Interfacing to RACEway: PitCREWjr

- Used to interface between FIFOs and the RACEway protocol.
- Drives/receives a RACEway port directly.
- Simple master control, automatic slave response.
- Moves data at 160 MByte/sec peak and 140 MByte/sec sustained throughput.
- Implemented in a Cypress CY7C384A, a 2K gate 100-pin FPGA.

Reference Documents

When using this application note refer to the following documents for more information:

- Cypress CY7C384A and pASIC380™ family data sheets.
- *RACEway Interlink − Data Link and Physical Layers*, VITA 5−1994, available from the VITA Standards organization (VSO)
- Cypress CY7C4245 4K x 18 Synchronous FIFO data sheet

General

PitCREWjr is a simple full-duplex on-ramp to the RACEway fabric. The device has a standard RACEway port and FIFO port. The controller functions either as a RACEway slave, moving data between RACEway and local FIFOs or as a RACEway master, again moving data between RACEway and local FIFOs. It connects to and drives a RACEway interlink port, directly providing all required handshaking and control signaling. PitCREWjr's local FIFO port consists of a 32-bit bidirectional data bus and control signals for moving data between PitCREWjr and industry-standard FIFO components. The data flow between the RACEway and FIFOs is shown in *Figure 1*. The PitCREWjr has no programmable internal registers. Internal PitCREWjr state machines assemble and disassemble the route, address, and data long words embedded in the RACEway protocol. RACEway mastering is accomplished by controlling a single input signal. *Figure 2* shows the block diagram for PitCREWjr and *Table 1* shows the driver and signal name description for each pin on the PitCREWjr controller.

