Interfacing to RACEway: PitCREW

PitCREW is intended for engineers who are designing an I/O circuit for use as an "on-ramp" to the RACEway switching fabric. This document illustrates a simple but complete FIFO interface to RACEway. This design can be used as described or as the starting point for custom RACEway interface development. This application note describes:

- The design specification for the PitCREW I/O Controller.
- Electrical information for designing a FIFO-based I/O circuit with the PitCREW Controller.

Reference Documents

Use this application note in conjunction with the latest Cypress data books and data sheets and related published standards documents. These resources are as follows:

- Cypress CY7C387P and pASIC380™ Family data sheets
- *Cypress Programmable Logic Data Book 1994/1995*. For more information on using pASIC380 Family devices, see the *Cypress Applications Handbook*
- *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
- *The VMEbus Specification, VITA 1–1994*
- Cypress CY74FCT162H501T data sheet
- Cypress CY7B9910 Low Skew Clock Buffer data sheet
- Front Panel Data Port Electrical and Physical Layers VITA 17 – 199x

RACEway On-Ramp System Overview

In general, this on-ramp is an I/O data port for a RACEway fabric. It defines a simple FIFO interface which is a slave to its RACEway port. Transactions cannot be initiated via the FIFO interface. Instead, the on-ramp has a DMA engine that moves blocks of data between RACEway nodes and its I/O port. This DMA engine is set in motion by commands received over the RACEway fabric. Data move instructions can be issued directly to the RACEway port, or placed in the memory of another RACEway node in the form of a linked list. This on-ramp should be considered a slave board whose function is controlled from a program executing on one or more RACEway nodes.

The on-ramp is comprised of the PitCREW I/O controller, an input FIFO, an output FIFO, and a bi-directional transceiver with synchronizing latches. *Figure 1* outlines the major components of the on-ramp.

The PitCREW Controller is implemented in a Cypress CY7C387P FPGA. All the logic required to control data movement between the FIFOs and the RACEway fabric resides in this device. PitCREW drives the RACEway fabric directly and implements the features described in the remaining sections. The architecture of a sample interface using PitCREW is shown in *Figure 2.*

Each FIFO is implemented with a pair of CY7C4245 4K x 18 Synchronous FIFOs. The trans-

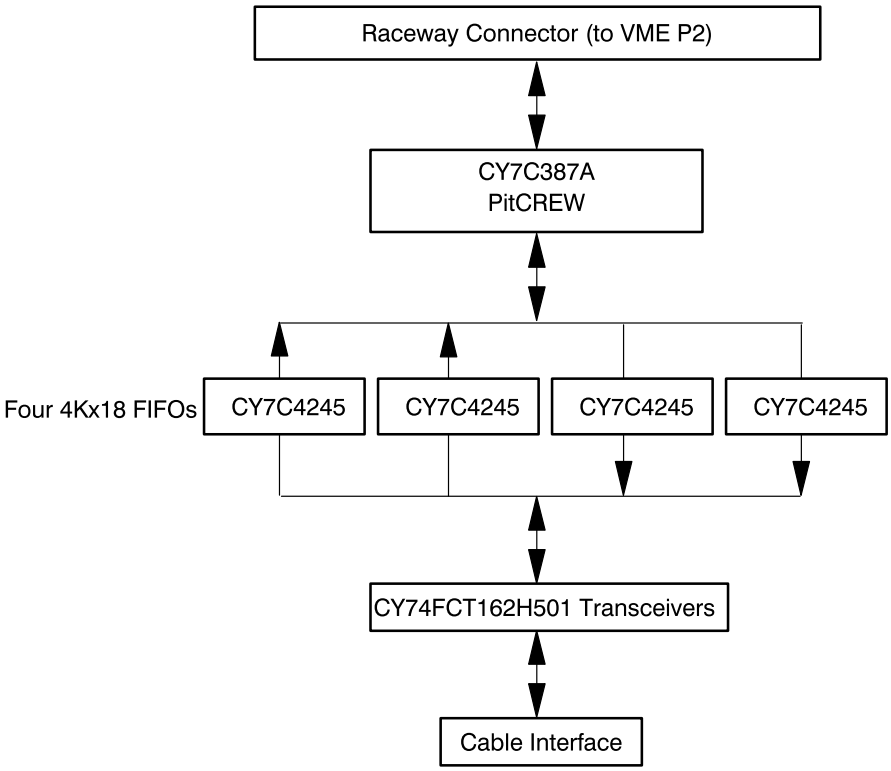


Figure 1. Components of a Sample I/O Interface

ceiver function is handled by a pair of CY74FCT162H501 registered transceivers. The FPGA and the FIFOs are available in 0.5-mm lead pitch TQFP packages (144-lead and 64-lead, respectively). The transceivers are available in a 56-lead SOIC pack.

Features

The on-ramp allows for autonomous DMA transfer through asynchronous data FIFOs. Transfers can be from RACEway to FIFO, FIFO to RACEway, or both (full duplex on the user side of the FIFOs, half duplex over RACEway). Features of the on-ramp circuit include:

- A DMA engine capable of routing a data stream between an external device and any node in the RACEway fabric.
- 160 MB/sec peak and 140 MB/sec sustained throughput.
- A 40-MHz, 32-bit cable interface, compatible with the Front Panel Data Port (FPDP) Standard.
- Flow control, synchronization signals, and user programmable bits available over the cable interface.
- Ability to write status to a RACEway memory location (for local polling) or to a mailbox location (to cause an interrupt).
- Optimal use of the crossbar network bandwidth by automatically buffering blocks of data for burst crossbar transfers at 160 MB/sec.
- Ability to act as a RACEway slave so a RACEway node anywhere on the network fabric can set up, control, or test the operation of the board.

Operation

The PitCREW Controller provides DMA operations on the RACEway Interlink, interfacing either an input FIFO, an output FIFO, or both to the

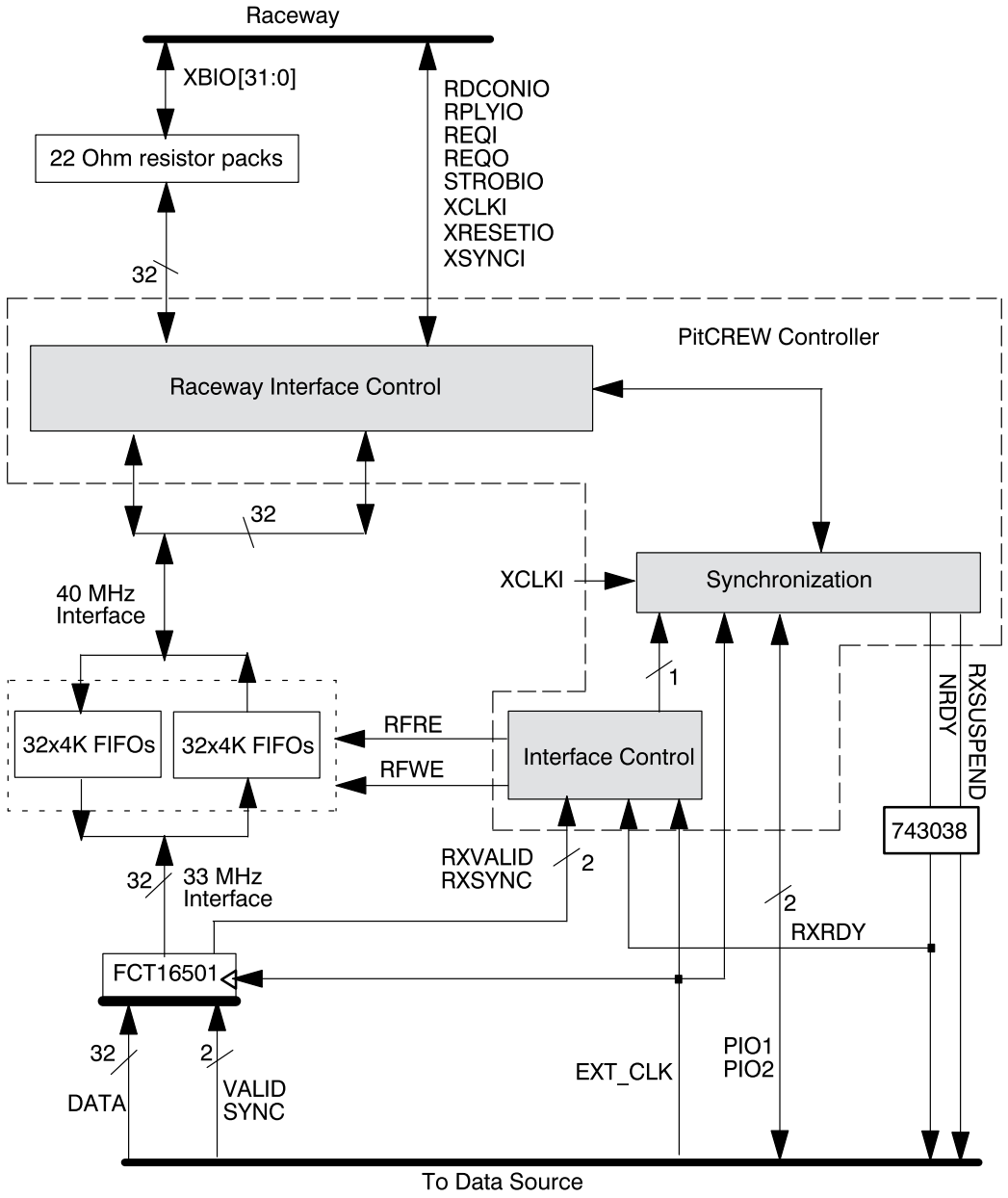


Figure 2. Architecture of a Sample Input Interface Using PitCREW

RACEway. Control signals are provided for the user side of the FIFOs, which can run asynchronously to the RACEway.

