AMBA IEEE1284 Parallel Port Interface

Data Sheet

ARM®

ARM DDI 0083A

AMBA IEEE1284 Parallel Port Interface
Data Sheet

Release Information

Change history

Date	Issue	Confidentiality	Change
September 1997	A	Non-Confidential	First release.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Contents

AMBA IEEE1284 Parallel Port Interface Data Sheet

Chapter 1
AMBA IEEE1284 Parallel Port Interface Data Sheet

This document describes the AMBA IEEE1284 Parallel Port Interface. It contains the following sections:

1.1 About the AMBA IEEE1284 Parallel Port Interface

The IEEE1284 Interface implements the IEEE specification providing a signalling method for bi-directional parallel communications with printers and other peripheral devices.

The IEEE1284 specification defines five modes of operation, namely:

- Compatibility mode
- Nibble mode
- Byte mode
- *Extended Capabilities Port* (ECP) mode
- *Enhanced Parallel Port* (EPP) mode.

All the above five modes are supported from a host viewpoint in some form, with registers defined for the control and status monitoring of the port in each mode. Interrupt request outputs have been provided that are asserted during handshaking and data transfers that occur during communication with the peripheral device.

Both forward (host-to-peripheral) and reverse (peripheral-to-host) data flows are supported through single register or FIFO access from the AMBA bus.

Note

This data sheet assumes that the reader is familiar with the following specification:

IEEE1284 Std-1284-1994 *IEEE Standard Signalling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers*.

1.2 Signal description

For clarity, the description of the multi-purpose pins of the IEEE Port I/O have been written for each of the five modes of operation of the IEEE1284 Interface:

- *IEEE port I/O: Compatibility mode*
- *IEEE port I/O: Nibble mode* on page 1-4
- *IEEE port I/O: Byte mode* on page 1-5
- *IEEE port I/O: Extended Capabilities Port (ECP) mode* on page 1-5
- *IEEE port I/O: Enhanced Parallel Port (EPP) mode* on page 1-7.

1.2.1 IEEE port I/O: Compatibility mode

Table 1-1 shows the pin functions for compatibility mode.

Table 1-1 IEEE Port I/O signals: Compatibility mode

Name	Type	Description
Data[8:1]	Bidir	Forward transfer byte wide databus.
Busy	In	When HIGH, indicates that the peripheral device is not ready to receive data.
Select	In	When HIGH, signifies that the peripheral is on-line.
nAck	In	Pulsed LOW by the peripheral to acknowledge transfer of a data byte from the host.
nFault	In	Set LOW by the peripheral to indicate that an error has occurred.
PError	In	When HIGH, signifies error has occurred in paper path. Peripherals shall set **nFault** LOW whenever they set **PError** HIGH.
nInit	Out	Pulsed LOW to reset the interface and force a return to Compatibility Mode idle phase.
nStrobe	Out	When LOW, data is transferred into the input latch of the peripheral. Data is valid while nStrobe is LOW.
nSelectIn	Out	When LOW, this peripheral is selected.
nAutoFd	Out	Interpretation varies between peripherals. When LOW, puts some printers into auto-line feed mode.
Buffen	Out	Bi-directional data pin and external buffer directional control signal.

1.2.2 IEEE port I/O: Nibble mode

Table 1-2 shows the pin functions for nibble mode.

Table 1-2 IEEE Port I/O signals: Nibble mode

Name	Type	Description
Data[8:1]	Bidir	Not Used.
Busy	In	Reverse data transfer phase: Data bit 3, then 7, then forward channel busy status.
Select	In	Reverse data transfer phase: Data bit 1, then 5.
nAck	In	Reverse data transfer phase: Used to qualify data being sent to the host. Reverse idle phase: Set LOW then HIGH by the peripheral to cause an interrupt indicating to the host that data is available.
nFault	In	Reverse data transfer phase: Set LOW to indicate that the peripheral has the data ready to send to the host. Then used to send Data bit 0, then 4. Reverse idle phase: Used to indicate that data is available.
PError	In	Reverse data transfer phase: Data bit 2, then 6. Reverse idle phase: Set HIGH until the host requests a transfer, then follows **nFault**.
nInit	Out	Set HIGH by the host.
nStrobe	Out	Set HIGH during transfers to avoid latching data into the peripheral.
nSelectIn	Out	Set HIGH by the host.
nAutoFd	Out	Reverse data transfer phase: Set LOW to indicate that the host can receive peripheral-to-host data, then set HIGH to acknowledge receipt of that nibble. Following a reverse channel transfer, the interface changes to idle phase when **nAutoFd** is set LOW and the peripheral has no data available. Reverse idle phase: Set HIGH in response to **nAck** LOW pulse to re-enter reverse data transfer phase. If set HIGH with **nSelectIn** set LOW, the IEEE1284 idle phase is aborted, and the interface returns to Compatibility Mode.
Buffen	Out	Bi-directional data pin and external buffer directional control signal.