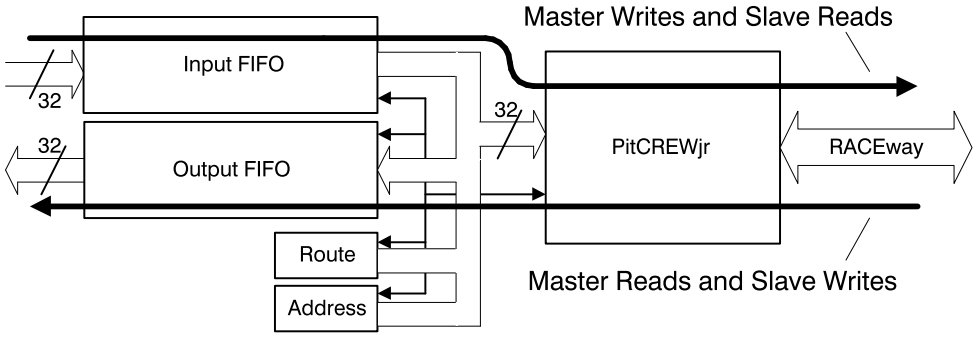

Figure 1. PitCREWjr Data Flow

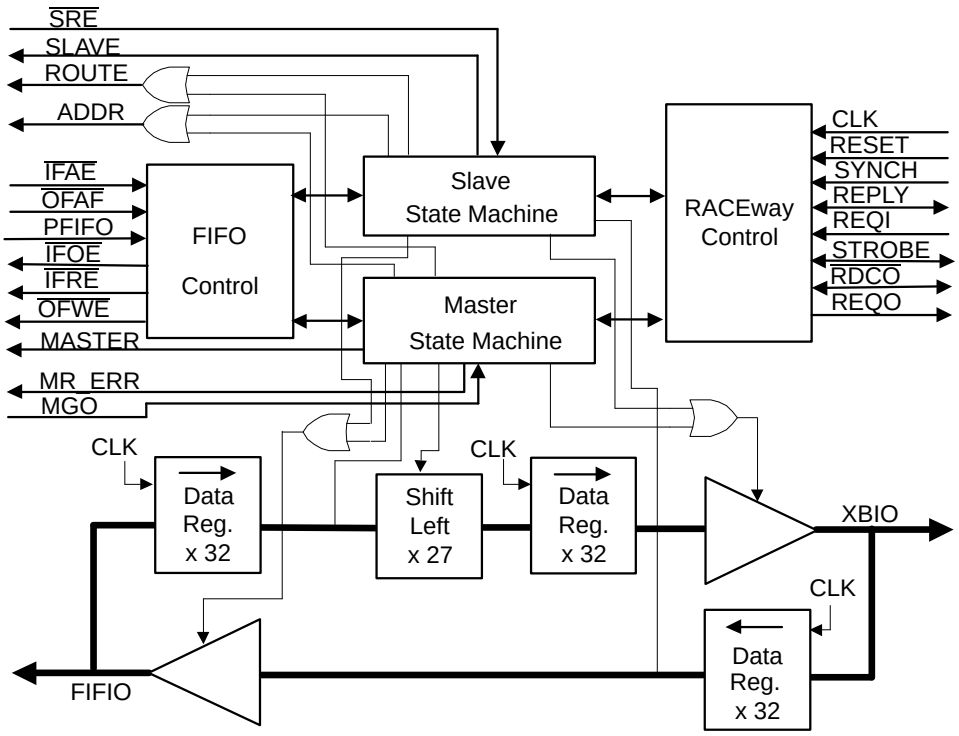

Figure 2. PitCREWjr Block Diagram

Table 1. PitCREWjr Interface Signals

Signal	Source	Function
FIFIO[31:0]	PitCREWjr/Input FIFO	FIFO Data Bus
XBIO[31:0]	PitCREWjr/RACEway	RACEway Data Bus
CLK	RACEway	Crossbar clock
RESET	RACEway	Reset from RACEway
SYNC	RACEway	Crossbar Sync − Provides control and phase information
REPLY	PitCREWjr/RACEway	Gives permission to send the address or data over the data bus
REQI	RACEway	Request In indicates the RACEway crossbar is requesting control of the data bus
STROBE	PitCREWjr/RACEway	Strobe indicates address or data is being sent on the data bus.
$\overline{\text{RDCO}}$	PitCREWjr/RACEway	Indicates to the crossbar to three-state the data bus so read data can be driven. It also indicates when a read error has occurred.
REQO	PitCREWjr	Request Out indicates the PitCREWjr is requesting control of the data bus
$\overline{\text{OFAF}}$	Output FIFO	Output FIFO almost full
$\overline{\text{OFWE}}$	PitCREWjr	Output FIFO write enable
PFIFO	User Hardware	Program output FIFO almost full flag
$\overline{\text{IFAE}}$	Input FIFO	Input FIFO almost empty
$\overline{\text{IFOE}}$	PitCREWjr	Input FIFO output enable
$\overline{\text{IFRE}}$	PitCREWjr	Input FIFO read enable
COUNT	PitCREWjr	Byte counter for master transfers
MR_ERR	PitCREWjr	Error occurred on a master read
MGO	User Hardware	Master GO − starts master state machine
SLAVE	PitCREWjr	Slave transaction in progress
$\overline{\text{SRE}}$	User Hardware	Slave read enable
ROUTE	PitCREWjr	PitCREWjr expecting route to be placed in FIFO data bus
ADDR	PitCREWjr	PitCREWjr expecting address to be placed on FIFO data bus
MASTER	PitCREWjr	Master transaction in progress

FIFOs

The timing generated by PitCREWjr is designed to match with CY7C4245 4K x 18 synchronous FIFOs. PitCREWjr signals can be connected directly to data and control signals of these FIFO components as shown in *Figure 3*. The input FIFO PAE flag should be set to 2. The output FIFO PAF flag should be set at least 16 entries from full.

Slave Function

The slave function of PitCREWjr is accessed whenever an incoming RACEway transaction is received on the RACEway port (REQI is asserted to PitCREWjr) During a slave transaction, the PitCREWjr asserts a status output pin called "SLAVE," which indicates that the PitCREWjr slave state machine is active. When a route word is received from the RACEway, it is driven onto the FIFO data bus. A PitCREWjr output called "ROUTE" is asserted for one XCLKI clock to indicate that a valid route word is present. When an address word is received from the RACEway, PitCREWjr drives this address word onto the FIFO data bus. An output called "ADDR" is asserted by PitCREWjr for one XCLKI clock to indicate that a valid address word is present on the FIFO data bus. PitCREWjr then acknowledges the RACEway with "REPLYIO."

