PRELIMINARY

CYP15G04K100V1-MGC
CYP15G04K200V2-MGC

Programmable Serial Interface™ (Frequency Agile Devices)

Features

- 200 Mbps–1.5 Gbps, 2.5 Gbps serial signaling rate
- Flexible parallel-to-serial conversion in transmit path
- Flexible serial-to-parallel conversion in receive path
- Multiple selectable loopback/loop-through modes
- 100k to 200k usable gates of CPLD logic
- 240k to 480k bits of integrated memory
  - 192k to 384k bits of synchronous or asynchronous SRAM
  - 48k to 96k bits of true Dual-Port or FIFO RAM
- Internal transmit and receive PLLs
- Logic dedicated Spread Aware PLL
- Transmit FIFO for flexible variable phase clocking
- Differential CML serial input with internal termination and DC-restoration
- Differential CML serial output with source matched impedance of 50Ω
- 160–240 user programmable I/Os
- AnyVolt™ I/O interface
  - Programmable as 1.8V, 2.5V, 3.3V
- Multiple I/O standards
  - LVCMOS, LVTTL, 3.3V PCI, SSTL2(I-II), SSTL3(I-II), HSTL(I-IV), and GTL+
- Direct interface to standard fiber-optic modules
- Designed to drive:
  - fiberoptic modules
  - copper cables
  - circuit board traces
  - backplane links
  - box-to-box links
  - chip-to-chip communication
- Supported standards:
  - Fibre Channel
  - Gigabit Ethernet
  - ESCON
  - DVB
  - SMPTE 259 and 292
- Extremely flexible clocking options
  - Four global clocks
  - Up to 192 additional product term clocks
  - Clock polarity at every register
- Carry chain logic for fast and efficient arithmetic operations
- PCI compliant (Rev. 2.2)
- JTAG programming interface with boundary scan support
- High-Speed (HS) or Frequency Agile (FA) Programmable Serial Interface™ (PSI™) versions available

Frequency Agile PSI Features

- 200 Mbps–1.5 Gbps serial signaling rate per channel
- Up to eight serial channels available to allow:
  - Frequency Agile
  - Redundancy
- Selectable input and output clocking options
- MultiFrame™ receive framer provides alignment to:
  - Bit, byte, half-word, word, multi-word
  - COMMA or Full K28.5 detect
  - Single or Multi-byte framer for byte alignment
  - Low-latency option
- Skew alignment support for multiple bytes of offset
- Serial Built-In-Self-Test (BIST) for at-speed link testing
- Per-channel Link Quality Indicator
  - Analog signal detect
  - Digital signal detect
  - Frequency range detect

High-Speed PSI Features[1]

- 2.5 Gbps/channel serial signaling rate
- Full Bellcore and ITU jitter compliance
- Power-saving mode
- Up to two serial channels available to allow:
  - High-Bandwidth
  - Redundancy
- Supported standards:
  - InfiniBand™
  - SONET/SDH OC-48

Development Software

- *Warp*®
  - IEEE 1076/1164 VHDL or IEEE 1364 Verilog context sensitive editing
  - Active-HDL FSM graphical finite state machine editor
  - Active-HDL SIM post-synthesis timing simulator
  - Architecture Explorer for detailed design analysis
  - Static Timing Analyzer for critical path analysis
  - Available on Windows 95, 98 & NT for $99
  - Supports all Cypress programmable logic products

Note:
1. For detailed data sheet see “High-Speed PSI data sheet.”

Table 1. PSI Quick Reference Selection Guide

Logic Gate Density	High-Speed/SONET/SDH PSI Serial Bandwidth		Frequency-Agile PSI Serial Bandwidth	
	1 x 2.5 Gbps	2 x 2.5 Gbps	4 x 0.2–1.5 Gbps	8 x 0.2–1.5 Gbps
100K	S25G01K100	S25G02K100	P15G04K100	
200K		S25G02K200	P15G04K200	P15G08K200

PSI Family Standards Supported

PSI Device		SONET/SDH (OC-48)	Infiniband	Fibre Channel	Gigabit Ethernet	ESCON	DVB	SMPTE 259	SMPTE 292
SONET/SDH	S25G01K100	X							
High Speed	P25G01K100		X						
Frequency Agile	P15G04K100			X	X	X	X	X	X
	P15G04K200			X	X	X	X	X	X
	P15G08K200			X	X	X	X	X	X

PSI Family General Selection Guide

Device	Typical Gates	Macro-cells	Cluster memory (Kbits)	Channel memory (Kbits)	Maximum User Programmable I/O	Package Offering
25G01K100	46K–144K	1536	192	48	240	456-BGA (35x35 mm, 1.27-mm pitch)
15G04K100	46K–144K	1536	192	48	206	456-BGA (35x35 mm, 1.27-mm pitch)
15G04K200	92K–288K	3072	384	96	332	700-BGA (40x40 mm, 1.27-mm pitch)
15G08K200	92K–288K	3072	384	96	206	700 BGA (40x40 mm, 1.27-mm pitch)

PSI Family Performance Selection Guide

Device	Channels & Link Speed	Total Bandwidth	f_{MAX2} (MHz)	Logic Speed — t_{PD} Pin-to-Pin (ns)	Standby I_{CC}[2]
25G01K100	1 x 2.5 Gbps	2.5 Gbps	222	7.5	16 mA
15G04K100	4 x 0.2 - 1.5 Gbps	6.0 Gbps	222	7.5	18 mA
15G04K200	4 x 0.2 - 1.5 Gbps	6.0 Gbps	181	8.5	18 mA
15G08K200	8 x 0.2 - 1.5 Gbps	12.0 Gbps	181	8.5	26 mA

Note:
2. Standby I_{CC} values are with PLL not utilized, no output load, and stable inputs.

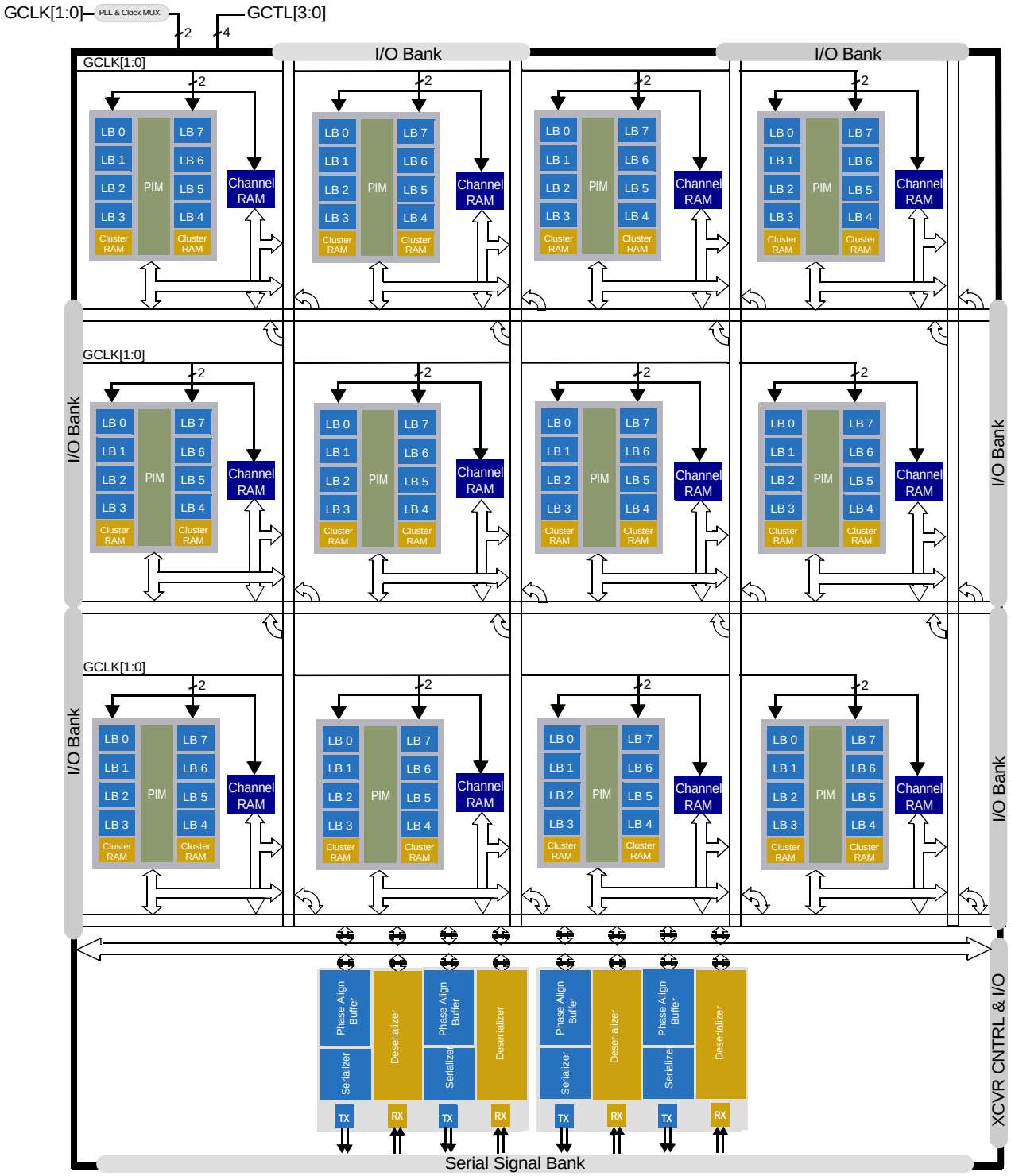


Figure 1. Frequency Agile PSI Block Diagram (CYP15G04K100) with I/O Bank Structure

PSI15G04K100V1 Pin Configuration (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	GND	OUTB2-	INB2-	VCC	OUTB1-	INB1-	VCC	OUTA2-	INA2-	VCC	OUTA1-	INA1-	VCC	OUTD2-	IND2-	VCC	OUTD1-	IND1-	VCC	OUTC2-	INC2-	VCC	OUTC1-	INC1-	GND	GND
B	GND	OUTB2+	INB2+	VCC	OUTB1+	INB1+	VCC	OUTA2+	INA2+	VCC	OUTA1+	INA1+	VCC	OUTD2+	IND2+	VCC	OUTD1+	IND1+	VCC	OUTC2+	INC2+	VCC	OUTC1+	INC1+	GND	GND
C	CONFIG_DO	BOND_INHn	RXRATE	VCC	RXLE	LPEN	VCC	RFEN	TXCLKA	VCC	MASTERn	SPDSEL	VCC	SDASEL	BOE_6	VCC	BOE_7	BOE_4	VCC	BOE_5	TRSTZn	VCC	RXCKSEL	BOND_ALL	GND	GND
D	DATA	BOE_1	BOE_0	VCC	LFIAn	TXMODE_0	VCC	TXMODE_1	RXMODE_0	VCC	RXMODE_1	TXRATE	VCC	INSELC	INSELD	VCC	RXCLKD+	INSELA	VCC	INSELB	RFMODE	VCC	LFICn	TXRSTn	BONDST_0	TXCKSEL
E	RECONFIG	BOE_3	BOE_2	VCC	VCC	VCC	VCC	VCC	RXCLKA+	VCC	REFCLK+	REFCLK-	VCC	TXCLKO+	VCC	VCC	VCC	VCC	VCC	VCC	VCC	VCC	RXCLKC+	LFIDn	BONDST_1	GND
F	RESET	VCCCNFG	BISTLE	FRAMCHAR	VCC																	VCC	IO/VREF0	IO/VREF0	IO0	IO0
G	CCLK	TDO	OELE	DECMODE	VCC																	VCC	IO0	IO0	IO0	IO0
H	CCE	MSEL	RXCLKB+	LFIBn	VCC																	VCC	IO7	IO/VREF7	IO7	IO7
J	IO/VREF2	IO2	IO2	IO2	VCC																	VCC	IO7	IO7	IO7	IO7
K	IO2	IO2	IO2	IO2	VCCO1																	VCCO0	IO6	IO6	IO6	IO7
L	IO/VREF2	IO2	IO2	IO2	VCCO1						GND	GND	GND	GND	GND	GND						VCCO0	IO6	IO6	IO/VREF6	IO6
M	IO2	IO2	IO2	IO2	VCCO1						GND	GND	GND	GND	GND	GND						VCCO0	GCLK0	IO6	IO6	IO6
N	IO2	IO/VREF2	IO2	IO2	VCCO1						GND	GND	GND	GND	GND	GND						VCCO0	GCTL0	IO/VREF6	IO6	IO6/Lock
P	IO2	IO2	IO2	IO2	VCCPRG						GND	GND	GND	GND	GND	GND						VCCore	IO6	GCTL3	IO6	IO6
R	IO2	IO2	IO2	IO/VREF2	VCCO2						GND	GND	GND	GND	GND	GND						VCCO7	IO6	GCLK3	IO6	IO6
T	IO2	IO/VREF2	IO2	IO2	VCCO2						GND	GND	GND	GND	GND	GND						VCCO7	IO6	IO6	IO/VREF6	IO6
U	IO2	IO2	IO2	IO2	VCCO2																	VCCO7	IO6	IO6	IO6	IO/VREF6
V	IO2	IO/VREF2	IO2	IO2	VCCO2																	VCCO7	IO6	IO/VREF6	IO6	IO6
W	IO3	IO3	IO2	IO2	VCCO2																	VCCO7	IO6	IO6	IO6	IO6
Y	IO3	IO3	IO3	IO/VREF3	VCCore																	VCCPLL	IO/VREF6	IO6	IO6	IO6
AA	IO3	IO3	IO3	IO3	VCCO3																	VCCO6	IO6	GCTL2	IO6	IO6
AB	IO/VREF3	IO3	IO3	IO3	VCCO3	VCCO3	VCCO3	VCCO3	VCCO4	VCCO4	VCCO4	VCCO4	VCCore	VCCore	VCCO5	VCCO5	VCCO5	VCCO5	VCCO6	VCCO6	VCCO6	VCCO6	GCTL1	GCLK2	VCCJTAG	TMS
AC	IO3	IO3	IO3	IO3	IO/VREF3	IO3	IO3	IO/VREF3	IO4	IO4	IO4	IO4	IO4	IO4	IO4	IO4	IO5	IO5	IO5	IO5	IO5	IO5	GCLK1	IO5	TDI	TCK
AD	IO3	IO3	IO3	IO/VREF3	IO3	IO/VREF3	IO3	IO3	IO4	IO4	IO/VREF4	IO4	IO/VREF4	IO4	IO4	IO4	IO5	IO5	IO5	IO5	IO5	IO5	IO/VREF5	IO5	GND	GND
AE	GND	GND	IO3	IO3	IO3	IO3	IO3	IO3	IO4	IO4	IO4	IO4	IO4	IO4	IO/VREF4	IO4	IO/VREF5	IO5	IO5	IO5	IO/VREF5	IO5	IO5	IO5	GND	GND
AF	GND	GND	IO3	IO3	IO3	IO3	IO3	IO3	IO/VREF4	IO4	IO4	IO4	IO4	IO4	IO4	IO5	IO5	IO5	IO/VREF5	IO5	IO5	IO5	IO5	IO5	GND	GND

Note:

3. Signals appended by “n” are active LOW signals.

Functional Description

The Programmable Serial Interface (PSI) family is a point-to-point or point-to-multipoint programmable communications building block allowing the manipulation and transfer of data over high-speed serial links at signaling speeds ranging from 200 Mbps to 1.5 and 2.5 Gbps per serial link. The PSI family is designed to combine the high speed, predictable timing, high density, low power, and ease-of-use of complex programmable logic devices (CPLDs) with the serializing/deserializing (SERDES) capability of high-speed serial transceivers. The family is divided into two groups: High-Speed PSI and Frequency Agile PSI. Both groups have unique transceiver characteristics that define the specific transceiver block operation of a given PSI device.

The architecture of the device is based on logic block clusters (LBC) and serial transceiver blocks that are connected by horizontal and vertical routing channels. Each LBC features eight individual logic blocks (LB) of 16 macrocells and two cluster memory blocks. Adjacent to each LBC is a channel memory block, which is externally accessible through the I/O interface. Each transmit channel of the transceiver accepts parallel characters, encodes each character for transport and converts it to serial data. Each receive channel accepts serial data and converts it to parallel data, decoding the data into characters and presents these characters to the routing channels of the PSI unit.

Frequency Agile Devices

The transceiver operation of the Frequency Agile Programmable Serial Interface devices is self-contained in a single block. It has separate transmit and receive PLLs and a clock and data recovery (CDR) unit for flexible clocking. The transmit channel accepts an 8-bit unencoded or 10-bit encoded input character from the routing channels and passes the character to an elasticity buffer. This character is then serialized and output on dual differential transmission-line drivers at the required bit-rate. The receive channel accepts a serial bit-stream from the two differential line receivers. This bit-stream is deserialized and an 8-bit unencoded or 10-bit encoded character is presented to the routing channels in the PSI device. The block also features Built-In Self Test (BIST) mode for simplified design debugging.

Global Routing Description

The routing architecture in the PLD block of a PSI device is made up of horizontal and vertical (H&V) routing channels. These routing channels allow signals to move among I/Os, logic blocks and memories. In addition to the horizontal and vertical routing channels that interconnect the I/O banks, channel memory blocks, transceiver blocks and logic block clusters, each LBC contains a Programmable Interconnect Matrix (PIM™), which is used to route signals among the logic blocks and the cluster memory blocks in the LBC.

Figure 3 is a block diagram of the routing channels that interface within the PSI architecture. The LBC is exactly the same for every member of the PSI family.

Transceiver Block

Each transceiver block of a given PSI device will have one serializer transmit path and one deserializer receive path operating at a speed from 200 Mbps to 1.5 Gbps. The transceiver block interfaces to the routing channels of the PSI device through highly configurable datapath cells. For specific architecture and operation of the transceiver blocks please refer to the Serial Transceiver Operation section (page 17).

Frequency Agile PSI Transceiver Blocks

Frequency Agile PSI devices include four or eight transceiver blocks operating at 0.2 to 1.5 Gbps per channel. They use the same reference clock.

The internal interfacing to the transceiver blocks of the device occur through the port definition of the transceiver block. The internal signals and their definition are described in the Pin and Signal Description section (page 62).

Standard Datapath Cell

Figure 2 is a block diagram of the PSI datapath cell. The datapath cell contains a three-state transmit buffer, a receive buffer, and a register that can be configured as an transmit or receive register.

The transceiver enable (TE) can be selected from one of the four global control signals or from one of two Output Control Channel (OCC) signals. The transmit enable can be configured as always enabled or always disabled or it can be controlled by one of the remaining inputs to the mux. The selection is done via a mux that includes V_{CC} and GND as inputs.

One of the global clocks can be selected as the clock for the datapath cell register. The clock mux output is an input to a clock polarity mux that allows the transmit/receive register to be clocked on either edge of the clock.

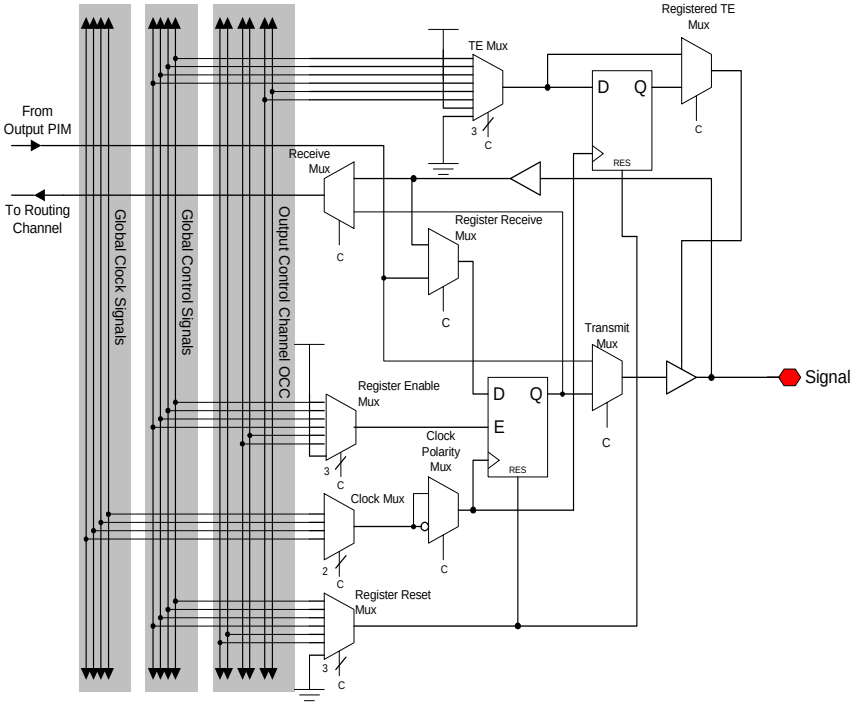


Figure 2. Block Diagram of a Standard Datapath Cell

Logic Block Cluster (LBC)

The PSI architecture consists of several logic block clusters, each of which have 8 Logic Blocks (LB) and 2 cluster memory blocks connected via a Programmable Interconnect Matrix (PIM) as shown in *Figure 4*. Each cluster memory block consists of 8-Kbit single-port RAM, which is configurable as synchronous or asynchronous. The cluster memory blocks can be cascaded with other cluster memory blocks within the same LBC as well as other LBCs to implement larger memory functions. If a cluster memory block is not specifically utilized by the designer, Cypress's *Warp*® software can automatically use it to implement large blocks of logic.

All LBCs interface with each other via horizontal and vertical routing channels.

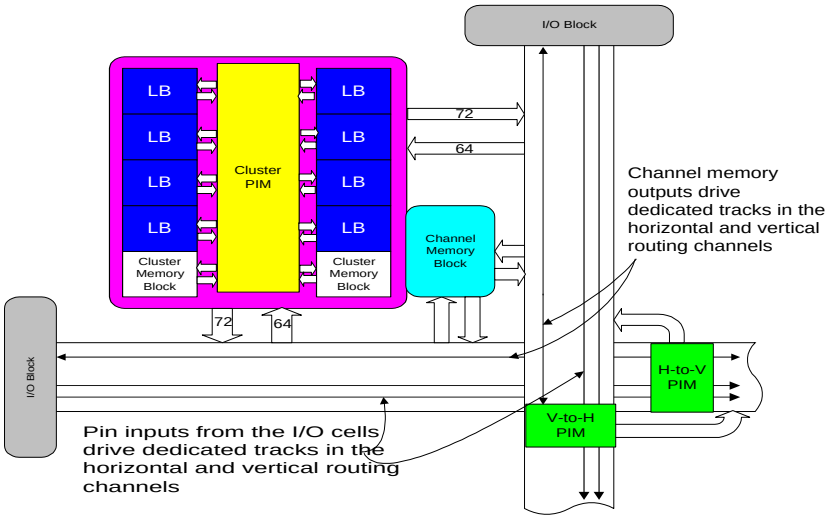


Figure 3. PSI Routing Interface

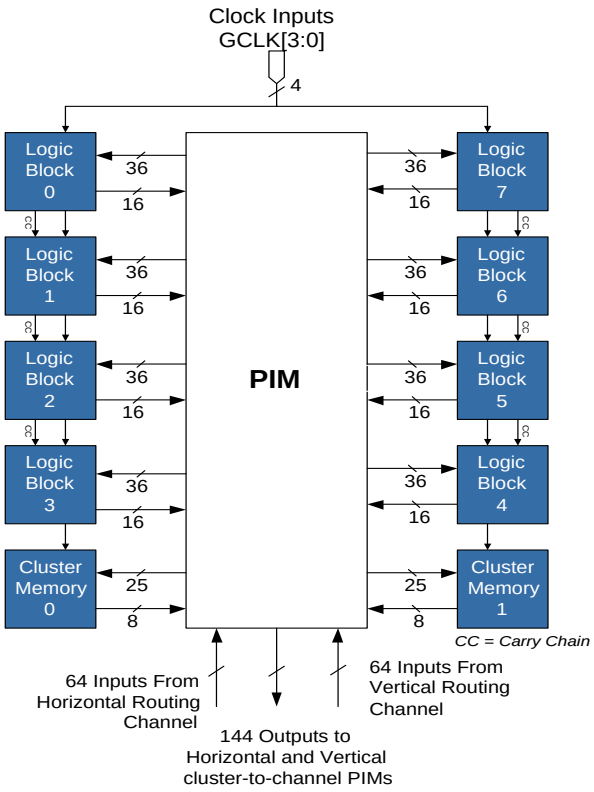


Figure 4. PSI Logic Block Cluster Diagram

Logic Block (LB)

The logic block is the basic building block of the PSI architecture. It consists of a product term array, an intelligent product-term allocator, and 16 macrocells.

Product Term Array

Each logic block features a 72 x 83 programmable product term array. This array accepts 36 inputs from the PIM. These inputs originate from device pins and macrocell feedbacks as well as cluster memory and channel memory feedbacks. Active LOW and active HIGH versions of each of these inputs are generated to create the full 72-input field. The 83 product terms in the array can be created from any of the 72 inputs.

Of the 83 product terms, 80 are for general-purpose use for the 16 macrocells in the logic block. Two of the remaining three product terms in the logic block are used as asynchronous set and asynchronous reset product terms. The final product term is the Product Term clock (PTCLK) and is shared by all 16 macrocells within a logic block.

Product Term Allocator

Through the product term allocator, *Warp* software automatically distributes the 80 product terms as needed among the 16 macrocells in the logic block. The product term allocator provides two important capabilities without affecting performance: product term steering and product term sharing.

Product Term Steering

Product term steering is the process of assigning product terms to macrocells as needed. For example, if one macrocell requires ten product terms while another needs just three, the product term allocator will “steer” ten product terms to one macrocell and three to the other. On PSI devices, product terms are steered on an individual basis. Any number between 1 and 16 product terms can be steered to any macrocell.

Product Term Sharing

Product term sharing is the process of using the same product term among multiple macrocells. For example, if more than one function has one or more product terms in its equation that are common to other functions, those product terms are only created once. The PSI product term allocator allows sharing across groups of four macrocells in a variable fashion. The software automatically takes advantage of this capability so that the user does not have to intervene.

Note that neither product term sharing nor product term steering have any effect on the speed of the product. All steering and sharing configurations have been incorporated in the timing specifications for the PSI devices.

Macrocell

Within each logic block there are 16 macrocells. Each macrocell accepts a sum of up to 16 product terms from the product term array. The sum of these 16 product terms can be output in either registered or combinatorial mode. *Figure 5* displays the block diagram of the macrocell. The register can be asynchronously preset or asynchronously reset at the macrocell level with the separate preset and reset product terms. Each of these product terms features programmable polarity. This allows the registers to be preset or reset based on an AND expression or an OR expression.

An XOR gate in the PSI macrocell allows for many different types of equations to be realized. It can be used as a polarity mux to implement the true or complement form of an equation in the product term array or as a toggle to turn the D flip-flop into a T flip-flop. The carry-chain input mux allows additional flexibility for the implementation of different types of logic. The macrocell can utilize the carry chain logic to implement adders, subtractors, magnitude comparators, parity tree, or even generic XOR logic. The output of the macrocell is either registered or combinatorial.