1.2.3 IEEE port I/O: Byte mode

Table 1-3 shows the pin functions for byte mode.

Table 1-3 IEEE Port I/O signals: Byte mode

Name	Type	Description
Data[8:1]	Bidir	Reverse channel byte wide data.
Busy	In	Forward channel busy status.
Select	In	Driven to indicate the response to a requested extensibility byte sent by the host.
nAck	In	Reverse data transfer phase: Used to qualify data being sent to the host. Reverse idle phase: Set LOW then HIGH by the peripheral to cause an interrupt indicating to the host that data is available.
nFault	In	Set LOW to indicate that the peripheral has data to send to the host.
PError	In	Reverse data transfer phase: Follows nFault. Reverse idle phase: Set HIGH until the host requests a data transfer, then follows nFault.
nInit	Out	Set HIGH by the host.
nStrobe	Out	Will be pulsed LOW during Byte Mode transfers. The peripheral shall ensure that nStrobe does not cause data to be latched when this happens.
nSelectIn	Out	Set HIGH by the host.
nAutoFd	Out	Reverse data transfer phase: Set LOW to indicate that the host can receive a peripheral-to-host byte, then set HIGH to acknowledge receipt of that byte. Following a reverse channel transfer, the interface transitions to idle phase when nAutoFd is set LOW and the peripheral has no data available. Reverse idle phase: Set HIGH in response to nAck LOW pulse to re-enter reverse data transfer phase. If set HIGH with nSelectIn set LOW, the IEEE 1284 idle phase is aborted and the interface returns to Compatibility Mode.
Buffen	Out	Bi-directional data pin and external buffer directional control signal.

1.2.4 IEEE port I/O: Extended Capabilities Port (ECP) mode

Table 1-4 on page 1-6 shows the pin functions for ECP mode.

Table 1-4 IEEE Port I/O signals: ECP mode

Name	Type	Description
Data[8:1]	Bidir	Host-to-peripheral or peripheral-to-host address or data.
Busy	In	The peripheral uses this signal for flow control in the forward direction. **nAck** also provides a ninth data bit used to determine whether command or data information is present on the data signals in the reverse direction.
Select	In	Driven to indicate the response to a requested extensibility byte sent by the host.
nAck	In	Used in a closed-loop handshake with **nAutoFd** to transfer data from the peripheral to the host.
nFault	In	During ECP mode, the peripheral may drive this pin LOW to request communication with the host. The request is merely a hint; the host has ultimate control over the transfer direction. This signal provides a mechanism for peer-to-peer communication. Typically, it is used to generate an interrupt to the host. This signal is valid in the forward and reverse directions.
PError	In	The peripheral drives this signal LOW to acknowledge nInit. The host relies upon **PError** to determine when it is permitted to drive the data signals.
nInit	Out	This signal is driven LOW to place the channel in reverse direction. While in the ECP Mode, the peripheral is only allowed to drive the bi-directional data signals when nInit is LOW and **nSelectIn** is HIGH.
nStrobe	Out	Used in a closed-loop handshake with Busy to transfer data or address information from the host to the peripheral.
nSelectIn	Out	Driven HIGH by the host while in ECP mode. Set LOW by the host to terminate ECP mode and return the Interface to Compatibility Mode.
nAutoFd	Out	The host drives this signal for flow control in the reverse direction. It is used in the interlocked handshake with **nAck**. **nAutoFd** also provides a ninth bit that is used to determine whether command or data information is present on the data signals in the forward direction.
Buffen	Out	Bi-directional data pin and external buffer directional control signal.

1.2.5 IEEE port I/O: Enhanced Parallel Port (EPP) mode

Table 1-5 shows the pin functions for EPP mode.

Table 1-5 IEEE Port I/O signals: EPP mode

Name	Type	Description
Data[8:1]	Bidir	Host-to-peripheral or peripheral-to-host address or data.
Busy	In	This signal should be driven inactive as a positive acknowledgement from the peripheral that transfer of data or address is completed.The signal is active LOW. It should be driven active as an indication that the device is ready for the next address or data transfer.
Select	In	This signal is manufacturer specific.
nAck	In	Used by the peripheral to interrupt the host. This signal is active HIGH and positive edge triggered.
nFault	In	This signal is manufacturer specific.
PError	In	This signal is manufacturer specific.
nInit	Out	When driven LOW, this signal initiates a termination cycle that results in the interface returning to Compatibility Mode.
nStrobe	Out	Set LOW to denote an address or data write operation to the peripheral. Set HIGH to denote an address or data read operation from the peripheral.
nSelectIn	Out	This signal is used to denote an address cycle. It is active LOW.
nAutoFd	Out	This signal is used to denote a data cycle. It is active LOW.
Buffen	Out	Bi-directional data pin and external buffer directional control signal.

1.2.6 Interrupt I/O

Table 1-6 shows the interrupt signals.