PitCREW always functions as a transaction master on the RACEway when it is moving data, and bursts at the full 160 megabyte per second rate. It can be operated in linked-list fashion, fetching a new command packet from the RACEway at the completion of the current one. Each command packet consists of a word count, the new contents of the Control Register, the data route and address, and the next command packet route and address.

The linked list of command packets is built in memory accessible over the RACEway fabric. The DMA engine is started by a RACEway master writing a load and go operation specifying the route and address of the first packet directly into the PitCREW Controller. The Controller then fetches and

executes from the linked list until a command packet is fetched with the GO bit reset. The linked-list structure is shown in *Figure 3*.

A simpler control alternative is to write the "Data Address," "Data Route," and "Word Count" registers each time a DMA transfer is desired. Writing the "Word Count" register will cause a DMA transfer to start.

For reads from the cable interface, as shown in *Figure 4*, the controller counts valid words as they are placed into the input FIFO. When the counter

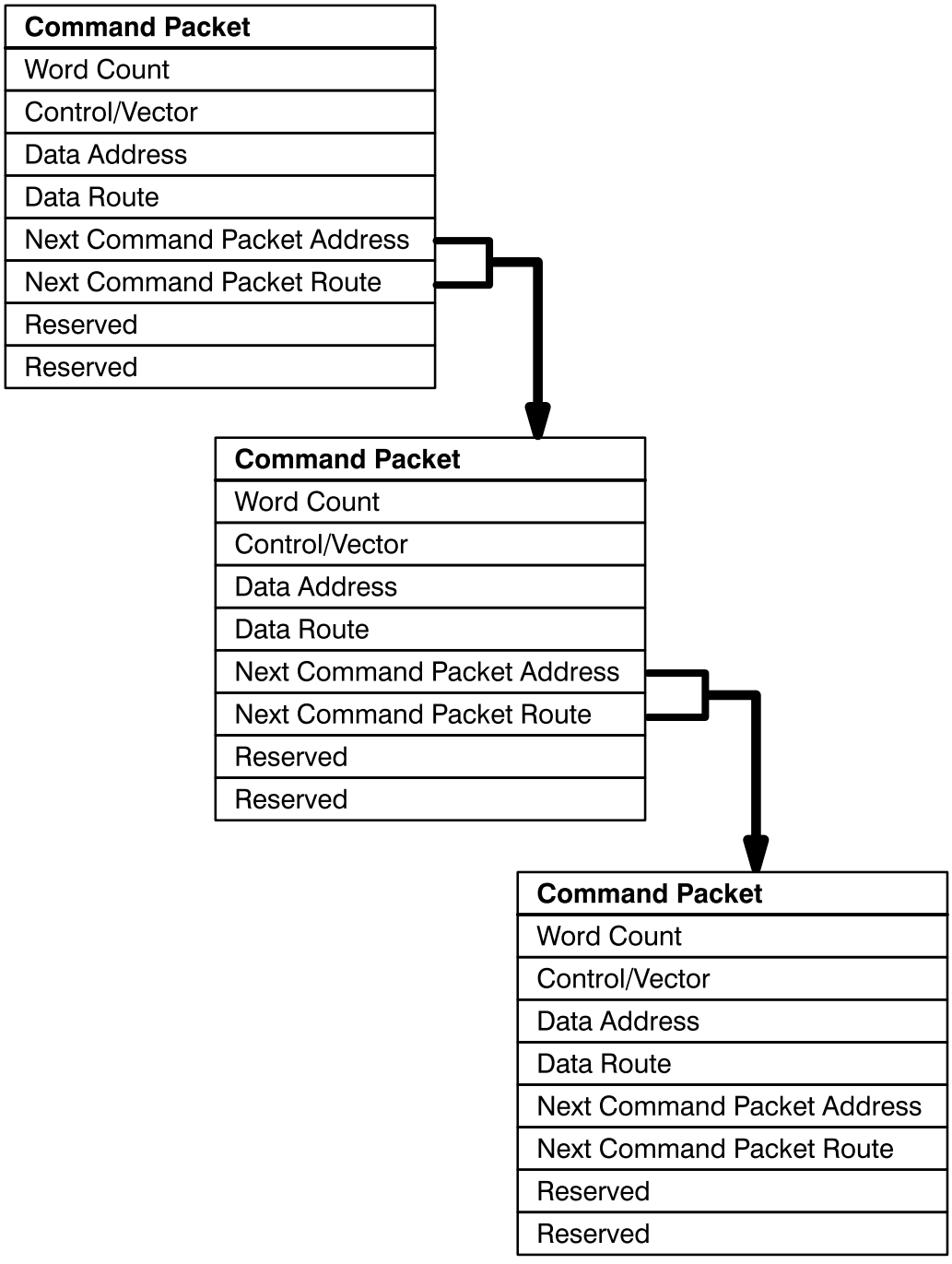


Figure 3. Linked List Operation

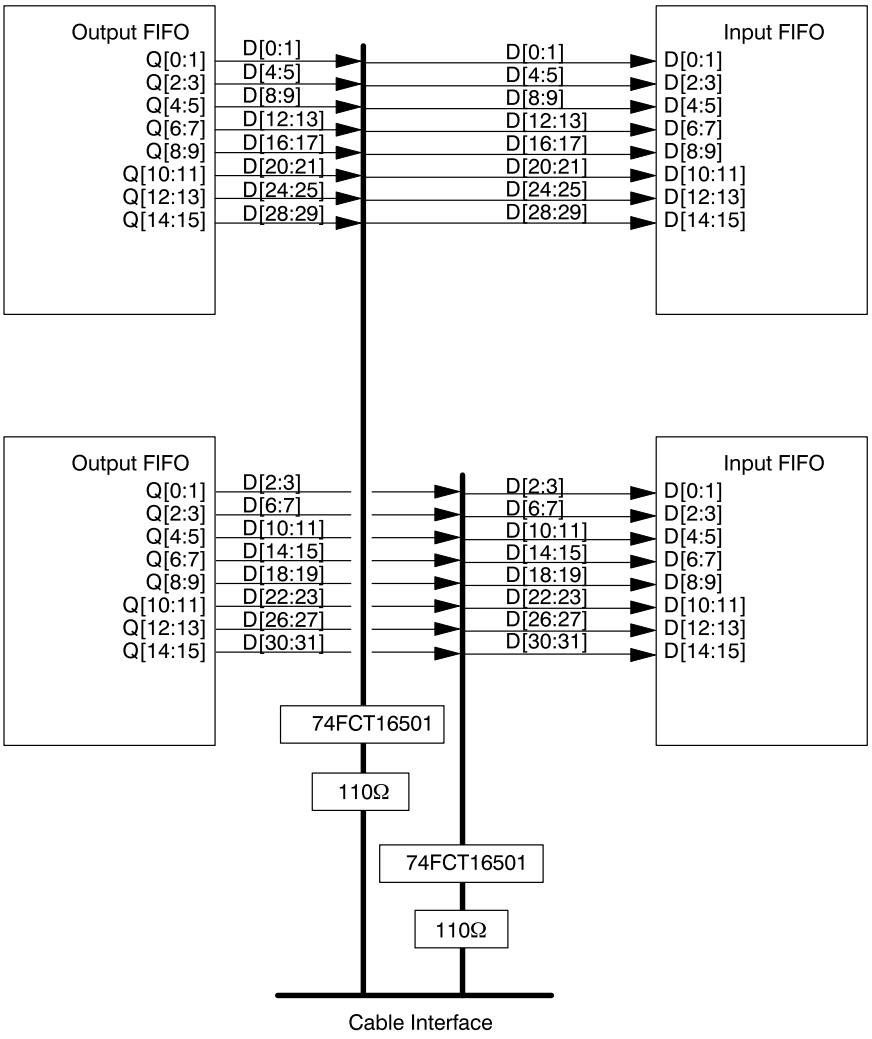


Figure 4. Example—Connecting the FIFOs to a Cable Interface

reaches 2K bytes, data is read from the input FIFO by PitCREW and written to the RACEway as a burst operation. The controller accepts a "data valid" input ($\overline{\text{RXVALID}}$) for qualifying input FIFO loading, as well as a sync input pin ($\overline{\text{RXSYNC}}$) allowing for an external event to start the acquisition.

For writes to the cable interface, a "suspend" signal ($\overline{\text{TXSUSPEND}}$) is provided for throttling the read operation of the cable side of the output FIFOs. When the Programmable Almost Full pin ($\overline{\text{TFPAF}}$) on the output FIFO indicates to PitCREW that there is room in the FIFO, a burst operation transfers data from the RACEway to the output FIFO to fill it up. PitCREW provides the output FIFO interface signals, as well as the ability of placing a sync marker (SET_SYNC) in the output FIFO for framing the data.

Two user-programmable I/O bits (PIO[2:1]), are available for data tagging or other application-specific purposes. These bits may be individually programmed via the PitCREW Control Register to be either inputs or outputs. These bits may be used to tag command packets as they are executed. For example, headers and data may be assigned different tags.

It is possible to perform a Status Write operation in which the DMA status is written to a memory location specified by the PitCREW data route and address registers. It is accomplished by controlling bit 25 of the word count field of a linked-list command

packet. If bit 25 is zero, the linked list entry is a "write status" command instead of a DMA move command. This feature is provided for semaphore operations, and is a mechanism for signaling DMA complete to a RACEway process.

Also provided is the ability to read and write the internal registers of the PitCREW, to write to the output FIFO, and to read from the input FIFO as a slave interface. These functions are all provided mainly for diagnostic purposes.

Connecting the FIFO Interface

Figure 5 describes the connections between the CY7C4245 FIFOs and the PitCREW Controller. For information on the CY7C4245 FIFO and its signals, see the Cypress CY7C4245 4K x 18 Synchronous FIFO data sheet.