Carry Chain Logic

The PSI macrocell features carry chain logic, which is used for fast and efficient implementation of arithmetic operations. The carry logic connects macrocells in up to 4 logic blocks for a total of 64 macrocells. Effective data path operations are implemented through the use of carry-in arithmetic, which drives through the circuit quickly. *Figure 5* shows that the carry chain logic within the macrocell consists of two product terms (CPT0 and CPT1) from the PTA and an input carry-in for carry logic. The inputs to the carry chain mux are connected directly to the product terms in the PTA. The output of the carry chain mux generates the carry-out for the next macrocell in the logic block as well as the local carry input that is connected to an input of the XOR input mux. Carry-in and a configuration bit are inputs to an AND gate. This AND gate provides a method of segmenting the carry chain in any macrocell in the logic block.

Macrocell Clocks

Clocking of the register is highly flexible. Four global synchronous clocks (GCLK[3:0]) and a Product Term clock (PTCLK) are available at each macrocell register. Furthermore, a clock polarity mux within each macrocell allows the register to be clocked on the rising or the falling edge (see macrocell diagram in *Figure 5*).

PRESET/RESET Configurations

The macrocell register can be asynchronously preset and reset using the PRESET and RESET mux. Both signals are active high and can be controlled by either of two Preset/Reset product terms (PRC[1:0] in *Figure 5*) or GND. In situations where the PRESET and RESET are active at the same time, RESET takes priority over PRESET.

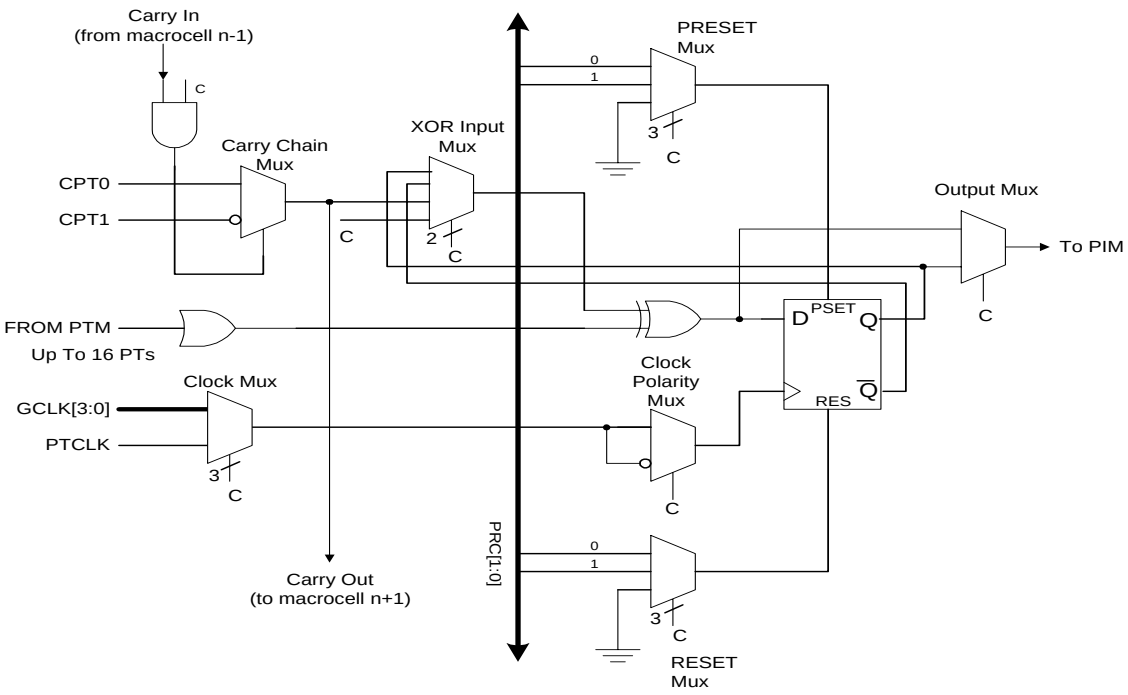


Figure 5. PSI Macrocell

Embedded Memory

Each member of the PSI family contains two types of embedded memory blocks. The channel memory block is placed at the intersection of horizontal and vertical routing channels. Each channel memory block is 4096 bits in size and can be configured as asynchronous or synchronous Dual-Port RAM, Single-Port RAM, Read-Only memory (ROM), or synchronous FIFO memory. The memory organization is configurable as 4Kx1, 2Kx2, 1Kx4 and 512x8. The second type of memory block is located within each LBC and is referred to as a cluster memory block. Each LBC contains two cluster memory blocks that are 8192 bits in size. Similar to the channel memory blocks, the cluster memory blocks can be configured as 8Kx1, 4Kx2, 2Kx4 and 1Kx8 and can be configured as either asynchronous or synchronous Single-Port RAM or ROM.

Cluster Memory

Each logic block cluster of the PSI device contains two 8192-bit cluster memory blocks. *Figure 6* is a block diagram of the cluster memory block and the interface of the cluster memory block to the cluster PIM.

The output of the cluster memory block can be optionally registered to perform synchronous pipelining or to register asynchronous read and write operations. The output registers contain an asynchronous RESET, which can be used in any type of sequential logic circuits (e.g., state machines)

There are four global clocks (GCLK[3:0]) and one local clock available for the input and the output registers. The local clock for the input registers is independent of the one used for the output registers. The local clock is generated in the user-design in a macrocell or comes from an I/O pin

Cluster Memory Initialization

The cluster memory powers up in an undefined state, but is set to a user-defined known state during configuration. To facilitate the use of look-up-table (LUT) logic and ROM applications, the cluster memory blocks can be initialized with a given set of data when the device is configured at power-up. For LUT and ROM applications, the user cannot write to memory blocks.

Channel Memory

The PSI architecture includes an embedded memory block at each crossing point of horizontal and vertical routing channels. The channel memory is a 4096-bit embedded memory block that can be configured as asynchronous or synchronous Single-Port RAM, Dual-Port RAM, ROM, or synchronous FIFO memory.

Data, address, and control inputs to the channel memory are driven from horizontal and vertical routing channels. All data and FIFO logic outputs drive dedicated tracks in the horizontal and vertical routing channels. The clocks for the channel memory block are selected from four global clocks and pin inputs from the horizontal and vertical channels. The clock muxes also include a polarity mux for each clock so that the user can choose an inverted clock.

Dual-Port (Channel Memory) Configuration

Each port has distinct address inputs, as well as separate data and control inputs that can be accessed simultaneously. The inputs to the Dual-Port memory are driven from the horizontal and vertical routing channels. The data outputs drive dedicated tracks in the routing channels. The interface to the routing is such that Port A of the Dual-Port interfaces primarily with the horizontal routing channel and Port B interfaces primarily with the vertical routing channel.

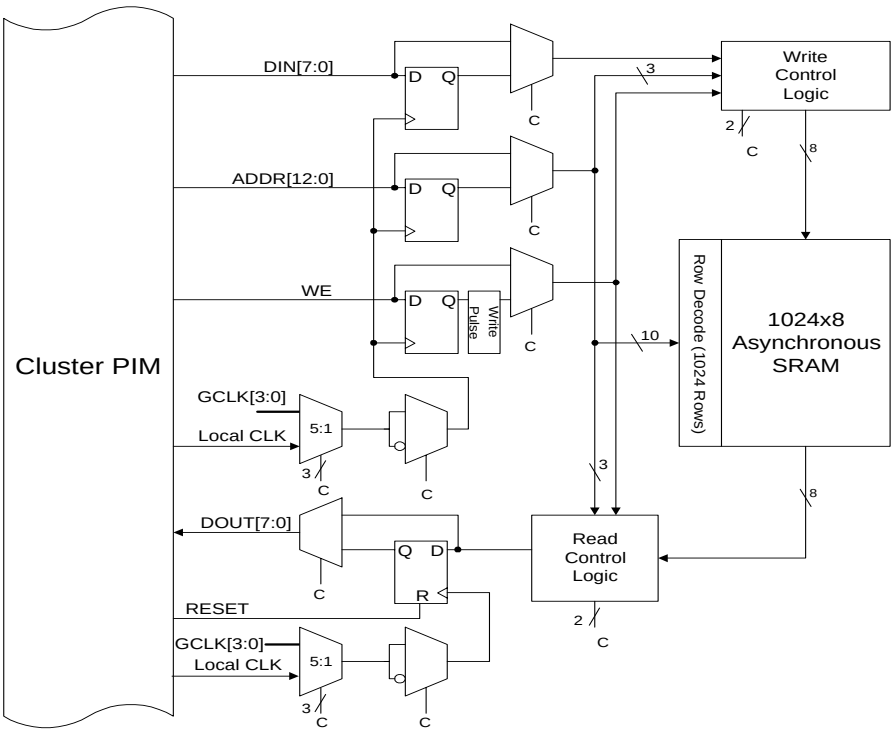


Figure 6. Block Diagram of Cluster Memory Block

The clocks for each port of the Dual-Port configuration are selected from four global clocks and two local clocks. One local clock is sourced from the horizontal channel and the other from the vertical channel. The data outputs of the dual-port memory can also be registered. Clocks for the output registers are also selected from four global clocks and two local clocks. One clock polarity mux per port allows the use of true or complement polarity for input and output clocking purposes.

Arbitration

The Dual-Port configuration of the Channel Memory Block provides arbitration when both ports access the same address at the same time. Depending on the memory operation being attempted, one port always gets priority. See *Table 2* for details on which port gets priority for read and write operations. An active-LOW 'Address Match' signal is generated when an address collision occurs.

Table 2. Arbitration Result: Address Match Signal Becomes Active

Port A	Port B	Result of Arbitration	Comment
Read	Read	No arbitration required	Both ports read at the same time
Write	Read	Port A gets priority	If Port B requests first then it will read the current data. The output will then change to the newly written data by Port A
Read	Write	Port B gets priority	If Port A requests first then it will read the current data. The output will then change to the newly written data by Port B
Write	Write	Port A gets priority	Port B is blocked until Port A is finished writing

FIFO (Channel Memory) Configuration

The channel memory blocks are also configurable as synchronous FIFO RAM. In the FIFO mode of operation, the channel memory block supports all normal FIFO operations without the use of any general-purpose logic resources in the device.

The FIFO block contains all of the necessary FIFO flag logic, including the read and write address pointers. The FIFO flags include an empty/full flag ($\overline{EF}$), half-full flag ($\overline{HF}$), and programmable almost-empty/full ($\overline{PAEF}$) flag output. The FIFO configuration has the ability to perform simultaneous read and write operations using two separate clocks. These clocks may be tied together for a single operation or may run independently for asynchronous read/write (w.r.t. each other) applications. The data and control inputs to the FIFO block are driven from the horizontal or vertical routing channels. The data and flag outputs are driven onto dedicated routing tracks in both the horizontal and vertical routing channels. This allows the FIFO blocks to be expanded by using multiple FIFO blocks on the same horizontal or vertical routing channel without any speed penalty.

In FIFO mode, the write and read ports are controlled by separate clock and enable signals. The clocks for each port are selected from four global clocks and two local clocks. One local clock is sourced from the horizontal channel and the other from the vertical channel. The data outputs from the read port of the FIFO can also be registered. One clock polarity mux per port allows using true or complement polarity for read and write operations. The write operation is controlled by the clock and the write enable pin. The read operation is controlled by the clock and the read enable pin. The enable pins can be sourced from horizontal or vertical channels.

Channel Memory Initialization

The channel memory powers up in an undefined state, but is set to a user-defined known state during configuration. To facilitate the use of look-up-table (LUT) logic and ROM applications, the channel memory blocks can be initialized with a given set of data when the device is configured at power up. For LUT and ROM applications, the user cannot write to memory blocks.

Channel Memory Routing Interface

Similar to LBC outputs, the channel memory blocks feature dedicated tracks in the horizontal and vertical routing channels for the data outputs and the flag outputs, as shown in *Figure 7*. This allows the channel memory blocks to be expanded easily. These dedicated lines can be routed to I/O pins as chip outputs or to other logic block clusters to be used in logic equations.

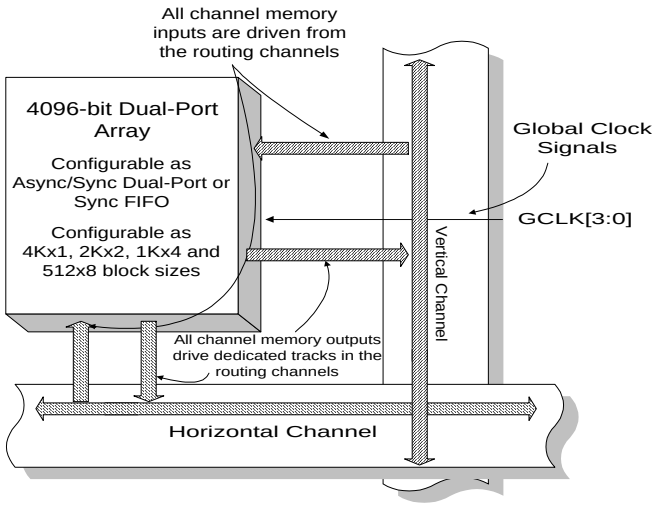


Figure 7. Block Diagram of Channel Memory Block

I/O Banks

The PSI interfaces the horizontal and vertical routing channels to the pins through I/O banks. There are several I/O banks per device as shown in *Figure 8* and all I/Os from an I/O bank are located in the same section of a package for PCB layout convenience. There are two kinds of I/O banks; fixed-signal I/O banks and user-programmable I/O banks.

The first fixed-signal bank is the Serial Signal Bank. This bank includes all differential serial data transmission and receive signals. The second bank is the Transceiver Control Bank. This bank includes all static signal pins required for the configuration and operation of the transceiver blocks in each of the PSI devices.

Each PSI device has several types of user-programmable I/O banks. *Table 3* indicates the availability of each type of programmable bank by device. Supported I/O standards for each bank are addressed by the appropriate V_{REF} and V_{CCIO} voltages. All the V_{REF} and V_{CCIO} pins in an I/O bank must be connected to the same V_{REF} and V_{CCIO} voltage respectively. This requirement restricts the number of I/O standards supported by an I/O bank at any given time. It also dictates the I/O standard used for the GCTL[3:0] pins.

The architecture defining each programmable I/O bank consists of several I/O cells, where each I/O cell contains an input/output register, an output enable register, programmable slew rate control, and programmable bus hold control logic. Each I/O cell drives a pin output of the device; the cell also supplies an input to the device that connects to a dedicated track in the associated routing channel.

There are four dedicated inputs (GCTL[3:0]) that are used as Global Control Signals available to every I/O cell. These global control signals may be used as output enables, register resets and register clock enables as shown in *Figure 9*.

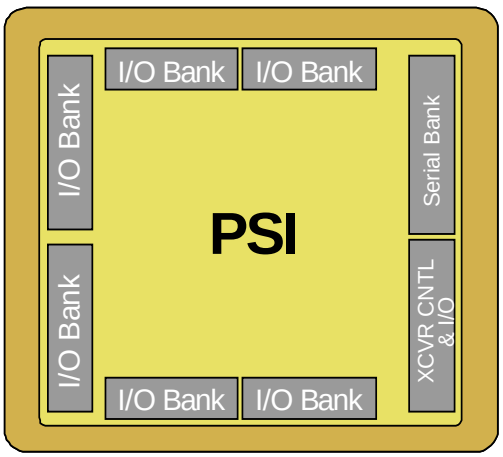


Figure 8. PSI I/O Bank Block Diagram

Table 3. IO Standards

I/O Standard	V_{REF} (V)		V_{CCIO}	Termination Voltage (V_{TT})
	Min.	Max.		
LVTTL	N/A		3.3V	N/A
LVCMOS			3.3V	N/A
LVCMOS3			3.0V	N/A
LVCMOS2			2.5V	N/A
LVCMOS18			1.8V	N/A
3.3V PCI			3.3V	N/A
GTL+	0.9	1.1	N/A	1.5
SSTL3 I	1.3	1.7	3.3V	1.5
SSTL3 II	1.3	1.7	3.3V	1.5
SSTL2 I	1.15	1.35	2.5V	1.25
SSTL2 II	1.15	1.35	2.5V	1.25
HSTL I	0.68	0.9	1.5V	0.75
HSTL II	0.68	0.9	1.5V	0.75
HSTL III	0.68	0.9	1.5V	1.5
HSTL IV	0.68	0.9	1.5V	1.5

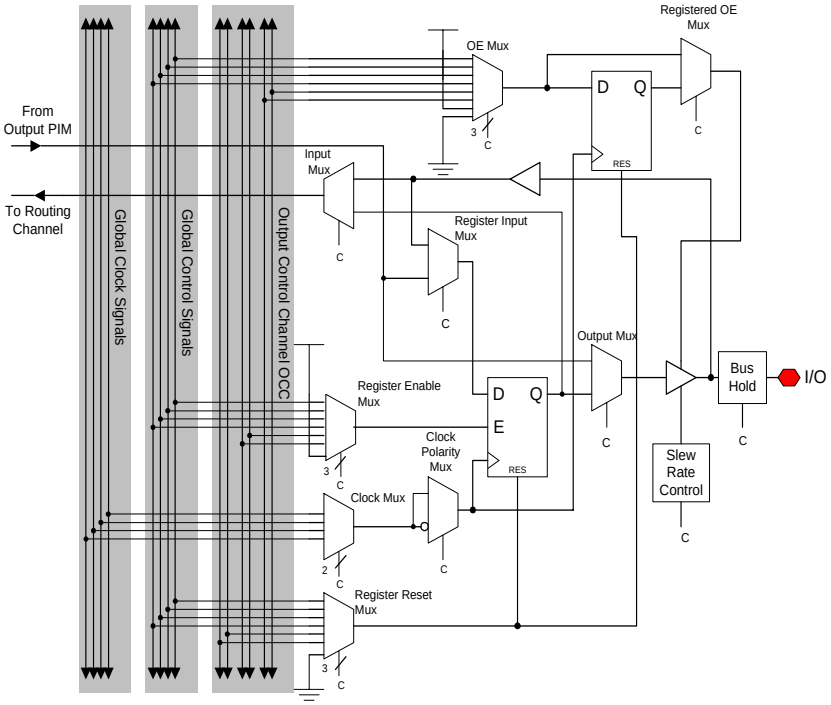


Figure 9. Block Diagram of I/O Cell

I/O Cell

Figure 9 is a block diagram of the PSI I/O cell. The I/O cell contains a three-state input buffer, an output buffer, and a register that can be configured as an input or output register. The output buffer has a slew rate control option that can be used to configure the output for a slower slew rate. The input of the device and the pin output can each be configured as registered or combinatorial, however only one path can be configured as registered in a given design.

The output enable can be selected from one of the four global control signals or from one of two Output Control Channel (OCC) signals. The output enable can be configured as always enabled or always disabled, or it can be controlled by one of the remaining inputs to the mux. The selection is done via a mux that includes V_{CC} and GND as inputs.

One of the global clocks can be selected as the clock for the I/O cell register. The clock mux output is an input to a clock polarity mux that allows the input/output register to be clocked on either edge of the clock.

Slew Rate Control

The output buffer has a slew rate control option. This allows the ouput buffer to slew at a fast rate (3 V/ns) or a slow rate (1 V/ns). All I/Os default to fast slew rate. For designs concerned with meeting FCC emissions standards the slow edge provides for lower system noise. For designs requiring very high performance the fast edge rate provides maximum system performance.

Programmable Bus Hold

On each I/O pin, user-programmable bus-hold is included. Bus-hold, which is an improved version of the popular internal pull-up resistor, is a weak latch connected to the pin that does not degrade the device's performance. As a latch, bus-hold maintains the last state of a pin when the pin is placed in a high-impedance state, thus reducing system noise in bus-interface applications. Bus-hold additionally allows unused device pins to remain unconnected on the board, which is particularly useful during prototyping as designers can route new signals to the device without cutting trace connections to V_{CC} or GND. For more information, see the application note "Understanding Bus-Hold – A Feature of Cypress CPLDs."

Clocks

PSI - Frequency Agile devices have four dedicated clock input pins (GCLK[3:0]) to accept system clocks. One of these clocks (GCLK[0]) may be selected to drive an on-chip Phase-Locked Loop (PLL) for frequency modulation (see *Figure 10* for details). The global clock tree for a PSI - Frequency Agile devices can be driven by a combination of the dedicated clock pins and/or the PLL-derived clocks. The global clock tree consists of four global clocks that go to every macrocell, memory block, and I/O cell.

Clock Tree Distribution

The global clock tree performs two primary functions. First, the clock tree generates the four global clocks by multiplexing four dedicated clocks from the package pins and four PLL driven clocks. Second, the clock tree distributes the four global clocks to every cluster, channel memory, and I/O block on the die. The global clock tree is designed such that the clock skew is minimized while maintaining an acceptable clock delay.

Spread Aware™ PLL

Each device in the PSI family features an on-chip PLL designed using Spread Aware™ technology for low EMI applications. In general, PLLs are used to implement time-division-multiplex circuits to achieve higher performance with fewer device resources.

For example, a system that operates on a 32-bit data path that runs at 40 MHz can be implemented with 16-bit circuitry that runs internally at 80 MHz. PLLs can also be used to take advantage of the positioning of the internally generated clock edges to shift performance towards improved setup, hold or clock-to-out times.

There are several frequency multiply (X1, X2, X4, X8) and divide (/1, /2, /3, /4, /5, /6. /8, /16) options available to create a wide range of clock frequencies from a single clock input (GCLK[0]). For increased flexibility, there are seven phase shifting options which allow clock skew/de-skew by 45°, 90°, 135°, 180°, 225°, 270° or 315°.

The Spread Aware feature refers to the ability of the PLL to track a spread-spectrum input clock such that its spread is seen on the output clock with the PLL staying locked. The total amount of spread on the input clock should be limited to 0.6% of the fundamental frequency. Spread Aware feature is supported only with X1, X2 and X4 multiply options.

The Voltage Controlled Oscillator (VCO), the core of the PSI PLL is designed to operate within the frequency range of 100 MHz to 266 MHz. Hence, the multiply option combined with input (GCLK[0]) frequency should be selected such that this VCO operating frequency requirement is met. This is demonstrated in *Table 4* (columns 1, 2, and 3).

Another feature of this PLL is the ability to drive the output clock (INTCLK) off the PSI chip to clock other devices on the board, as shown in *Figure 11* below. This off-chip clock is half the frequency of the output clock as it has to go through a register (I/O register or a macrocell register).

This PLL can also be used for board deskewing purpose by driving a PLL output clock off-chip, routing it to the other devices on the board and feeding it back to the PLL's external feedback input (GCLK[1]). When this feature is used, only limited multiply, divide and phase shift options can be used.

Table 4 describes the valid multiply and divide options that can be used without an external feedback. *Table 5* describes the valid multiply & divide options that can be used with an external feedback.

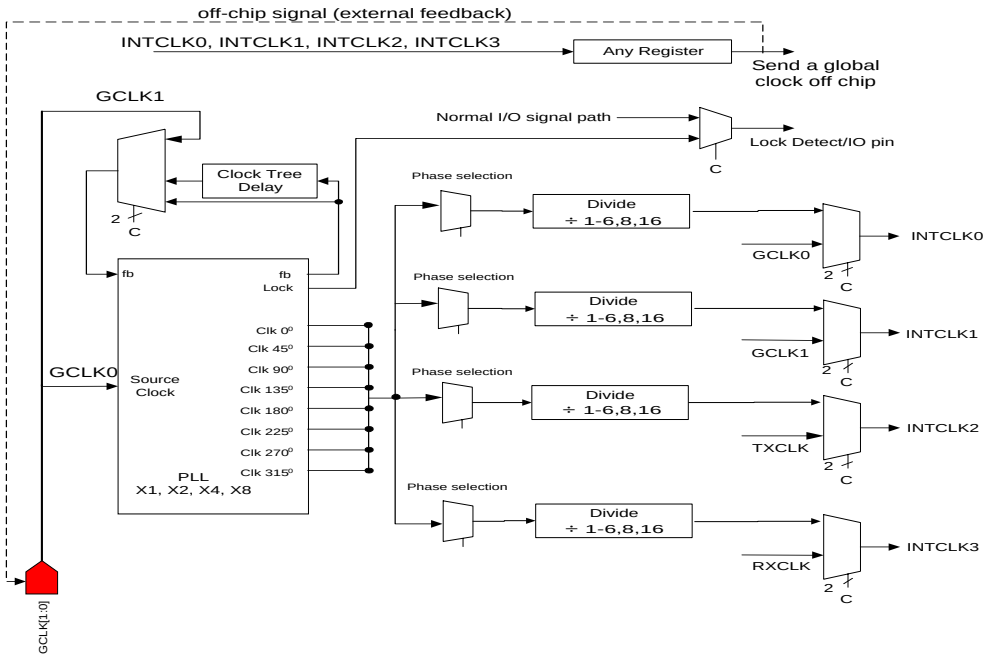


Figure 10. Block Diagram of Spread Aware PLL for CYS25G01K100

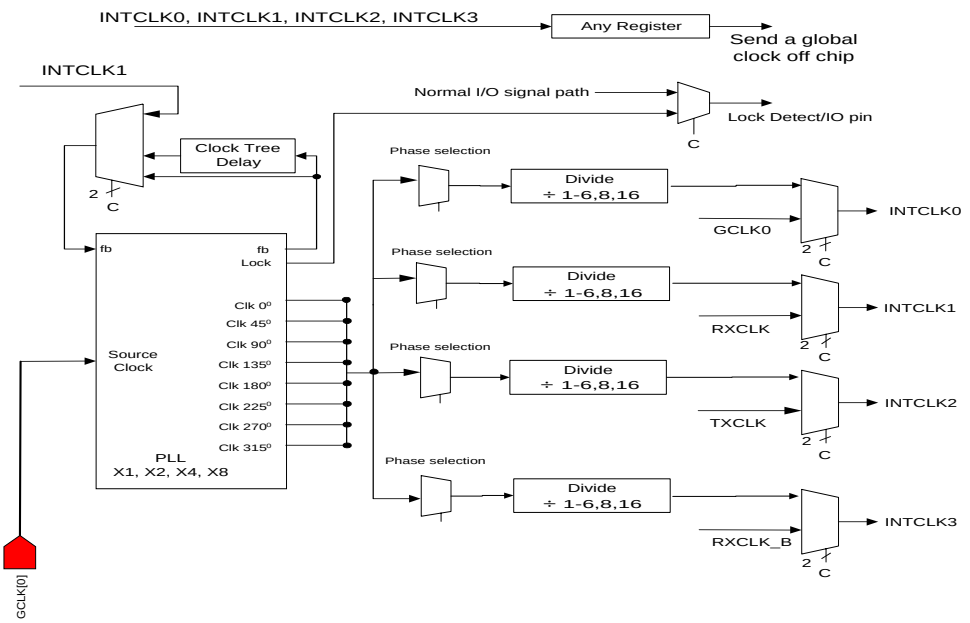


Figure 11. Block Diagram of Spread Aware PLL for CYS25G02K100

Table 4. PLL Multiply and Divide Options—without INTCLK1 Feedback

Input Frequency (GCLK[0]) f_{PLLI} (MHz)	Valid Multiply Options		Valid Divide Options		
	Value	VCO Output Frequency (MHz)	Value	Output Frequency (INTCLK[3:0]) f_{PLLO} (MHz)	Off-chip Clock Frequency
12.5-25	8	100-200	1–6, 8, 16	6.25–200	3.12-100
25–33	8	200–266	1–6, 8, 16	12.5–266	6.25–133
	4	100–133	1–6, 8, 16	6.25–133	3.12–66
33–50	4	133–200	1–6, 8, 16	8.33–200	4.16–100
50–66	4	200–266	1–6, 8, 16	12.5–266	6.25–133
	2	100–133	1–6, 8, 16	6.25–133	3.12–66
66–100	2	133–200	1–6, 8, 16	8.3–200	4.16–100
100–133	2	200–266	1–6, 8, 16	12.5–266	6.25–133
	1	100–133	1–6, 8, 16	6.25–133	3.12–66

Table 5. PLL Multiply and Divide Options—with External Feedback

Input (GCLK) Frequency f_{PLLI} (MHz)	Valid Multiply Options		Valid Divide Options		
	Value	VCO Output Frequency (MHz)	Value	Output (INTCLK) Frequency f_{PLLO} (MHz)	Off-chip Clock Frequency
50–66	1	100–133	1	100–133	50–66
66–100	1	133–200	1	133–200	66–100
100–133	1	200–266	1	200–266	100–133

Table 6 describes the valid phase shift options that can be used with or without an external feedback.