Table 1-6 Interrupt signals

Name	Type	Description
IntReqTrc	Out	Active HIGH signal asserted on a successful forward/reverse data or address transfer.
IntReqTim	Out	Active HIGH signal asserted when the peripheral fails to provide a response to a host-driven handshake.
IntReqRav	Out	Active HIGH signal asserted as a result of a peripheral input control signal changing state.
IntReqInt	Out	Active HIGH signal set when the peripheral asserts the 'Interrupt' input in EPP mode.
IntReqEmp	Out	Active HIGH signal asserted if the Forward Interface FIFO is emptied in the course of a data block transfer.
IntReqDat	Out	Active HIGH signal asserted whenever the forward or reverse data FIFOs require service.

1.2.7 Miscellaneous signals

Table 1-7 shows the miscellaneous signals.

Table 1-7 Miscellaneous signals

Name	Type	Description
RefClk	In	48 MHz Reference Clock.

1.3 Functional description

The IEEE1284 Parallel Port Interface is described in:

- *Functionality*
- *Communication sequence* on page 1-10
- *Compatibility mode* on page 1-12
- *FIFO transfers* on page 1-14.

1.3.1 Functionality

Figure 1-1 shows a block diagram of the IEEE Interface.

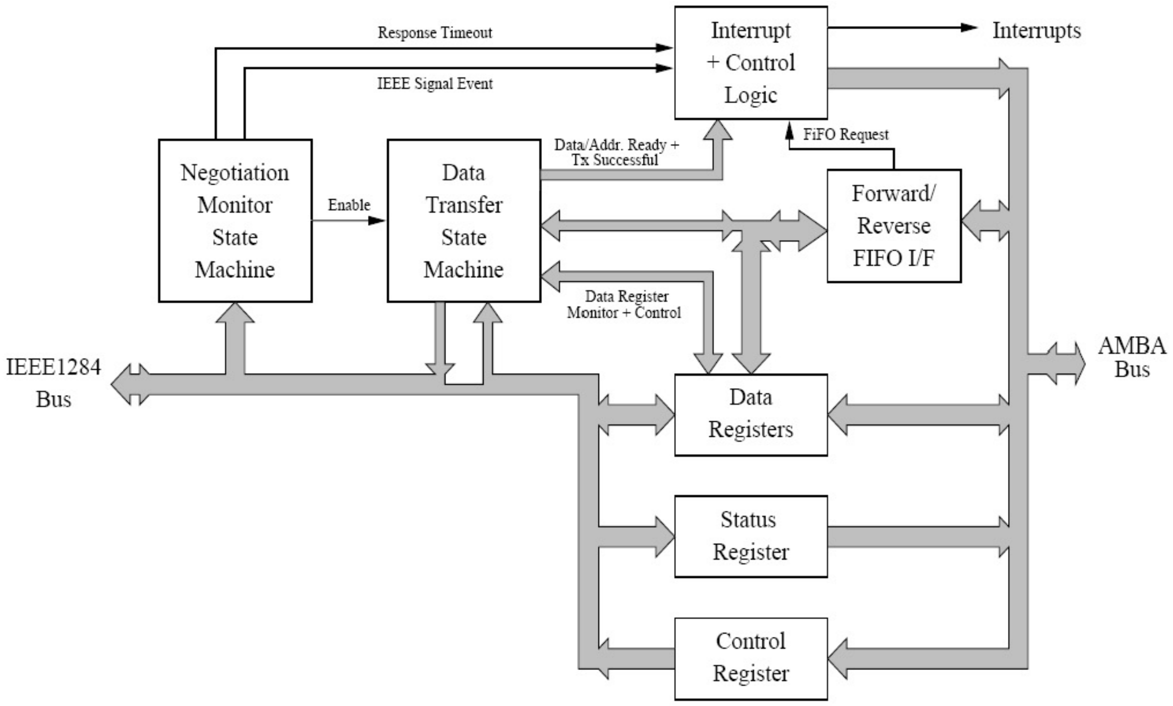
Response Timeout
IEEE Signal Event
Interrupt + Control Logic
Interrupts
FiFO Request
Data/Addr. Ready + Tx Successful
Negotiation Monitor State Machine
Enable
Data Transfer State Machine
Forward/ Reverse FIFO I/F
Data Register Monitor + Control
IEEE1284 Bus
AMBA Bus
Data Registers
Status Register
Control Register

Figure 1-1 IEEE interface block diagram

The negotiation Monitor State Machine monitors IEEE Bus signal events, either generated by the CPU via the AMBA bus or from the peripheral. A response (or lack of a response) from the peripheral to a CPU initiated handshake results in an Interrupt Request being generated.

When the host-peripheral handshake process enters the data transfer phase of one the five IEEE modes of operation, a control register bit is set to enable the Data Transfer State Machine. The Data Transfer State Machine controls and monitors data flow between the IEEE data bus and the AMBA bus. Interrupt Requests are generated when data is received from the peripheral or on completion of a successful forward data transfer.