The RACEway protocol communicates data direction in bit 1 of the address word. PitCREWjr's slave state machine branches on this bit value. If the direction of the data is from the RACEway to the local FIFO, the transaction is a slave write (bit 1 of address word is false). As data arrives from the RACEway, it is registered and driven onto the FIFO data bus. (See *Figure 2*.) The PitCREWjr writes the data received from the RACEway to the output FIFO by asserting "$\overline{\text{OFWE}}$" each time a valid word is ready on the FIFO data bus. A PitCREWjr input called "$\overline{\text{OFAF}}$" is used to indicate to PitCREWjr that the output FIFO is full. Assertion of "$\overline{\text{OFAF}}$" causes PitCREWjr to send a kill request to the RACEway master, effectively ending the RACEway transaction. "$\overline{\text{OFAF}}$" would typically be connected to the output FIFO programmable almost full flag. On completion of the RACEway data transfer, PitCREWjr three-states the FIFO data bus and deasserts the "SLAVE" status output.

If the direction of the data is from the input FIFO to the RACEway (a slave read, bit 1 of address word is true), then the FIFO data bus is three-stated by PitCREWjr and PitCREWjr asserts the signal "$\overline{\text{IFRE}}$" and then "$\overline{\text{IFOE}}$" to enable data from the input FIFO onto the FIFO data bus. PitCREWjr asserts this signal pair each time a new word is required from the FIFO. If the input FIFO becomes empty, as signaled by the "$\overline{\text{IFAE}}$" PitCREWjr input, PitCREWjr stops reading the input FIFO for the balance of that transaction and issues an error signal to the RACEway master on completion of the transaction. The kill request is also sent in this case, so that the master ends the transaction soon after the

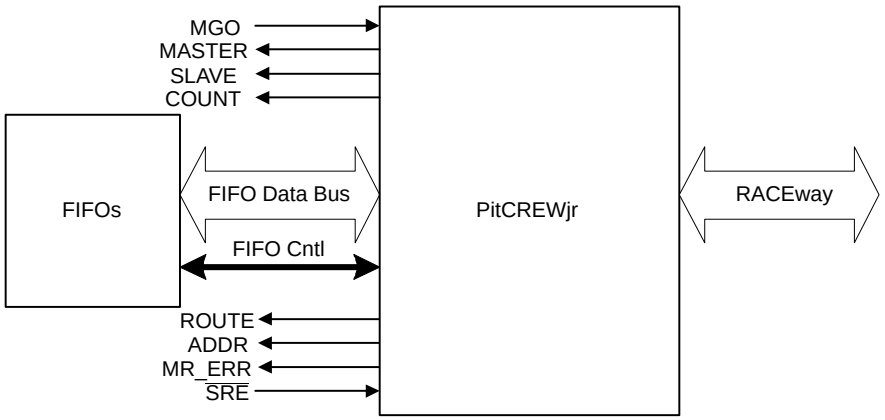

Figure 3. PitCREWjr Signals

underflow. On completion of the RACEway data transfer, PitCREWjr deasserts the "SLAVE" status output.

The intent of the "SLAVE" pin is to indicate a slave transaction in progress. It can be used to tag incoming data, select a data destination, or as a board logic control input.

Note that PitCREWjr will NOT cause route and address header words received from the RACEway to be written to the output FIFO. External logic would be required to place address and/or route words in the output FIFO.