Table 6. PLL Phase Shift Options—with and without INTCLK1 Feedback

Without External Feedback	With External Feedback
0°,45°, 90°, 135°, 180°, 225°, 270°, 315°	0°

Table 7 is an example of the effect of all the available divide and phase shift options on a VCO output of 250 MHz. It also shows the effect of division on the duty cycle of the resultant clock. Note that the duty cycle is 50-50 when a VCO output is divided by an even number. Also note that the phase shift applies to VCO output and not to the divided output

Table 7. Timing of Clock Phases for all Divide Options for a VCO Output Frequency of 250 MHz

Divide Factor	Period (ns)	Duty Cycle%	0° (ns)	45° (ns)	90° (ns)	135° (ns)	180° (ns)	225° (ns)	270° (ns)	315° (ns)
1	4	40-60	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
2	8	50	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
3	12	33-67	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
4	16	50	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
5	20	40-60	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
6	24	50	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
8	32	50	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
16	64	50	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5

Timing Model

One important feature of the PSI family is the simplicity of its timing. All combinatorial and registered/synchronous delays are worst case and system performance is static (as shown in the AC specs section) as long as data is routed through the same horizontal and vertical channels. *Figure 12* illustrates the true timing model for the 200-MHz devices. For synchronous clocking of macrocells, a delay is incurred from macrocell clock to macrocell clock of separate Logic Blocks within the same cluster, as well as separate Logic Blocks within different clusters. This is shown as t_{SCS} and t_{SCS2} in *Figure 12.* For combinatorial paths, any input to any output (from corner to corner on the device), incurs a worst-case delay in the 100K gate PSI regardless of the amount of logic or which horizontal and vertical channels are used. This is the t_{PD} shown in *Figure 12.* For synchronous systems, the input set-up time to the output macrocell register and the clock-to-output time are shown as the parameters t_{MCS} and t_{MCCO} shown in the *Figure 12.* These measurements are for any output and synchronous clock, regardless of the logic placement.

PSI features:

- no dedicated vs. I/O pin delays
- no penalty for using 0–16 product terms
- no added delay for steering product terms
- no added delay for sharing product terms
- no output bypass delays

The simple timing model of the PSI family eliminates unexpected performance penalties.

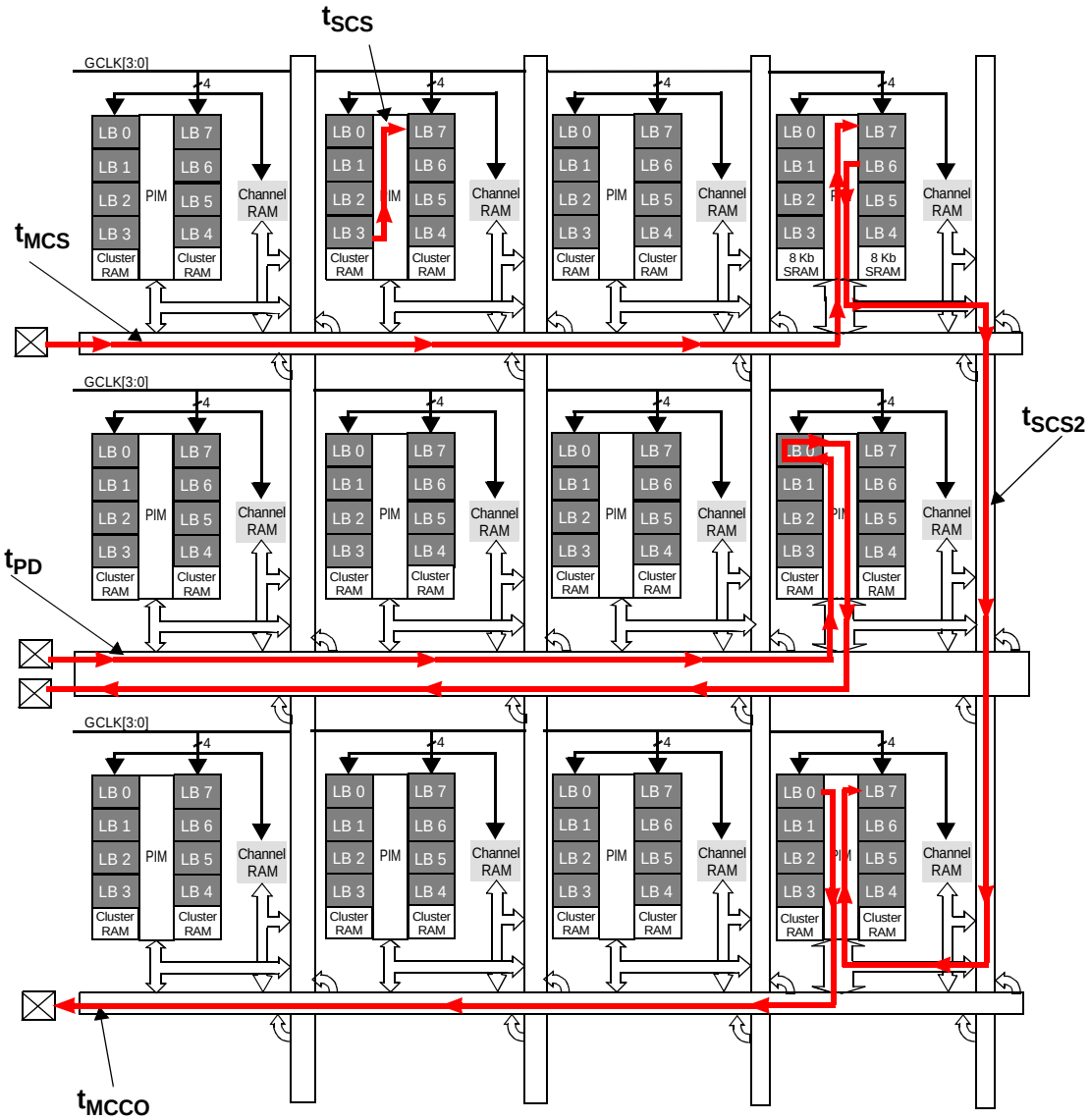


Figure 12. Timing Model for 100K gate PSI Devices

Serial Transceiver Operation

The PSI transceiver block is a highly configurable device designed to support reliable transfer of large quantities of data, using high-speed serial links, from one or multiple sources to one or multiple destinations. This device supports four single-byte or single-character channels that may be combined to support transfer of wider buses.

Frequency Agile PSI Transmit Data Path

Operating Modes

The transmit path of the PSI transceiver block supports four character-wide data paths. These data paths are used in multiple operating modes as controlled by the TXMODE[1:0] inputs.

Input Register

Within these operating modes, the bits in the Input Register for each channel support different bit assignments, based on whether the character is unencoded, encoded with two control bits, or encoded with three control bits. These assignments are shown in *Table 8*.

Table 8. Input Register Bit Assignments

Signal Name	Unencoded	Encoded
		2-bit Control
TXDx[0] (LSB)	DINx[0]	TXDx[0]
TXDx[1]	DINx[1]	TXDx[1]
TXDx[2]	DINx[2]	TXDx[2]
TXDx[3]	DINx[3]	TXDx[3]
TXDx[4]	DINx[4]	TXDx[4]
TXDx[5]	DINx[5]	TXDx[5]
TXDx[6]	DINx[6]	TXDx[6]
TXDx[7]	DINx[7]	TXDx[7]
TXCTx[0]	DINx[8]	TXCTx[0]
TXCTx[1] (MSB)	DINx[9]	TXCTx[1]

Each input register captures a minimum of eight data bits and two control bits on each input clock cycle. When the encoder is bypassed, the control bits are part of the pre-encoded 10-bit character.

When the Encoder is enabled (TXMODE[1] ≠ L), the TXCTx[1:0] bits are interpreted along with the associated TXDx[7:0] character to generate the specific 10-bit transmission character.

Phase-Align Buffer

Data from the input registers is passed either to the encoder or to the associated Phase-Align buffer. When the transmit paths are operated synchronous to REFCLK↑ (TXCKSEL = L and TXRATE = LOW), the Phase-Align Buffers are bypassed and data is passed directly to the encoder blocks to reduce latency.

When an Input-Register clock with an uncontrolled phase relationship to REFCLK is selected (TXCLSEL = H) or if data is captured on both edges of REFCLK (TXRATE = HIGH), the Phase-Align Buffers are enabled. These buffers are used to absorb clock phase differences between the presently selected input clock and the internal character clock.

Initialization of these phase-align buffers takes place when the TXRST input is sampled LOW by TXCLKA↑. When TXRST is returned HIGH, the present input clock phase relative to REFCLK↑ is set. TXRST is an asynchronous input, but is sampled internally to synchronize it to the internal transmit path state machines. TXRST must be sampled LOW by a minimum of two consecutive TXCLKA↑ clocks to ensure the reset operation is initiated correctly on all channels.

Once set, the TXCLKA is allowed to skew in time up to half a character period in either direction relative to REFCLK↑; i.e., ±180°. This time shift allows the delay paths of the character clocks (relative to REFLCK↑) to change due to operating voltage and temperature, while not affecting the design operation.

If the phase offset, between the initialized location of the input clock and REFCLK↑, exceeds the skew-handling capabilities of the Phase-Align Buffer, an error is reported on the associated TXPERx output. This output will indicate a continuous error until the Phase-Align Buffer is reset. While the error remains active, the transmitter for the associated channel will output a continuous C0.7 character to indicate to the remote receiver that an error condition is present in the link.

In specific transmit modes it is also possible to reset the Phase-Align Buffers individually and with minimal disruption of the serial data stream. When the transmit interface is configured for generation of atomic Word Sync Sequences (TXMODE[1] = M) and a Phase-Align Buffer error is present, the transmission of a Word Sync Sequence will re-center the buffer and clear the error condition.

> **NOTE**: One or more K28.5 characters may be added or lost from the data stream during this reset operation. When used with non-Cypress devices that require a complete 16-character Word Sync Sequence for proper receive Elasticity Buffer alignment, it is recommend that the sequence be followed by a second Word Sync Sequence to ensure proper operation.

Encoder

The character, received from the input register or phase-align buffer is then passed to the Encoder logic. This block interprets each character and any associated control bits, and outputs a 10-bit transmission character.

Depending on the configured operating mode, the generated transmission character may be

- the 10-bit pre-encoded character accepted in the input register
- the 10-bit equivalent of the 8-bit Data character accepted in the input register
- the 10-bit equivalent of the 8-bit Special Character code accepted in the input register
- the 10-bit equivalent of the C0.7 SVS character if a Phase-Align Buffer overflow or underflow error is present
- a character that is part of the 511-character BIST sequence
- a K28.5 character generated as an individual character or as part of the 16-character Word Sync Sequence.

The selection of the specific characters generated are controlled by the TXMODE[1:0], TXCTx[1:0], and TXDx[7:0] inputs for each character.

Data Encoding

Raw data, as received directly from the Transmit Input Register, is seldom in a form suitable for transmission across a serial link. The characters must usually be processed or transformed to guarantee

- a minimum transition density (to allow the serial receive PLL to extract a clock from the data stream)
- a DC-balance in the signaling (to prevent baseline wander)
- run-length limits in the serial data (to limit the bandwidth of the link)
- the remote receiver a way of determining the correct character boundaries (framing).

When the Encoder is enabled (TXMODE[1] ≠ L), the characters to be transmitted are converted from Data or Special Character codes to 10-bit transmission characters (as selected by their respective TXCTx[1:0] and SCSEL inputs), using an integrated 8B/10B encoder. When directed to encode the character as a Special Character code, it is encoded using the Special Character encoding rules listed in *Table 30*. When directed to encode the character as a Data character, it is encoded using the Data Character encoding rules in *Table 29*.

The 8B/10B encoder is standards compliant with ANSI/NCITS ASC X3.230-1994 (Fibre Channel), IEEE 802.3z (Gigabit Ethernet), the IBM® ESCON® and FICON™ channels, and ATM Forum standards for data transport.

Many of the Special Character codes listed in *Table 30* may be generated by more than one input character. The PSI transceiver block is designed to support two independent (but non-overlapping) Special Character code tables. This allows the PSI transceiver block to operate in mixed environments with other PSI transceiver blocks using the enhanced Cypress command code set and the reduced command sets of other non-Cypress devices. Even when used in an environment that normally uses non-Cypress Special Character codes, the selective use of Cypress command codes can permit operation where running disparity and error handling must be managed.

Following conversion of each input character from 8 bits to a 10-bit transmission character, it is passed to the Transmit Shifter and is shifted out LSB first, as required by ANSI and IEEE standards for 8B/10B coded serial data streams.

Transmit Modes

The operating mode of the transmit path is set through the TXMODE[1:0] inputs. These three-level select inputs allow one of nine transmit modes to be selected. Within each of these operating modes, the actual characters generated by the Encoder logic block are also controlled both by these and other static and dynamic control signals. The transmit modes are listed in *Table 9*.

The encoded modes (TX Modes 3 through 8) support multiple encoding tables. These encoding tables vary by the specific combinations of TXCTx[1] and TXCTx[0] that are used to control the generation of data and control characters. These multiple encoding forms allow maximum flexibility in interfacing to legacy applications, while also supporting numerous extensions in capabilities.

Table 9. Transmit Operating Modes

TX Mode		Operating Mode		
Mode Number	TXMODE [1:0]	Word Sync Sequence Support		TXCTx Function
0	LL	None	None	Encoder Bypass
1	LM	None	None	Reserved for test
2	LH	None	None	Reserved for test
3	ML	Atomic	Word Sync	Encoder Control
4	MM	Atomic	Special Character	Encoder Control
5	MH	Atomic	Word Sync + Special Character	Encoder Control
6	HL	Interruptible	Word Sync	Encoder Control
7	HM	Interruptible	Special Character	Encoder Control
8	HH	Interruptible	Word Sync + Special Character	Encoder Control

TX Mode 0—Encoder Bypass

When the Encoder is bypassed, the character captured in the TXDx[7:0] and TXCTx[1:0] inputs is passed directly to the transmit shifter without modification. With the encoder bypassed, the TXCTx[1:0] inputs are considered part of the data character and do not perform a control function that would otherwise modify the interpretation of the TXDx[7:0] bits. The bit usage and mapping of these control bits when the Encoder is bypassed is shown in *Table 10*.

Table 10. Encoder Bypass Mode (TXMODE[1:0] = LL)

Signal Name	Bus Weight	10B Name
TXDx[0] (LSB)	2^0	a[4]
TXDx[1]	2^1	b
TXDx[2]	2^2	c
TXDx[3]	2^3	d
TXDx[4]	2^4	e
TXDx[5]	2^5	i
TXDx[6]	2^6	f
TXDx[7]	2^7	g
TXCTx[0]	2^8	h
TXCTx[1] (MSB)	2^9	j

In this mode the SCSEL input is not interpreted. All clocking modes interpret the data the same, with no internal linking between channels.

TX Modes 1 and 2—Factory Test Modes

These modes enable specific factory test configurations. They are not considered normal operating modes of the device. En-

Note:
4. LSB is shifted out first.

try or configuration into these test modes will not damage the device.

TX Mode 3—Atomic Word Sync and Control of Special Codes

When configured in TX Mode 3, TXCTx[1:0] data control inputs are captured. These bits combine to control the interpretation of the TXDx[7:0] bits and the characters generated by them. These bits are interpreted as listed in *Table 11*.

Table 11. TX Modes 3 and 6 Encoding

TXCTx[1]	TXCTx[0]	Characters Generated
X	0	Encoded data character
0	1	K28.5 fill character
1	1	16-character Word Sync Sequence

Word Sync Sequence

When TXCTx[1:0] = 11, a 16-character sequence of K28.5 characters, known as a Word Sync Sequence, is generated on the associated channel. This sequence of K28.5 characters may start with either a positive or negative disparity K28.5 (as determined by the current running disparity and the 8B/10B coding rules). The disparity of the second and third K28.5 characters in this sequence are reversed from what normal 8B/10B coding rules would generate. The remaining K28.5 characters in the sequence follow all 8B/10B coding rules. The disparity of the generated K28.5 characters in this sequence would follow a pattern of either ++−−+−+−+−+−+−+− or −−++−+−+−+−+−+.

When TXMODE[1] = M (open, TX modes 3, 4, and 5), the generation of this character sequence is an atomic (non-interruptible) operation. Once it has been successfully started, it cannot be stopped until all 16 characters have been generated. The content of the associated input register(s) is ignored for the duration of this 16-character sequence. At the end of this sequence, if the TXCTx[1:0] = 11 condition is sampled again, the sequence restarts and remains uninterruptible for the following 15 character clocks.

When TXMODE[1] = H (TX modes 6, 7, and 8), the generation of the Word Sync Sequence becomes an interruptible operation. In TX Mode 6, this sequence is started as soon as the TXCTx[1:0] = 11 condition is detected on a channel. In order for the sequence to continue on that channel, the TXCTx[1:0] inputs must be sampled as 00 for the remaining 15 characters of the sequence.

If at any time a sample period exists where TXCTx[1:0] ≠ 00, the Word Sync Sequence is terminated, and a character representing the associated data and control bits is generated by the Encoder. This resets the Word Sync Sequence state machine such that it will start at the beginning of the sequence at the next occurrence of TXCTx[1:0] = 11.

When TXCKSEL = L, the input registers for all four transmit channels are clocked by REFCLK [36]. When TXCKSEL = H, the input registers for all four transmit channels are clocked with TXCLKA↑.

NOTE: When operated in any configuration where receive channels are bonded together, TXCKSEL must be either LOW or HIGH (not MID) to ensure that associated characters are transmitted in the same character cycle.

TX Mode 4—Atomic Word Sync and Control of Word Sync Sequence Generation

When configured in TX Mode 4, TXCTx[1:0] data control inputs are captured. These bits combine to control the interpretation of the TXDx[7:0] bits and the characters generated by them. These bits are interpreted as listed in *Table 12*.

Table 12. TX Modes 4 and 7 Encoding

TXCTx[1]	TXCTx[0]	Characters Generated
X	0	Encoded data character
0	1	K28.5 fill character
1	1	Special character code

TX Mode 5—Atomic Word Sync

In addition to the standard character encodings, both with and without atomic Word Sync Sequence generation, two additional encoding mappings are controlled by the Channel Bonding selection made through the RXMODE[1:0] inputs.

For non-bonded operation, the TXCTx[1:0] inputs for each channel control the characters generated by that channel. The specific characters generated by these bits are listed in *Table 13*.

Table 13. TX Modes 5 and 8 Encoding, Non-Bonded

TXCTx[1]	TXCTx[0]	Characters Generated
0	0	Encoded data character
0	1	K28.5 fill character
1	0	Special character code
1	1	16-character Word Sync Sequence

TX Mode 5 also has the capability of generating an Atomic Word Sync Sequence. For the sequence to be started, the TXCTx[1:0] inputs must both be sampled HIGH. With the exception of the combination of control bits used to initiate the sequence, the generation and operation of this Word Sync Sequence is the same as that documented for TX Mode 3.

Two additional encoding maps are provided for use when receive channel bonding is enabled. When dual-channel bonding is enabled (RXMODE[1] = M), the PSI transceiver block is configured such that channels A and B are bonded together to form a two-character-wide path, and channels C and D are bonded together to form a second two-character-wide path.

When operated in this two-channel bonded mode, the TXCTA[0] and TXCTB[0] inputs control the interpretation of the data on both the A and B channels, while the TXCTC[0] and TXCTD[0] inputs control the interpretation of the data on both the C and D channels. The characters on each half of these bonded channels are controlled by the associated TXCTx[1] bit. The specific characters generated by these control bit combinations are listed in *Table 14*. Note especially that any time TXCTB[0] is sampled HIGH, both channels A and B start generating an Atomic Word Sync Sequence, regardless of the

Table 14. TX Modes 5 and 8, Dual-Channel Bonded

TXCTA[1]	TXCTA[0]	Characters Generated
0	0	Encoded data character on channel A
0	1	K28.5 fill character on channel A
1	0	Special character code on channel A
1	1	16-character word sync on channel A
X	0	Encoded data character on channel B
X	1	K28.5 fill character on channel B
X	0	Special character code on channel B
X	1	16-character word sync on channel B

state of any of the other bits in the A or B input registers. In a similar fashion, anytime TXCTD[0] is sampled HIGH, both the C and D channels start generation of an Atomic Word Sync Sequence.

When RXMODE[1] = H, the PSI transceiver block is configured for quad-channel bonding, such that channels A, B, C, and D are bonded together to form a four-character-wide path. When operated in this mode, the TXCTA[0] and TXCTB[0] inputs control the interpretation of the data on all four channels. The characters generated on these bonded channels are controlled by the associated TXCTx[1] bit. The specific characters generated by these bits are listed in *Table 15*.

Unlike dual-channel modes, when all four channels are bonded together, the TXCTC[0] and TXCTD[0] inputs are not interpreted.

Transmit BIST

The transmitter interfaces contain internal pattern generators that can be used to validate both device and link operation. These generators are enabled by the associated BOE[x] signals listed in *Table 16* (when the BISTLE latch enable input is HIGH). When enabled, a register in the associated transmit channel becomes a signature pattern generator by logically converting to a Linear Feedback Shift Register (LFSR). This LFSR generates a 511-character sequence that includes all Data and Special Character codes, including the explicit violation symbols. This provides a predictable yet pseudo-random sequence that can be matched to an identical LFSR in the attached Receiver(s).

Table 15. TX Modes 5 and 8, Quad-Channel Bonded

TXCTA[1]	TXCTA[0]	Characters Generated
0	0	Encoded data character on channel A
0	1	K28.5 fill character on channel A
1	0	Special character code on channel A
1	1	16-character word sync on channel A
X	0	Encoded data character on channel B
X	1	K28.5 fill character on channel B
X	0	Special character code on channel B
X	1	16-character word sync on channel B
X	0	Encoded data character on channel C
X	1	K28.5 fill character on channel C
X	0	Special character code on channel C
X	1	16-character word sync on channel C
X	0	Encoded data character on channel D
X	1	K28.5 fill character on channel D
X	0	Special character code on channel D
X	1	16-character word sync on channel D

When the BISTLE signal is HIGH, any BOE[x] input that is LOW enables the BIST generator in the associated transmit channel (or the BIST checker in the associated receive channel). When BISTLE returns LOW, the values of all BOE[x] signals are captured in the BIST Enable Latch. These values remain in the BIST Enable Latch until BISTLE is returned high to open the latch again. All captured signals in the BIST Enable Latch are set HIGH (i.e., BIST is disabled) following a a device reset ($\overline{\text{TRSTZ}}$ is sampled LOW).

All data and data-control information present at the associated TXDx[7:0] and TXCTx[1:0] inputs are ignored when BIST is active on that channel. If the receive channels are configured for common clock operation (RXCKSEL ≠ MID) each pass is preceded by a 16-character Word Sync Sequence to allow Elasticity Buffer alignment and management of clock-frequency variations.

Serial Output Drivers

The serial interface Output Drivers make use of high-performance differential CML (Current Mode Logic) to provide a source-matched driver for the transmission lines. These drivers accept data from the Transmit Shifters. These outputs have signal swings equivalent to that of standard PECL drivers, and are capable of driving AC-coupled optical modules or AC-coupled transmission lines.

When configured for local loopback (LPEN = HIGH), the output drivers for all enabled ports are configured to drive a static differential logic-1.

Each output can be enabled or disabled separately through the BOE[7:0] inputs, as controlled by the OELE latch-enable signal. When OELE is HIGH, the signals present on the BOE[7:0] inputs are passed through the Serial Output Enable latch to control the serial output drivers. The BOE[7:0] input associated with a specific OUTxy± driver is listed in *Table 16*.

Table 16. Output Enable, BIST, and Receive Channel Enable Signal Map

BOE Input	Output Controlled (OELE)	BIST Channel Enable (BISTLE)	Receive PLL Channel Enable (RXLE)
BOE[7]	OUTD2±	Transmit D	X
BOE[6]	OUTD1±	Receive D	Receive D
BOE[5]	OUTC2±	Transmit C	X
BOE[4]	OUTC1±	Receive C	Receive C
BOE[3]	OUTB2±	Transmit B	X
BOE[2]	OUTB1±	Receive B	Receive B
BOE[1]	OUTA2±	Transmit A	X
BOE[0]	OUTA1±	Receive A	Receive A

When OELE is HIGH and BOE[x] is HIGH, the associated serial driver is enabled to drive any attached transmission line. When OELE is HIGH and BOE[x] is LOW, the associated driver is disabled and internally configured for minimum power dissipation. If both outputs for a channel are in this disabled state, the associated internal logic for that channel is also configured for lowest power operation. When OELE returns LOW, the values present on the BOE[7:0] inputs are latched in the Output Enable Latch, and remain there until OELE returns HIGH to opened the latch again.

> **Note:** When a disabled transmit channel (i.e., both outputs disabled) is re-enabled, the data on the serial outputs may not meet all timing specifications for up to 10 ms.

Transmit PLL Clock Multiplier

The Transmit PLL Clock Multiplier accepts a character-rate or half-character-rate external clock at the REFCLK input, and multiples that clock by 10 or 20 (as selected by TXRATE) to generate a bit-rate clock for use by the transmit shifter. It also provides a character-rate clock used by the transmit paths.

The clock multiplier PLL can accept a REFCLK input between 10 MHz and 150 MHz, however, this clock range is limited by the operating mode of the PSI transceiver block clock multiplier (controlled by TXRATE) and by the level on the SPDSEL input.

SPDSEL is a three-level select (ternary) input that selects one of three operating ranges for the serial data outputs and inputs. The operating serial signaling-rate and allowable range of REFCLK frequencies are listed in *Table 17*.