Inherent within the design is a multi-purpose timer. It provides a safety mechanism to time-out transactions that do not get responses from the IEEE1284 interface, and also enables the state machines to ensure that data setup and hold times are met with respect to the IEEE1284 strobe signals.

The methods of host-peripheral data transfer used in each of the five IEEE modes are different in each case. This, and other aspects of the IEEE modes, are discussed in the following sections.

1.3.2 Communication sequence

Hosts may re-initialize the interface at any time by asserting **nInit** LOW in conjunction with **nSelectIn** LOW. Resetting the interface may also reset the peripheral.

The interface is always initialized to the Compatibility Mode - a conventional, unidirectional host-to-peripheral interface. From the Compatibility Mode, the host may:

- direct the interface to a mode capable of transmitting data from the peripheral to the host.
- transmit data to the peripheral using the Compatibility Mode, or

Hosts and peripherals indicate their readiness to communicate by asserting their Host and Peripheral Logic High signals respectively.

An IEEE1284 compliant host negotiates with the peripheral to determine whether or not the peripheral is IEEE1284 compliant and if so, the host will request a communication mode for the peripheral, otherwise the host will remove its stimulus.

The compliant IEEE1284 device will acknowledge the communication request based on its capabilities and execute or reject the request as appropriate. At the discretion of the host, the interface can be returned to Compatibility Mode at any time.

If the request fails, the host will return the link to the Compatibility Mode and renegotiate for another transfer mode. If successful, there are several possible outcomes of the negotiation sequence, depending on the mode requested, the response of the peripheral, and subsequent host action.

If Nibble or Byte Mode was requested, the peripheral responds correctly and peripheral-to-host data is available, the host may:

- proceed with a reverse channel data transfer using the mode requested by the host
- remain in the host busy, data available phase
- terminate and return to Compatibility Mode.

If Nibble or Byte Mode was requested and the peripheral responds correctly, but no peripheral-to-host data is available, the host may:

- set **nAutoFd** LOW, which puts the interface into idle phase
- terminate and return to Compatibility Mode.

If ECP Mode was requested and the peripheral responds correctly, the link, will enter the ECP Mode forward idle phase, following a setup phase. From this state:

- the host can request to send data to the peripheral
- the peripheral can request to send data to the host
- the host can terminate the ECP Mode and return to Compatibility Mode.

If EPP Mode was requested and the peripheral responds correctly, the link will enter the EPP Mode idle phase. From this phase the host can:

- write data to the peripheral
- read data from the peripheral
- write an address value to the peripheral
- read an address value from the peripheral
- terminate EPP Mode and return to Compatibility Mode.

This initiates one of two types of termination:

- to terminate either the Nibble, Byte or ECP IEEE1284 Modes, the host sets **nSelectIn** LOW and **nAutoFd** HIGH if not set already, which will initiate one of two types of termination:
    - — a handshake, which allows the peripheral to tell the host when it has returned to Compatibility Mode.
    - — an immediate abort, with no guarantee of interface integrity. If the peripheral was in a valid state, which is any state where a reverse data transfer is not in progress, the peripheral will perform the handshake. Otherwise the peripheral will abort immediately and the current byte in transit is lost, however the byte remains in the peripheral output buffer and will be transmitted the next time the host performs a reverse channel transfer.
- to terminate EPP Mode, the host asserts the nInit signal and the peripheral will reset to its initial Compatibility Mode.

1.3.3 Compatibility mode

This mode provides a byte-wide forward channel with data, clock and status lines to the peripheral device. This mode is defined by the IEEE1284 specification to be the default for the IEEE port, and all other modes (Nibble, Byte, ECP, EPP) revert to Compatibility Mode after host termination of the other modes.

This mode is supported through the Data Transfer State Machine built into the IEEE Interface with a minimal amount of CPU intervention required. The state machine controls all outputs, and responds to inputs from the peripheral device, asserting the Interrupt request line when successful data transfer has been completed.

Data access can be programmed to be through single register access or FIFO interface from the AMBA bus.

There are four modes defined as capable of providing peripheral-to-host data transfers:

- *Nibble mode*
- *Byte mode*
- *ECP mode* on page 1-13
- *EPP mode* on page 1-13.

Nibble mode

This mode provides a reverse data channel under control of the host where data bytes are transmitted as two sequential 4-bit nibbles across unused control lines. The mode is entered through a negotiation sequence with the peripheral, and a further handshaking sequence is required before data is transferred.

All negotiation and handshaking sequences are under control of the CPU through the use of a writeable Control Register. Peripheral responses are monitored and an Interrupt Request signal asserted when control lines from the peripheral change state. A readable Status register is provided so that the CPU can determine the source of the Interrupt and monitor the control lines from the peripheral.