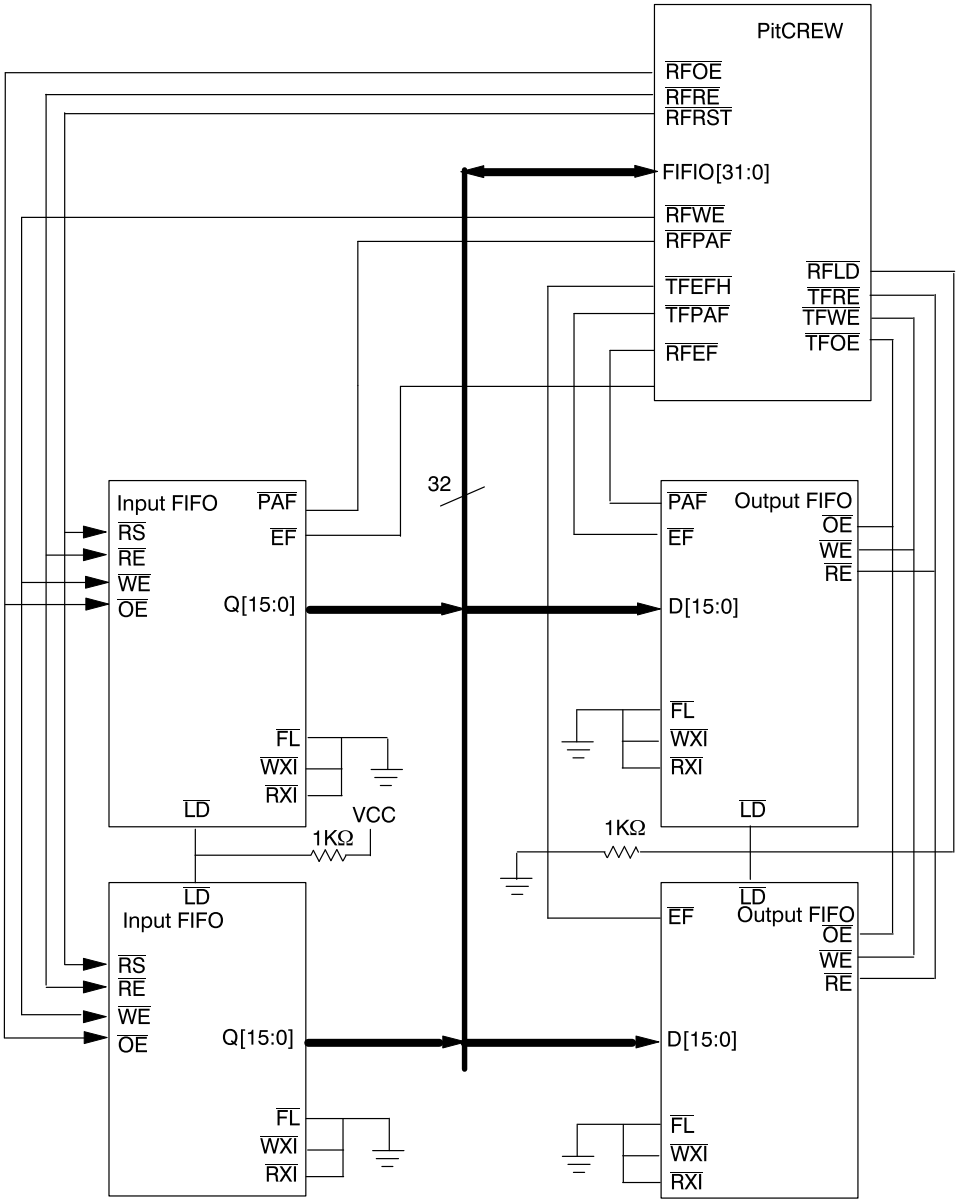


Figure 5. Connecting the FIFOs to the PitCREW Controller

Registers

Register Address Map

The following two tables display the addresses for the PitCREW registers for writing and reading separately. Most of the registers are 32 bits wide but mapped into 64-bit address space, since this is the granularity of a single cycle on the RACEway (there is no address bit 2). A few of the registers are true 64-bit registers as discussed below.

In the Register Write Address Map, entries designated NA (not available) are not writable locations. To perform a write operation to any of the register locations, with the exception of address 0x10, either a 64-bit or a 32-bit write should be specified with the data located in bits 63 through 32.

Address 0x10 is a special address to allow a 64-bit *load and go* operation. If a 64-bit write is specified to address 0x10, the Command Address register is loaded from bits 63 through 32 and the Command Route register is loaded from bits 31 through 0. After the load-and-go write, the Controller will fetch the command packet pointed to by the route and address in the load and go, and execute that packet (this assumes that the GO bit is set in the Command Address register data).

A second method of initiating a transfer is to perform a 32-bit write of the Command Route register data at address 0x10 (with route data located on bits 31 through 0) followed by a write to address 0x18 of the Command Address register (with the GO bit set).

DMA transfers can also be initiated by directly writing the Word Count register after loading appropriate values in Data Route and Data Address registers. This method circumvents use of the linked-list convention of the PitCREW.

It is possible to write directly from a RACEway master to the output FIFO via address 0x28. Users are warned that the last long word of any RACEway write to this address will NOT be written to the output FIFO.

It is also possible to read from address 0x28 to move data from the input FIFO to the RACEway.

Register Write Address Map

Address[5:3]	Bits 63.....32	Bits 31.....0
000	NA	NA
001	Control	NA
010	Command Address	Command Route
011	Command Address	NA
100	Word Count	NA
101	TX FIFO	NA
110	Data Route	NA
111	Data Address	NA

Register Read Address Map

Address[5:3]	Bits 63.....32	Bits 31.....0
000	Status	Status
001	Control	Control
010	Command Route	Command Route
011	Command Address	Command Address
100	Word Count	Word Count
101	RX FIFO entry n	RX FIFO entry n+1
110	Data Route	Data Route
111	Data Address	Data Address

Reading from all addresses except 0x28 will return the same data replicated on the upper and lower 32-bit words. Reading from address 0x28 will return the next two consecutive input FIFO entries (64 bits). This is primarily for diagnostic purposes.

Command Route Register

31	28	25	22	19	16	13	10	7	4 3	2 1	0
Route 0	Route 1	Route 2	Route 3	Route 4	Route 5	Route 6	Route 7	Route 8	Broadcast Accept. Code	Routing Priority	0

The Command Route register is used by the PitCREW to retrieve the next command packet in the linked list. The format of this register is the standard format from the RACEway interlink standard VITA 5−1994. Bit 0 must always be reset to zero in this register.

Command Address Register

31 28	27 3	2	1	0
Width/Alignment	Address	Go	Read	Locked

The Command Address register is used to specify the address of the next command packet in the linked list. The Width/Alignment, Address, and Locked fields are the same format as specified in the RACEway interlink standard. When a command packet is fetched (or written into the registers) with the Go bit set, the next command packet will be fetched at the completion of the current command packet. The last command packet fetched in a linked list should have the Go bit reset.

The Read bit must always be set to a one to specify reading a command packet. Also, the Locked bit should always be set to a one, specifying that the fetch is not locked.

Data Route Register

31	28	25	22	19	16	13	10	7	4 3	2 1	0
Route 0	Route 1	Route 2	Route 3	Route 4	Route 5	Route 6	Route 7	Route 8	Broadcast Accept. Code	Routing Priority	Broadcast/ Single

The Data Route register contains the route for the data packet to be transferred. The format is the same as specified in the RACEway interlink standard.

Data Address Register

31 28	27 3	2	1	0
Width/Alignment	Address	Reserved	Transmit	Locked

The Data Address register contains the address for the data packet to be transferred. The format is the same format as specified in the RACEway interlink standard. The Transmit bit specifies the direction of the transfer: when it is set, data is read from the RACEway and written to the output FIFO. When it is reset, data is read from the input FIFO and written to RACEway.

Status Register

Bit	Function	Active HIGH	Query Control	Description
31:29	Reserved			
28	Read Error	Yes	S/LL	Error reading command or data
27:26	PIO[2:1]		S/LL	User controlled input bits
25:24	Reserved			
23	Output FIFO Greater Than Zero	Yes	S/LL	Data present in output FIFO ($\overline{\text{TFEFL}}$ pin)
22	Ready In	Yes	S/LL	Cable interface ready (RXRDY pin)
21	Valid Packet	Yes	S	Command Packet has valid format. “Not Valid” cleared by a correct packet
20	Overflow	Yes	S/LL	Input FIFO overflow
19	Input Suspended	Yes	S/LL	Input FIFO almost full ($\overline{\text{RFPAF}}$)
18	Input FIFO Greater Than Zero	Yes	S/LL	Data present in input FIFO (dynamic)
17	Reserved			Reserved − read as zero
16	Reserved			Reserved − read as zero
15:4	Board Type		S/LL	0x010=PitCREW
3:0	Board Rev		S/LL	Board Revision

The **Query Control** column displays whether the bit can be queried under either slave (**S**) control, linked-list control (**LL**), or both (**S/LL**). The following paragraphs discuss the different fields in the status register.

The **Read Error** bit is set when an error is detected during a RACEway transfer. It is cleared either by hardware reset or by writing the control register.

The **PIO[2:1]** field is used to read the state of the PIO pins when these pins are operated in input mode.

The **Output FIFO Greater Than Zero** bit is connected directly to the $\overline{\text{TFEFL}}$ pin of PitCREW.

The **Ready In** bit is connected directly to the RXRDY input pin of PitCREW.

The **Valid Packet** bit gets set when a valid packet is fetched. A valid packet is defined as containing a valid packet field in the Word Count register.

The **Overflow** bit is set when a input FIFO overflow occurs. This bit can be cleared by a hardware reset or a software reset of the Input FIFO in the Control register.

The **Input Suspended** bit is essentially the PitCREW $\overline{\text{RFPAF}}$ pin synchronized to the EXT_CLK.

The **Input FIFO Greater Than Zero** bit is an internally generated input FIFO not empty signal.