Table 17. Operating Speed Settings

SPDSEL	TXRATE	REFCLK Frequency (MHz)	Signaling Rate (MBaud)
LOW	1	10–20	200–400
	0	20–40	
MID (Open)	1	20–40	400–800
	0	40–80	
HIGH	1	40–75	800–1500
	0	80–150	

The REFCLK± input is a non-standard input. It is implemented as a differential input with each input internally biased to $V_{CC}/2$. If the REFCLK+ input is connected to a TTL, LVTTL, or LVCMOS clock source, the input signal is recognized when it passes through the internally biased reference point.

When both the REFCLK+ and REFCLK– inputs are connected, the clock source must be a differential clock. This can be either a differential LVPECL clock that is DC- or AC-coupled, or a differential LVTTL or LVCMOS clock.

By connecting the REFCLK– input to an external voltage source or resistive voltage divider, it is possible to adjust the reference point of the REFCLK+ input for alternate logic levels. When doing so it is necessary to ensure that the 0V-differential crossing point remain within the parametric range supported by the input.

Frequency Agile PSI Receive Data Path

Serial Line Receivers

Two differential line receivers, INx1± and INx2±, are available on each channel for accepting serial data streams. The active line receiver on a channel is selected using the associated INSELx input. The serial line receiver inputs are all differential, and can accommodate wire interconnect and filtering losses or transmission line attenuation greater than 16 dB ($V_{DIF} \geq 100$ mV, or 200 mV peak-to-peak differential) or can be DC- or AC-coupled to +3.3V powered fiber-optic interface modules (any ECL/PECL logic family, not limited to 100K PECL) or AC-coupled to +5V powered optical modules. The common-mode tolerance of these line receivers accommodates a wide range of signal termination voltages. Each receiver provides internal DC-restoration, to the center of the receiver's common mode range, for AC-coupled signals.

The local loopback input (LPEN) allows the serial transmit data outputs to be routed internally back to the Clock and Data Recovery circuit associated with each channel. When configured for local loopback, all transmit serial driver outputs are forced to output a differential logic-1. This prevents local diagnostic patterns from being broadcast to attached remote receivers.

Signal Detect / Link Fault

Each selected Line Receiver (i.e., that routed to the Clock and Data Recovery PLL) is simultaneously monitored for

- analog amplitude

- transition density
- range controls report the received data stream inside normal frequency range (±200 ppm)
- receive channel enabled

All of these conditions must be valid for the Signal Detect block to indicate a valid signal is present. This status is presented on the LFIx (Link Fault Indicator) output associated with each receive channel, which changes synchronous to the selected receive interface clock.

Table 18. Analog Amplitude Detect Valid Signal Levels

SDASEL	Typical signal with peak amplitudes above
LOW	140 mV p-p differential
MID (Open)	280 mV p-p differential
HIGH	420 mV p-p differential

Analog Amplitude

While the majority of these signal monitors are based on fixed constants, the analog amplitude level detection is adjustable to allow operation with highly attenuated signals, or in high-noise environments. This adjustment is made through the SDASEL signal, a three-level select (ternary) input, which sets the trip point for the detection of a valid signal at one of three levels, as listed in *Table 18*. This control input effects the analog monitors for all receive channels.

The Signal Detect monitors are active for the present line receiver, as selected by the associated INSELx input. When configured for local loopback (LPEN = HIGH), no line receivers are selected, and the LFI output for each channel reports only the receive VCO frequency out-of-range and transition density status of the associated transmit signal. When local loopback is active, the analog amplitude monitors are disabled.

Transition Density

The transition detection logic checks for the absence of any transitions spanning greater than six transmission characters (60 bits). If no transitions are present in the data received on a channel (within the referenced period), the transition detection logic for that channel will assert LFIx. The LFIx output remains asserted until at least one transition is detected in each of three adjacent received characters.

Range Controls

The receive-VCO range-control monitors do more than just report the frequency status of the received signal. They also determine if the receive Clock/Data Recovery circuits (CDR) should align the receive VCO clock to the data stream or to the local REFCLK input. This function prevents the receive VCO from tracking an out-of-specification received signal.

When the range-control monitor for a channel indicates that the signaling rate is within specification, the phase detector in the receive PLL is configured to track the transitions in the received data stream. In this mode the LFIx output for the associated channel is HIGH (unless one of the other status monitors indicates that the received signal is out of specification). If the range-control monitor indicates that the received data stream signaling-rate is out of specification, the phase detector is configured to track the local REFCLK input, and the associated LFIx output is asserted LOW.

The specific trip points for this compare function are listed in *Table 19*. Because the compare function operates with two asynchronous clocks, there is a small uncertainty in the measurement. The switch points are asymmetric to provide hysteresis to the operation.

Table 19. Receive Signaling Rate Range Control criteria

Current RX PLL Tracking Source	Frequency Difference Between Transmit Character Clock & RX VCO	Next RX PLL Tracking Source
Selected data stream (LFIx = HIGH)	<1708 ppm	Data Stream
	1708–1953 ppm	Indeterminate
	>1953 ppm	REFCLK
REFCLK (LFIx = LOW)	<488 ppm	Data Stream
	488–732 ppm	Indeterminate
	>732 ppm	REFCLK

Receive Channel Enabled

The Frequency Agile PSI device contains four receive channels that can be independently enabled and disabled. Each channel can be enabled or disabled separately through the BOE[7:0] inputs, as controlled by the RXLE latch-enable signal. When RXLE is HIGH, the signals present on the BOE[7:0] inputs are passed through the Receive Channel Enable latch to control the PLLs and logic of the associated receive channel. The BOE[7:0] input associated with a specific receive channel is listed in *Table 16*.

When RXLE is HIGH and BOE[x] is HIGH, the associated receive channel is enabled to receive and decode a serial stream from the selected line receiver. When RXLE is HIGH and BOE[x] is LOW, the associated receive channel is disabled and internally configured for minimum power dissipation. If a single channel of a bonded-pair or bonded-quad is disabled, this will impact the ability of the receive channels to bond correctly. In addition, if the disabled channel is selected as the master channel for insert/delete functions, or for recovered clock select, these functions will not work correctly. Any disabled channel will indicate a constant /LFIx output. When RXLE returns LOW, the values present on the BOE[7:0] inputs are latched in the Receive Channel Enable Latch, and remain there until RXLE returns HIGH to opened the latch again.

> **Note:** When a disabled receive channel is re-enabled, the status of the associated LFIx output and data on the parallel outputs for the associated channel may be indeterminate for up to 10ms.

Clock/Data Recovery

The extraction of a bit-rate clock and recovery of bits from each received serial stream is performed by a separate Clock/Data Recovery (CDR) block within each receive channel. The clock extraction function is performed by high-performance embedded phase-locked loops (PLLs) that track the frequency of the transitions in the incoming bit streams and align the phase of their internal bit-rate clocks to the transitions in the selected serial data streams.

Each CDR accepts a character-rate (bit-rate ÷ 10) or half-character-rate (bit-rate ÷ 20) reference clock from the REFCLK input. This REFCLK input is used to

- ensure that the VCO (within each CDR) is operating at the correct frequency (rather than some harmonic of the bit-rate)
- to improve PLL acquisition time
- and to limit unlocked frequency excursions of the CDR VCO when no data is present at the selected serial inputs.

Regardless of the type of signal present, the CDR will attempt to recover a data stream from it. If the frequency of the recovered data stream is outside the limits set by the range control monitors, the CDR PLL will track REFCLK instead of the data stream. When the frequency of the selected data stream returns to a valid frequency, the CDR PLL is allowed to track the received data stream. The frequency of REFCLK is required to be within ±200 ppm of the frequency of the clock that drives the REFCLK input of the *remote* transmitter to ensure a lock to the incoming data stream.

For systems using multiple or redundant connections, the $\overline{\text{LFIx}}$ output can be used to select an alternate data stream. When an $\overline{\text{LFIx}}$ indication is detected, external logic can toggle selection of the associated INx1± and INx2± inputs through the associated INSELx input. When a port switch takes place, it is necessary for the receive PLL for that channel to reacquire the new serial stream and frame to the incoming character boundaries. If channel bonding is also enabled, a channel alignment event is also required before the output data may be considered usable.

Deserializer/Framer

Each CDR circuit extracts bits from the associated serial data stream and clocks these bits into the Shifter/Framer at the bit-clock rate. When enabled, the Framer examines the data stream looking for one or more COMMA or K28.5 characters at all possible bit positions. The location of this character in the data stream is used to determine the character boundaries of all following characters.

Framing Character

The PSI transceiver block allows selection of one of three combinations of framing characters to support requirements of different interfaces. The selection of the framing character is made through the FRAMCHAR input.

FRAMCHAR is a 3-level select input that allows selection of one of three different framing characters or character combinations. The specific bit combinations of these framing characters are listed in *Table 20*. When the specific bit combination of the selected framing character is detected by the framer, the boundaries of the characters present in the received data stream are known.

Table 20. Framing Character Selector

FRAMCHAR	Bits detected in framer	
	Character Name	Bits Detected
LOW	COMMA+	00111110XX [5]
MID (Open)	COMMA+ COMMA–	00111110XX [5] or 11000001XX
HIGH	–K28.5 +K28.5	0011111010 or 1100000101

Note:
5. The standard definition of a COMMA contains only seven bits. However, since all valid COMMA characters within the 8B/10B character set also have the 8th bit as an inversion of the 7th bit, the compare pattern is extended to a full eight bits to reduce the possibility of a framing error.

Framer

The framer on each channel operates in one of three different modes, as selected by the RFMODE input. In addition, the framer itself may be enabled or disabled through the RFEN input. When RFEN = LOW, the framers in all four receive paths are disabled, and no combination of bits in a received data stream will alter the character boundaries. When RFEN = HIGH, the framer selected by RFMODE is enabled on all four channels.

When RFMODE = LOW, the low-latency framer is selected. This framer operates by stretching the recovered character clock until it aligns with the received character boundaries. In this mode the framer starts its alignment process on the first detection of the selected framing character. To reduce the impact on external circuits that make use of a recovered clock, the clock period is not stretched by more than two bit-periods in any one clock cycle. When operated in with a character-rate output clock (RXRATE = LOW), the output of properly framed characters may be delayed by up to nine character-clock cycles from the detection of the selected framing character. When operated with a half-character-rate output clock (RXRATE = HIGH), the output of properly framed characters may be delayed by up to 14 character-clock cycles from the detection of the selected framing character.

When RFMODE is MID (open) the Cypress-mode multi-byte framer is selected. The required detection of multiple framing characters makes the associated link much more robust to incorrect framing due to aliased SYNC characters in the data stream. In this mode, the framer does not adjust the character clock boundary, but instead aligns the character to the already recovered character clock. This ensures that the recovered clock will not contain any significant phase changes or hops during normal operation or framing, and allows the recovered clock to be replicated and distributed to other external circuits or components using PLL-based clock distribution elements. In this framing mode the character boundaries are only adjusted if the selected framing character is detected at least twice within a span of 50 bits, with both instances on identical 10-bit character boundaries.

When RFMODE = HIGH, the alternate-mode multi-byte framer is enabled. Like the Cypress-mode multi-byte framer, multiple framing characters must be detected before the character boundary is adjusted. In this mode, the data stream must contain a minimum of four of the selected framing characters, received as consecutive characters, on identical 10-bit boundaries, before character framing is adjusted.

> **Note:** Except for the K29.7 character, the 8B/10B running disparity rules prohibit the presence of multiple COMMA+ characters as consecutive characters. Because of this, the combination of FRAMCHAR LOW and RFMODE = HIGH is not recommended. While framing can still take place while following all 8B/10B coding rules, this configuration prevents framing to the K28.5 character.

> **Note:** The receive Elasticity Buffers require detection of four of the selected framing character to enable buffer alignment and centering. Because these characters must occur as consecutive characters, the combination of FRAMCHAR LOW and RFMODE = HIGH is not recommended for receive modes that use the Elasticity Buffers.

Framing for all channels is enabled when RFEN = HIGH. If RFEN = LOW, the framer for each channel is disabled. When the framers are disabled, no changes are made to the recov-

ered character boundaries on any channel, regardless of the presence of framing characters in the data stream.

10B/8B Decoder Block

The decoder logic block performs two primary functions:

- decoding the received transmission characters back into Data and Special Character codes,
- comparing generated BIST patterns with received characters to permit at-speed link and device testing,

10B/8B Decoder

The framed parallel output of each deserializer shifter is passed to the 10B/8B Decoder where, if the Decoder is enabled (DECMODE ≠ LOW), it is transformed from a 10-bit transmission character back to the original Data and Special Character codes. This block uses the 10B/8B decoder patterns in *Tables 29* and *30* of this data sheet. Valid data characters are indicated by a 000b bit-combination on the associated RXSTx[2:0] status bits, and Special Character codes are indicated by a 001b bit-combination on these same status outputs. Framing characters, Invalid patterns, disparity errors, and synchronization status are presented as alternate combinations of these status bits.

The 10B/8B decoder operates in two normal modes, and can also be bypassed. The operating mode for the decoder is controlled by the DECMODE input.

When DECMODE = LOW, the decoder is bypassed and raw 10-bit characters are passed to the output register. In this mode, channel bonding is not possible, the receive Elasticity Buffers are bypassed, and RXCKSEL must be MID. This clock mode generates separate RXCLKx+ outputs for each receive channel.

When DECMODE is MID (or open), the 10-bit transmission characters are decoded using *Tables 29* and *30*. Received Special Code characters are decoded using the Cypress column of *Table 30*.

When DECMODE = HIGH, the 10-bit transmission characters are decoded using *Tables 29* and *30*. Received Special Code characters are decoded using the Alternate column of *Table 30*.

In all settings where the decoder is enabled, the receive paths may be operated as separate channels or bonded to form various multi-channel buses.

Receive BIST Operation

The receiver interfaces contain internal pattern generators that can be used to validate both device and link operation. These generators are enabled by the associated BOE[x] signals listed in *Table 16* (when the BISTLE latch enable input is HIGH). When enabled, a register in the associated receive channel becomes a signature pattern generator and checker by logically converting to a Linear Feedback Shift Register (LFSR). This LFSR generates a 511-character sequence that includes all Data and Special Character codes, including the explicit violation symbols. This provides a predictable yet pseudo-random sequence that can be matched to an identical LFSR in the attached Transmitter(s). When synchronized with the received data stream, the associated receiver checks each character in the Decoder with each character generated by the LFSR and indicates compare errors and BIST status at the RXSTx[2:0] bits of the output register.

When the BISTLE signal is HIGH, any BOE[x] input that is LOW enables the BIST generator/checker in the associated receive channel (or the BIST generator in the associated transmit channel). When BISTLE returns LOW, the values of all BOE[x] signals are captured in the BIST Enable Latch. These values remain in the BIST Enable Latch until BISTLE is returned high to open the latch again. All captured signals in the BIST Enable Latch are set HIGH (i.e., BIST is disabled) following a device reset ($\overline{\text{TRSTZ}}$ is sampled LOW).

The LFSR is initialized by the BIST hardware once the BIST enable for that receive channel is present at the output of the BIST Enable Latch, and is recognized. This sets the BIST LFSR to the BIST-loop start-code of D0.0 (D0.0 is sent only once per BIST loop). The status of the BIST progress and any character mismatches is presented on the RXSTx[2:0] status outputs.

Code rule violations or running disparity errors that occur as part of the BIST loop do not cause an error indication. RXSTx[2:0] indicates 010b or 100b for one character period per BIST loop to indicate loop completion. This status can be used to check test pattern progress. These same status values are presented when the decoder is bypassed and BIST is enabled on a receive channel.

The specific status reported by the BIST state machine are listed in *Table 26*. These same codes are reported on the receive status outputs regardless of the state of DECMODE.

The specific patterns checked by each receiver are described in detail in the Cypress application note “HOTLink Built-In Self-Test.” The sequence compared by the PSI transceiver block is identical to that in the CY7B933 and CY7C924DX, allowing interoperable systems to be built when used at compatible serial signaling rates.

If the number of invalid characters received ever exceeds the number of valid characters by 16, the receive BIST state machine aborts the compare operations and resets the LFSR to the D0.0 state to look for the start of the BIST sequence again.

When the receive paths are configured for common clock operation (RXCKSEL ≠ MID) each pass must be preceded by a 16-character Word Sync Sequence to allow output buffer alignment and management of clock frequency variations. This is automatically generated by the transmitter when its local RXCKSEL ≠ MID.

The BIST state machine requires the characters to be correctly framed for it to detect the BIST sequence. If the framer is enabled and configured for low-latency operation (RFMODE = LOW), the framer can align to characters within the BIST sequence. If either of the multi-byte framers are enabled (RFMODE ≠ LOW), it is generally necessary to frame the receiver before BIST is enabled. If the receive outputs are clocked relative to REFCLK (RXCKSEL = LOW), the transmitter precedes every 511 character BIST sequence with a 16-character Word Sync Sequence. This sequence will frame the receiver regardless of the setting of RFMODE.

Receive Elasticity Buffer

Each receive channel contains an Elasticity Buffer that is designed to support multiple clocking modes. These buffers allow data to be read using an Elasticity Buffer read-clock that is asynchronous in both frequency and phase from the Elasticity Buffer write clock, or to use a read clock that is frequency coherent but with uncontrolled phase relative to the Elasticity Buffer write clock.

Each Elasticity Buffer is a minimum of 10-characters deep, and supports a 11-bit wide data path. It is capable of supporting a decoded character and three status bits for each character present in the buffer. The write clock for these buffers is always the recovered clock for the associated read channel.

The read clock for the Elasticity Buffers may come from one of three selectable sources. It may be a

- character-rate REFCLK↑
- recovered clock from the same receive channel
- recovered clock from an alternate receive channel

These Elasticity Buffers are also used to align the output data streams when multiple channels are bonded together.

Receive Modes

The operating mode of the receive path is set through the RXMODE[1:0] inputs. These RXMODE[1:0] inputs are only interpreted when the decoder is enabled (DECMODE ≠ LOW). These modes determine the type (if any) of channel bonding and status reporting. The different receive modes are listed in *Table 21*.

Table 21. Receive Operating Modes

RX Mode		Operating Mode	
Mode Number	RXMODE [1:0]	Channel Bonding	RXSTx Status Reporting
0	LL	Independent	Status A
1	LM		Reserved for test
2	LH	Independent	Status B
3	ML	Dual	Status A
4	MM		Reserved for test
5	MH	Dual	Status B
6	HL	Quad	Status A
7	HM		Reserved for test
8	HH	Quad	Status B

Independent Channel Modes

In independent channel modes (RX Modes 0 and 2, where RXMODE[1] = LOW), all four receive paths may be clocked in any clock mode selected by RXCKSEL.

When RXCKSEL = LOW, all four receive channels are clocked by REFCLK. RXCLKB+ and RXCLKD+ outputs are disabled (High-Z), and the RXCLKA+ and RXCLKC+ outputs present a buffered and delayed form of REFCLK. In this mode, the receive Elasticity Buffers are enabled. For REFCLK↑ clocking, the Elasticity Buffers must be able to insert K28.5 characters and delete framing characters as appropriate.

The insertion of a K28.5 or deletion of a framing character can occur at any time on any channel, however, the actual timing on these insertions and deletions is controlled in part by the how the transmitter sends its data. Insertion of a K28.5 character can only occur when the receiver has a framing character in the Elasticity Buffer. Likewise, to delete a framing character, one must also be in the Elasticity Buffer. To prevent a receive buffer overflow or underflow on a receive channel, a minimum density of framing characters must be present in the received data streams.

Prior to reception of valid data, at least one Word Sync Sequence (or that portion of one necessary to align the receive buffers) must be received to allow the receive Elasticity Buffer to be centered. The Elasticity buffer may also be set by a device reset operation initiated through the TRSTZ input, however, following such an event the Frequency Agile PSI device will normally require a framing event before it will correctly decode characters.

When RXCKSEL is MID (or open), each received channel output register is clocked by the recovered clock for that channel. Since no characters may be added or deleted, the receiver Elasticity Buffer is bypassed.

When RXCKSEL = HIGH, all channels are clocked by the selected recovered clock. This selection is made using the RXCLKB+ and RXCLKD+ signals as inputs per *Table 22*. This selected clock is always output on RXCLKA+ and RXCLKC+. In this mode the receive Elasticity Buffers are enabled. When data is output using a recovered clock (RXCKSEL = HIGH), receive channels are not allowed to insert and delete characters, except as necessary for Elasticity Buffer alignment.

Table 22. Independent and Quad Channel Bonded Recovered Clock Select

RXCLKB+	RXCLKD+	RXCLKA+/RXCLKC+ Clock Source
0	0	RXCLKA
0	1	RXCLKB
1	0	RXCLKC
1	1	RXCLKD

Prior to reception of valid data, at least one Word Sync Sequence (or that portion of one necessary to align the receive buffers) must be received to allow the receive Elasticity Buffers to be centered. The Elasticity buffer may also be set by a device reset operation initiated through the TRSTZ input, however, following such an event the Frequency Agile PSI device will normally require a framing event before it will correctly decode characters. Since the Elasticity buffer is not allowed to insert or delete framing characters, the transmit clocks on the channels must all be from a common source.

Dual-Channel Bonded Modes

In dual-channel bonded modes (RX Modes 3 and 5, where RXMODE[1] = MID or open), the associated receive channel pair output registers must be clocked by a common clock. This mode does not operate when RXCKSEL = MID.

Proper operation in this mode requires that the associated transmit data streams are clocked from a common reference with no long-term character slippage between the bonded channels. In dual-channel mode this means that channels A and B must be clocked from a common reference, and channels C and D must be clocked from a common reference (all four transmit channels *may* be clocked from the same source, but that is not a requirement).

Prior to reception of valid characters, at least one Word Sync Sequence (or that portion of one necessary to align the receive buffers) must be received on the bonded channels (within the allowable inter-channel skew window) to allow the re-

ceive Elasticity Buffers to be centered. While normal characters may be output prior to this alignment event, they are not necessarily aligned within the same boundaries that they were transmitted.

When RXCKSEL = LOW, all four receive channels are clocked by REFCLK. RXCLKB+ and RXCLKD+ outputs are disabled (High-Z), and RXCLKA+ and RXCLKC+ present a buffered and delayed form of REFCLK. In this mode, the receive Elasticity Buffers are enabled. For REFCLK clocking, the Elasticity Buffers must be able to insert K28.5 characters and delete framing characters as appropriate. While these insertions and deletions can take place at any time, they must occur at the same time on both channels that are bonded together. This is necessary to keep the data in the bonded channel-pairs properly aligned. This insert and delete process is controlled by the channel selected using the RXCLKB+ and RXCLKD+ inputs using the decodes listed in *Table 23*.

When RXCKSEL = HIGH, the A and B channels are clocked by the selected recovered clock, and the C and D channels are clocked by the selected recovered clock, as shown in *Table 23*. The output clock for the channel A/B bonded-pair is output continuously on RXCLKA+. The clock source for this output is selected from the recovered clock for channel A or channel B using the RXCLKB+ input. The output clock for the channel C/D bonded-pair is output continuously on RXCLKC+. The clock source for this output is selected from recovered clock for channel C or channel D using the RXCLKD+ input.

Table 23. Dual-Channel Bonded Recovered Clock Select

RXCLKB+	RXCLKD+	Clock Source	
		RXCLKA+	RXCLKC+
0	X	RXCLKA	
1	X	RXCLKB	
X	0		RXCLKC
X	1		RXCLKD

When data is output using a recovered clock (RXCKSEL = HIGH), receive channels are not allowed to insert and delete characters, except as necessary for Elasticity Buffer alignment.

Quad-Channel Modes

In quad-channel modes (RX modes 6 and 7, where RXMODE[1] = HIGH), all four receive channel output registers must be clocked by a common clock. This mode does not operate when RXCKSEL = MID.

Proper operation in this mode requires that the four transmit data streams are clocked from a common reference with no long-term character slippage between the bonded channels. In quad-channel modes this means that the transmit channels A, B, C, and D must all be clocked from a common reference.

Prior to reception of valid data, at least one Word Sync Sequence (or that portion of one necessary to align the receive buffers) must be received on all four bonded channels (within the allowable inter-channel skew window) to allow the receive Elasticity Buffers to be centered and aligned.

When RXCKSEL = LOW, all four receive channels are clocked by the internal derivative of REFCLK. RXCLKB+ and RXCLKD+ outputs are disabled (High-Z), and RXCLKA+ and RXCLKC+ present a buffered and delayed form of REFCLK. In this mode the receive Elasticity Buffers are enabled. For REFCLK clocking, the Elasticity Buffers must be able to insert K28.5 characters and delete framing characters as appropriate. While these insertions and deletions can take place at any time, they must occur at the same time on all four channels. This is necessary to keep the data in the four bonded channels properly aligned. This insert and delete process is controlled by the channel selected using the RXCLKB+ and RXCLKD+ inputs using the decode listed in *Table 22*.

When RXCKSEL = HIGH, all four receive-channel output registers are clocked by the selected recovered clock. The clock select for quad channel mode is the same as that for independent channel operation. This selection is made using the RXCLKB+ and RXCLKD+ inputs, as shown in *Table 22*. The output clock for the four bonded channels is output continuously on RXCLKA+ and RXCLKC+.

When data is output using a recovered clock (RXCKSEL = HIGH), receive channels are not allowed to insert and delete characters, except as necessary for Elasticity Buffer alignment.

Multi-Device Bonding

When configured for quad-channel bonding (RXMODE[1] = HIGH) it is also possible to bond channels across multiple devices. This form of channel bonding is only possible when RXCKSEL = LOW, selecting REFCLK as the output clock for all channels on all devices.

In this mode, the BONDST[1:0] signals are used to pass channel bonding status between the different devices. This is necessary to keep the data on all bonded devices in common alignment. One device must be selected as the controlling device by driving the MASTER pin on that device LOW. All other devices must have their MASTER pin HIGH to prevent having multiple active drivers on the BONDST bus. Within the master device, a single receive channel is selected as the controlling channel for generation of the different BONDST[1:0] status. This selection is made using the RXCLKB+ and RXCLKD+ inputs, as shown in *Table 22*. This allows the master channel selection to be dynamically changed through external control of the MASTER, RXCLKB+, and RXCLKD+ inputs.

Note: Any change in master device or channel should be followed by assertion of TRSTZ to properly initialize the devices.

Output Bus

Each receive channel presents a 11-signal output bus consisting of

- an 8-bit data bus
- a 3-bit status bus

The signals present on this output bus are modified by the present operating mode of the PSI transceiver block as selected by DECMODE. This mapping is shown in *Table 24*.

When the 10B/8B decoder is bypassed (DECMODE = LOW), the framed 10-bit value is presented to the associated output register, along with a status output (COMDETx) indicating if the character in the output register is one of the selected framing characters. The bit usage and mapping of the external signals to the raw 10B coded character is shown in *Table 25*.