A control register bit is used to indicate that the data transfer phase has been entered and is used to activate a state machine to affect the data transfer. When the data byte from the peripheral has been read, an interrupt request is generated and the process returns to the handshake phase.Data access can be programmed to be through single register access or FIFO interface from the AMBA bus.

Byte mode

This mode is identical to Nibble Mode except that reverse data is transferred byte-wide across the bi-directional data lines provided by the IEEE port.

ECP mode

This mode provides a byte-wide bi-directional channel controlled by an interlocked handshake between host and peripheral. Data transfers in this mode can be optionally Run Length Encoded (RLE), with a separate control line between host and peripheral indicating whether a data byte is a command word or data word.

The mode is entered through a negotiation sequence with the peripheral which is under control of the CPU through the use of a writeable Control Register. Peripheral responses are monitored and an Interrupt Request signal asserted when control lines from the peripheral change state. A readable status register is provided so that the CPU can determine the source of the Interrupt and monitor the control lines from the peripheral.

A control register bit is used to indicate that the data transfer phase has been entered and is used to activate a state machine to affect data transfers. Another control bit is used to indicate the direction of data flow. Data access can be programmed to be through single register or FIFO interface from the AMBA bus.

ECP Mode has a ninth bit which may be used to tag the accompanying byte as being data or a command. For example, if Run Length Encoding (RLE) is used, it may be used to tag the byte as being a channel address, or as a run length of consecutive data bytes of the same value.

This mode provides a byte-wide bi-directional channel controlled by the host through the use of separate time-multiplexed address and data cycles over the eight data lines of the IEEE interface.

EPP mode

The mode is entered through a negotiation sequence with the peripheral which is under control of the CPU through the use of a writeable Control Register. Peripheral responses are monitored and an Interrupt Request signal asserted when control lines from the peripheral change state. A readable Status register is provided so that the CPU can determine the source of the Interrupt and monitor the control lines from the peripheral.

A control register bit is used to indicate that the EPP Idle phase has been entered and is used to activate state machines to perform one of the four following IEEE1284 transfers:

- Data Write
- Data Read
- Address Write
- Address Read.

Address transfers can be controlled by reading/writing to an address location.

Address read and write operations performed on the ASB registers cause an Interrupt request to be generated if successful. This method of data transfer has been adopted due to the relatively slow peripheral response time (10us) allowed by the IEEE specification, and saves tying up the AMBA bus for long periods of time.

1.3.4 FIFO transfers

Data access through FIFO is supported in all the data transfer modes supported by the IEEE1284 Interface. The FIFO interface block provides FIFO access in both forward and reverse directions. The forward direction uses a 16 word FIFO. Data throughput in the forward direction is enabled by the CPU setting a bit of a programmable register resulting in the generation of Interrupt Requests whenever the fill of the FIFO is less than 9 words or 16 words (programmable).

The reverse direction also uses a 16 word FIFO. Reverse data from the peripheral is written to the FIFO and an Interrupt Request is made when an 8 or 1 word fill threshold (programmable) is reached. At the end of a data block transfer the IEEE1284 interface is placed into Compatibility Mode which causes the last data word of the IEEE transfer to be written to the FIFO.

1.4 Programmer's model

Table 1-8 shows the registers, along with their offset addresses and widths, used by the IEEE1284 Interface.

The Reserved bits of registers should be written to as 0 and read as "Don't Care" values. The physical implementation may not be 32 bits wide.

Table 1-8 IEEE interface register summary

Address offset	Type	Width	Name	Description
`0x00`	R/W	8	Config	IEEE mode selection and programmable attributes
`0x04`	R/W	4	Control	Controls the states of IEEE port control outputs
`0x40–0x7C`	R/W	32	Data	Forward (host-to-peripheral) transfer data register, 16 locations allocated for burst mode support
`0x08`	R/W	8	Addr	Forward (host-to-peripheral) transfer address register
`0x0C`	R	9	Status	Port input/output signal status register
`0x10`	R/W[a]	9	IntStatus	Port interrupts status register
`0x14`	R/W	2	FifoLevels	Programs the RX & TX FIFO interrupt generation levels
`0x18`	R/W	22	InitTime	Forward (host-to-peripheral) timeout counter initial value
`0x1C`	R	22	TimerStatus	Forward (host-to-peripheral) timeout counter current value
`0x20`	W	0	FifoReset	A write to this location will reset the forward transfer FIFO

a. A write of a logic 1 to any bit position will clear the interrupt. A write of logic 0 will leave it unchanged.

1.4.1 IEEE Config Register

This is an 8-bit register containing programmable bits relating to configuration of the IEEE1284 Interface.