Control Register

Bit	Function	Active HIGH	Load Control	Description
31:30	PIO Enable[1:0]	Yes	S/LL	User bits direction: 0 = In, 1 = Out
29:28	PIO[2:1] Data		S/LL	User controlled data output
27	Reserved			
26	Output Reset	Yes	S/LL	Self-pulsed output FIFO reset
25	PIO Cntl Enable	Yes	S/LL	Mask for controlling user outputs
24	Input Reset	Yes	S/LL	Self-pulsed input FIFO reset
23	Sync Wait	Yes	S/LL	Self-pulsed Wait For Sync trigger for the input FIFO logic
22	Ready Out	Yes	S	Enable transfers
21	Stop DMA	Yes	S	Stop operation in progress—current packet data may be corrupted
20	Reserved			Reserved for future use
19	Sync Out	Yes	S/LL	Self-pulsed signal setting Send Sync with next output FIFO data
18	Reserved			Reserved for future use
17	$\overline{\text{RSVD2}}$ Out	No	S/LL	For use as a general purpose output pin.
16	Reserved			Reserved for future use
15:0	Rupt Vector		S/LL	Interrupt control

The **Load Control** column displays whether the bit can be loaded under either slave (**S**) control, linked−list control (**LL**), or both (**S/LL**). The following paragraphs discuss the different fields in the Control Register.

The **PIO Enable[1:0]** field provides individual direction control over the two PitCREW programmable I/O pins. When a PIO Enable[1:0] bit is defined as output, the value driven out of that PIO pin is specified in the **PIO[2:1] Data** field. In order to change either the PIO[2:1] Enable or PIO[2:1] Data fields the **PIO Cntl Enable** bit must be set. Writes and link-list loads to the control register with the PIO Cntl Enable bit reset will not affect the PIO[2:1] Enable and PIO[2:1] Data fields.

The **Output Reset** bit performs a reset of the output FIFOs and the associated logic internal to the PitCREW Controller. To perform a reset, a one is written to the Output Reset bit. It is not necessary to follow this with a write of zero—the Output Reset bit is self-pulsed. This reset will be followed by an output FIFO load cycle to load the watermark value of the programmable flags.

The **Input Reset** bit performs a reset of the input FIFOs and the associated logic internal to the PitCREW Controller. Like the Output Reset bit, the Input Reset bit is self-pulsed. Also, a programmable flag load cycle is not performed for the input FIFO since the PitCREW Controller does not have access to the data input of the input FIFO devices.

The **Sync Wait** bit is a self-pulsed bit that puts the input FIFO interface logic in the armed state. Input FIFO write enable ($\overline{\text{RFWE}}$) will not go active until a sync pulse is input on the $\overline{\text{RXSYNC}}$ pin (synchronous to EXT_CLK).

The **Ready Out** bit is used to set and reset the NRDY output pin. The pin will be inverted from the register bit. It is intended that the cable be driven through an inverting open-collector buffer, and then brought back into the RXRDY pin. Note that this bit can only be modified by performing a slave write—not via linked-list load.

The **Stop DMA** bit will stop a transfer in process. The transfer can then be continued or aborted. The integrity of the data packets may be corrupted if used in conjunction with the Output Reset or Input Reset bits in this register (an abort of the command packet).

The **Sync Out** bit allows for a sync marker to be written into the output FIFO to tag the beginning of a data frame. This sync marker moves through the FIFO with the data.

The **$\overline{\text{RSVD2}}$ Out** bit is inverted and connected to the $\overline{\text{RSVD2}}$ pin of the PitCREW. It is for use as a programmable output pin.

Word Count Register

Bit	Word Count Reg	Function
31:27	Reserved	Reserved
26	Bit Bucket	Discard output data
25	Write Type	1= Data Write, 0 = Status Write
24:21	Valid Packet Field	Must be equal to binary ‘0010’
20	Reserved	Reserved
19:0	Word Count [19:0]	Number of 8−byte words to write (Up to 8 Mbytes)

The Word Count register can be loaded linked-list style, or it may be written or read directly via the RACEway. The following paragraphs describe the Word Count Register fields in greater detail.

The **Bit Bucket** bit, when set, will cause the PitCREW output logic to discard the output data. Data will be read from the RACEway but not written into the output FIFOs. This is useful for diagnostic purposes.

The **Write Type** bit is normally set to a one to perform data writes, however by resetting this bit, a status write will be performed. During this write, bits 31 through 16 of the Status register are concatenated with bits 15 through 0 of the Control register (the Rupt field) and written to the route and address specified in the Data Route and Data Address registers. This is useful for end of transfer notification, and is also a method of performing RACEway interrupts.

The **Valid Packet Field** is used to detect runaway linked lists. The Word Count register is the first register loaded in a linked list. If the Valid Packet Field fetched in the command packet is not equal to binary ‘0010’, then the data transfer is never started and the Valid Packet bit in the Status register is cleared indicating an error.

The **Word Count** field is loaded with the number of 8-byte (64-bit) words to be transferred. In the output direction, the PitCREW Controller checks the $\overline{\text{TFPAF}}$ signal to see if there are 576 empty slots (2304 bytes) available in the output FIFOs. An output transfer cycle will be initiated when the number of available slots is at least 576 (which is 512 data slots plus 64 sync marker slots corresponding to 2304 bytes). The size of the transfer will be equal to the lessor of 2K bytes or the value programmed into the Word Count register. For input cycles, the Controller actually counts the number of entries in the input FIFOs by counting the number of EXT_CLK rising edges with $\overline{\text{RXVALID}}$ active. The pins RXRDY and $\overline{\text{RXSYNC}}$ are also used to define valid input data entries. An input transfer cycle is initiated if the number of entries in the input FIFO is equal to the value in the Word Count Register or 2 Kbytes, whichever is lower.

Signals

The 108 signal pins of the PitCREW controller can be divided into six main groups:

- RACEway interface signals
- Output FIFO interface signals
- Output FIFO control signals
- Input FIFO interface signals
- Input FIFO control signals
- Cable interface signals

The RACEway interface signals provide a port to the RACEway fabric with full 160 megabytes per second capability. These signals are synchronous to the RACEway clock. The RACEway clock frequency is 40 MHz. *Table 1* lists these signals.

The output and input FIFO interface groups provide strobes to reset, read, and write both sets of FIFOs. The input and output FIFOs share the 32-bit FIFO data bus (FIFIO[31:0]) and the asynchronous external clock (EXT_CLK). Pins are provided to interface to the input and output FIFO status flags and to set the initial value of the programmable status flag in the output direction. Setting this value is not possible in the input direction since there is no data bus connection to the inputs of the input FIFOs. Instead, PitCREW counts valid entries as data is clocked into the input FIFOs.

The output FIFO control group provides signals to control data being read from the output FIFOs (TXRDY and $\overline{\text{TXSUSPEND}}$), to provide indication that valid data has been read from the FIFOs (TXVALID), and to generate a start of frame marker to be placed into the output FIFOs ($\overline{\text{TXSYNC}}$).

The input FIFO control group provides signals to control data being placed into the input FIFOs ($\overline{\text{RXVALID}}$, RXRDY), two indicators (opposite polarities) that show the input FIFOs are almost full (RXSUSPEND, $\overline{\text{RXSUSPEND}}$), and a start of frame indicator which allows for the acquisition of data frames based upon an external event ($\overline{\text{RXSYNC}}$). Also provided are two programmable data bits used for data tagging under software control (PIO[2:1]). An overflow pin is provided for input FIFO error indication ($\overline{\text{OVFLOW}}$).

Table 1. PitCREW RACEway Signals

Signal	I/O	Source	Function
RDCONIO	I/O	PitCREW or 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.
RPLYIO	I/O	PitCREW or RACEway	Reply gives the RACEway crossbar permission to send the address or data over the data bus.
REQI	I	RACEway	Request In indicates that the RACEway crossbar is requesting control of the data bus.
REQO	O	PitCREW	Request Out is asserted by the master to request access to the crossbar data bus.
STROBIO	I/O	PitCREW or RACEway	Strobe indicates that address or data is being sent on the data bus. Strobe is sent by the master node after asserting REQO.
XBIO[31:0]	I/O	PitCREW or RACEway	Crossbar Address/Data. These lines must each have 22Ω series termination.
XCLKI	I	RACEway	Crossbar Clock provides the RACEway timing.
XRESETIO	I	RACEway	Reset input from the RACEway connecting port.
XSYNCI	I	RACEway	Crossbar Sync provides control phase information to the crossbar.

Output FIFO Interface

The PitCREW Controller provides interfaces to the cable side of both the input FIFO and the output FIFO, also referred to as the user side of the FIFOs. The following section discusses the output FIFO interface, both the user and RACEway sides.

The output FIFOs can be reset by either the assertion of the $\overline{\text{XRESETIO}}$ pin, or by writing to bit 26 in the PitCREW Control register. Either of these events will cause the $\overline{\text{TFRST}}$ signal to go LOW. An output FIFO reset is always followed by a programmable flag load cycle where the flag data is presented on the FIFIO data bus and the $\overline{\text{TFLD}}$ and $\overline{\text{TFWE}}$ signals asserted. The flag data consists of 0x240, which corresponds to 2304 bytes. This byte size is determined by allocating 2 Kbytes (512 32-bit entries) for data and 256 bytes (64 entries) for sync markers. Note that this places an upper limit of 64 sync markers for every 256 data words which must be adhered to.

The programmable flag load cycle requires that bits 11 through 0 of the FIFO data bus (FIFIO[11:0]) must be connected to bits 11 through 0 of the FIFO that is used to send $\overline{\text{TFPAF}}$ to the Controller (only one of the $\overline{\text{TFPAF}}$ output FIFO flags needs to be connected to the Controller). Also, $\overline{\text{TFLD}}$ must be connected to both output FIFOs in order to prevent an extra write from being registered in the FIFO which does not supply $\overline{\text{TFPAF}}$ (the $\overline{\text{TFWE}}$ pin goes active during a programmable flag load cycle).