The COMDETx status outputs operate the same regardless of the bit combination selected for character framing by the FRAMCHAR input. They are HIGH when the character in the

Table 24. Output Register Bit Assignments

Signal Name	DECMODE = LOW	DECMODE = MID or HIGH
RXSTx[2] (LSB)	COMDETx	RXSTx[2]
RXSTx[1]	DOUTx[0]	RXSTx[1]
RXSTx[0]	DOUTx[1]	RXSTx[0]
RXDx[0]	DOUTx[2]	RXDx[0]
RXDx[1]	DOUTx[3]	RXDx[1]
RXDx[2]	DOUTx[4]	RXDx[2]
RXDx[3]	DOUTx[5]	RXDx[3]
RXDx[4]	DOUTx[6]	RXDx[4]
RXDx[5]	DOUTx[7]	RXDx[5]
RXDx[6]	DOUTx[8]	RXDx[6]
RXDx[7] (MSB)	DOUTx[9]	RXDx[7]

Table 25. Decoder Bypass Mode (DECMODE = LOW)

Signal Name	Bus Weight	10B Name
RXSTx[2] (LSB)	COMDETx	
RXSTx[1]	2^0	a
RXSTx[0]	2^1	b
RXDx[0]	2^2	c
RXDx[1]	2^3	d
RXDx[2]	2^4	e
RXDx[3]	2^5	i
RXDx[4]	2^6	f
RXDx[5]	2^7	g
RXDx[6]	2^8	h
RXDx[7] (MSB)	2^9	j

output register contains the selected framing character at the proper character boundary, and LOW for all other bit combinations.

When the low-latency framer and half-rate receive port clocking are also enabled (RFMODE = LOW, RXRATE = HIGH, and RXCKSEL ≠ LOW), the framer will stretch the recovered clock to the nearest 20-bit boundary such that the rising edge of RXCLKx+ occurs when COMDETx is present on the associated output bus.

When the standard framer is enabled and half-rate receive port clocking are also enabled (RFMODE ≠ LOW and RXRATE = HIGH), the output clock is not modified when framing is detected, but a single pipeline stage may be added or subtracted from the data stream by the framer logic such that the rising edge of RXCLKx+ occurs when COMDET is present on the associated output bus.

This adjustment only occurs when the framer is enabled (RFEN = HIGH). When the framer is disabled, the clock boundaries are not adjusted, and COMDETx may be active during the rising edge of RXCLKx– (if an odd number of characters were received following the initial framing).

Receive Status Bits

When the 10B/8B decoder is enabled (DECMODE ≠ LOW), each character presented at the output register includes three associated status bits. These bits are used to identify

- if the contents of the data bus are valid,
- the type of character present,
- the state of receive BIST operations (regardless of the state of DECMODE),
- character violations,
- and channel bonding status

These conditions normally overlap; i.e., a valid data character received with incorrect running disparity is not reported as a valid data character. It is instead reported as a decoder violation of some specific type. This implies a hierarchy or priority level to the various status bit combinations. The hierarchy and value of each status is listed in *Table 26*.

Table 26. Receive Character Status Bits

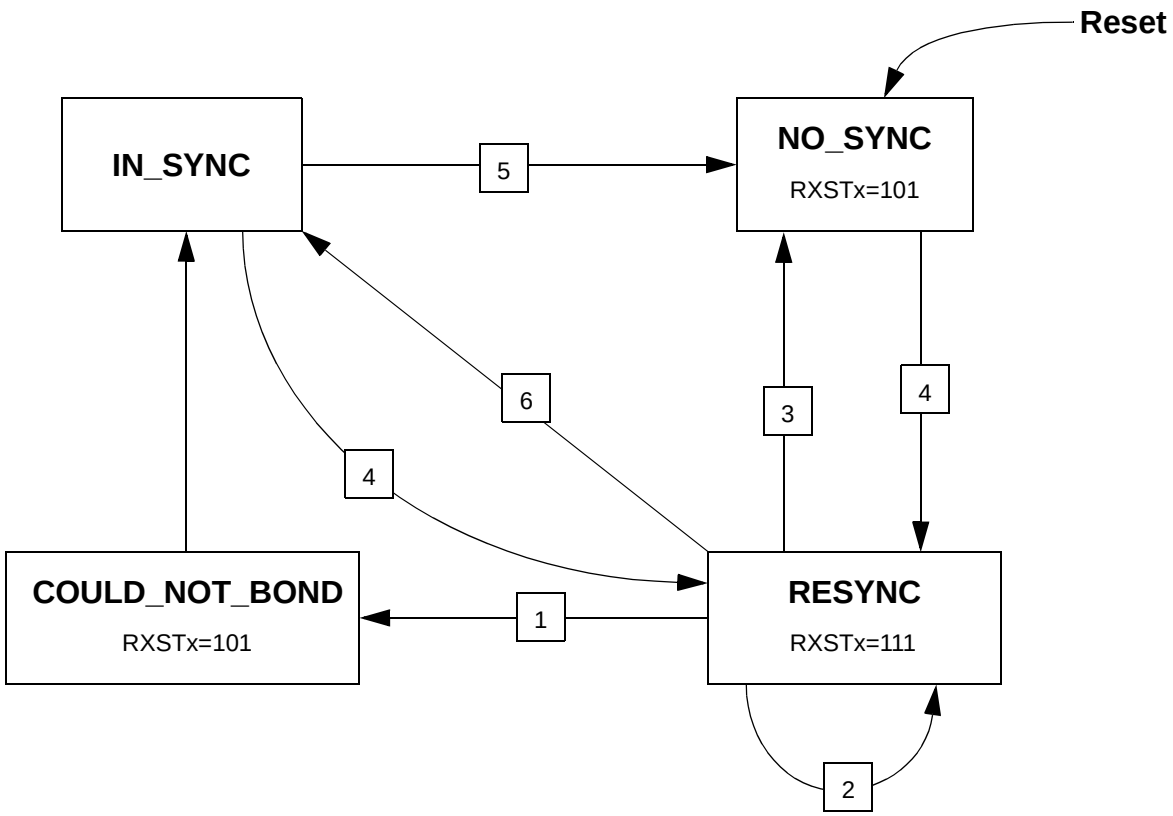
RXSTx[2:0]	Priority	Description		
		Type-A Status	Type-B Status	Receive BIST Status (Receive BIST = Enabled)
000	7	**Normal Character Received.** The valid Data character on the output bus meets all the formatting requirements of Data characters listed in *Table 29*.		**BIST Data Compare.** Character compared correctly
001	7	**Special Code Detected.** The valid special character on the output bus meets all the formatting requirements of Special Code characters listed in *Table 30*, but is not the presently selected framing character or a decoder violation indication.		**BIST Command Compare.** Character compared correctly
010	2	**Receive Elasticity Buffer Underrun/Overrun Error.** The receive buffer was not able to add/drop a K28.5 or framing character.	**Channel Lock Detected.** Asserts when the bonded channels have detected RESYNC within the allotted window. Presented only on the last cycle before aligned data is presented.	**BIST Last Good.** Last Character of BIST sequence detected and valid.
011	5	**Framing Character detected.** This indicates that a character matching the patterns identified as a framing character (as selected by FRAMCHAR) was detected. The decoded value of this character is present in the associated output bus.		
100	4	**Codeword Violation.** The character on the output bus is a C0.7. This indicates that the received character cannot be decoded into any valid character.		**BIST Last Bad.** Last Character of BIST sequence detected invalid.
101	1	**Loss of Sync.** The character on the bus is invalid, due to an event that has caused the receive channels to lose synchronization. When channel bonding is enabled, this indicates that one or more channels have either lost bit synchronization (loss of character framing), or that the bonded channels are no longer in proper character alignment. When the channels are operated independently (with the decoder enabled), this indicates a PLL Out of Lock condition.	**Loss of Sync.** The character on the bus is invalid, due to an event that has caused the receive channels to lose synchronization. When channel bonding is enabled, this indicates that one or more channels have either lost bit synchronization (loss of character framing), or that the bonded channels are no longer in proper character alignment. When the channels are operated independently (with the decoder enabled), this indicates a loss of character framing. Also used to indicate receive Elasticity Buffer underflow/overflow errors.	**BIST Start.** Receive BIST is enabled on this channel, but character compares have not yet commenced. This also indicates a PLL Out of Lock condition, and Elasticity Buffer overflow/underflow conditions.
110	6	**Running Disparity Error.** The character on the output bus is a C4.7, C1.7, or C2.7.		**BIST Error.** While comparing characters, a mismatch was found in one or more of the decoded character bits.
111	3	**Resync.** The receiver state machine is in the Resynchronization state. In this state the data on the output bus reflects the presently decoded FRAMCHAR.		**BIST Wait.** The receiver is comparing characters. but has not yet found the start of BIST character to enable the LFSR.

Within these status decodes, there are three forms of status reporting. The two normal or data status reporting modes (Type A and Type B) are selectable through the RXMODE[0] input. These status types allow compability with legacy systems, while allowing full reporting in new systems. The third status type is used for reporting receive BIST status and progress. These status values are generated in part by the Receive Synchronization State Machine, and are listed in *Table 26*.

Receive Synchronization State Machine

Each receive channel contains a Receive Synchronization state machine. This machine handles loss and recovery of bit, channel, and word framing, and part of the control for channel bonding. This state machine is enabled whenever the receive channels are configured for channel bonding (RXMODE[1] ≠ LOW). Separate forms of the state machine exist for the two different types of status reporting.

When operated without channel bonding (RXMODE[1] = LOW, RX Modes 0 and 2), these state machines are disabled and



#	State Transition Conditions
1	(BOND_INH = LOW) AND (Deskew Window Expired)
2	FRAMCHAR Detected
3	(Elasticity Buffer Under/Overrun) OR (RX PLL Loss of Lock) OR (Any Decoder Error)
4	Four Consecutive FRAMCHAR Detected
5	(Elasticity Buffer Under/Overrun) OR (RX PLL Loss of Lock) OR (Four Consecutive Decoder Errors) OR (Invalid Minus Valid = 4)
6	Valid Character other than a FRAMCHAR

Figure 13. Status Type-A Receive State Machine

characters are decoded directly. In RX Mode 0 the RESYNC (111b) status is never reported. In RX Mode 2, neither the RESYNC (111b) or Channel Lock Detected (010b) status are reported.

Status Type-A Receive State Machine

This machine has four primary states: NO_SYNC, RESYNC, COULD_NOT_BOND, and IN_SYNC, as shown in *Figure 13*. The IN_SYNC state can respond with multiple status types, while others can respond with only one type.

Status Type-B Receive State Machine

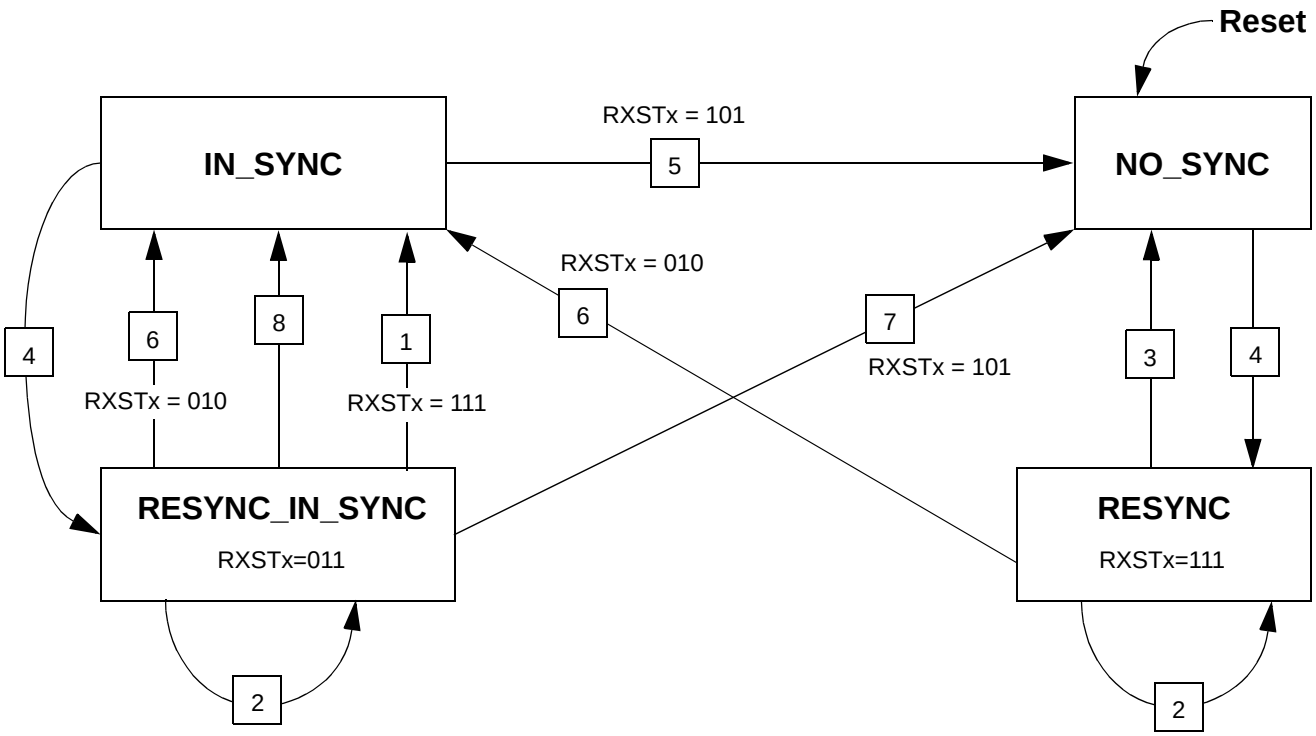
This machine has four primary states: NO_SYNC, RESYNC, IN_SYNC, and COULD_NOT_BOND, as shown in *Figure 14*. Some of these state can respond with only one status value, while others can respond with multiple status types.

BIST Status State Machine

When a receive path is enabled to look for and compare the received data stream with the BIST pattern, the RXSTx[2:0] bits identify the present state of the BIST compare operation.

The BIST state machine has multiple states, as shown in *Table 26*. When the receive PLL detects an out-of-lock condition, the BIST state is forced to the Start-of-BIST state, regardless of the present state of the BIST state machine. If the number of detected errors ever exceeds the number of valid matches by greater than 16, the state machine is forced to the WAIT_FOR_BIST state where it monitors the interface for the first character of the next BIST sequence (D0.0). Also, if the Elasticity Buffer ever hits and overflow/underflow condition, the status is forced to the BIST_START until the buffer is re-centered (approximately nine character periods).

To ensure compatibility between the source and destination BIST operating modes, the sending and receiving ends of the BIST sequence must both have RXCKSEL = MID or both have RXCKSEL ≠ MID.



#	Condition
1	(BOND_INH = LOW OR Master Channel Did Not Bond) AND Deskew Window Expired
2	FRAMCHAR Detected
3	(Elasticity Buffer Under/Overrun) OR (RX PLL Loss of Lock) OR (Any Decoder Error) OR ((BOND_INH = LOW OR Master Channel Did Not Bond) AND (Deskew Window Expired))
4	Four Consecutive FRAMCHAR Detected
5	(Elasticity Buffer Under/Overrun) OR (RX PLL Loss of Lock) OR (Four Consecutive Decoder Errors) OR (Invalid Minus Valid = 4)
6	Last FRAMCHAR Before a Valid Character AND Bonded to Master Channel
7	(Elasticity Buffer Under/Overrun) OR (RX PLL Loss of Lock)
8	Decoder Error

Figure 14. Status Type-B Receive State Machine

IEEE 1149.1 Compliant JTAG Operation

The PSI family has an IEEE std 1149.1 JTAG interface for both Boundary Scan and ISR operations.

Four dedicated pins are reserved on each device for use by the Test Access Port (TAP).

Boundary Scan

The PSI family supports Bypass, Sample/Preload, Extest, Intest, Idcode and Usercode boundary scan instructions. The JTAG interface is shown in *Figure 15*.

Frequency Agile devices also allow system-level diagnosis of transceiver interface and interconnect. Boundary scan is supported on the LVCMOS signals, inputs and outputs. The high-speed serial inputs are not part of the JTAG test chain.

In-System Reprogramming (ISR)

In-System Reprogramming is the combination of the capability to program or reprogram a device on-board, and the ability to support design changes without changing the system timing or device pinout. This combination means design changes during debug or field upgrades do not cause board respins. The PSI family implements ISR by providing a JTAG compliant interface for on-board programming, robust routing resources for pinout flexibility, and a simple timing model for consistent system performance.

Configuration

The CPLD block in each device of the PSI family is designed with Self-Boot capability. An embedded on-chip EEPROM is used to store configuration data. For PSI devices, programming is defined as the loading of a user's design into the inter-

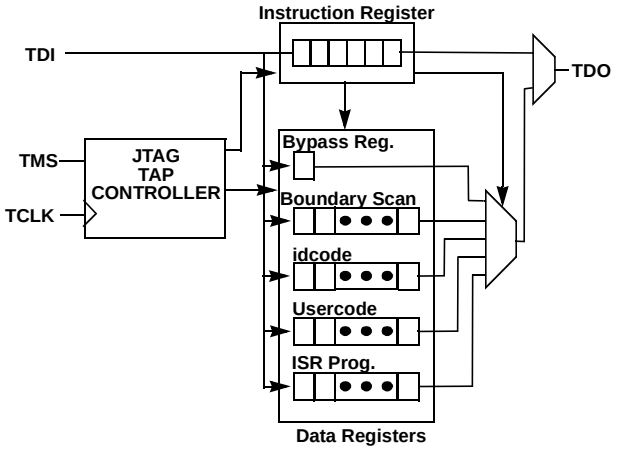


Figure 15. JTAG Interface

nal EEPROM. Configuration, on the other hand, is defined as the loading of a user's design into the volatile CPLD block.

Configuration can begin in two ways. It can be initiated by toggling the *Reconfig* pin from LOW to HIGH, or by issuing the appropriate IEEE std 1149.1 JTAG instruction to the PSI device via the JTAG interface. There are two IEEE std 1149.1 JTAG instructions that initiate configuration of the PSI. The *Self Config* instruction causes the PSI to (re)configure with data store in the internal EEPROM. The *Load Config* instruction causes the PSI to (re)configure with data provided by other sources such as a PC, automatic test equipment (ATE), or an embedded microcontroller/processor via the JTAG port.

There are multiple configuration options available for issuing the IEEE std 1149.1 JTAG instructions to the PSI. The first method is to use a PC with the C3 ISR programming cable and software. With this method, the ISR pins of the PSI devices in the system are routed to a connector at the edge of the printed circuit board. The C3 ISR programming cable is then connected between the PC and this connector. A simple configuration file instructs the ISR software of the programming operations to be performed on the PSI devices in the system. The ISR software then automatically completes all of the necessary data manipulations required to accomplish configuration, reading, verifying, and other ISR functions. For more information on the Cypress ISR interface, see the Programming/ISR application notes at http://www.cypress.com/pld/pldappnotes.html.

For systems with embedded controllers/processors, a controller/processor may be used to configure the PSI. The PSI ISR software assists in this method by converting the device HEX file into the ISR serial stream that contains the ISR instruction information and the addresses and data of locations to be configured. The controller/processor then simply directs this ISR stream to the chain of PSI devices to complete the desired reconfiguration or diagnostic operations. Contact your local sales office for information on availability of this option.

Programming

The on-chip EEPROM device of the CPLD block is programmed by issuing the appropriate IEEE std 1149.1 JTAG instruction. This can be done automatically using ISR/STAPL software. The configuration bits are sent from a PC through the JTAG port into the PSI via the C3 ISR programming cable. The data is then passed to the internal EEPROM through the Non-Volatile (NV) port of the CPLD block. For more information on how to program the PSI through ISR/STAPL, please refer to the ISR/STAPL User Guide.

Third-Party Programmers

Cypress support is available on a wide variety of third-party programmers. All major programmers (including BP Micro, System General, Hi-Lo) support the PSI family.

Development Software Support

Warp

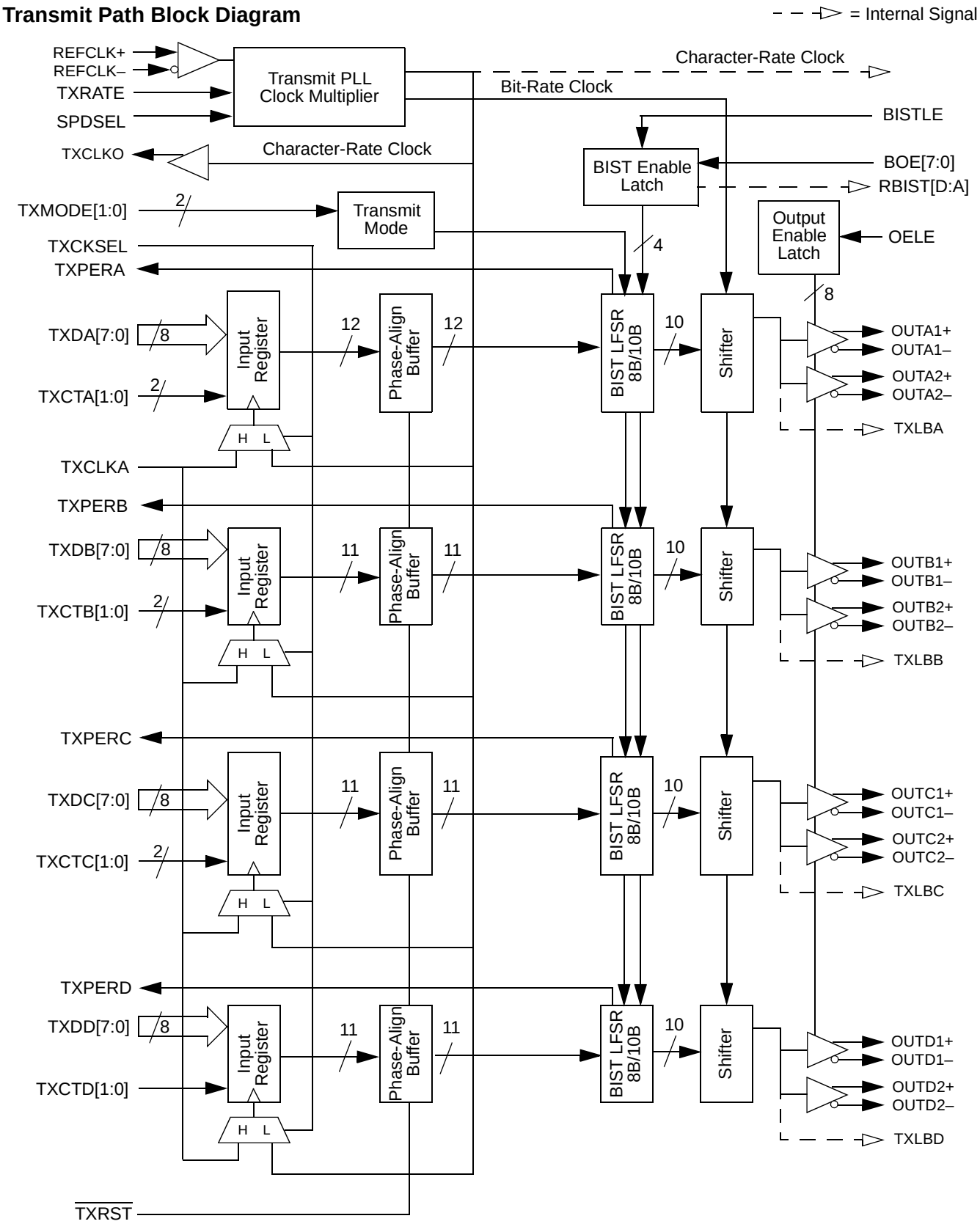
Warp is a state-of-the-art design environment for designing with Cypress programmable logic. *Warp* utilizes a subset of IEEE 1076/1164 VHDL and IEEE 1364 as the Hardware Description Language (HDL) for design entry. *Warp* accepts VHDL or Verilog input, synthesizes and optimizes the entered design, and outputs a configuration bitstream for the desired Delta39K device. For simulation, *Warp* provides a graphical waveform simulator as well as VHDL and Verilog Timing Models.

VHDL and Verilog are open, powerful, non-proprietary Hardware Description Languages (HDLs) that are standards for behavioral design entry and simulation. HDL allows designers to learn a single language that is useful for all facets of the design process.

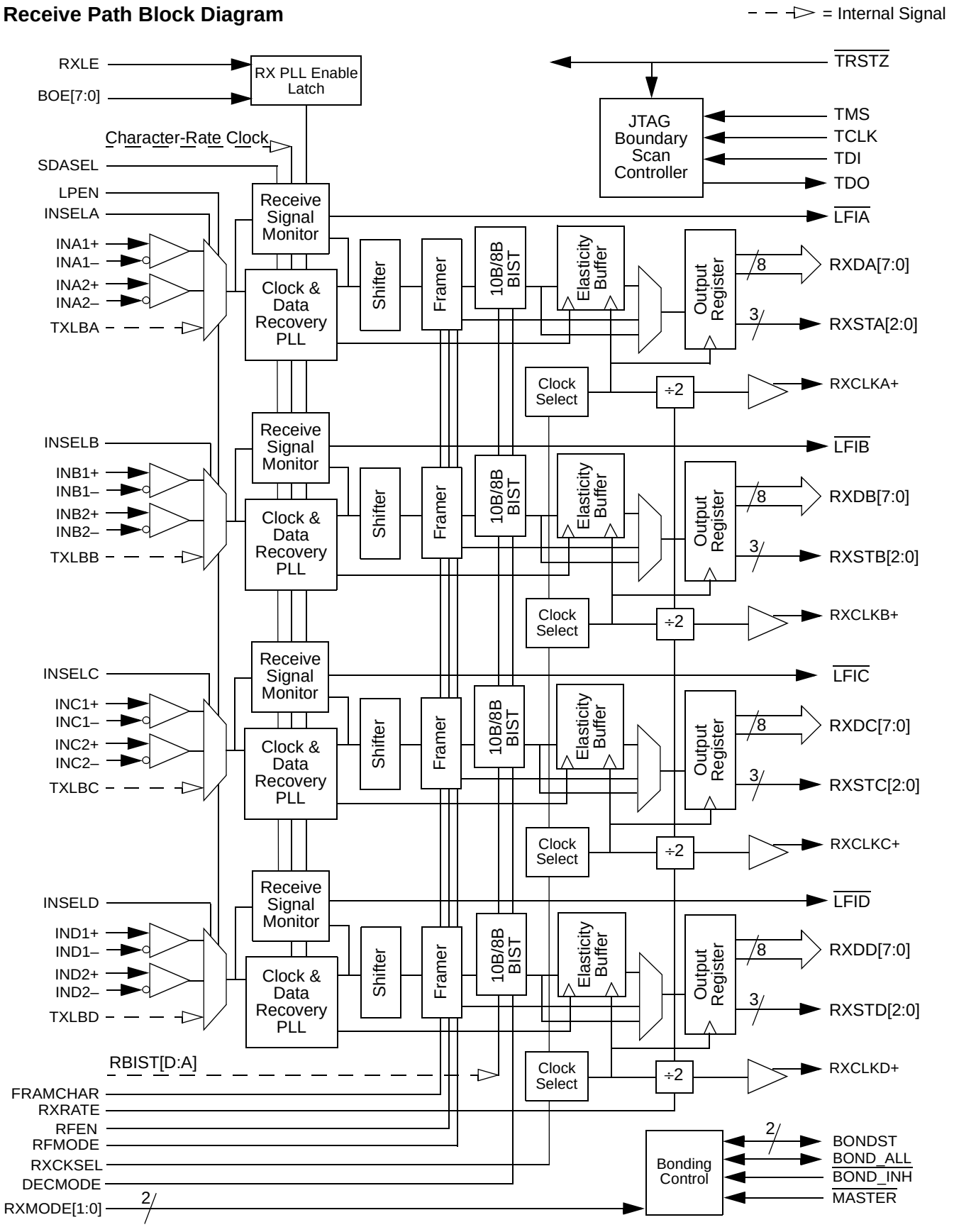
Third-Party Software

Cypress products are supported in a number of third-party design entry and simulation tools. Refer to the third-party software data sheet or contact your local sales office for a list of currently supported third party vendors.