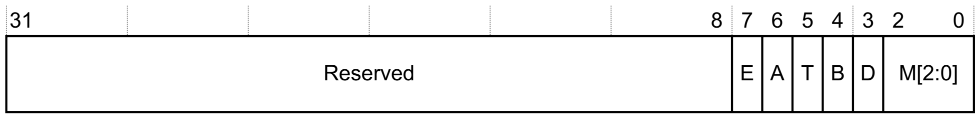

Figure 1-2 IEEE Config Register bit coding

Table 1-9 IEEE Config Register bit functions

Bits	Name	Function
31:8	-	Reserved
7	E	Timer Enable: 0 - Timer Disabled 1 - Timer Enabled
6	A	Data transfer direction: 0 - peripheral-to-host (reverse) 1 - host-to-peripheral
5	T	Data Transfer Enable: 0 - Transfer Disabled 1 - Transfer Enabled
4	B	9 Bit Word Enable: 0 - Disable 9 bit mode 1 - Enable 9 bit mode
3	D	FIFO Access Enable: 0 - Register Access 1 - FIFO Access
2:0	M[2:0]	Mode Select: 1XX - Compatibility Mode 000 - Nibble Mode 001 - Byte Mode 010 - ECP Mode 011 - EPP Mode

1.4.2 Control Register

This is a 4-bit register containing programmable bits that directly control the states of IEEE port control outputs. It should be noted that in some modes, control of some IEEE port outputs is handed over to state machines built in to the IEEE Interface (for example, the **nStrobe** output in Compatibility Mode). When this occurs, the IEEE1284 port output does not necessarily mirror the state defined by this register. The status of IEEE1284 port outputs can be read from the Status register in these cases.

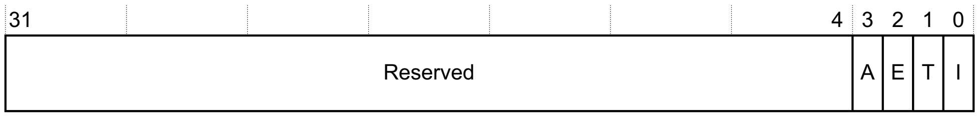

Figure 1-3 Control Register bit coding

Table 1-10 Control Register bit functions

Bits	Name	Function
31:4	-	Reserved
3	A	IEEE1284 nAutoFd Output
2	E	IEEE1284 nSelectin Output
1	T	IEEE1284 nStrobe Output
0	I	IEEE1284 Port nInit Output

1.4.3 Data Register

This is a 32-bit register containing data relating to forward FIFO transfers used for AMBA access.

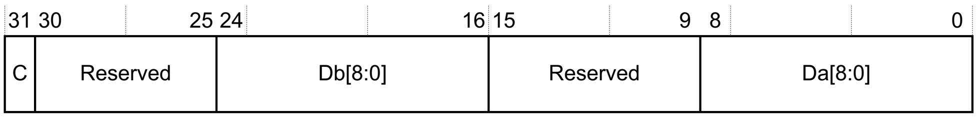

Figure 1-4 Data Register bit coding

Table 1-11 Data Register bit functions

Bits	Name	Function
31	C	Byte Count: 0 - only byte 1 valid 1 - bytes 1 and 2 valid.
30:25	-	Reserved
24:16	Db[8:0]	Data byte 2
8:0	Da[8:0]	Data byte 1

1.4.4 Addr Register

This is an 8-bit register containing forward transfer address to the peripheral device used in EPP mode.

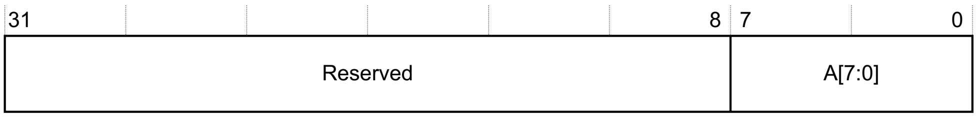

Figure 1-5 Addr Register bit coding

Table 1-12 Addr Register bit functions

Bits	Name	Function
31:8	-	Reserved
7:0	A[7:0]	EPP Mode Forward Address Transfer Byte

1.4.5 Status Register

This is a 9-bit register containing the status of all inputs and outputs of the IEEE1284 port at the time of reading the register.

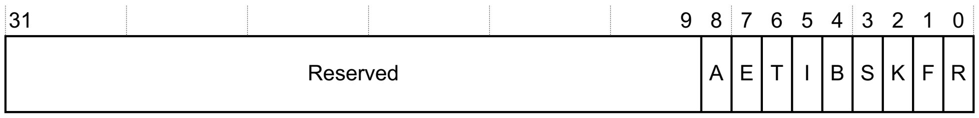

Figure 1-6 Status Register bit coding

Table 1-13 Status Register bit functions

Bits	Name	Function
31:9	-	Reserved
8	A	nAutoFd Port Output status
7	E	nSelectIn Port Output status
6	T	nStrobe Port Output status
5	I	nInit Port Output status
4	B	Busy Port Input status
3	S	Select Port Input status
2	K	nAck Port Input status
1	F	nFault Port Input status
0	R	PError Port Input status

1.4.6 IntStatus Register

This is a 9-bit register containing the status of all inputs and outputs of the IEEE1284 port at the time of reading the register.