The generation of the output FIFO read signal, $\overline{\text{TFRE}}$, is based upon the $\overline{\text{TXSUSPEND}}$, TXRDY, $\overline{\text{TFEFL}}$, and $\overline{\text{TFEFH}}$ input signals. If all of these signals are high then $\overline{\text{TFRE}}$ goes active. The $\overline{\text{TXVALID}}$ output will go LOW in response to the read, if the sync input signal INV_SYNC is not active ($\overline{\text{TXVALID}}$ only goes active for valid data items—not sync markers). $\overline{\text{TXSUSPEND}}$ must be returned to the Controller synchronous to EXT_CLK. In the case of the cable interface, an external synchronizing flip-flop is recommended between the cable signal $\overline{\text{SUSPEND}}$ and the $\overline{\text{TXSUSPEND}}$ pin on the Controller. $\overline{\text{TFRE}}$ is guaranteed to go inactive within four EXT_CLKs from the rising edge of $\overline{\text{TXSUSPEND}}$.

The output FIFO programmable almost full flag $\overline{\text{TFPAF}}$ is used by the PitCREW Controller to burst data over the RACEway. A RACEway burst read cycle is initiated when there are at least 576 (2304 ÷ 4) empty locations in the output FIFO. The size of the burst is equal to the lesser of 2 Kbytes or the value programmed into the PitCREW Word Count register.

It is not required to use the PitCREW output FIFO interface. The data output side of the output FIFO may be clocked asynchronously to the EXT_CLK as long as the $\overline{\text{TFPAF}}$, and both of the FIFO empty flags $\overline{\text{TFEFL}}$ and $\overline{\text{TFEFH}}$, are connected to the Controller.

A sync marker may be placed in the output FIFO using the SET_SYNC, INV_SYNC, and $\overline{\text{TXSYNC}}$ pins. By setting the Sync Out bit in the Control register, a sync marker will be driven out on the SET_SYNC pin. It is intended that this be connected to one of the unused data inputs on the output FIFOs (assuming that the FIFOs are organized 18 bits wide). The output from the data bit should be connected to the INV_SYNC input pin. The $\overline{\text{TXSYNC}}$ output pin is simply an inversion of the INV_SYNC pin going active when the sync marker is read out of the FIFO. Also, the $\overline{\text{TXVALID}}$ output is gated by the INV_SYNC pin and will not go active for the sync marker FIFO read. *Table 2* summarizes the output FIFO interface signals, and *Table 3* summarizes the output FIFO control signals.

Table 2. PitCREW Output FIFO Interface Signals

Signal	I/O	Source	Function
EXT_CLK	I	External	External clock synchronous to FIFO interface. Is common to both TX and RX FIFO logic.
FIFIO[31:0]	I/O	PitCREW	Data lines to the output FIFOs and from the input FIFOs
$\overline{\text{TFLD}}$	O	PitCREW	Output FIFO load for programmable flags.
$\overline{\text{TFOE}}$	O	PitCREW	Output FIFO output enable
$\overline{\text{TFRE}}$	O	PitCREW	Output FIFO read enable
$\overline{\text{TFRST}}$	O	PitCREW	Output FIFO reset
$\overline{\text{TFWE}}$	O	PitCREW	Output FIFO write enable
$\overline{\text{TFEFL}}$,$\overline{\text{TFEFH}}$	I	FIFO	Output FIFO empty flags from both FIFOs for PitCREW
$\overline{\text{TFEF}}$	I	FIFO	Output FIFO empty flag for PitCREW status register reads—connect to either FIFO flag
$\overline{\text{TFPAF}}$	I	FIFO	Output FIFO programmable almost full flag

Table 3. PitCREW Output FIFO Control Signals

Signal	I/O	Source	Function
PIO[2:1]	I/O	PitCREW	Programmable data bits used for software handshaking.
NRDY	O	PitCREW	Generates Ready out to cable interface. This HIGH-active signal should go through an inverting open-collector buffer to drive the cable $\overline{\text{NRDY}}$ signal.
TXRDY	I	External	Should be connected to the output of the NRDY open-collector buffer (to NRDYN). When active indicates that data can be read out of the output FIFO on the next EXT_CLK.
$\overline{\text{TXSUSPEND}}$	I	External	May be asserted to suspend reading out of the output FIFO (to throttle output data).
SET_SYNC	O	PitCREW	Sync (top of frame) marker output to connect to output FIFO data input to tag start of data frame in FIFO.
INV_SYNC	I	FIFO	Sync marker input from output FIFO (output of FIFO input signal SET_SYNC).
$\overline{\text{TXSYNC}}$	O	PitCREW	Indicates the start of a data frame when asserted. Is inverted INV_SYNC for use in driving the cable interface SYNCN signal.
$\overline{\text{TXVALID}}$	O	PitCREW	Indicates valid data has been read out of the FIFOs. May be used to drive the cable interface VALIDN signal.

Input FIFO Interface

The input FIFO interface may be reset either by the assertion of the $\overline{\text{XRESETIO}}$ pin, or by writing to bit 24 in the Control Register. Either of these will cause the $\overline{\text{RFRST}}$ signal to go LOW. Loading of the programmable flags is not performed in the input FIFO interface because the Controller does not have access to the input FIFO input data path.

The generation of the input FIFO write enable $\overline{\text{RFWE}}$ is based upon the RXRDY and $\overline{\text{RXVALID}}$

pins, as well as the sync logic utilizing the $\overline{\text{RXSYNC}}$ pin. Ignoring the sync logic for the moment, if the $\overline{\text{RXVALID}}$ pin is LOW and the RXRDY pin is HIGH, the $\overline{\text{RFWE}}$ signal will go active (LOW). Note that the path from either of these two input signals to the $\overline{\text{RFWE}}$ output is purely combinatorial. $\overline{\text{RXVALID}}$ is intended to be used to gate off individual writes into the input FIFO and RXRDY is intended to be tied to the output of the open-collector buffer driven by the NRDY output (stating that the cable is ready).

The above example assumes that the sync logic is disabled, that is the Sync Wait bit in the PitCREW Control register has not been set. If the Sync Wait bit is set, the logic generating $\overline{\text{RFWE}}$ will wait until a single EXT_CLK pulse on the $\overline{\text{RXSYNC}}$ is detected. The first write will occur on the clock following the assertion of the $\overline{\text{RXSYNC}}$ pin, if RXRDY and $\overline{\text{RXVALID}}$ are also active as described above.

The PitCREW Controller does not use the input FIFO flags to determine when to initiate a RACEway write transfer. Instead, it counts valid input FIFO entries as defined in the above criteria and initiates a RACEway transfer upon detecting the lessor of 2 Kbytes or the value programmed into the PitCREW Word Count register. Data will be read from the FIFO and written to the RACEway until the word count reaches zero, the FIFO empties, a 2-Kbyte boundary is reached, or the RACEway Request In signal is raised (indicating a “Kill” condition). Any of these conditions will cause the Request Out signal to be deasserted.

When the word count reaches zero, the next command packet is fetched and operation continues if the GO bit of the PitCREW Command Address register is set. Using two 4K X 16 FIFOs yields 16 Kbytes of buffering which, at 120 MB/sec, corresponds to 136 microseconds. *Table 4* summarizes the input FIFO interface signals and *Table 5* summarizes the input FIFO control signals. Two other signals, $\overline{\text{RSVD2}}$ and TXDIR, are described in *Table 6*.

Table 4. PitCREW Input FIFO Interface Signals

Signal	I/O	In From/ Out To	Function
EXT_CLK	I	External	External clock synchronous to FIFO interface. Is common to both TX and RX FIFO logic.
FIFIO[31:0]	I/O	FIFO	Data lines to the FIFO.
RFLD	O	PitCREW	Input FIFO load for programmable flags. This pin is a static high-level (no programmable load function performed).
RFOE	O	PitCREW	Input FIFO output enable.
$\overline{\text{RFRE}}$	O	PitCREW	Input FIFO read enable.
$\overline{\text{RSTRF}}$	O	PitCREW	Input FIFO reset.
$\overline{\text{RFWE}}$	O	PitCREW	Input FIFO write enable.
$\overline{\text{RFPAF}}$	I	FIFO	Programmable Almost Full Flag from FIFO.
$\overline{\text{RFEF}}$	I	FIFO	Input FIFO empty flag.

Table 5. PitCREW Input FIFO Control Signals

Signal	I/O	In From/ Out To	Function
$\overline{\text{OVFLOW}}$	O	PitCREW	Indicates a input FIFO overflow has occurred.
PIO[2:1]	I/O	PitCREW	Programmable data bits used for software handshaking.
PIOEN[2:1]	O	PitCREW	Indicate (when LOW) that the PIO[2:1] pins are enabled.
RXRDY	I	External	Should be connected to the output of the NRDY open-collector buffer (to $\overline{\text{NRDY}}$ on the cable interface). When active (HIGH) allows data to be written into the input FIFO.
RXSUSPEND	O	FIFO	Asserted HIGH when the FIFO is almost full (127 words from full).
$\overline{\text{RXSUSPEND}}$	O	FIFO	Asserted LOW when the FIFO is almost full (127 words from full).
$\overline{\text{RXSYNC}}$	I	External	Indicates the start of a data frame when asserted.
$\overline{\text{RXVALID}}$	I	External	Indicates valid data is available to write to the FIFOs when low. Is used to dynamically qualify each data word written into the input FIFOs.

Table 6. Miscellaneous Control Signals

Signal	I/O	In From/ Out To	Function
$\overline{\text{RSVD2}}$	O	PitCREW	Set/reset from bit 17 of the Control Register. This bit is inverted from the value programmed into the Control Register.
TXDIR	O	PitCREW	Used to indicate the direction of data transfer on the cable interface.

Cable Interface Signal Description

The PitCREW Controller can be connected to a bi-directional cable interface compatible with FPDP (see Reference Documents section for related standard). This interface consists of a 32-bit data bus, two user-defined data bits for data tagging, a free-running clock, and five control signals. The cable interface supports multiple destinations, but the required arbitration is not described in this note. The following paragraphs describe how cable interface signals are related to PitCREW control signals.