Master Function

The master function of PitCREWjr is accessed whenever the "MGO" PitCREWjr input is asserted. The assertion of "MGO" launches the PitCREWjr master state machine. This state machine is clocked by the RACEway data clock "XCLKI". Two clocks after "MGO" is sampled asserted, PitCREWjr asserts its "ROUTE" output. Local board hardware should use "ROUTE" to enable a route word onto the FIFO data bus. PitCREWjr asserts its "MASTER" output when it drives this route word onto the RACEway and then drives the "shifted route" prescribed by the RACEway protocol. "MGO" should be deasserted once PitCREWjr's "MASTER" output is true. This is because "MGO" will cause a slave in progress to issue a kill over the RACEway. When "change to address" reply is received from the RACEway, "ROUTE" is deasserted, and one clock later "ADDR" is asserted. Local board hardware should use "ADDR" to enable an address word onto the FIFO data bus. PitCREWjr relays the address word to the RACEway and waits for a "$\overline{\text{DSE}}$" reply from the RACEway. When the reply is received, PitCREWjr deasserts the "ADDR" signal.

The RACEway protocol communicates data direction in bit 1 of the address word. PitCREWjr's master state machine branches on this bit value. If the direction of the data is from the local FIFO to the RACEway (a master write, bit 1 of address word is false), then data is read from the local input FIFO, registered inside the PitCREWjr, and driven onto the RACEway XBIO bus. The PitCREWjr FIFO data bus pins remain three-stated and PitCREWjr asserts the signals "$\overline{\text{IFRE}}$" and "$\overline{\text{IFOE}}$" to enable the input FIFO data onto the FIFO data bus. PitCREWjr asserts this signal pair each time a new word is required from the FIFO. If the input FIFO becomes empty, as signaled by the "$\overline{\text{IFAE}}$" PitCREWjr input, PitCREWjr stops reading the input FIFO and ends the RACEway transaction.

If the direction of data is from the RACEway to the local FIFO (a master read, bit 1 of address word is true), then as data arrives from the RACEway, it is registered inside the PitCREWjr and driven onto the FIFO data bus. The PitCREWjr writes the data received from the RACEway to the output FIFO by asserting "OFWEN" each time a valid word is ready on the FIFO data bus. A PitCREWjr input called "$\overline{\text{OFAF}}$" is used to indicate to PitCREWjr that the output FIFO is full. Assertion of "$\overline{\text{OFAF}}$" causes PitCREWjr to suspend transfer requests to the RACEway slave, effectively stalling the RACEway transaction until the signal is deasserted. "$\overline{\text{OFAF}}$" would typically be connected to the output FIFO programmable almost full flag. On completion of the RACEway data transfer as indicated by the deassertion of "MASTER," PitCREWjr three-states the FIFO data bus.

Additional Features

A slave read enable input "$\overline{\text{SRE}}$" is provided to lock out slave access from the RACEway side of the interface. This signal may be used to "protect" data in the input FIFO when that FIFO is being used for both master and slave data. Slave read can be disallowed when data is being queued up in the input FIFO for a master write.

The "MR_ERR" output of the PitCREWjr is an indicator that a master read operation received an error response from its target slave. The signal is a "one-shot", pulsing HIGH for one XCLKI clock period at the end of a master read access for which the RACEway slave signaled a read error.

The "COUNT" output signal strobes each time an 8-byte data beat occurs on the raceway when PitCREWjr is master. For writes, "COUNT" is as-

serted for each 8 bytes sent. For reads, "COUNT" is asserted for each 8 bytes requested.

The "PFIFO" input is used to assist in loading the output FIFO almost full flag. When "PFIFO" is asserted, PitCREWjr three-states its FIFIO data bus drivers, and asserts "$\overline{\text{OFWE}}$." The signal that connects to "PFIFO" can also be used to enable the "almost empty" value onto the FIFIO data bus.

PitCREWjr Operation

Figure 4 illustrates master write behavior. The "MGO" PitCREWjr input is asserted to start RACEway master (read or write) function. It should be deasserted when PitCREWjr asserts "MASTER". Master write is stopped by asserting "$\overline{\text{IFAE}}$" to the PitCREWjr. Notice that two data words are read after "$\overline{\text{IFAE}}$" is asserted. "ROUTE" and "ADDR" are shown enabling route and address information respectively onto the FIFIO data bus from external hardware. The "COUNT" PitCREWjr output pulses once for each 8 bytes sent over the RACEway.

Figure 5 illustrates master read behavior. Data arriving from the RACEway is to be taken from the FIFIO data bus on the rising edge of the RACEway data clock "CLK". Again "COUNT" pulses once for each 8 bytes requested from the RACEway. Master read is stopped by asserting the PitCREWjr input "$\overline{\text{OFAF}}$." Note that eight data values are delivered after "$\overline{\text{OFAF}}$" is signalled. This figure shows the timing when data traverses one RACEway crossbar. Latency will increase by two for each additional crossbar in the data path.

Figures 6 and *7* illustrate slave timing. "ROUTE" and "ADDR" PitCREWjr outputs mark the timing of valid route and address information on the FIFIO data bus. Bit 1 of the RACEway address field is captured by PitCREWjr, causing the appropriate FIFO control signalling for the data direction. For writes, "$\overline{\text{OFWE}}$" is asserted as data is driven by PitCREWjr onto the FIFIO data bus. For reads, "IFOEN" and "$\overline{\text{IFRE}}$" are asserted as shown and data is sampled from the FIFIO data bus on the rising edge of the RACEway data clock, "CLK".