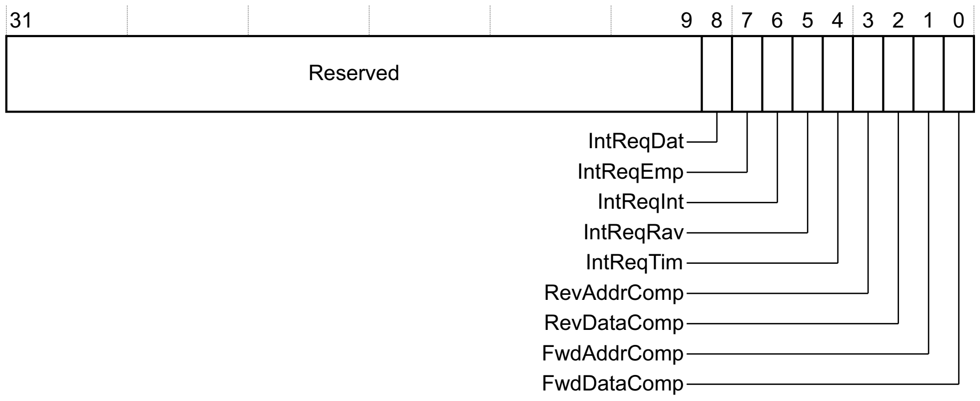

Figure 1-7 IntStatus Register bit coding

Table 1-14 IntStatus Register bit functions

Bits	Name	Function
31:9	-	Reserved
8	IntReqDat	Active HIGH signal, asserted whenever the forward or reverse FIFOs require servicing.
7	IntReqEmp	Active HIGH signal, asserted if the forward interface FIFO is emptied in the course of a data block transfer.
6	IntReqInt	Active HIGH signal, set when the peripheral asserts the interrupt signal in EPP mode.
5	IntReqRav	Active HIGH signal, asserted as a result of a peripheral input control signal changing state.
4	IntReqTim	Active HIGH signal, asserted when the peripheral fails to provide a response to a host-driven handshake.
3	RevAddrComp	Active HIGH signal, asserted on completion of a successful reverse address transfer.
2	RevDataComp	Active HIGH signal, asserted on completion of a successful reverse data transfer.
1	FwdAddrComp	Active HIGH signal, asserted on completion of a successful forward address transfer.
0	FwdDataComp	Active HIGH signal, asserted on completion of a successful forward data transfer.

1.4.7 FifoLevels Register

This is a 2-bit register used to program the RX and TX FIFO interrupt generation levels.

In the reverse direction, when the RevFifoLevel bit is set to 1, the interrupt signal, **IntReqDat**, goes HIGH whenever the RX FIFO contains at least one full location. When the RevFifoLevel bit is set to 0, **IntReqDat** goes HIGH whenever the RX FIFO contains eight or more full locations.

In the forward direction, when the FwdFifoLevel bit is set to 1, the interrupt signal, **IntReqDat**, goes HIGH whenever the TX FIFO has 9 or more empty locations. When the FwdFifoLevel bit is set to 0, **IntReqDat** goes HIGH whenever the TX FIFO has fifteen or more empty locations.

Note

The signal **IntReqEmp** goes HIGH when the TX FIFO has sixteen empty locations.

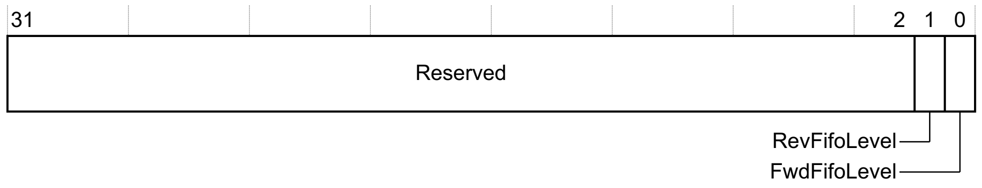

Figure 1-8 FifoLevels Register bit coding

Table 1-15 FifoLevels Register bit functions

Bits	Name	Function
31:2	-	Reserved
1	RevFifoLevel	Reverse FIFO Level: 0 = 8 words 1 = 1 words
0	FwdFifoLevel	Forward FIFO Level: 0 =16 words 1 = 9 words

1.4.8 InitTime Register

This is a 22-bit register and defines the initial timer value following an interrupt.

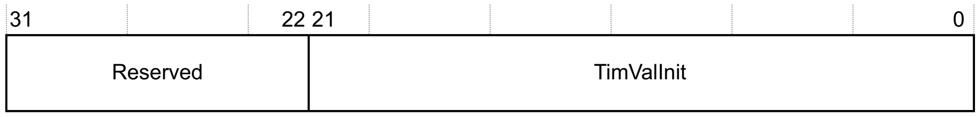

Figure 1-9 InitTime Register bit coding

Table 1-16 InitTime Register bit functions

Bits	Name	Function
31:22	-	Reserved
21:0	TimValInit	Initial Timer Value following an interrupt

1.4.9 TimerStatus Register

This is a 22-bit register and is the current timer value.