The source of the data (transmitter) drives the signal $\overline{\text{DIR}}$ LOW to indicate the direction is from the cable interface to the input FIFOs. This signal is included on the PitCREW Controller. All sources and destinations drive the open-collector signal NRDY, which indicates that the cable is ready. This signal is also included on the PitCREW Controller. Sources of data are required to read NRDY in hardware to ascertain that the interface is in the ready state. This is performed via the RXRDY pin on the PitCREW Controller. The free-running clock, STROB, is sourced by the source of the data bus and drives the EXT_CLK pin of PitCREW.

A data synchronization signal, $\overline{\text{SYNC}}$, is provided to frame data at the input FIFO. The input FIFO will wait until a single pulse of $\overline{\text{SYNC}}$ is detected, and then start to acquire data on the next assertion of $\overline{\text{VALID}}$. The $\overline{\text{VALID}}$ signal is used to indicate that valid data is available to be input on a particular rising edge of STROB. $\overline{\text{SYNC}}$ and $\overline{\text{VALID}}$ are synchronized to STROB (EXT_CLK) and connected to PitCREW pins $\overline{\text{RXSYNC}}$ and $\overline{\text{RXVALID}}$ respectively.

A suspend signal is provided to inform the data transmitter to stop sending data. The receiver asserts the $\overline{\text{SUSPEND}}$ signal when its buffer is almost full. The $\overline{\text{RXSUSPEND}}$ output of PitCREW provides this signaling.

Pins

Table 7 identifies the CY7C387P pinout for the PitCREW Controller.

Table 7. Cypress CY7C387P Pinout

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	FIFIO17	37	XBIO26	73	RXSUSPEND	109	NU/GND
2	XBIO17	38	NU/GND	74	$\overline{\text{TXSUSPEND}}$	110	NU/GND
3	FIFIO14	39	XBIO27	75	TXRDY	111	NU/GND
4	XBIO12	40	XBIO25	76	$\overline{\text{RFRST}}$	112	NU/GND
5	XBIO14	41	XBIO30	77	NU/GND	113	$\overline{\text{RXSYNC}}$
6	FIFIO12	42	VCC	78	$\overline{\text{TFEFH}}$	114	VCC
7	VCC	43	FIFIO30	79	VCC	115	$\overline{\text{RFWE}}$
8	XBIO16	44	XBIO31	80	XRPLYIO	116	NRDY
9	FIFIO16	45	FIFIO31	81	NU/GND	117	XBIO5
10	FIFIO11	46	XBIO28	82	$\overline{\text{TFRST}}$	118	XBIO3
11	XBIO11	47	XBIO29	83	XSTROBIO	119	XBIO6
12	FIFIO18	48	FIFIO28	84	XREQ_O	120	XBIO9
13	XBIO18	49	FIFIO29	85	NU/GND	121	FIFIO6
14	FIFIO19	50	GND	86	$\overline{\text{TFPAF}}$	122	GND
15	GND	51	PIO2	87	GND	123	FIFIO7
16	XBIO19	52	PIOEN2	88	TXDIR	124	XBIO7
17	$\overline{\text{XRESETIO}}$	53	FIFIO0	89	XREQ_I	125	XBIO8
18	EXT_CLK	54	GND	90	XCLKI	126	GND
19	VCC	55	FIFIO5	91	VCC	127	FIFIO9
20	RXRDY	56	FIFIO2	92	$\overline{\text{RFPAF}}$	128	PIOEN1
21	$\overline{\text{RXVALID}}$	57	FIFIO4	93	$\overline{\text{TFEFL}}$	129	PIO1
22	VCC	58	VCC	94	VCC	130	VCC
23	XBIO20	59	XBIO2	95	XRDCONIO	131	FIFIO8
24	FIFIO21	60	FIFIO1	96	XSYNCI	132	FIFIO3
25	FIFIO22	61	XBIO4	97	$\overline{\text{OVFLOW}}$	133	RSVD2
26	FIFIO23	62	XBIO0	98	$\overline{\text{RFRE}}$	134	XBIO10
27	FIFIO20	63	$\overline{\text{TFLD}}$	99	$\overline{\text{RFOE}}$	135	NU/GND
28	XBIO22	64	XBIO1	100	NU/GND	136	FIFIO10
29	FIFIO24	65	SET_SYNC	101	NU/GND	137	FIFIO13
30	GND	66	GND	102	GND	138	GND
31	XBIO21	67	$\overline{\text{TFWE}}$	103	NU/GND	139	XBIO15
32	XBIO23	68	INV_SYNC	104	NU/GND	140	XBIO13
33	FIFIO27	69	$\overline{\text{TXVALID}}$	105	NU/GND	141	XTDI
34	FIFIO26	70	$\overline{\text{TXSYNC}}$	106	NU/GND	142	XTDO
35	FIFIO25	71	$\overline{\text{TFRE}}$	107	$\overline{\text{TFOE}}$	143	$\overline{\text{RFLD}}$
36	XBIO24	72	$\overline{\text{RXSUSPEND}}$	108	NU/GND	144	FIFIO15

PitCREW Programming Considerations

Direct access to the PitCREW DMA channel is gained by writing the Word Count register. Writing this register initiates a DMA transfer. Values previously written to the Data Route and Data Address registers are used for RACEway direction and header information. The Word Count register specifies the transfer length.

The PitCREW Controller can also operate in linked-list fashion, fetching a new command packet at the completion of the current one, as shown in *Figure 3*. Each command packet consists of a word count, the new contents of the Control register, the data route and address, and the next command packet route and address.

The linked list of command packets is built in memory and a load-and-go operation specifying the route and address of the first packet is written to the 64-bit location at address 0x10 (the combined Command Route and Address register) to prime the operation. The PitCREW then executes each element in the linked list until a command packet is fetched with the GO bit reset. The GO bit in the Command Address register instructs the Controller to fetch the next command packet at the destination specified the Command Address and Command Route registers. The last command packet in the linked list should have the GO bit reset.

The Read bit in the Command Address register should always be set to a one, to specify reading the command packet from RACEway memory. All Lock bits should always be one specifying that operations are not locked.

The Data Route and Address registers specify the location in RACEway memory where the data packet will be stored (input operation) or fetched (output operation). The output bit in the Data Address register specifies the direction of operation to be either input (reset to zero) or output (set to a one).

Two user programmable I/O bits, PIO[2:1] are provided for data tagging or other application specific purposes. These bits may be individually programmed via the Control register to be either inputs or outputs, and the programming may be accomplished through direct writes to the Control register or under linked-list control. Assigning the data values under linked-list control allows for the tagging of the command packets as they are executed. For example, the first packet may be assigned a value that tags it as a header and subsequent packets may be tagged as data.

In order to program the PIO bits, the PIO Control Enable bit must be set. Writes to the Control register with the PIO Control Enable bit reset will not affect either the direction (PIO Enable[1:0]) or the value (PIOdata[1:0]) fields. It is intended that the PIO bits be connected to the 33rd and 34th bits of the FIFOs. In this way they may be used by custom hardware to distinguish data packets. They cannot, however, be transferred to memory, which has a 32-bit data organization.

The Stop DMA bit may be set to pause or abort a DMA operation in progress. When set by a slave write operation to the Control register, the current DMA operation will be in a paused state. Resetting the Stop DMA bit at a later time will resume the operation. Setting Output Reset and Input Reset bits while in the paused state will cause an abort to take place. Note that the integrity of the data packets may be violated after aborting an operation.

The PitCREW Controller has the ability to write the status of the Controller to the RACEway memory location specified in the Data Route and Address registers. This is accomplished under linked-list control as a separate command packet and is the basic mechanism used for notification of end of packet.

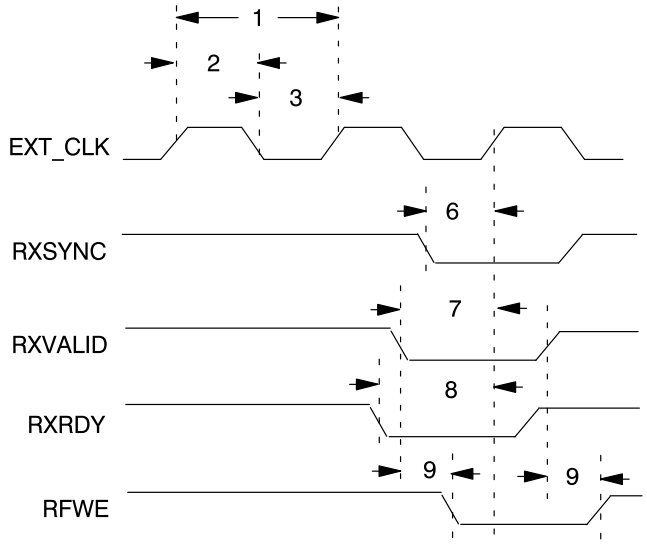
When a command packet is fetched that has bit 25 in the Word Count register reset, a Status Write operation will be performed. In this operation, bits 31 to 16 of the Status register will be concatenated with bits 15 to 0 of the Control register (the Rupt vector) and written to RACEway memory. Note that the Transmit bit in the Data Address register must be reset to zero specifying a write to RACEway memory as the direction.

Status Write operations may be interspersed with actual data transfers in the linked list as a method of sending end of packet status to a controlling process. The location specified in the Status Write operation may be polled by the controlling process, or the location may be specified to be a mailbox interrupt location for a given process on the RACEway. In this way, an end of packet interrupt may be generated to the requesting process via the linked list. If only the Rupt field is desired, the Data Width and Alignment bits in the Data Address register may be used.

