floppy drive (via external decoding) as shown in Table 47. Note that the drive motor can be enabled separately without selecting the drive. This permits the motor to come up to speed before selecting the drive. Note also that only one drive can be selected at a time. However, the drive should not be selected without enabling the appropriate drive motor via bits[7:4] of this register. |

**Table 46. Output Pin Status for Four Disk Drives**

| | FDC DOR Register Bits | | | | | | Signal Pins | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Description** | **ME3** | **ME2** | **ME1** | **ME0** | **DS1** | **DS0** | **MDS1#** | **MDS0#** | **DSEN#** | **MEEN#** |
| ME0 and DS0 enable | X | X | X | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| ME1 and DS1 enable | X | X | 1 | X | 0 | 1 | 0 | 1 | 0 | 0 |
| ME2 and DS2 enable | X | 1 | X | X | 1 | 0 | 1 | 0 | 0 | 0 |
| ME3 and DS3 enable | 1 | X | X | X | 1 | 1 | 1 | 1 | 0 | 0 |
| ME0 enable only | X | X | X | 1 | DS[1:0]≠00 | | 0 | 0 | 1 | 0 |
| ME1 enable only | X | X | 1 | 0 | DS[1:0]≠01 | | 0 | 1 | 1 | 0 |
| ME2 enable only | X | 1 | 0 | 0 | DS[1:0]≠10 | | 1 | 0 | 1 | 0 |
| ME3 enable only | 1 | 0 | 0 | 0 | DS[1:0]≠11 | | 1 | 1 | 1 | 0 |
| No ME or DS enable | 0 | 0 | 0 | 0 | X | X | 1 | 1 | 1 | 1 |

**NOTE:**
To enable a particular drive motor and select the drive, the value for DS[1:0] must match the appropriate motor enable bit selected as indicated in the first four rows of the table. For example, to enable the drive 0 motor and select the drive, ME0 is set to 1 and DS[1:0] must be set to 00. To enable the drive motor and keep the drive de-selected the value for DS[1:0] must not match the particular motor enable as shown in the first four rows. For example, to enable the motor for drive 0 while the drive remains de-selected, ME0 is set to 1 and DS[1:0] is set to 01, 10, or 11.