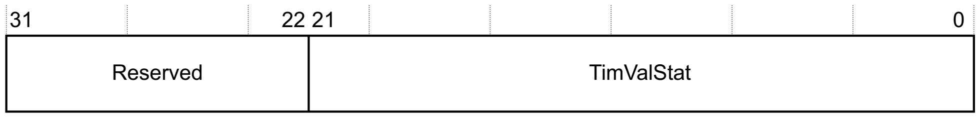

Figure 1-10 TimerStatus Register bit coding

Table 1-17 TimerStatus Register bit functions

Bits	Name	Function
31:22	-	Reserved
21:0	TimValInit	Initial Timer Value following an interrupt

1.4.10 FIFO reset location

A write of any value to this address location will reset the Transmit FIFO.

1.5 Test registers

Additional registers are provided in the IEEE1284 Parallel Port Interface to provide efficient test coverage.

Note

These registers are for production test purposes only and should not be accessed during normal operation.

Table 1-18 provides a summary of their function and corresponding memory map location.

Table 1-18 Test registers

Address offset	Type	Width	Name	Description
`0x24`	R/W	4	TestControl	Enables efficient testing of the timer counter and test clock programmability
`0x28`	R/W	8	TestDataIn	Register containing a value that can be applied to the internal data bus in test mode
`0x2C`	R/W	1	TestDataInEn	When HIGH, the value of TestDataIn register value is applied to the internal databus in testmode.
`0x30`	R/W	5	TestCtrlIn	Register containing signal values that can be used to drive the control signals in test mode.
`0x34`	R/W	1	TestCtrlInEn	When HIGH, the value of the TestCtrlIn register is applied to the internal control signals in testmode.
`0x38`	R	8	TestDataStat	Current IEEE I/O data bus value. Note that this is not a register, it is the value on the peripheral pins for forward transfers and the data register during reverse transfers

1.5.1 TestControl Register

This is a 4-bit register used to control the test features of the peripheral. The large 22-bit timer counter can be partitioned into nibbles and, along with the flexibility of several clocking sources, enables efficient testing of the peripheral.

Table 1-19 TestControl Register bit functions

Bits	Name	Function
31:4	-	Reserved
3	RegClk	Registered Clock Bit Value: This bit is written with LOW/HIGH values to generate a test mode clock.
2:1	Clock Select	00 - External Reference Clock 01 - Bus Clock 10 - Strobe Clock (generated by read of Test Control Register) 11 - Registered Clock Bit
0	TimerTestModeEnable	When HIGH, the 22-bit timer counter is partitioned into a 2-4-4-4-4-4 nibble configuration, allowing the counter to be efficiently tested.

1.5.2 TestDataIn Register

This is an 8-bit register which can be programmed with values to be applied to the internal data bus in test mode. The value of the internal data bus can also be read through this register.

Table 1-20 TestDataIn Register bit functions

Bits	Name	Function
31:8	-	Reserved
7:0	TestDataIn	In test mode, byte-wide data can be written to and read from the data bus through this register.

1.5.3 TestDataInEn Register

This is a 1-bit register used to control whether the internal data bus is driven from the external data bus or by the value contained within the TestDataIn register.

Table 1-21 TestDataInEn Register bit functions

Bits	Name	Function
31:1	-	Reserved
0	TestDataInEn	Test Data In Enable: 0 - External data drives the internal data bus 1 - TestDataIn register value drives the internal data bus

1.5.4 TestCtrlIn Register

This is a 5-bit register whose value can be used to drive the major internal control signals of the peripheral in test mode. The status of the control signals can also be read through this register.

Table 1-22 TestCtrlIn Register bit functions

Bits	Name	Function
31:5	-	Reserved
4	PError	Mode dependent, see *Signal description* on page 1-3.
3	nFault	Mode dependent, see *Signal description* on page 1-3.
2	nAck	Mode dependent, see *Signal description* on page 1-3.
1	PSel	Mode dependent, see *Signal description* on page 1-3.
0	Busy	Mode dependent, see *Signal description* on page 1-3.

1.5.5 TestCtrlInEn Register

This is a 1-bit register used to control whether the major internal control signals are driven from their source or by the value contained within the TestCtrlIn register.

Table 1-23 TestCtrlInEn Register bit functions

Bits	Name	Function
31:1	-	Reserved
0	TestCtrlInEn	Control Value In Enable: 0 - Control bus driven by internal control signals 1 - Control bus driven by TestCtrlIn register value

1.5.6 TestDataStat Location

This is an 8-bit read-only location that can be used to provide the current IEEE I/O data bus value of the peripheral pins. It is the value of the peripheral pins for forward transfers, and the data register value during reverse transfers.