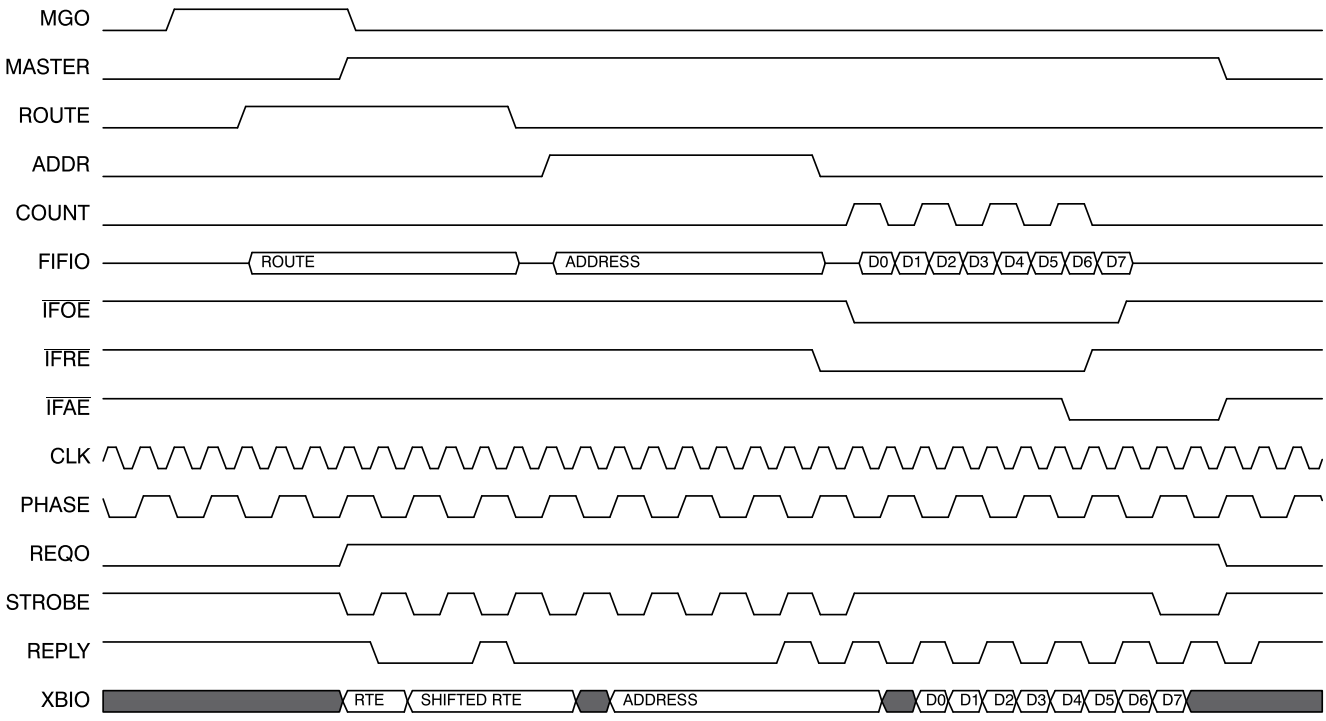

Figure 4. Master Write

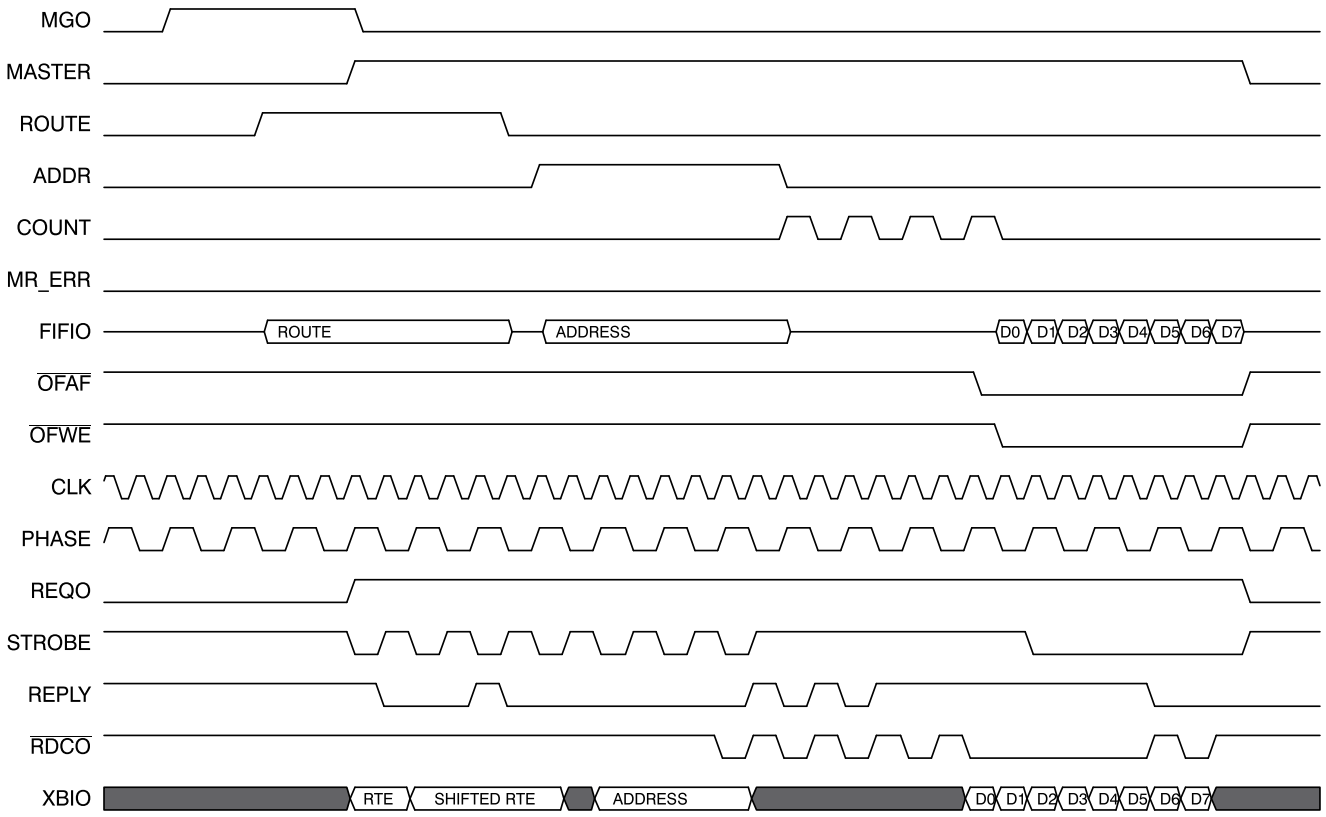

Figure 5. Master Read

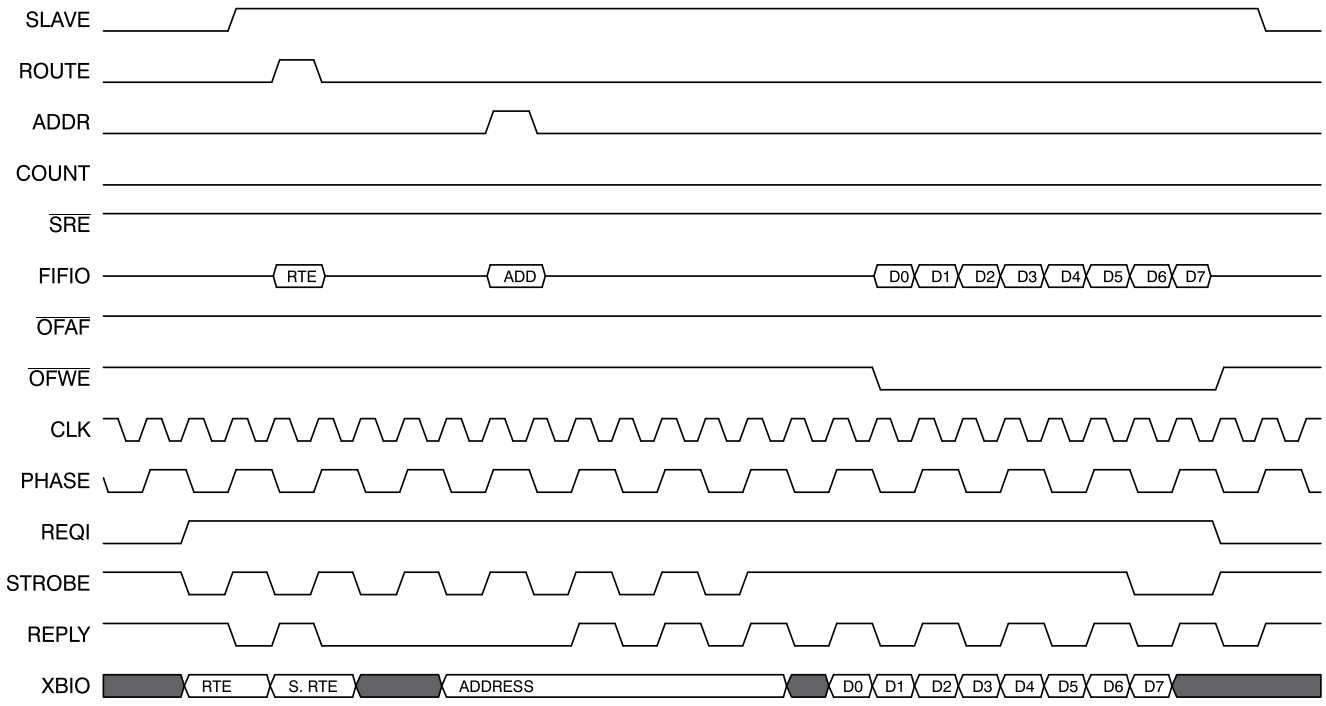

Figure 6. Slave Write

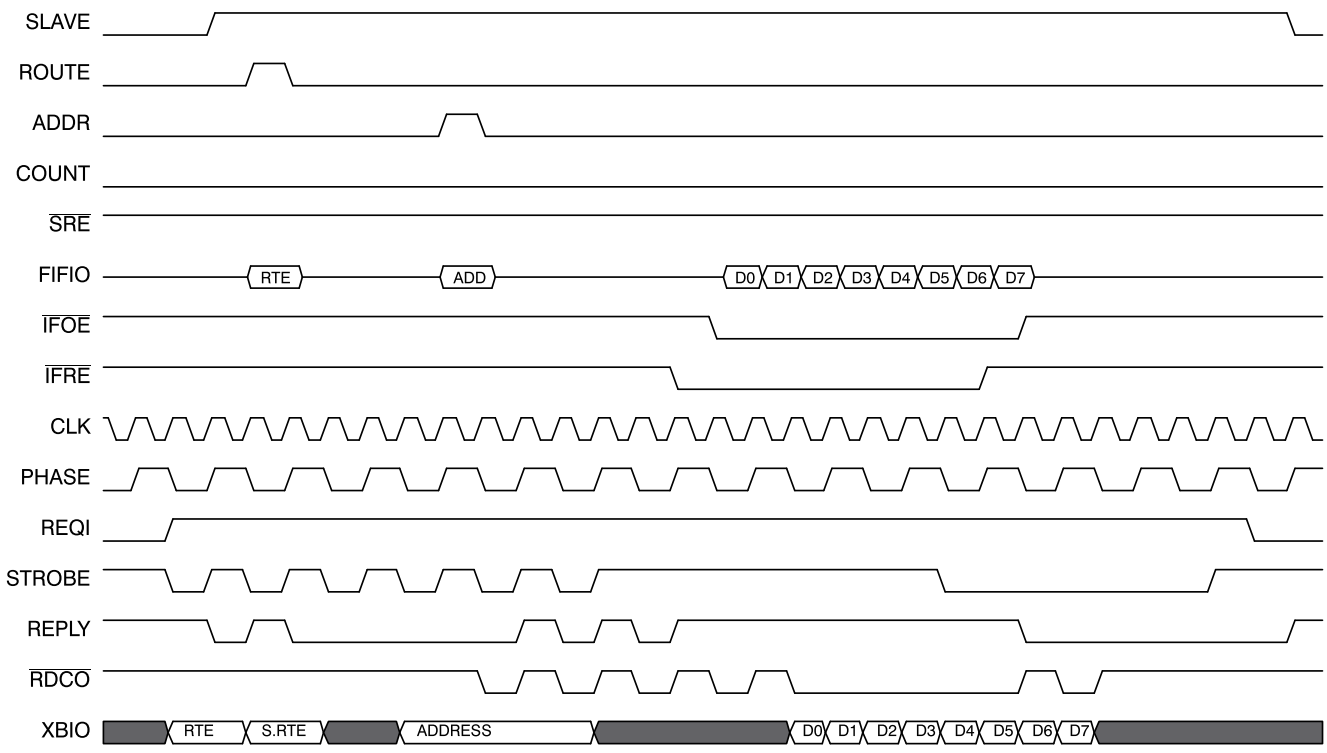

Figure 7. Slave Read

Figure 8 shows a PitCREWjr master writing to a PitCREWjr slave across one RACEway crossbar. The slave signals have an (S) suffix. In this example, the slave PitCREWjr input "$\overline{\text{OFAF}}$" signals that the slave is "almost full". The slave PitCREWjr signals "REQO" (a RACEway protocol kill). This kill propagates through the intervening RACEway crossbar to the PitCREWjr master, terminating the master transaction. The amount of data the slave must absorb after "$\overline{\text{OFAF}}$" is signalled is shown for a single intervening crossbar. Two additional FIFO write cycles will be required for each additional "crossbar hop".

Figure 9 shows the utility of the "$\overline{\text{SRE}}$" PitCREWjr input. It can be used to block PitCREWjr's response to a slave read from the RACEway. This feature allows the input FIFO facility to be multiplexed between master write and slave read without coordinating with the remote master across the RACEway. Master data being queued up in the input FIFO can be "protected" from a slave read operation as shown. The timing of "MGO" assertion with respect to slave arrival from the RACEway is arbitrary. "$\overline{\text{SRE}}$" may be deasserted any time after the assertion of "MASTER" by the PitCREWjr.

Figure 10 shows a PitCREWjr master reading from a PitCREWjr slave across one RACEway crossbar. The slave signals have an (S) suffix. In this example, the slave PitCREWjr input "$\overline{\text{IFAE}}$" signals that the slave is "almost empty." The slave PitCREWjr signals "REQO" (a RACEway protocol kill). This kill propagates through the intervening RACEway crossbar to the PitCREWjr master, terminating the master transaction. The slave PitCREWjr stops reading from its input FIFO two clocks after "$\overline{\text{IFAE}}$" is asserted; however, the RACEway protocol compels the slave to send until the RACEway master stops. By the time the PitCREWjr master responds to the kill, several long words of bad data have been written to the PitCREWjr master's output FIFO. The PitCREWjr output "MR_ERR" pulses HIGH for one data clock to signal that this error has occurred.