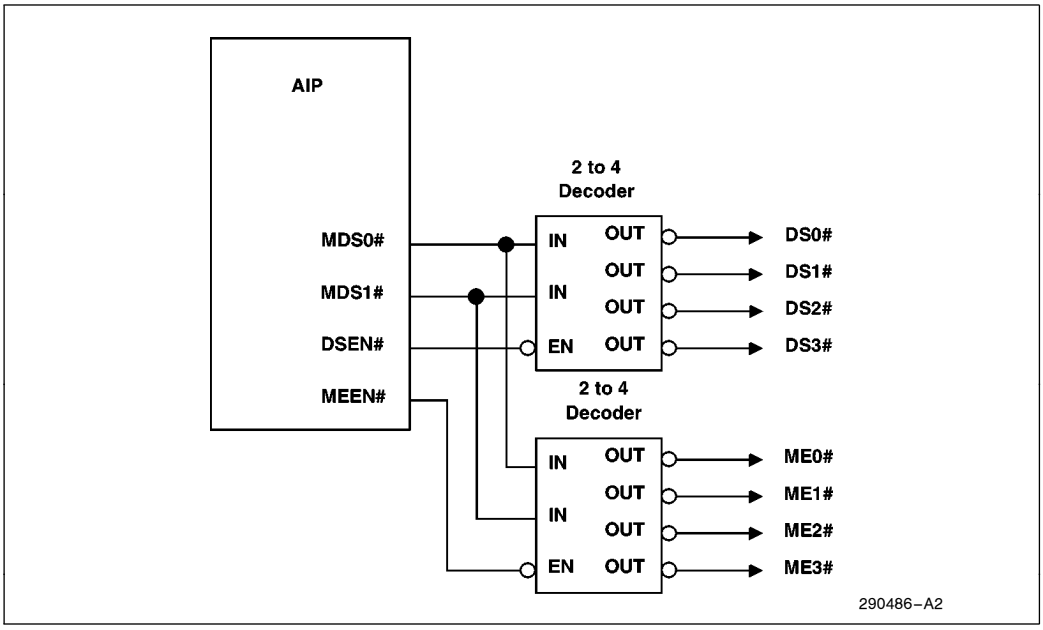



**Figure 99. Example External Decoder (Four Drive System)**

## A.3 TDR—Enhanced Tape Drive Register

I/O Address: Base +3h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. A hardware reset sets all bits in this register to 0 making drive 0 not available for tape support. A software reset via bit 2 of the DOR does not affect this register. Drive 0 is reserved for the floppy boot drive. Bits[7:2] are only available when EREG EN = 1; otherwise the bits are tri-stated.

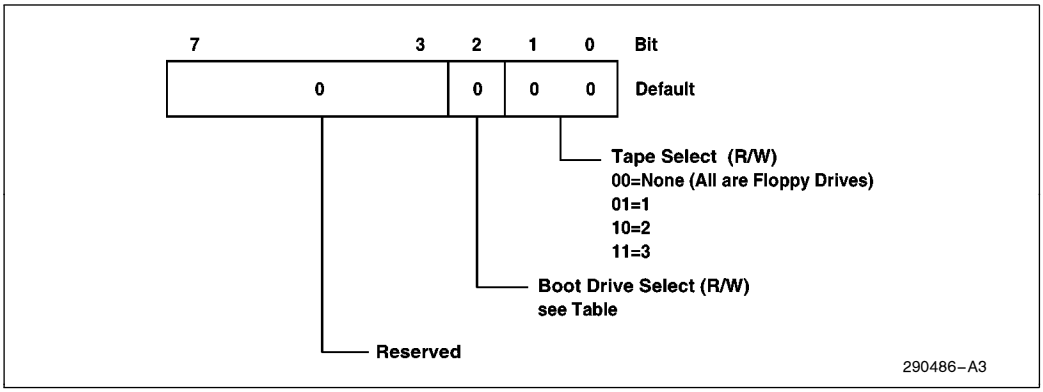



**Figure 100. Enhanced Tape Drive Register**

| Bit | Description |
|---|---|
| 7:3 | **Reserved:** |
| 2 | **Boot Drive Select (BOOTSEL):** The BOOTSEL bit is used to remap the drive selects and motor enables. The functionality is shown below:<br>**BOOTSEL** **Mapping**<br>0 DS0 → FDS0, ME0 → FDME0 (default)<br>DS1 → DS1, ME1 → FDME1<br>1 DS0 → DS1, ME0 → FDME1<br>DS1 → FDS0, ME1 → FDME0<br>Only drive 0 or drive 1 can be selected as the boot drive. |
| 1:0 | **Tape Select (TAPESEL[1:0]):** These two bits are used by software to assign a logical drive number to be a tape drive. Other than adjusting precompensation delays for tape support, these two bits do not affect the FDC hardware. They can be written and read by software as an indication of the tape drive assignment. Drive 0 is not available as a tape drive and is reserved as the floppy disk boot drive. The tape drive assignments are as follows:<br>**Bits[1:0]** **Drive Selected**<br>0 0 None (all are floppy disk drives)<br>0 1 1<br>1 0 2<br>1 1 3 |

## A.4 MSR—Main Status Register

I/O Address: Base + 4h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This read only register provides FDC status information. This information is used by software to control the flow of data to and from the FIFO (accessed via the FDCFIFO Register). The MSR indicates when the FDC is ready to send or receive data through the FIFO. During non-DMA transfers, this register should be read before each byte is transferred to or from the FIFO.

After a hard or soft reset or recovery from a powerdown state, the MSR is available to be read by the host. The register value is 00h until the oscillator circuit has stabilized and the internal registers have been initialized. When the FDC is ready to receive a new command, MSR[7:0] = 80h. The worst case time allowed for the MSR to report 80h (i.e., RQM is set to 1) is 2.5 $\mu$s after a hard or soft reset.