Transmit Path Block Diagram



Receive Path Block Diagram



Maximum Ratings

(Above which the useful life may be impaired. For user guidelines, not tested.)

Storage Temperature .....–65°C to +150°C

Soldering Temperature.....220°C

Ambient Temperature with Power Applied.....–40°C to +85°C

Junction Temperature.....135°C

V_{CC} relative to Ground Potential..... –0.5V to 4.2V

V_{CCIO} relative to Ground Potential..... –0.5V to 4.6V

DC Voltage Applied to Outputs in High Z State –0.5V to 4.5V

Output Current into LVCMOS Outputs (LOW)..... 30 mA

DC Input voltage.....–0.5V to 4.5V

DC Current into Outputs..... .....± 20 mA[6]

Static Discharge Voltage.....> 2001 V (per MIL-STD-883, Method 3015)

Latch-Up Current.....> 200 mA

Operating Range

Range	Ambient Temperature	V_{CC}	V_{DDQ}
Commercial	0°C to +70°C	3.3V ± 10%	1.4V to 1.6V

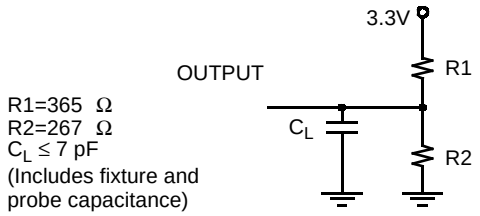
Operating Range

Range	Ambient Temperature	Junction Temperature	Output Condition	V_{CCIO}	V_{CC}	V_{CCJTAG}/ V_{CCCNFG}	V_{CCPLL}	V_{CCPRG}
Commercial	0°C to +70°C	0°C to +85°C	3.3V	3.3V ± 0.3V	3.3V ± 0.3V	Same as V_{CCIO}	Same as V_{CC}	3.3V ± 0.3V
			2.5V	2.5V ± 0.2V				
			1.8V	1.8V ± 0.15V				
			1.5V	1.5V ± 0.1V				

Notes:

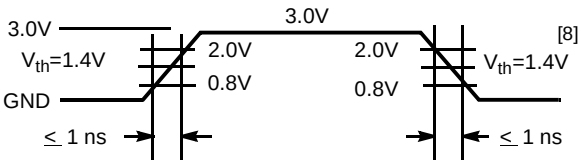
6. DC current into outputs is 36 mA with HSTL III and 48 mA with HSTL IV

AC Test Loads and Waveforms

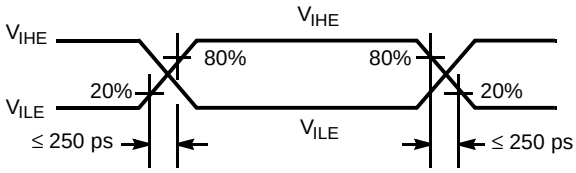


(a) LVTTL AC Test Load [7]

(b) CML AC Test Load [7]



(c) LVTTL Input Test Waveform



(d) PECL Input Test Waveform

Notes:

7. Cypress uses constant current (ATE) load configurations and forcing functions. This figure is for reference only.
8. The LVTTL switching threshold is 1.4V. All timing references are made relative to the point where the respective rising or falling signal edge crosses this threshold voltage.

Electrical Characteristics Over the Operating Range

DC Characteristics

Parameter	Description	Test Conditions	V_{CCIO} = 3.3V		V_{CCIO} = 2.5V		V_{CCIO} = 1.8V		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
V_{DRINT}	Data Retention V_{CC} Voltage (config data may be lost below this)		1.5		1.5		1.5		V
V_{DRIO}	Data Retention V_{CCIO} Voltage (config data may be lost below this)		1.2		1.2		1.2		V
I_{IX}	Input Leakage Current	$GND \le V_I \le 3.6V$	–10	10	–10	10	–10	10	µA
I_{OZ}	Output Leakage Current	$GND \le V_O \le V_{CCIO}$	–10	10	–10	10	–10	10	µA
I_{OS}[9]	Output Short Circuit Current	V_{CCIO} = Max., V_{OUT} = 0.5V		–160		–160		–160	mA
I_{BHL}	Input Bus Hold LOW Sustaining Current	V_{CC} = Min., $V_{PIN} = V_{IL}$	+40		+30		+25		µA
I_{BHH}	Input Bus Hold HIGH Sustaining Current	V_{CC} = Min., $V_{PIN} = V_{IH}$	–40		–30		–25		µA
I_{BHLO}	Input Bus Hold LOW Overdrive Current	V_{CC} = Max.		+250		+200		+150	µA
I_{BHHO}	Input Bus Hold HIGH Overdrive Current	V_{CC} = Max.		–250		–200		–150	µA

Capacitance[10]

Parameter	Description	Test Conditions	Min.	Max.	Unit
$C_{I/O}$	Input/Output Capacitance	$V_{in} = V_{CCIO}$ @ f = 1 MHz 25°C		12	pF
C_{PCI}	PCI compliant I/O Capacitance	$V_{in} = V_{CCIO}$ @ f = 1 MHz 25°C		8	pF
C_{CLK}	Clock Signal Capacitance	$V_{in} = V_{CCIO}$ @ f = 1 MHz 25°C	5	12	pF
C_{INPECL}	PECL Input Capacitance	V_{CC} = 3.3V @ f = 1 MHz 25°C		4	pF
C_{SD1}	SD Pin Input Capacitance	V_{CC} = 3.3V @ f = 1 MHz 25°C		5	pF
C_{INTTL}	TTL Input Capacitance	V_{CC} = 3.3V @ f = 1 MHz 25°C		7	pF

DC Specifications - Power

Parameter	Device	Description	Test Conditions	Standby	Typical	Unit
I_{CC2}[11]	P15G04K100	Active Power Supply Current	Frequency = Max Commercial	11	1800	mA
	P15G04K200	Active Power Supply Current	Frequency = Max Commercial	11	2000	mA
	P15G08K200	Active Power Supply Current	Frequency = Max Commercial	11	3000	mA

Notes:
9. Not more than one output should be tested at a time. Duration of the short circuit should not exceed 1 second. V_{OUT}=0.5V has been chosen to avoid test problems caused by tester ground degradation. Tested initially and after any design or process changes that may affect these parameters.
10. Tested initially and after any design or process changes that may affect these parameters, but not 100% tested.
11. Typical I_{CC} is measured with V_{CC} = 3.3V, T_A = 25°C, RFEN = LOW, and outputs unloaded.

DC Characteristics (I/O)

Input/ Output Standard	V_{REF} (V)		V_{CCIO} (V)	V_{OH} (V)		V_{OL} (V)		V_{IH} (V)		V_{IL} (V)	
	Min.	Max.		@ I_{OH} =	V_{OH} (min.)	@ I_{OL} =	V_{OL} (max.)	Min.	Max.	Min.	Max.
LVTTL			3.3	–4 mA	2.4	4 mA	0.4	2.0V	V_{CCIO}+0.3	–0.3V	0.8V
LVCMOS			3.3	–0.1 mA	V_{CCIO}–0.2V	0.1 mA	0.2	2.0V	V_{CCIO}+0.3	–0.3V	0.8V
LVCMOS3			3.0	–0.1 mA	V_{CCIO}–0.2V	0.1 mA	0.2	2.0V	V_{CCIO}+0.3	–0.3V	0.8V
LVCMOS2			2.5	–0.1 mA	2.1	0.1 mA	0.2	1.7V	V_{CCIO}+0.3	–0.3V	0.7V
				–1.0 mA	2.0	1.0 mA	0.4				
				–2.0 mA	1.7	2.0 mA	0.7				
LVCMOS18			1.8	–0.1 mA	V_{CCIO}–0.2V	0.1 mA	0.2	0.65V_{CCIO}	V_{CCIO}+0.3	–0.3V	0.35V_{CCIO}
				– 2 mA	V_{CCIO}– 0.45V	2.0 mA	0.45				
3.3V PCI			3.3	–0.5 mA	0.9V_{CCIO}	1.5 mA	0.1V_{CCIO}	0.5V_{CCIO}	V_{CCIO}+0.5	–0.5V	0.3V_{CCIO}
GTL+	0.9	1.1	Note 12			Note 13	0.6	V_{REF}+0.2			V_{REF}–0.2
SSTL3 I	1.3	1.7	3.3	–8 mA	V_{CCIO}–1.1V	8 mA	0.7	V_{REF}+0.2	V_{CCIO}+0.3	–0.3V	V_{REF}–0.2
SSTL3 II	1.3	1.7	3.3	–16 mA	V_{CCIO}–0.9V	16 mA	0.5	V_{REF}+0.2	V_{CCIO}+0.3	–0.3V	V_{REF}–0.2
SSTL2 I	1.15	1.35	2.5	–7.6 mA	V_{CCIO}– 0.62V	7.6 mA	0.54	V_{REF}+1.8	V_{CCIO}+0.3	–0.3V	V_{REF}–0.18
SSTL2 II	1.15	1.35	2.5	–15.2 mA	V_{CCIO}– 0.43V	15.2 mA	0.35	V_{REF}+1.8	V_{CCIO}+0.3	–0.3V	V_{REF}–0.18
HSTL I	0.68	0.9	1.5	–8 mA	V_{CCIO}–0.4V	8 mA	0.4	V_{REF}+1.0	V_{CCIO}+0.3	–0.3V	V_{REF}–0.1
HSTL II	0.68	0.9	1.5	–16 mA	V_{CCIO}–0.4V	16 mA	0.4	V_{REF}+1.0	V_{CCIO}+0.3	–0.3V	V_{REF}–0.1
HSTL III	0.68	0.9	1.5	–8 mA	V_{CCIO}–0.4V	24 mA	0.4	V_{REF}+1.0	V_{CCIO}+0.3	–0.3V	V_{REF}–0.1
HSTL IV	0.68	0.9	1.5	–8 mA	V_{CCIO}–0.4V	48 mA	0.4	V_{REF}+1.0	V_{CCIO}+0.3	–0.3V	V_{REF}–0.1

Notes:
12. See “Power-up Sequence Requirements” for V_{CCIO} requirement.
13. 25Ω resistor terminated to termination voltage of 1.5V.

Parameter	Description	Test Conditions	Min.	Max.	Unit
LVTTL Compatible Outputs					
V_{OHT}	Output HIGH Voltage	I_{OH} = – 4 mA, V_{CC} = Min.	2.4	V_{CC}	V
V_{OLT}	Output LOW Voltage	I_{OL} = 4 mA, V_{CC} = Min.	0	0.4	V
I_{OST}	Output Short Circuit Current	V_{OUT} = 0V [14]	–30	–100	mA
I_{OZL}	High-Z Output Leakage Current		–20	20	µA
LVTTL Compatible Inputs					
V_{IHT}	Input HIGH Voltage		2.0	V_{CC}+0.3	V
V_{ILT}	Input LOW Voltage		–0.5	0.8	V
I_{IHT}	Input HIGH Current	REFCLK Input, V_{IN} = V_{CC}		+40	µA
		Other Inputs, V_{IN} = V_{CC}		+40	µA
I_{ILT}	Input LOW Current	REFCLK Input, V_{IN} = 0.0V		–500	µA
		Other Inputs, V_{IN} = 0.0V		–40	µA
I_{IHPDT}	Input HIGH Current with internal pull-down	V_{IN} = V_{CC}		+200	µA
I_{ILPUT}	Input LOW Current with internal pull-up	V_{IN} = 0.0V		–200	µA

Parameter	Description	Test Conditions		Min.	Max.	Unit
LVDIFF Inputs: REFCLK±						
V_{DIFF} [15]	Input Differential Voltage			400	V_{CC}	mV
V_{IHHP}	Highest Input HIGH Voltage			1.0	V_{CC}	V
V_{ILLP}	Lowest Input LOW voltage			GND	V_{CC}–0.4V	V
V_{COM} [16]	Common Mode Range			1.0	V_{CC}–1.2V	V
3-Level Inputs						
V_{IHH}	Three-Level Input HIGH Voltage	Min. ≤ V_{CC} ≤ Max.		0.87 * V_{CC}	V_{CC}	V
V_{IMM}	Three-Level Input MID Voltage	Min. ≤ V_{CC} ≤ Max.		0.47 * V_{CC}	0.53 * V_{CC}	V
V_{ILL}	Three-Level Input LOW Voltage	Min. ≤ V_{CC} ≤ Max.		0.0	0.13 * V_{CC}	V
I_{IHH}	Input HIGH Current	V_{IN} = V_{CC}			200	µA
I_{IMM}	Input MID current	V_{IN} = V_{CC}/2		–50	50	µA
I_{ILL}	Input LOW current	V_{IN} = GND			–200	µA
Differential CML Serial Outputs: OUTA1±, OUTA2±, OUTB1±, OUTB2±, OUTC1±, OUTC2±, OUTD1±, OUTD2±						
V_{OHC}	Output HIGH Voltage (V_{CC} referenced)	100Ω differential load		V_{CC}–0.85	V_{CC}–0.2	V
		150Ω differential load		V_{CC}–0.85	V_{CC}–0.2	V
V_{OLC}	Output LOW Voltage (V_{CC} referenced)	100Ω differential load		V_{CC}–1.1	V_{CC}–0.7	V
		150Ω differential load		V_{CC}–1.1	V_{CC}–0.7	V
V_{ODIF}	Output Differential Voltage |(OUT+) – (OUT–)|	100Ω differential load		300	800	mV
		150Ω differential load		450	1200	mV
Differential Serial Line Receiver Inputs: INA1±, INA2±, INB1±, INB2±, INC1±, INC2±, IND1±, IND2±						
V_{DIFF}	Input Differential Voltage |(IN+) – (IN–)|			100	1200	mV
V_{IHE}	Highest Input HIGH Voltage			V_{CC}–1.2	V_{CC}	V
V_{ILE}	Lowest Input LOW Voltage			V_{CC}–2.0	V_{CC}–1.45	V
I_{IHE}	Input HIGH Current	V_{IN} = V_{IHE} Max.			1300	µA
I_{ILE}	Input LOW Current	V_{IN} = V_{ILE} Min.		–600		µA
Miscellaneous				**Typ.**	**Max.**	
I_{CC} [17]	Power Supply Current	Freq. = Max.	Commercial	1800	2000	mA

Notes:

14. Tested one output at a time, output shorted for less than one second, less than 10% duty cycle.
15. This is the minimum difference in voltage between the true and complement inputs required to ensure detection of a logic-1 or logic-0.
16. The common mode range defines the allowable range of REFCLK+ and REFCLK– (relative to the associated power rail) when |(REFCLK+) – (REFCLK–)| = 0V. This marks the zero-crossing between the true and complement inputs as the signal switches between HIGH and LOW.
17. Maximum I_{CC} is measured with V_{CC} = MAX, RFEN = LOW, with all serial channels sending a constant alternating 01 pattern, and outputs unloaded. Typical I_{CC} is measured under similar conditions except with V_{CC} = 3.3V, T_A = 25°C.

Configuration Parameters

Parameter	Description	Min.	Units.
$t_{RECONFIG}$	*Reconfig* pin LOW time before it goes HIGH	200	ns

Power-up Sequence Requirements

- Upon power-up, all the outputs remain three-stated until all the V_{CC} pins have powered-up to the nominal voltage and the part has completed configuration.
- The part will not start configuration until V_{CC}, V_{CCIO}, V_{CCJTAG}, V_{CCCNFG}, V_{CCPLL} and V_{CCPRG} have reached nominal voltage.
- V_{CC} pins can be powered up in any order. This includes V_{CC}, V_{CCIO}, V_{CCJTAG}, V_{CCCNFG}, V_{CCPLL} and V_{CCPRG}.
- All V_{CCIO}s on a bank should be tied to the same potential and powered up together.
- All V_{CCIO}s (even the unused banks) need to be powered up to at least 1.5V before configuration has completed.
- Maximum ramp time for all V_{CC}s should be 0V to nominal voltage in 100 ms.

Switching Characteristics

Timing Parameter Values

Parameter	Description	200		Unit
		Min.	Max.	
Combinatorial Mode Parameters				
t_{PD}	Delay from any pin input, through any cluster on the channel associated with that pin input, to any pin output on the horizontal or vertical channel associated with that cluster		7.5	ns
t_{EA}	Global control to output enable		5.0	ns
t_{ER}	Global control to output disable		5.0	ns
t_{PRR}	Asynchronous macrocell RESET or PRESET recovery time from any pin input on the horizontal or vertical channel associated with the cluster the macrocell is in	6.0		ns
t_{PRO}	Asynchronous macrocell RESET or PRESET from any pin input on the horizontal or vertical channel associated with the cluster that the macrocell is in to any pin output on those same channels	10		ns
t_{PRW}	Asynchronous macrocell RESET or PRESET minimum pulse width, from any pin input to a macrocell in the farthest cluster on the horizontal or vertical channel the pin is associated with	3.6		ns
Synchronous Clocking Parameters				
t_{MCS}	Set-up time of any input pin to a macrocell in any cluster on the channel associated with that input pin, relative to a global clock	3.0		ns
t_{MCH}	Hold time of any input pin to a macrocell in any cluster on the channel associated with that input pin, relative to a global clock	0.0		ns
t_{MCCO}	Global clock to output of any macrocell to any output pin on the horizontal or vertical channel associated with the cluster that macrocell is in		6.0	ns
t_{IOS}	Set-up time of any input pin to the I/O cell register associated with that pin, relative to a global clock	1.0		ns
t_{IOH}	Hold time of any input pin to the I/O cell register associated with that pin, relative to a global clock	1.0		ns
t_{IOCO}	Clock to output of an I/O cell register to the output pin associated with that register		4.0	ns
t_{SCS}	Macrocell clock to macrocell clock through array logic within the same cluster	4.0		ns
t_{SCS2}	Macrocell clock to macrocell clock through array logic in different clusters on the same channel	5.0		ns
t_{ICS}	I/O register clock to any macrocell clock in a cluster on the channel the I/O register is associated with	5.0		ns
t_{OCS}	Macrocell clock to any I/O register clock on the horizontal or vertical channel associated with the cluster that the macrocell is in	5.0		ns
t_{CHZ}	Clock to output disable (high-impedance)		3.5	ns
t_{CLZ}	Clock to output enable (low-impedance)	2.0		ns
f_{MAX}	Maximum frequency with internal feedback—within the same cluster		250	MHz
f_{MAX2}	Maximum frequency with internal feedback—within different clusters at the opposite ends of a horizontal or vertical channel		200	MHz
Product Term Clocking Parameters				
t_{MCSPT}	Set-up time for macrocell used as input register, from input to product term clock	3.0		ns
t_{MCHPT}	Hold time of macrocell used as an input register	1.0		ns
t_{MCCOPT}	Product term clock to output delay from input pin		8.0	ns
t_{SCS2PT}	Register to register delay through array logic in different clusters on the same channel using a product term clock	6.5		ns

Switching Characteristics

Timing Parameter Values (continued)

Parameter	Description	200		Unit
		Min.	Max.	
Channel Interconnect Parameters				
t_{CHSW}	Adder for a signal to switch from a horizontal to vertical channel and vice-versa		1.0	ns
t_{CL2CL}	Cluster to Cluster delay adder (through channels and channel PIM)		2.0	ns
Miscellaneous Parameters				
t_{CPLD}	Delay from the input of a cluster PIM, through a macrocell in the cluster, back to a cluster PIM input. This parameter can be added to the t_{PD} and t_{SCS} parameters for each extra pass through the AND/OR array required by a given signal path		3.0	ns
t_{MCCD}	Adder for carry chain logic per macrocell		0.25	ns
PLL Parameters				
t_{MCCJ}	Maximum cycle to cycle jitter time		0.50	ns
t_{DWSA}	PLL delay with skew adjustment		0.35	ns
t_{DWOSA}	PLL delay without any skew adjustment		0.35	ns
t_{LOCK}	Lock time for the PLL		3.0	ms
f_{PLLO}[18]	Output frequency of the PLL	6.2	266	MHz
f_{PLLI}[18]	Input frequency of the PLL	25	133	MHz

Note:
18. Tested initially and after any design or process changes that may affect these parameters, but not 100% tested.

Cluster Memory Timing Parameter Values

Parameter	Description	200		Unit
		Min.	Max.	
Asynchronous Mode Parameters				
t_{CLMAA}	Cluster memory access time. Delay from address change to read data out		11	ns
t_{CLMPWE}	Write enable pulse width	6.0		ns
t_{CLMSA}	Address set-up to the beginning of write enable	2.0		ns
t_{CLMHA}	Address hold after the end of write enable with both signals from the same I/O block	1.0		ns
t_{CLMSD}	Data set-up to the end of write enable	6.0		ns
t_{CLMHD}	Data hold after the end of write enable	0.5		ns
Synchronous Mode Parameters				
$t_{CLMCYC1}$	Clock cycle time for flow-through read and write operations (from macrocell register through cluster memory back to a macrocell register in the same cluster)	10		ns
$t_{CLMCYC2}$	Clock cycle time for pipelined read and write operations (from cluster memory input register through the memory to cluster memory output register)	5.0		ns
t_{CLMS}	Address, data, and WE set-up time of pin inputs, relative to a global clock	3.0		ns
t_{CLMH}	Address, data, and WE hold time of pin inputs, relative to a global clock	0.0		ns
t_{CLMDV1}	Global clock to data valid on output pins for flow through data		11	ns
t_{CLMDV2}	Global clock to data valid on output pins for pipelined data		7.5	ns
$t_{CLMMACS1}$	Cluster memory input clock to macrocell clock in the same cluster	8.0		ns
$t_{CLMMACS2}$	Cluster memory output clock to macrocell clock in the same cluster	5.0		ns
$t_{MACCLMS1}$	Macrocell clock to cluster memory input clock in the same cluster	4.0		ns
$t_{MACCLMS2}$	Macrocell clock to cluster memory output clock in the same cluster	6.5		ns
$t_{CLMCLAA}$	Asynchronous cluster memory access time from input of cluster to output of cluster	6.0		ns

Channel Memory Timing Parameter Values

Parameter	Description	200		Unit
		Min.	Max.	
Dual-Port Asynchronous Mode Parameters				
t_{CHMAA}	Channel memory access time. Delay from address change to read data out		11	ns
t_{CHMPWE}	Write enable pulse width	6.0		ns
t_{CHMSA}	Address set-up to the beginning of write enable	2.0		ns
t_{CHMHA}	Address hold after the end of write enable with both signals from the same I/O block	1.0		ns
t_{CHMSD}	Data set-up to the end of write enable	6.0		ns
t_{CHMHD}	Data hold after the end of write enable	0.5		ns
t_{CHMBA}	Channel memory asynchronous dual port address match (busy access time)		9.0	ns
Dual-Port Synchronous Mode Parameters				
$t_{CHMCYC1}$	Clock cycle time for flow through read and write operations (from macrocell register through channel memory back to a macrocell register in the same cluster)	10		ns
$t_{CHMCYC2}$	Clock cycle time for pipelined read and write operations (from channel memory input register through the memory to channel memory output register)	5.0		ns
t_{CHMS}	Address, data, and WE set-up time of pin inputs, relative to a global clock	3.3		ns
t_{CHMH}	Address, data, and WE hold time of pin inputs, relative to a global clock	0.0		ns
t_{CHMDV1}	Global clock to data valid on output pins for flow through data		11	ns
t_{CHMDV2}	Global clock to data valid on output pins for pipelined data		7.5	ns
t_{CHMBDV}	Channel memory synchronous dual-port address match (busy, clock to data valid)		9.0	ns
$t_{CHMMACS1}$	Channel memory input clock to macrocell clock in the same cluster	9.0		ns
$t_{CHMMACS2}$	Channel memory output clock to macrocell clock in the same cluster	5.0		ns
$t_{MACCHMS1}$	Macrocell clock to channel memory input clock in the same cluster	5.0		ns
$t_{MACCHMS2}$	Macrocell clock to channel memory output clock in the same cluster	7.0		ns
Synchronous FIFO Data Parameters				
t_{CHMCLK}	Read and write minimum clock cycle time	5.0		ns
t_{CHMFS}	Data, read enable, and write enable set-up time relative to pin inputs	4.0		ns
t_{CHMFH}	Data, read enable, and write enable hold time relative to pin inputs	0.0		ns
$t_{CHMFRDV}$	Data access time to output pins from rising edge of read clock (read clock to data valid)	7.0		
$t_{CHMMACS}$	Channel memory FIFO read clock to macrocell clock for read data	5.0		ns
$t_{MACCHMS}$	Macrocell clock to channel memory FIFO write clock for write data	5.0		ns
t_{CHMFO}	Read or write clock to respective flag output at output pins	11		ns
$t_{CHMMACF}$	Read or write clock to macrocell clock with FIFO flag	9		ns
t_{CHMFRS}	Master Reset Pulse Width	5.0		ns
$t_{CHMFRSR}$	Master Reset Recovery Time		4.0	ns
$t_{CHMFRSF}$	Master Reset to Flag and Data Output Time		10.0	ns
$t_{CHMSKEW1}$	Read/Write Clock Skew Time for Full Flag		2.0	ns
$t_{CHMSKEW2}$	Read/Write Clock Skew Time for Empty Flag		2.0	ns
$t_{CHMSKEW3}$	Read/Write Clock Skew Time for Boundary Flags		5.0	ns
Internal Parameters				
$t_{CHMCHAA}$	Asynchronous channel memory access time from input of channel memory to output of channel memory	7.0		ns

Transmitter LVTTL Switching Characteristics

Parameter	Description	Min.	Max	Unit
f_{TS}	TXCLKx Clock Cycle Frequency	20	150	MHz
t_{TXCLK}	TXCLKx Period	6.66	50	ns
t_{TXCLKH}	TXCLKx HIGH Time	2.2		ns
t_{TXCLKL}	TXCLKx LOW Time	2.2		ns
t_{TXCLKR} [19, 20, 21]	TXCLKx Rise Time	0.7	5	ns
t_{TXCLKF} [19, 20, 21]	TXCLKx Fall Time	0.7	5	ns
t_{TXDS}	Transmit Data Set-Up Time to TXCLKx↑ (TXCKSEL ≠ LOW)	1.5		ns
t_{TXDH}	Transmit Data Hold Time from TXCLKx↑ (TXCKSEL ≠ LOW)	1		ns
f_{TOS}	TXCLKO Clock Cycle Frequency (=1x or 2x REFCLK Frequency)	20	150	MHz
t_{TXCLKO}	TXCLKO Period	6.66	50	ns
$t_{TXCLKOD}$	TXCLKO Duty Cycle	47	53	%
t_{TXOH}	TXCLKO HIGH Time	1.5		ns
t_{TXOL}	TXCLKO LOW Time	1.5		ns

Receiver LVTTL Switching Characteristics

Parameter	Description	Min.	Max.	Unit
f_{RS}	RXCLKx Clock Output Frequency	10	150	MHz
t_{RXCLKP}	RXCLKx Period	6.66	100	ns
t_{RXCLKH}	RXCLKx HIGH Time (RXRATE = LOW)	1.5	51	ns
	RXCLKx HIGH Time (RXRATE = HIGH)	5	25	ns
t_{RXCLKL}	RXCLKx LOW Time (RXRATE = LOW)	1.5	51	ns
	RXCLKx HIGH Time (RXRATE = HIGH)	5	25	ns
t_{RXCLKD}	RXCLKx Duty Cycle	47	53	%
t_{RXCLKR} [19]	RXCLKx Rise Time	0.5	1.2	ns
t_{RXCLKF} [19]	RXCLKx Fall Time	0.5	1.2	ns
t_{RXDS} [22]	Status and Data Set-up Time From RXCLKx↑	2.0		ns
t_{RXDH} [22]	Status and Data Hold Time From RXCLKx↑	1.0		ns

Notes:

19. Parallel data output specifications are only valid if all outputs are loaded with similar DC and AC loads.
20. For a given operating frequency, neither rise or fall specification can be greater than 20% of the clock-cycle period or the data sheet maximum time.
21. The duty cycle specification is a simultaneous condition with the t_{REFH} and t_{REFL} parameters. This means that at faster character rates the REFCLK duty cycle cannot be as large as 30%-70%.
22. The ratio of rise time to falling time must not vary by greater than 2:1.