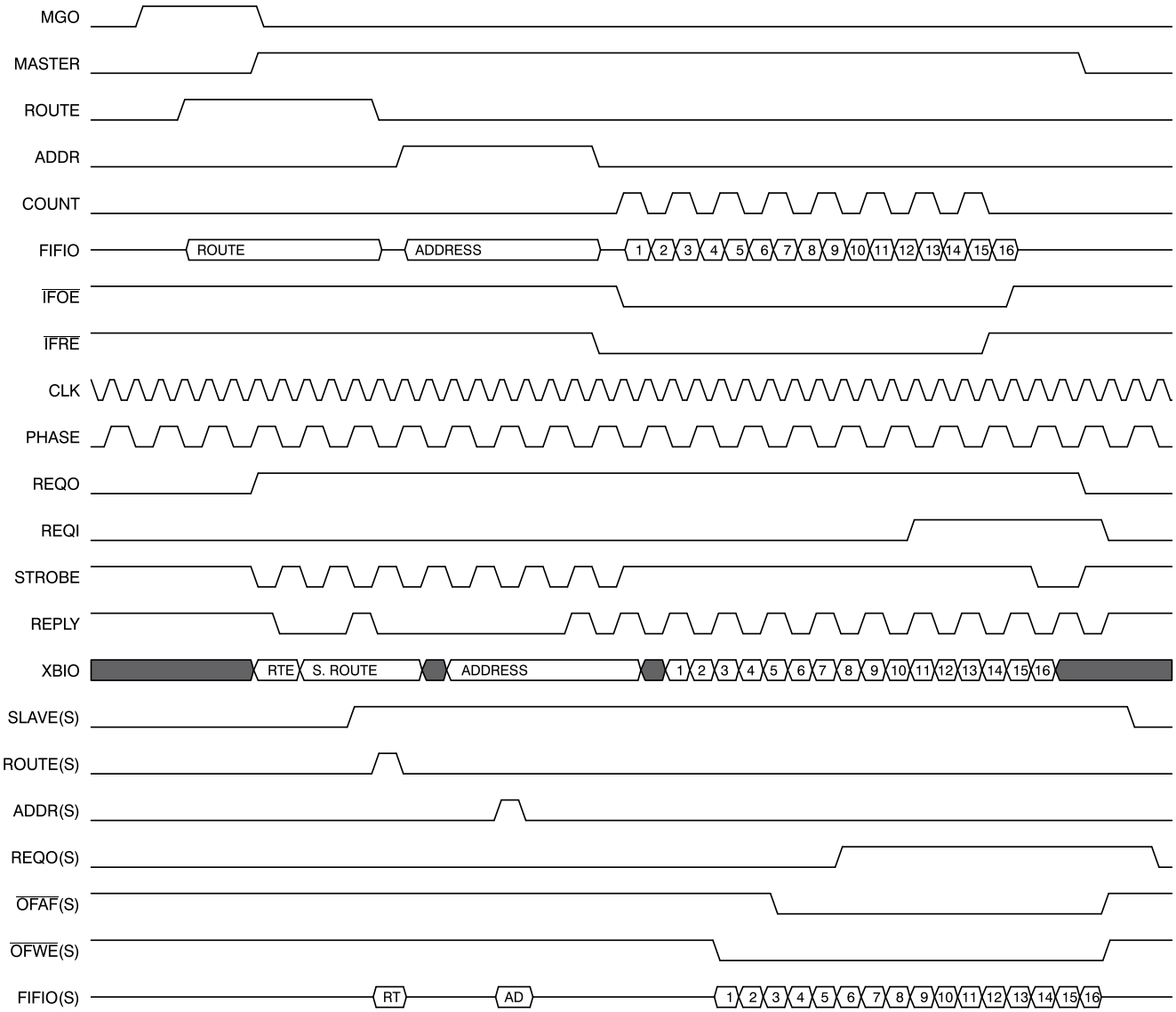

Figure 8. Master Write Overflow

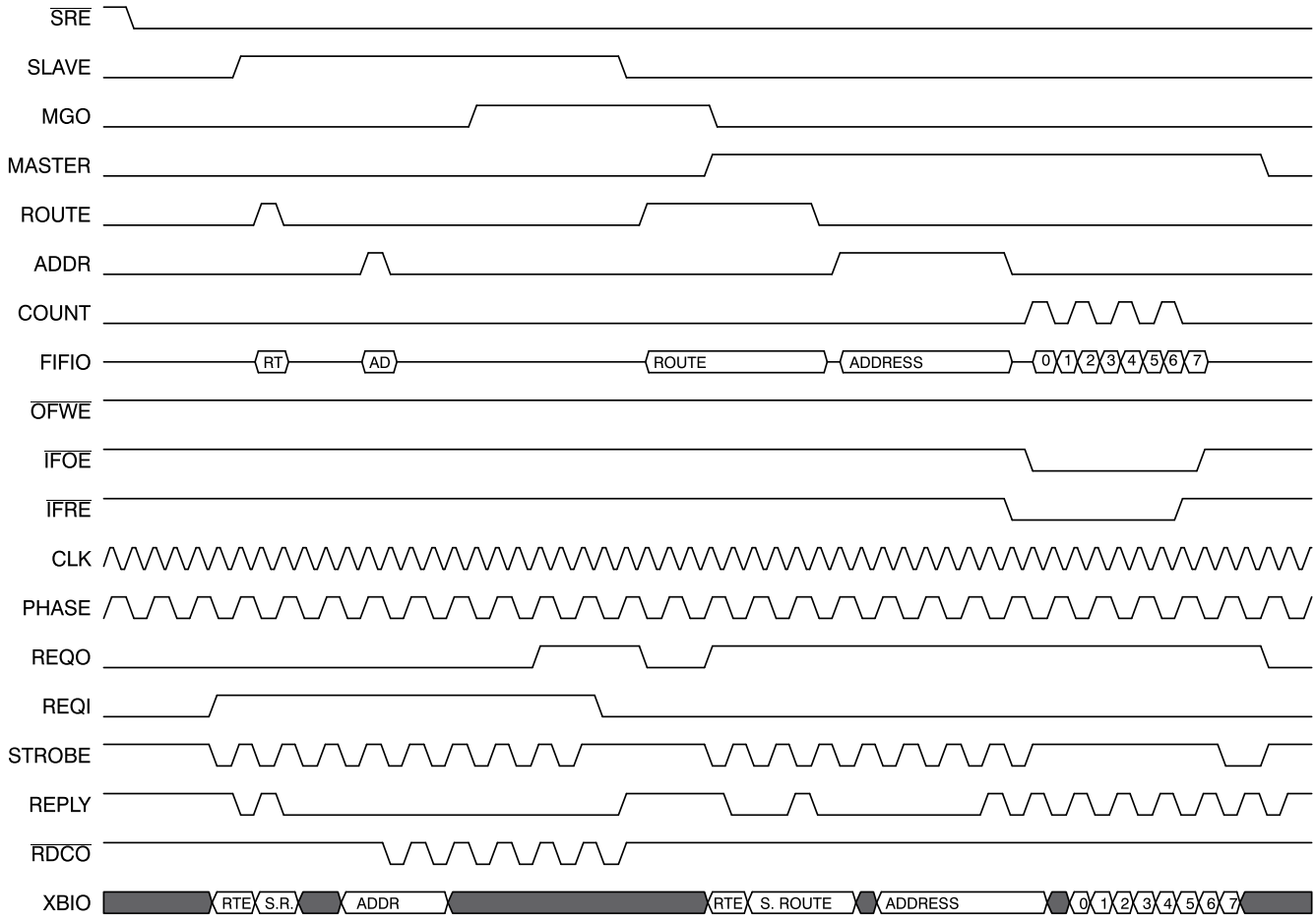

Figure 9. $\overline{\text{SRE}}$ Function

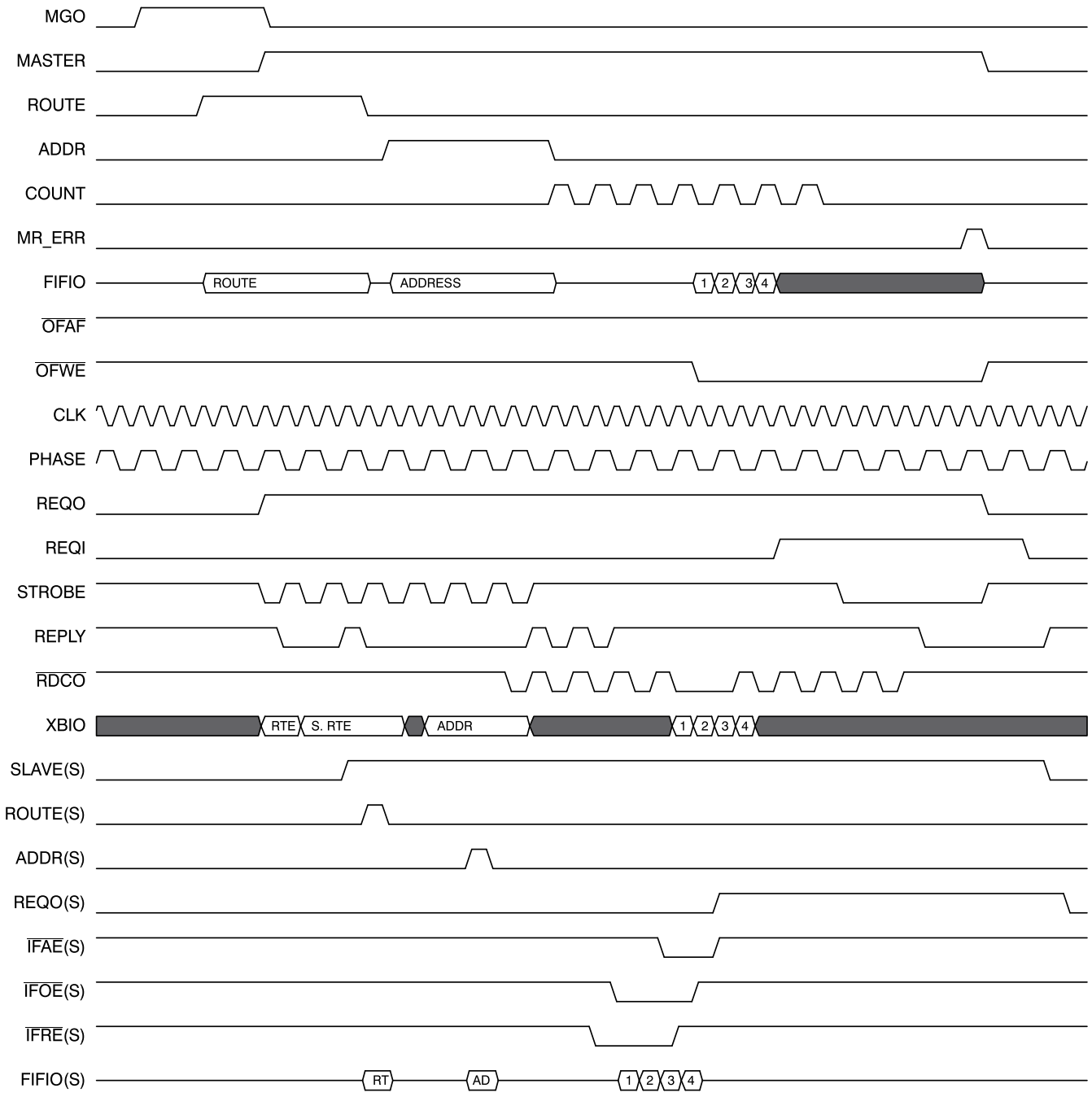

Figure 10. Master Read Error

CY7C384A Pin Table

Signal	Type	Pin No.
FIFIO_16	INOUT	1
XBIO_19	INOUT	2
FIFIO_19	INOUT	3
FIFIO_22	INOUT	4
XBIO_22	INOUT	5
FIFIO_25	INOUT	6
XBIO_25	INOUT	7
FIFIO_24	INOUT	8
VSS	–––––	9
XBIO_24	INOUT	10
MGO	INPUT	11
CLK	INPUT	12
VCC	–––––	13
UNUSED	INPUT	14
OFAF	INPUT	15
VCC	–––––	16
XBIO_23	INOUT	17
FIFIO_23	INOUT	18
FIFIO_26	INOUT	19
XBIO_26	INOUT	20
XBIO_31	INOUT	21
FIFIO_31	INOUT	22
FIFIO_30	INOUT	23
XBIO_30	INOUT	24
FIFIO_27	INOUT	25
XBIO_27	INOUT	26
FIFIO_28	INOUT	27
XBIO_28	INOUT	28
FIFIO_21	INOUT	29
FIFIO_29	INOUT	30
XBIO_29	INOUT	31
XBIO_21	INOUT	32
PFIFO	INOUT	33
$\overline{\text{OFWE}}$	OUTPUT	34
VSS	–––––	35
SYNC	INOUT	36
REQO	OUTPUT	37
VSS	–––––	38
MR_ERR	OUTPUT	39
STROBE	INOUT	40
$\overline{\text{RDCO}}$	INOUT	41
VCC	–––––	42
ROUTE	OUTPUT	43
COUNT	OUTPUT	44