Main Status Register is used for controlling command input and result output for all commands. Some example values of the MSR are:

- MSR = 80H; The controller is ready to receive a command.
- MSR = 90H; Executing a command or waiting for the host to read status bytes (assume DMA mode).
- MSR = D0H; Waiting for the host to write status bytes.

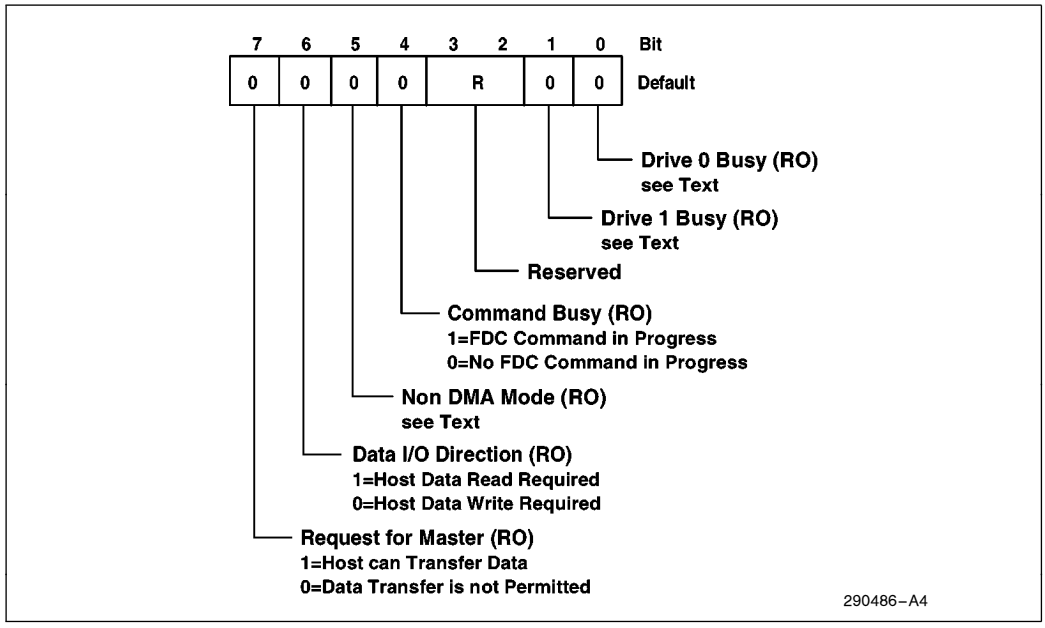



**Figure 101. Main Status Register**

| Bit | Description |
|---|---|
| 7 | **Request For Master (RQM):** When RQM = 1, the FDC is ready to send/receive data through the FIFO (FDCFIFO Register). The FDC sets this bit to 0 after a byte transfer and then sets the bit to 1 when it is ready for the next byte. During non-DMA execution phase, RQM indicates the status of IRQ6. |
| 6 | **Direction I/O (DIO):** When RQM = 1, DIO indicates the direction of a data transfer. When DIO = 1, the FDC is requesting a read of the FDCFIFO. When DIO = 0, the FDC is requesting a write to the FDCFIFO. |
| 5 | **NON-DMA (NONDMA):** Non-DMA mode is selected via the SPECIFY Command. In this mode, the FDC sets this bit to a 1 during the execution phase of a command. This bit is for polled data transfers and helps differentiate between the data transfer phase and the reading of result bytes. |
| 4 | **Command Busy (CMDBUSY):** CMDBUSY indicates when a command is in progress. When the first byte of the command phase is written, the FDC sets this bit to 1. CMDBUSY is set to 0 after the last byte of the result phase is read. If there is no result phase (e.g., SEEK or RECALIBRATE Commands), CMDBUSY is set to 0 after the last command byte is written. |
| 3 | **Drive 3 Busy (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 3. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive. |
| 2 | **Drive 2 Busy (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 2. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive. |
| 1 | **Drive 1 Busy (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 1. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive. |
| 0 | **Drive 0 Busy (DRV0BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 0. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive. |