REFCLK Switching Characteristics

Parameter	Description	Min.	Max.	Unit
f_{REF}	REFCLK Clock Frequency	10	150	MHz
t_{REFCLK}	REFCLK Period	6.6	100	ns
t_{REFH}	REFCLK HIGH Time (TXRATE = HIGH)	5.9	70	ns
	REFCLK HIGH Time (TXRATE = LOW)	2.9	35	ns
t_{REFL}	REFCLK LOW Time (TXRATE = HIGH)	5.9	70	ns
	REFCLK LOW Time (TXRATE = LOW)	2.9	35	ns
t_{REFD}[23]	REFCLK Duty Cycle	30	70	%
t_{REFR}[19, 20, 21]	REFCLK Rise Time (20%-80%)	0.3	5	ns
t_{REFF}[19, 20, 21]	REFCLK Fall Time (20%-80%)	0.3	5	ns
t_{TREFDS}	Transmit Data or TXRST Setup Time to REFCLK (TXCKSEL = LOW)	1.5		ns
t_{TREFDH}	Transmit Data or TXRST Hold Time from REFCLK (TXCKSEL = LOW)	1		ns
t_{RREFDA}	Receive Data Access Time from REFCLK (RXCKSEL = LOW)		9.5	ns
t_{RREFDH}	Receive Data Hold Time from REFCLK (RXCKSEL = LOW)	4.0		ns
t_{REFADS}	Received Data Setup Time to RXCLKA (RXCKSEL = LOW)	2		ns
t_{REFADH}	Received Data Hold Time from RXCLKA (RXCKSEL = LOW)	1.5		ns
t_{REFCDS}	Received Data Setup Time to RXCLKC (RXCKSEL = LOW)	3		ns
t_{REFCDH}	Received Data Hold Time from RXCLKC (RXCKSEL = LOW)	0.5		ns
t_{REFRX}	REFCLK Frequency Referenced to Received Clock Period[23]	–0.02	+0.02	%

Transmit Serial Outputs and TX PLL Characteristics

Parameter	Description	Condition	Min.	Max.	Unit
t_B	Bit Time		5000	660	ps
t_{RISE}	CML Output Rise Time 20–80% (CML Test Load)[18]	SPDSEL = HIGH	50	250	ps
		SPDSEL = MID	100	500	ps
		SPDSEL = LOW	200	1000	ps
t_{FALL}	CML Output Fall Time 80–20% (CML Test Load)[18]	SPDSEL = HIGH	50	250	ps
		SPDSEL = MID	100	500	ps
		SPDSEL = LOW	200	1000	ps
t_{DJ}	Deterministic Jitter (peak-peak)[18, 24]			0.1	UI
t_{RJ}	Random Jitter (σ)[18, 25]			0.3	UI
t_{TXLOCK}	Transmit PLL Lock to REFCLK	TBD	TBD	ns	

Notes:

23. REFCLK has no phase or frequency relationship with the recovered clock(s) and only acts as a centering reference to reduce clock synchronization time. REFCLK must be within ±200 PPM (±0.02%) of the transmitter PLL reference (REFCLK) frequency, necessitating a ±100-PPM crystal.
24. While sending continuous K28.5s, outputs loaded to a balanced 100Ω load, over the operating range.
25. While sending continuous K28.7s, after 100,000 samples measured at the cross point of differential outputs, time referenced to REFCLK input, over the operating range.

Receive Serial Inputs and CDR PLL Characteristics

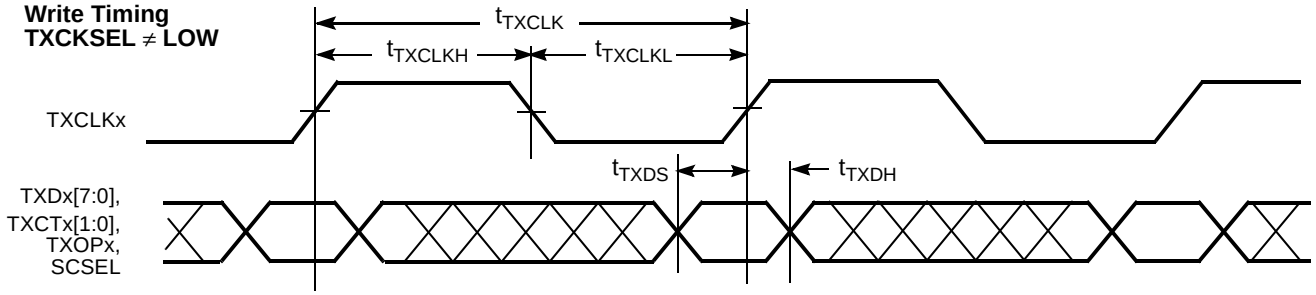
Parameter	Description	Min.	Max.	Unit
t_{RXLOCK}	Receive PLL lock to input data stream (cold start)		10	ms
	Receive PLL lock to input data stream		2500	UI
$t_{RXUNLOCK}$	Receive PLL Unlock Rate	TBD	TBD	ns
t_{SA}	Static Alignment[19, 26]			ps
t_{EFW}	Error Free Window [19, 27, 28]	0.5		UI

Notes:

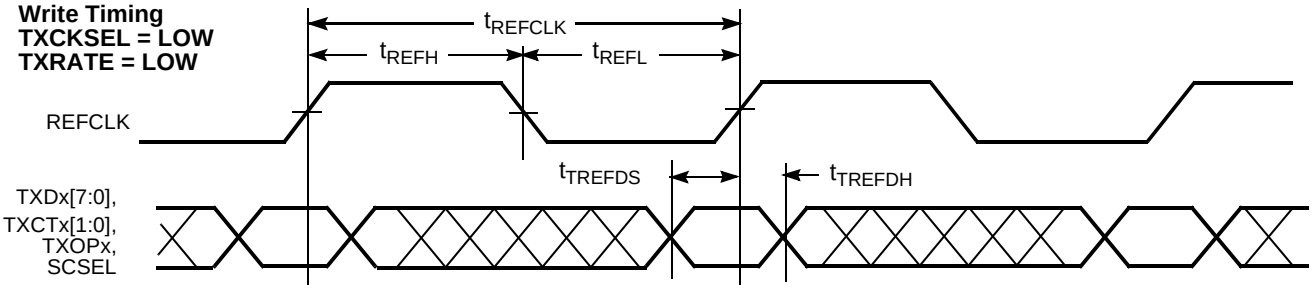
26. Static alignment is a measure of the alignment of the Receiver sampling point to the center of a bit. Static alignment is measured by sliding one bit edge in 3,000 nominal transitions until a character error occurs.
27. Receiver UI (Unit Interval) is calculated as $1/(f_{REF} * 20)$ (when RXRATE = HIGH) or $1/(f_{REF} * 10)$ (when RXRATE = LOW) if no data is being received, or $1/(f_{REF} * 20)$ (when RXRATE = HIGH) or $1/(f_{REF} * 10)$ (when RXRATE = LOW) of the remote transmitter if data is being received. In an operating link this is equivalent to t_B.
28. Error Free Window is a measure of the time window between bit centers where a transition may occur without causing a bit sampling error. EFW is measured over the operating range, input jitter < 50% Dj.

Transmitter Switching Waveforms

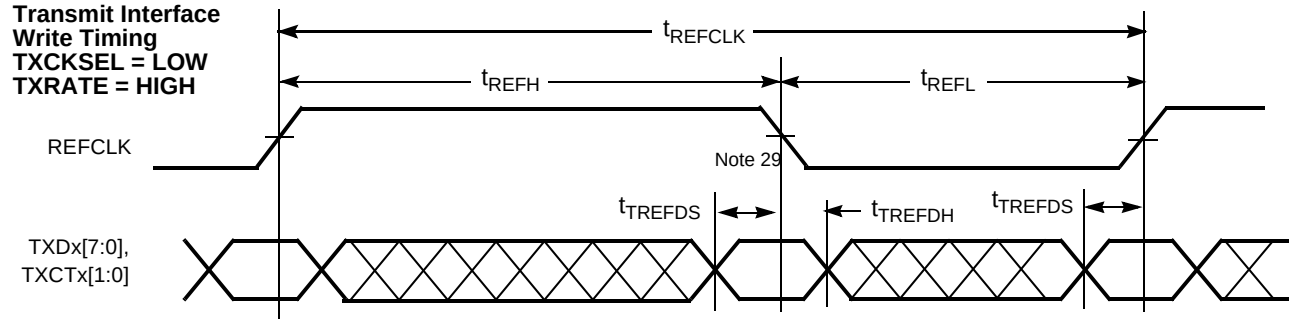
Transmit Interface Write Timing TXCKSEL ≠ LOW



Transmit Interface Write Timing TXCKSEL = LOW TXRATE = LOW

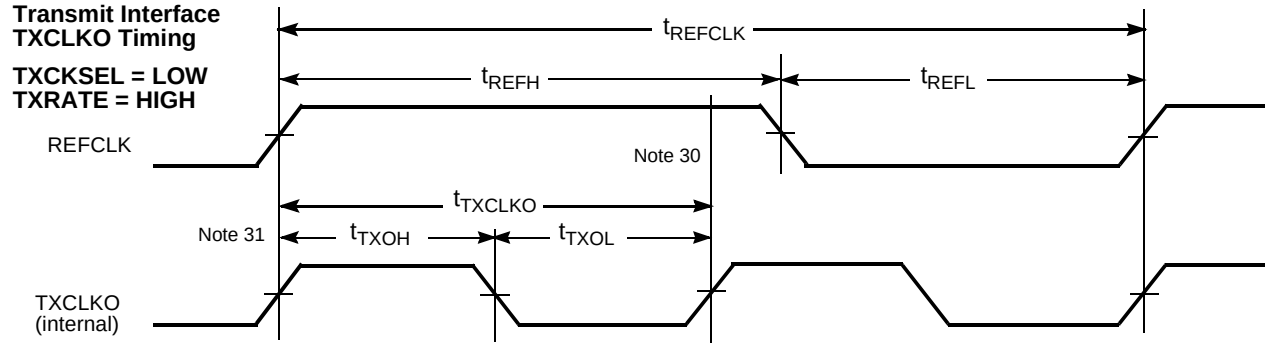


Transmit Interface Write Timing TXCKSEL = LOW TXRATE = HIGH



Transmit Interface TXCLKO Timing

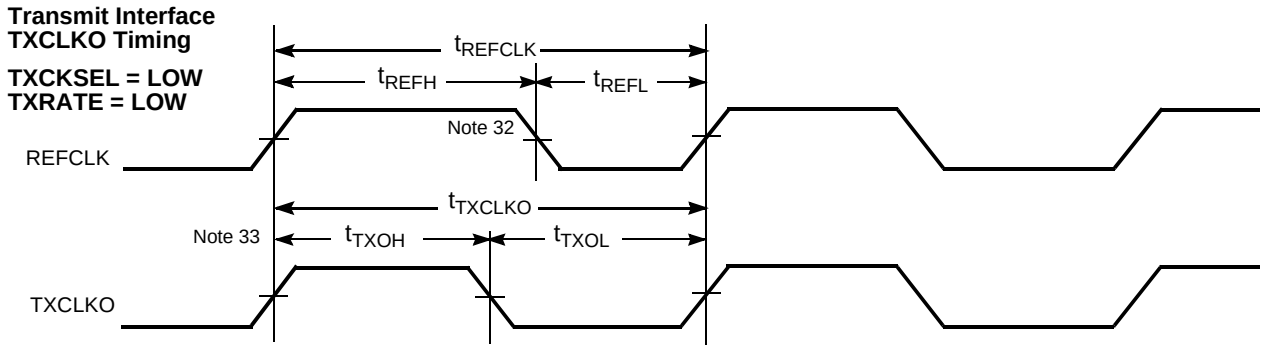
TXCKSEL = LOW TXRATE = HIGH



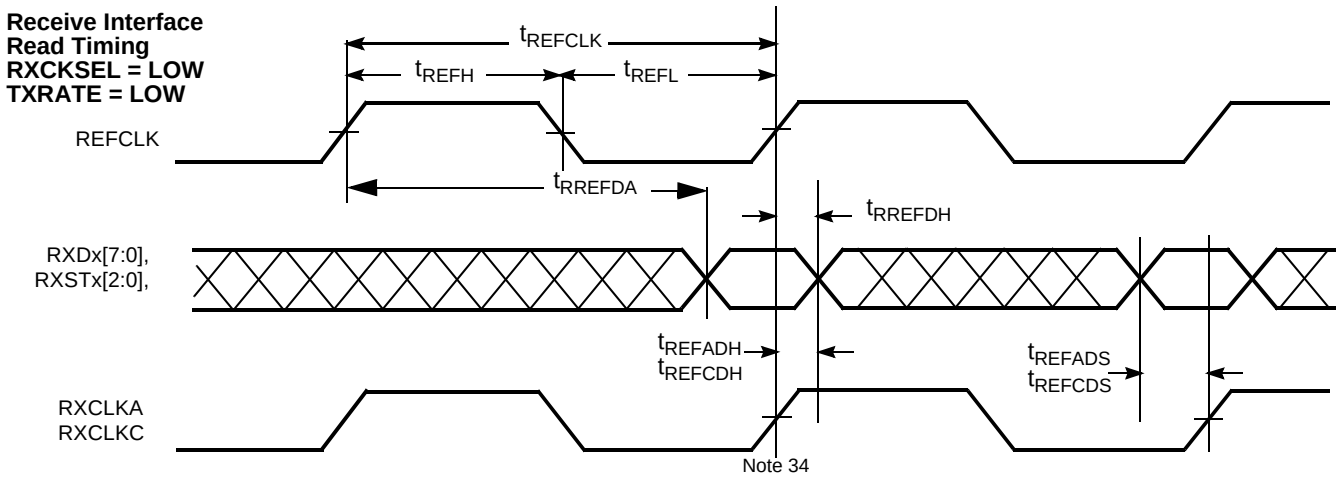
Note:

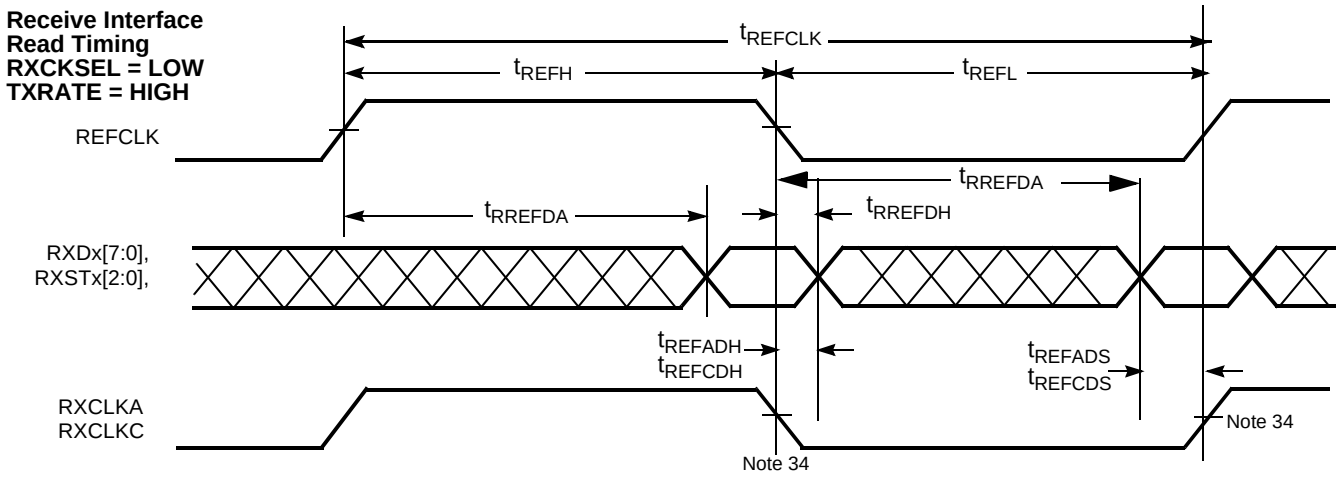
29. When REFCLK is configured for half-rate operation (TXRATE = HIGH) and data is captured using REFCLK instead of a TXCLKx clock (TXCKSEL = LOW), data is captured using both the rising and falling edges of REFCLK.
30. The TXCLKO output remains at the character rate regardless of the state of TXRATE and does not follow the duty cycle of REFCLK.
31. The rising edge of TXCLKO output has no direct phase relationship to the REFCLK input.

Transmitter Switching Waveforms (continued)



Receiver Switching Waveforms

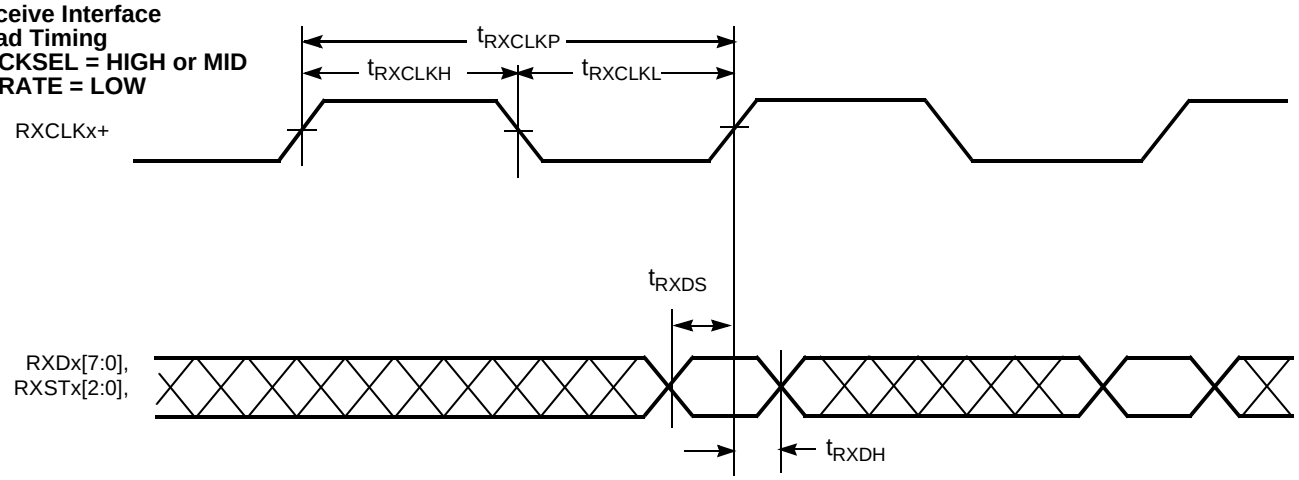




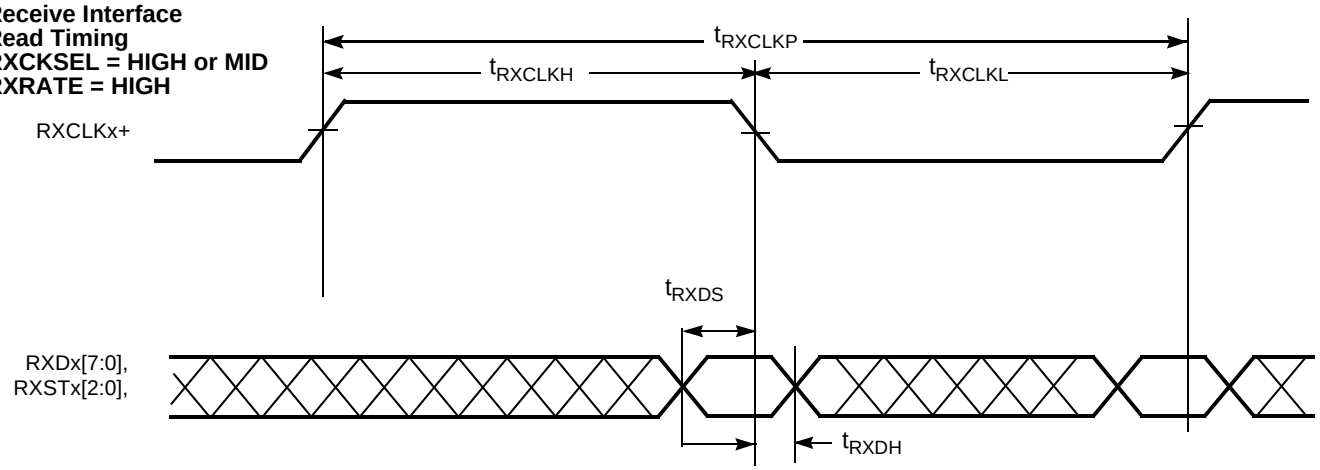
Note:
32. RXCLKA and RXCLKC are delayed in phase from REFCLK, and are different in phase from each other.
33. When operated with a half-rate REFCLK, the setup and hold specifications for data relative to RXCLKA and RXCLKC are relative to both rising and falling edges of the respective clock output

Receiver Switching Waveforms (continued)

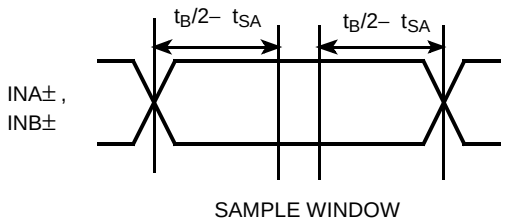
Receive Interface Read Timing RXCKSEL = HIGH or MID RXRATE = LOW



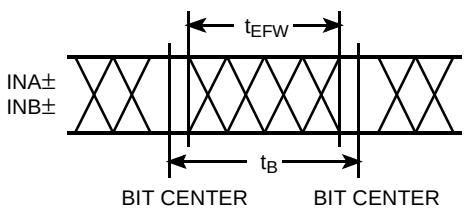
Receive Interface Read Timing RXCKSEL = HIGH or MID RXRATE = HIGH



Static Alignment



Error-Free Window



Input & Output Standard Timing Delay Adjustments

All the timing specifications in this data sheet are specified based on 3.3V PCI compliant inputs and outputs (fast slew rates[34]). Apply following adjustments if the inputs and outputs are configured to operate at other standards.

Input/Output Standard	Output Delay Adjustments			Input Delay Adjustments		
	t_{IOD}	t_{EA}	t_{ER}	t_{IOIN}	t_{CKIN}	$t_{IOREGPIN}$
LVTTL	0.2	0	0	0	0	0
LVCMOS	0.2	0	0	0	0	0
LVCMOS3	0.3	0.05	0	0.1	0.1	0.2
LVCMOS2	0.5	0.1	0	0.2	0.2	0.4
LVCMOS18	2.1	0.7	0.1	0.5	0.4	0.3
3.3V PCI	0	0	0	0	0	0
GTL+	0.6[35]	0.6[35]	0.9[35]	0.5	0.4	0.2
SSTL3 I	–0.3	0.3	0.1	0.5	0.3	0.3
SSTL3 II	–0.4	0.2	0	0.5	0.3	0.3
SSTL2 I	–0.1	0.4	0	0.9	0.5	0.6
SSTL2 II	–0.2	0.2	0	0.9	0.5	0.6
HSTL I	0.6	0.9	0.5	0.5	0.5	0.3
HSTL II	0.4	0.8	0.5	0.5	0.5	0.3
HSTL III	0.6	0.5	0.1	0.5	0.5	0.3
HSTL IV	0.7	0.6	0	0.5	0.5	0.3

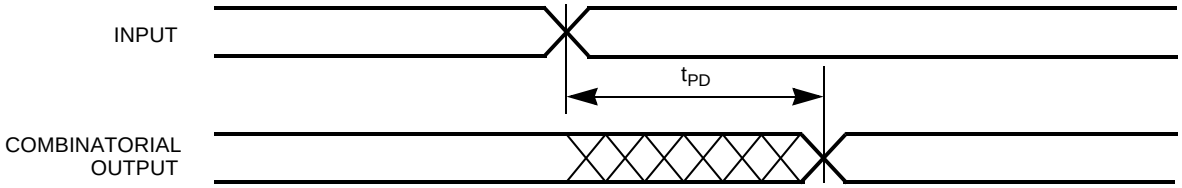
Notes:

34. For “slow slew rate” output delay adjustments, refer to *Warp* software’s static timing analyzer results.
35. These delays are based on falling edge output. The rising edge delay depends on the size of pull up resistor and termination voltage.