REPLY	INOUT	45
MASTER	OUTPUT	46
SRE	INOUT	47
$\overline{\text{IFRE}}$	OUTPUT	48
$\overline{\text{IFOE}}$	OUTPUT	49
XBIO_2	INOUT	50

Signal	Type	Pin No.
FIFIO_2	INOUT	51
FIFIO_4	INOUT	52
XBIO_4	INOUT	53
XBIO_5	INOUT	54
FIFIO_5	INOUT	55
XBIO_3	INOUT	56
FIFIO_3	INOUT	57
FIFIO_1	INOUT	58
VSS	–––––	59
XBIO_1	INOUT	60
$\overline{\text{IFAE}}$	INPUT	61
RESET	INPUT	62
VCC	–––––	63
REQI	INPUT	64
UNUSED	INPUT	65
VCC	–––––	66
XBIO_7	INOUT	67
FIFIO_7	INOUT	68
XBIO_0	INOUT	69
FIFIO_0	INOUT	70
XBIO_8	INOUT	71
FIFIO_8	INOUT	72
FIFIO_9	INOUT	73
XBIO_9	INOUT	74
FIFIO_6	INOUT	75
XBIO_6	INOUT	76
XBIO_12	INOUT	77
ADDR	OUTPUT	78
XBIO_11	INOUT	79
FIFIO_11	INOUT	80
FIFIO_12	INOUT	81
XBIO_10	INOUT	82
XBIO_13	INOUT	83
FIFIO_10	INOUT	84
VSS	–––––	85
XBIO_15	INOUT	86
FIFIO_13	INOUT	87
VSS	–––––	88
XBIO_14	INOUT	89
FIFIO_15	INOUT	90
SLAVE	OUTPUT	91
VCC	–––––	92
XBIO_18	INOUT	93
FIFIO_14	INOUT	94
FIFIO_18	INOUT	95
XBIO_17	INOUT	96
FIFIO_17	INOUT	97
XBIO_20	INOUT	98
XBIO_16	INOUT	99
FIFIO_20	INOUT	100

pASIC is a trademark of Quicklogic.