Timings

Input Timing

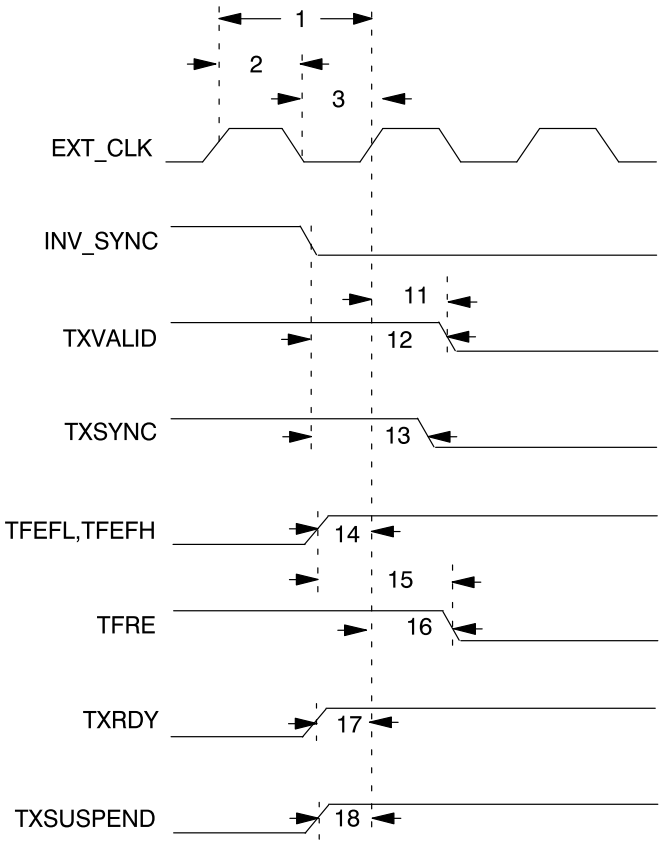
The following timing diagram describes the worst-case timing parameters for the input interface.



Symbol	Parameter	Min.	Max.	Note
1	EXT_CLK clock period	25		
2	EXT_CLK high width	10		
3	EXT_CLK low width	10		
6	$\overline{\text{RXSYNC}}$ set-up time to EXT_CLK	8		
7	$\overline{\text{RXVALID}}$ set-up time to EXT_CLK	19		A
8	RXRDY set-up time to EXT_CLK	21		B
9	$\overline{\text{RXVALID}}$ to $\overline{\text{RFWE}}$ delay		13	A

Output Timing

The following timing diagram describes the worst-case timing parameters for the output interface.



Symbol	Parameter	−1 Min.	−1 Max.	−2 Min.	−2 Max.	Notes
1	EXT_CLK clock period	30		25		
2	EXT_CLK high width	13		10		
3	EXT_CLK low width	13		10		
11	EXT_CLK to $\overline{\text{TXVALID}}$ delay		10		8	
12	INV_SYNC to $\overline{\text{TXVALID}}$ delay		13		11	
13	INV_SYNC to $\overline{\text{TXSYNC}}$		10		8	
14	$\overline{\text{TFEFH}}$, $\overline{\text{TFEFL}}$ set-up time to EXT_CLK	10		8		
15	$\overline{\text{TFEFH}}$, $\overline{\text{TFEFL}}$ to $\overline{\text{TFRE}}$ delay		12		10	
16	EXT_CLK to $\overline{\text{TFRE}}$ delay		10		8	
17	TXRDY set-up time to EXT_CLK	10		8		
18	$\overline{\text{TXSUSPEND}}$ set-up time to EXT_CLK	10		8		A, C

Notes

A. $\overline{\text{RXVALID}}$ to $\overline{\text{RFWE}}$ is a combinatorial path used to dynamically mask writes to the input FIFO. The delay for the path is specified above in symbol 9. Symbol number 7, $\overline{\text{RXVALID}}$ set-up time to EXT_CLK, includes a FIFO set-up time of 6 ns.

B. RXRDY is intended to be a static signal displaying the ready status of the cable interface.

C. $\overline{\text{TXSUSPEND}}$ must meet the set-up time specified in symbol 18. An external synchronizing flip-flop is recommended for the cable interface. The $\overline{\text{TFRE}}$ signal is guaranteed to transition to the inactive state within four EXT_CLK periods from the rising edge of $\overline{\text{TXSUSPEND}}$.

Design Considerations

This section describes the minimum requirements for the design of input and output interfaces.

Basic Input Interface

The simplest input interface requires only EXT_CLK and data signals. However, the basic interface must meet the following conditions:

- EXT_CLK is a free-running clock.
- The data stream must be continuous; the relative starting point within the data stream is arbitrary.
- The aggregate data rate must not exceed the overall sustainable bandwidth.
- Unused control lines must be set to the appropriate state. In a basic synchronous interface, tie both $\overline{\text{RXVALID}}$ and $\overline{\text{RXSYNC}}$ LOW. With $\overline{\text{RXVALID}}$ tied LOW, data is valid on all cycles. With $\overline{\text{RXSYNC}}$ tied LOW, data transfers are not synchronized.

Figure 6 illustrates a basic interface.

Input Data Qualification with $\overline{\text{RXVALID}}$

The $\overline{\text{RXVALID}}$ signal should be asserted LOW when valid data is to be input. *Figure 7* illustrates the use of $\overline{\text{RXVALID}}$. Note that the user should monitor the RXSUSPEND signal, which is a doubly-synchronized version of the RFPAF pin, and stop writing into the input FIFO when RXSUSPEND goes active.

Input Data Qualification with $\overline{\text{RXSYNC}}$ and $\overline{\text{RXVALID}}$

Figure 8 illustrates buffered interface using $\overline{\text{RXSYNC}}$ and $\overline{\text{RXVALID}}$. If using the sync wait mode, data will not be written to the FIFO until the cycle after the first SYNC pulse is received.

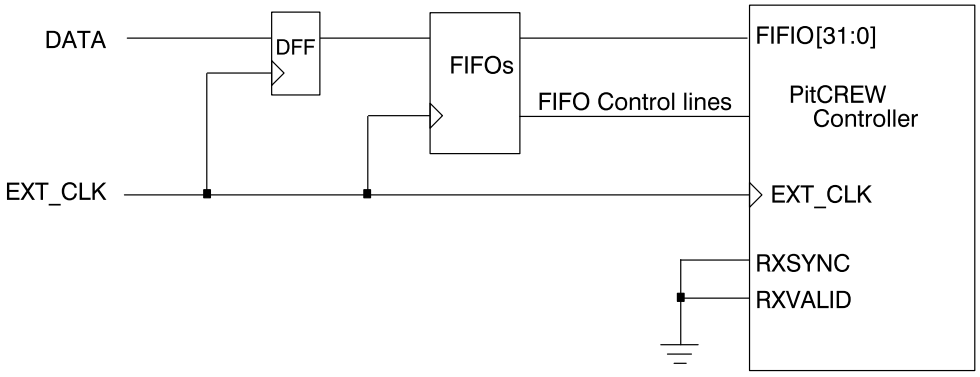


Figure 6. Basic Input Interface

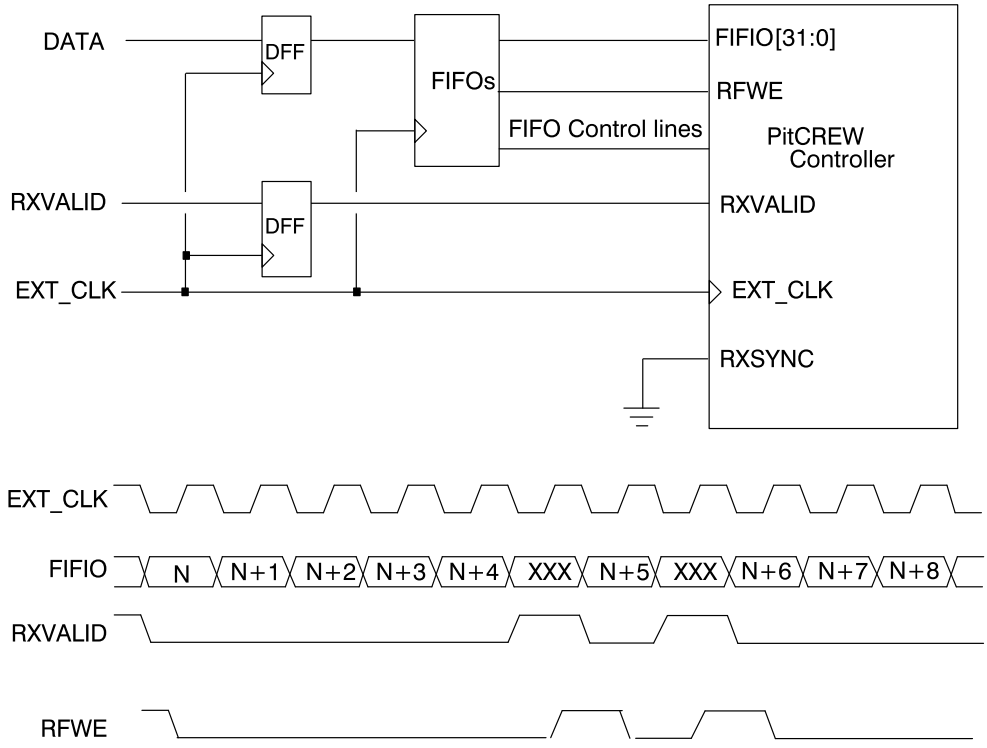


Figure 7. Input Data Qualification with $\overline{\text{RXVALID}}$

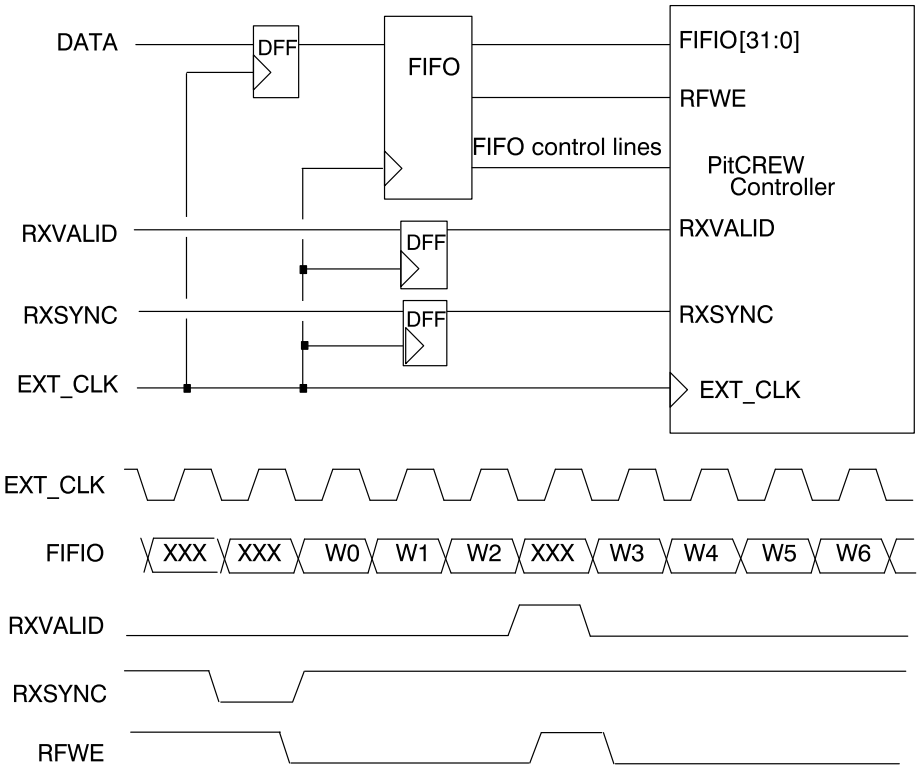


Figure 8. Input Data Qualification With $\overline{\text{RXSYNC}}$ and $\overline{\text{RXVALID}}$

Basic Output Interface

The simplest output interface requires only EXT_CLK and data signals. However, the basic interface must meet the following conditions:

- EXT_CLK is a free-running clock.
- The data stream must be continuous; the relative starting point within the data stream is arbitrary.
- The aggregate data rate must not exceed the overall sustainable bandwidth.
- Unused control lines must be set to the appropriate state.

In a basic synchronous interface, tie both $\overline{\text{TXSUSPEND}}$ and INV_SYNC LOW. With $\overline{\text{TXSUSPEND}}$ tied LOW, data will be continuously read out of the output FIFO. With INV_SYNC tied LOW, a start of frame sync is not generated. *Figure 9* illustrates a basic interface.

Controlling Data Transmission with $\overline{\text{TXSUSPEND}}$ and $\overline{\text{TXSYNC}}$

Figure 10 illustrates an output interface with full controls. Reading out of the output FIFO can be controlled dynamically with the $\overline{\text{TXSUSPEND}}$ pin. The $\overline{\text{TXVALID}}$ pin will be active when valid data is output from the FIFO. With the Sync Out bit set in the Control register, a sync marker will be written into the output FIFO. When the sync marker is later read out, the $\overline{\text{TXSYNC}}$ pin will go active and the $\overline{\text{TXVALID}}$ pin will be invalid.

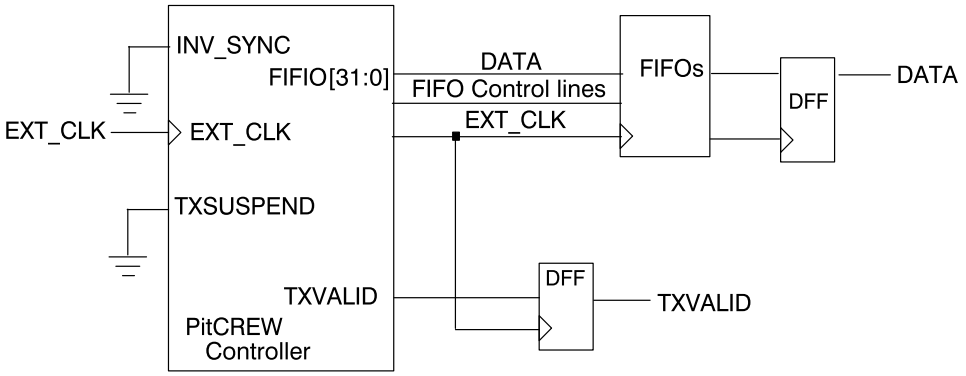


Figure 9. Basic Output Interface

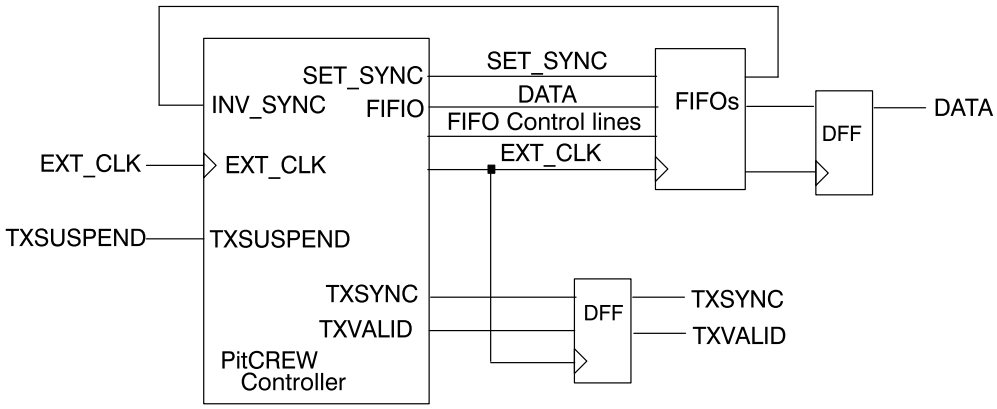


Figure 10. Output Interface with $\overline{\text{TXSUSPEND}}$ and $\overline{\text{TXSYNC}}$

Miscellaneous Design Information

Clocking

The crossbar clock, XCLKI, runs directly from the connector to the Cypress CY7B9910 Low Skew Clock Buffer chip. The clock outputs of this device are used by all on-board components that operate on the 40 MHz RACEway clock frequency. These outputs should be series terminated through 22-ohm resistors. All loads on XCLKI should be connected in series with the daughtercard or P2 connector as the source, and all loads should be within two inches of each other. The ideal configuration is illustrated in *Figure 11.*

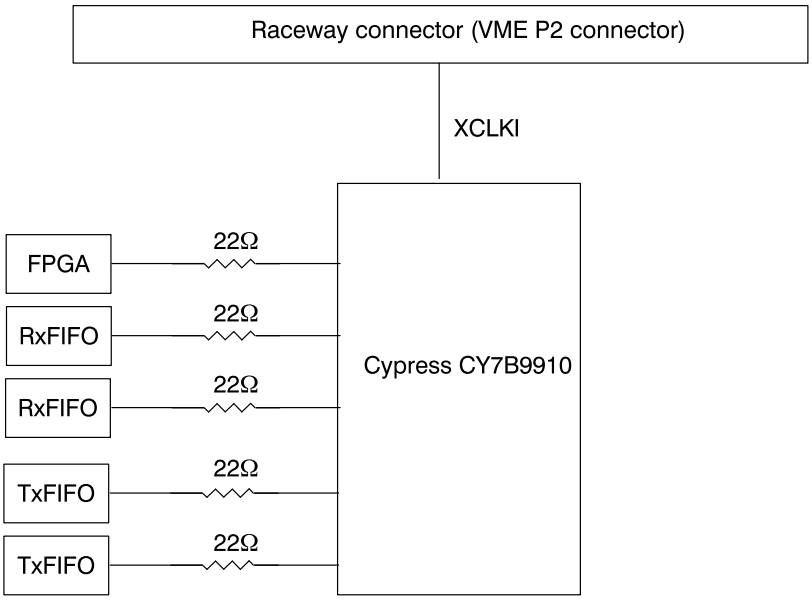


Figure 11. Distributing the XCLKI Clock

RACEway VME J2/P2 Connector

Table 8 describes the use of the VME J2/P2 connector pins for implementing RACEway.

Table 8. RACEway VME J2/P2 Pin Assignments

Pin	Signal	Pin	Signal	Pin	Signal
A1	XCLKI	B1	+5 VOLTS	C1	XRESETIO
A2	GND	B2	GND	C2	Reserved
A3	XBIO9	B3		C3	XSYNCI
A4	XBIO8	B4		C4	GND
A5	GND	B5		C5	XBIO7
A6	XBIO6	B6		C6	GND
A7	GND	B7		C7	XBIO11
A8	XBIO10	B8		C8	GND
A9	XBIO4	B9		C9	STROBIO
A10	GND	B10		C10	RPLYIO
A11	XBIO5	B11		C11	GND
A12	XBIO3	B12	GND	C12	REQI
A13	GND	B13	+5 VOLTS	C13	REQO
A14	RDCONIO	B14		C14	GND
A15	Reserved	B15		C15	XBIO2
A16	GND	B16		C16	XBIO1
A17	XBIO0	B17		C17	GND
A18	XBIO15	B18		C18	XBIO12
A19	GND	B19		C19	XBIO25
A20	XBIO24	B20		C20	GND
A21	XBIO31	B21		C21	XBIO29
A22	GND	B22		C22	XBIO30
A23	XBIO28	B23		C23	GND
A24	XBIO27	B24		C24	XBIO26
A25	GND	B25		C25	XBIO23
A26	XBIO22	B26		C26	GND
A27	XBIO20	B27		C27	XBIO19
A28	GND	B28		C28	XBIO21
A29	XBIO18	B29		C29	GND
A30	XBIO17	B30		C30	XBIO16
A31	GND	B31	GND	C31	XBIO14
A32	XBIO13	B32	+5 VOLTS	C32	GND

pASIC is a trademark of Quicklogic.