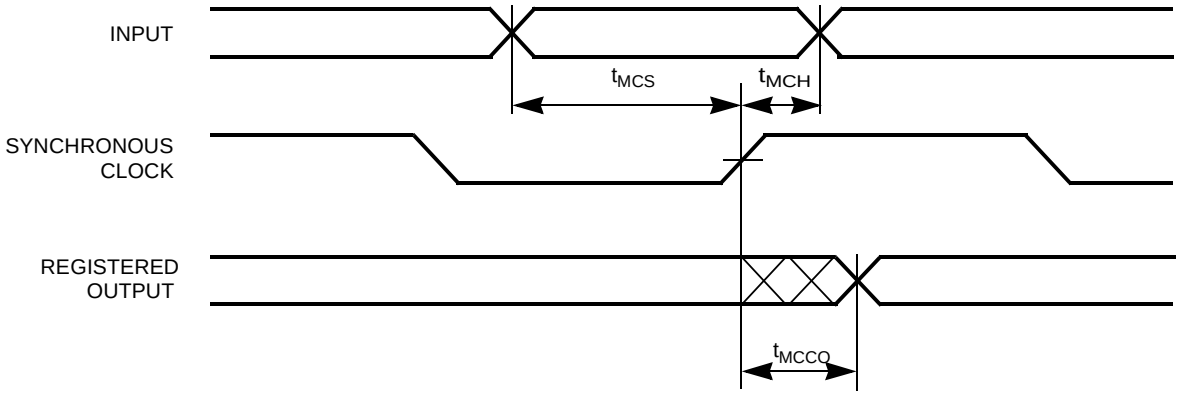
Switching Waveforms

General Switching Waveforms

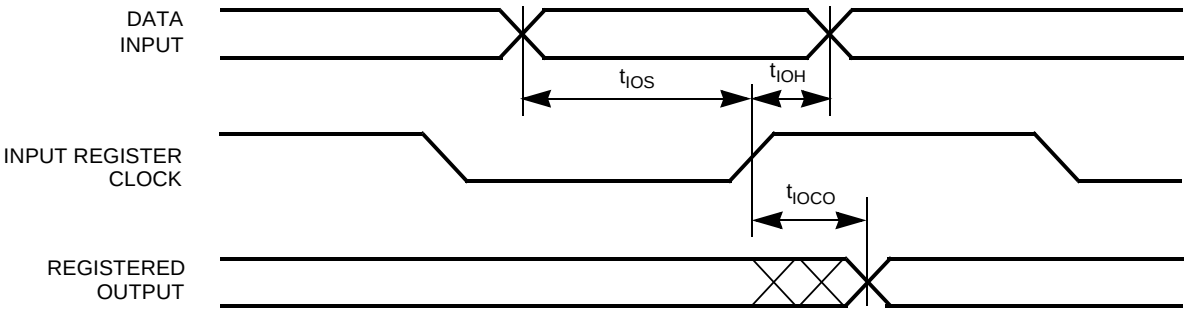
Combinatorial Output



Registered Output with Synchronous Clocking (Macrocell)

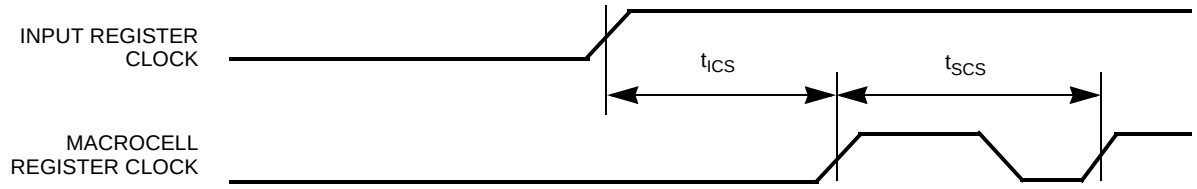


Registered Input in I/O Cell

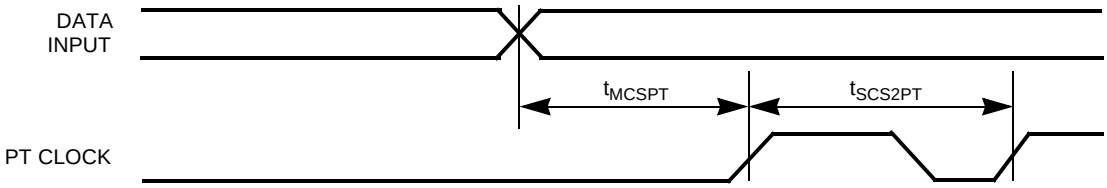


Switching Waveforms (continued)

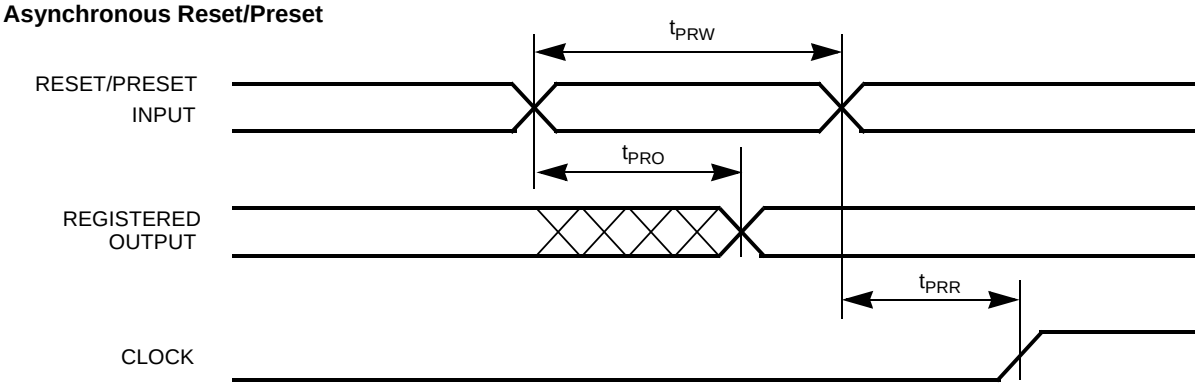
Clock to Clock



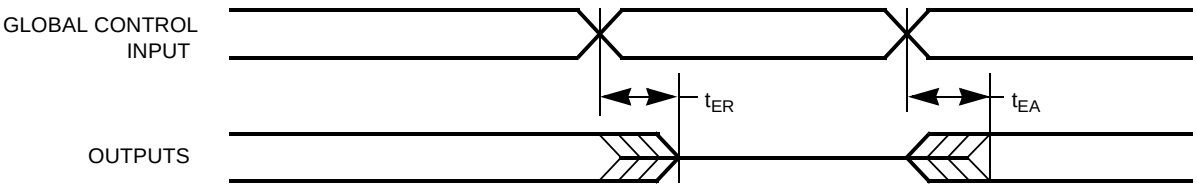
PT Clock to PT Clock



Asynchronous Reset/Preset

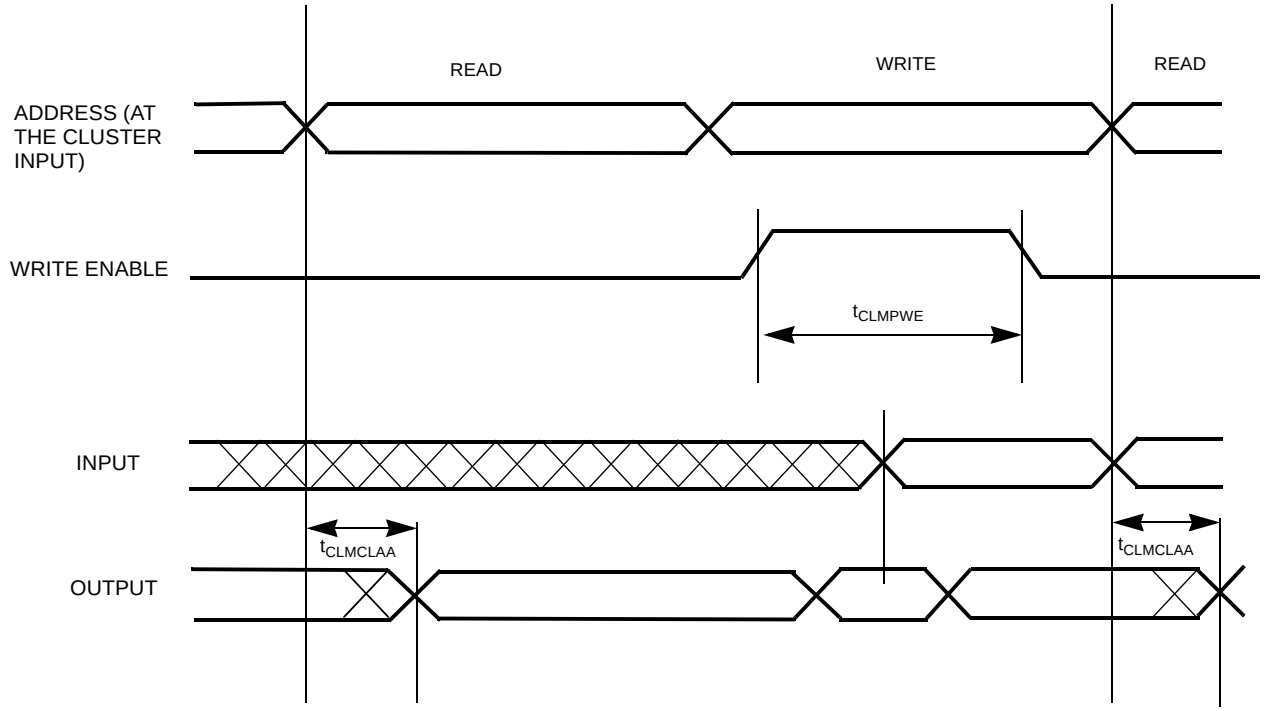


Output Enable/Disable

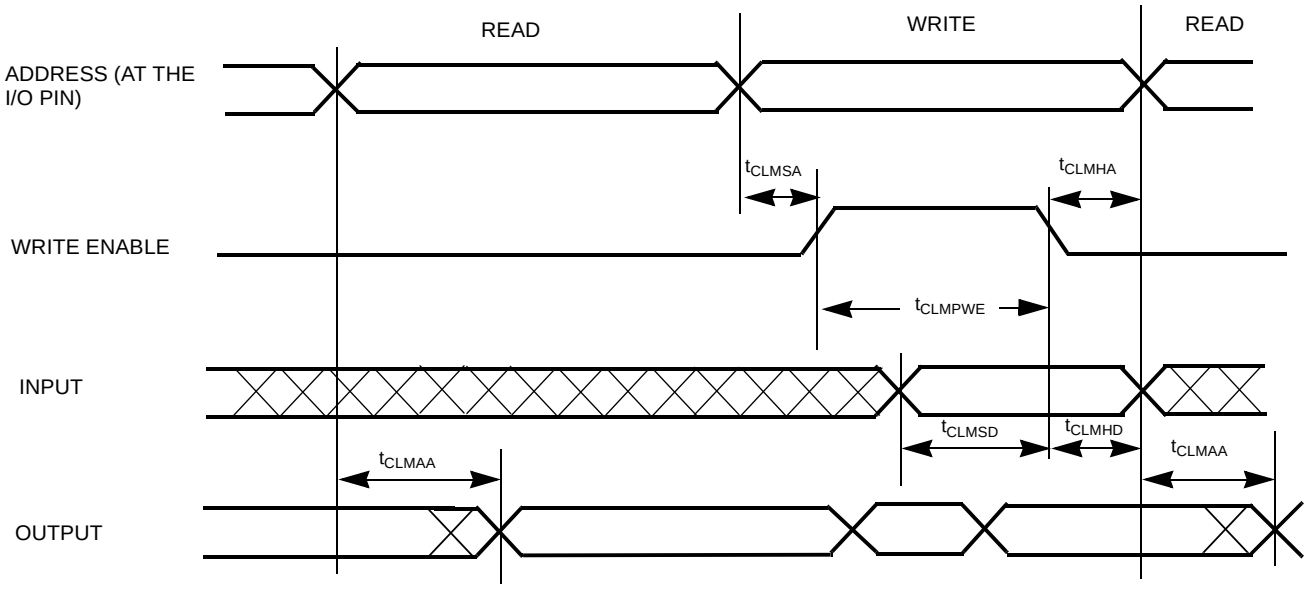


Switching Waveforms (continued)

Cluster Memory Asynchronous Timing

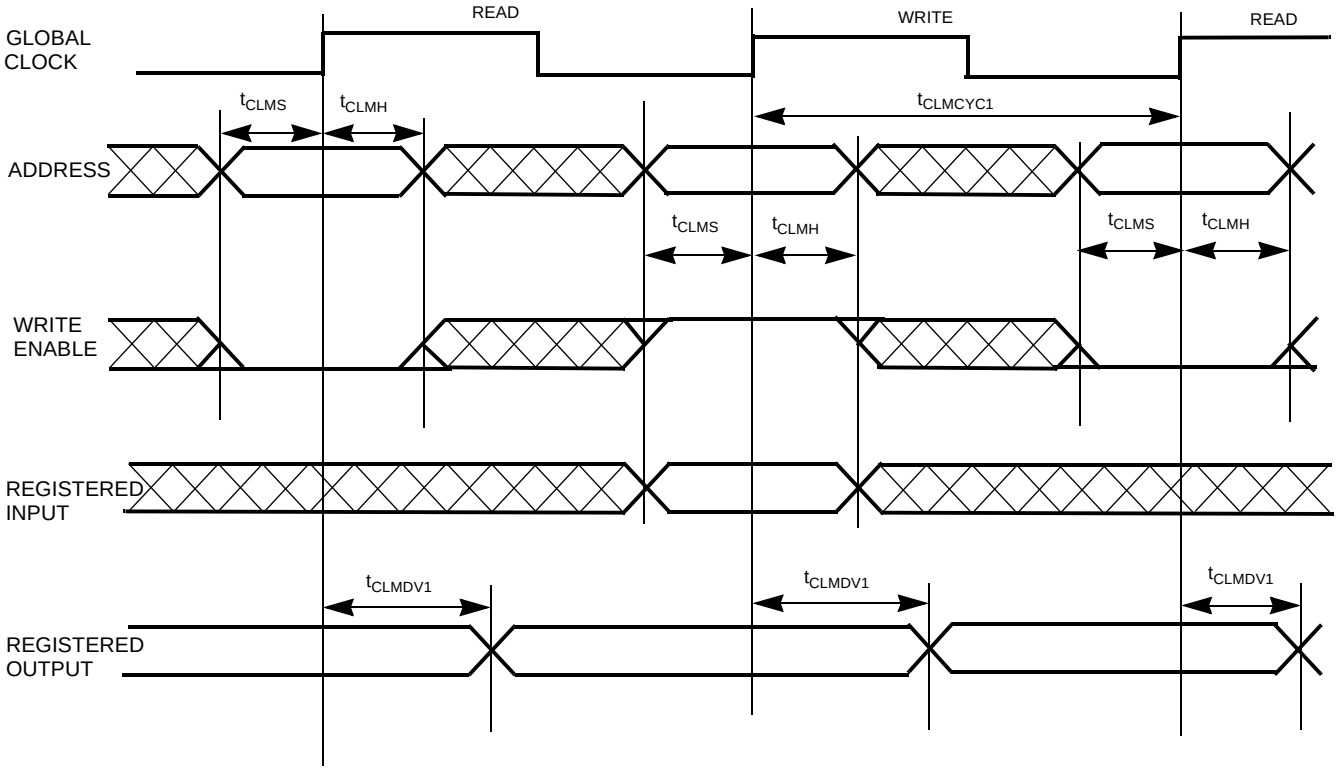


Cluster Memory Asynchronous Timing 2

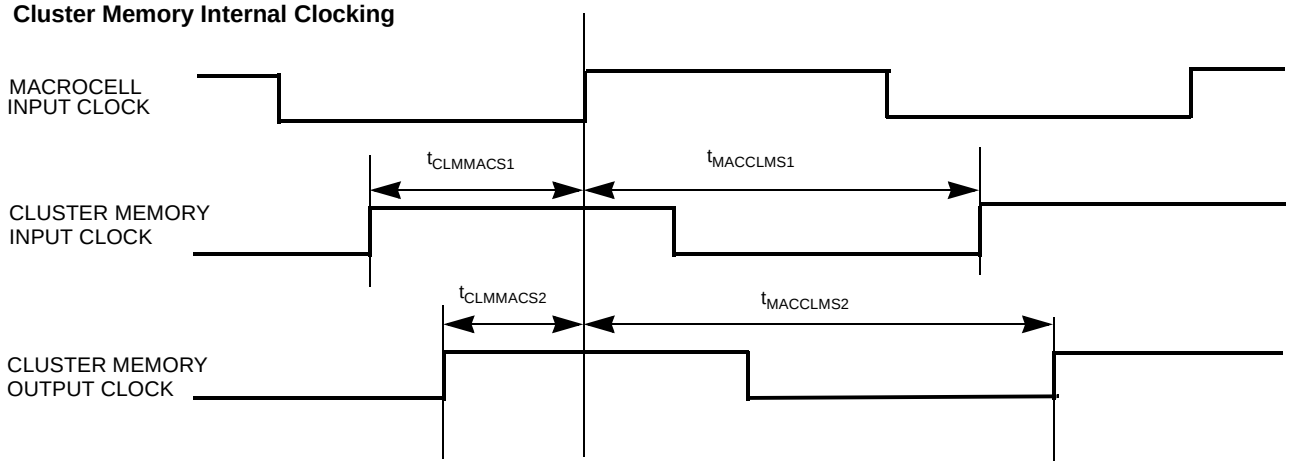


Switching Waveforms (continued)

Cluster Memory Synchronous Timing

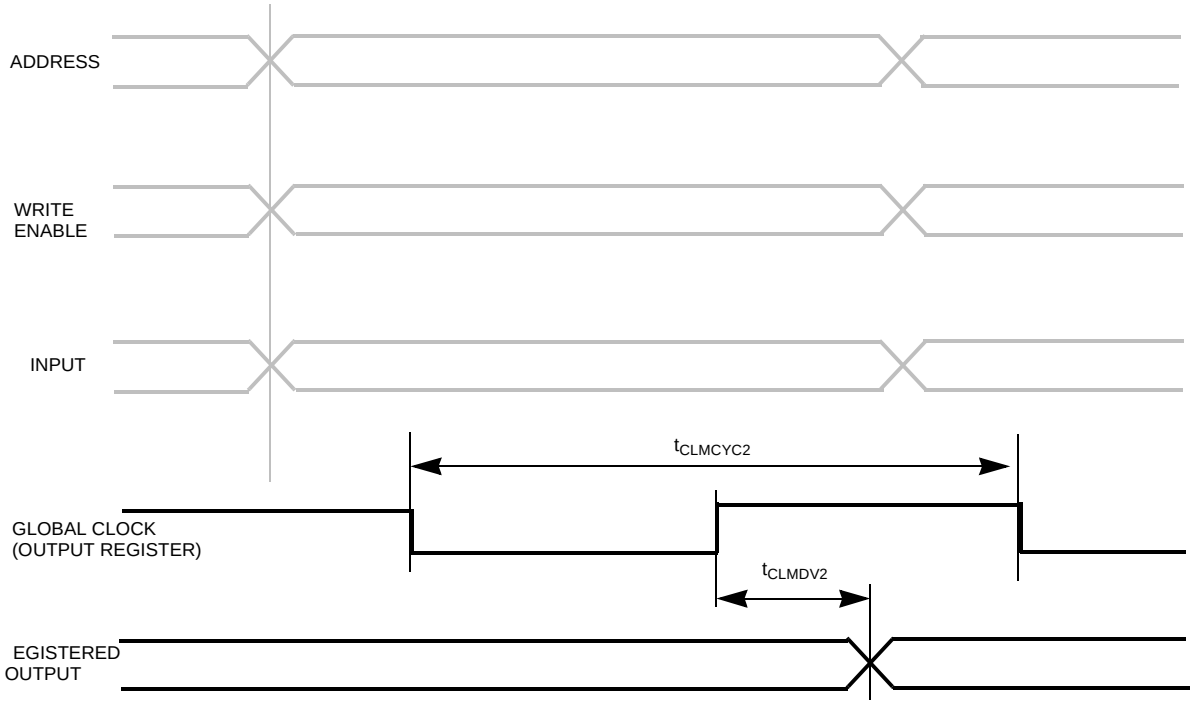


Cluster Memory Internal Clocking

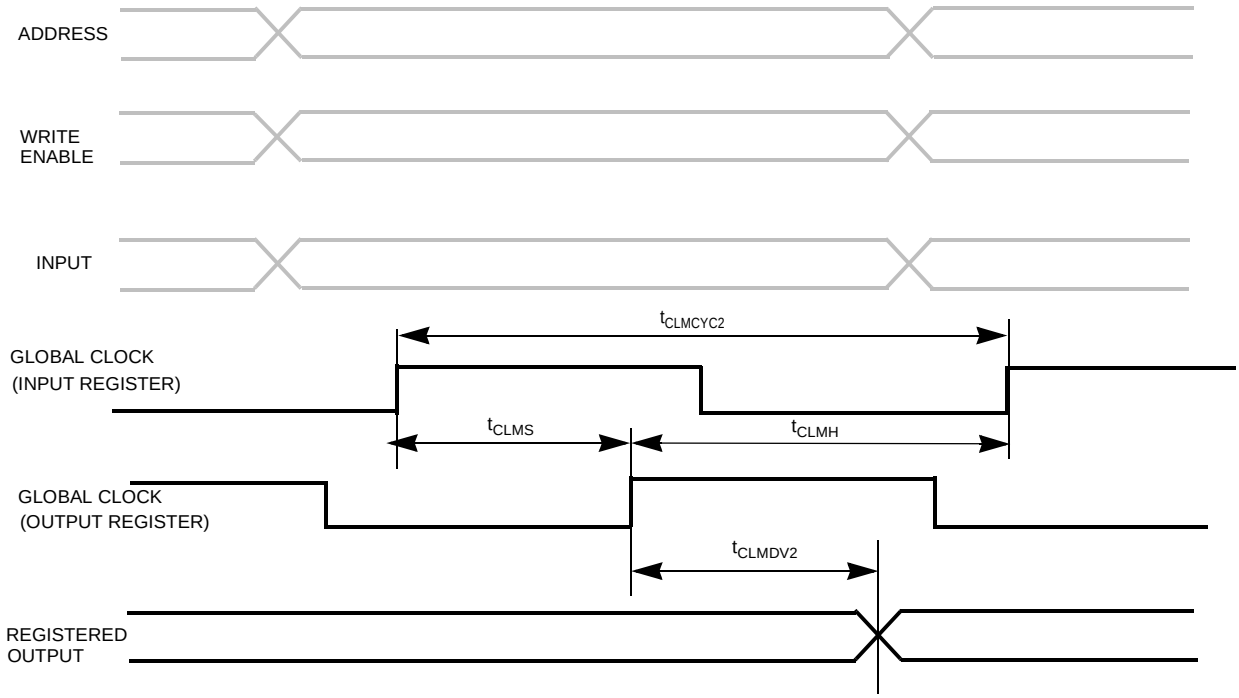


Switching Waveforms (continued)

Cluster Memory Output Register Timing (Asynchronous Inputs)

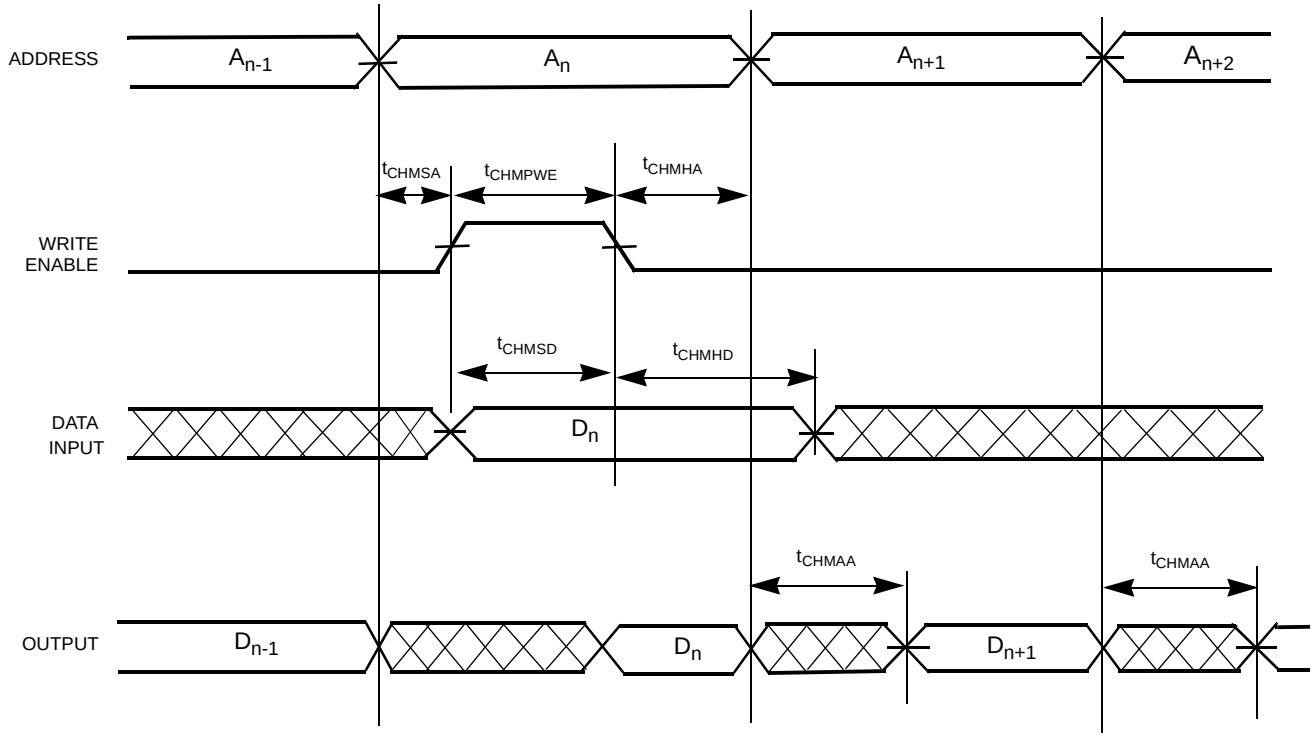


Cluster Memory Output Register Timing (Synchronous Inputs)

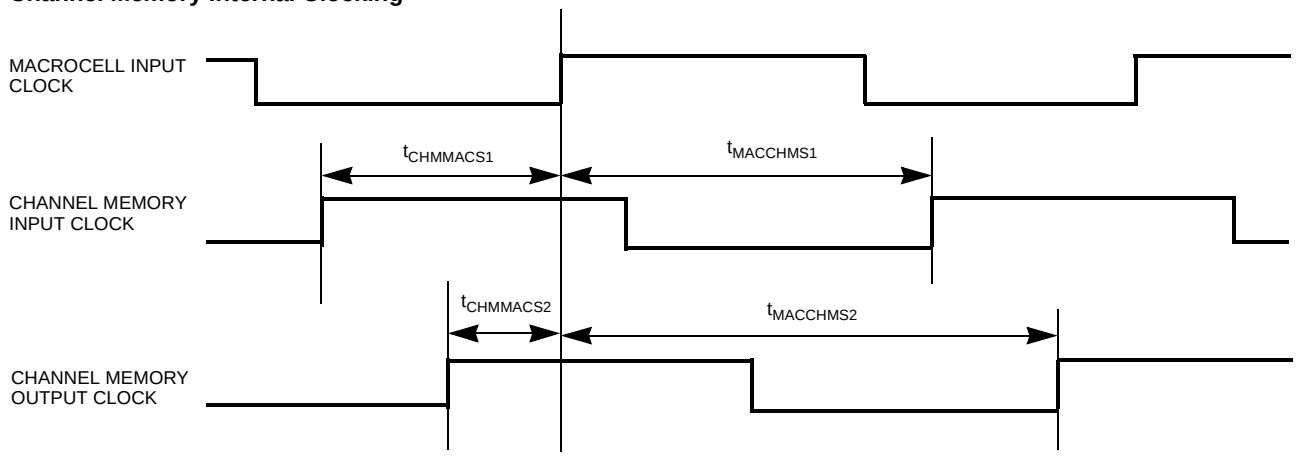


Switching Waveforms (continued)

Channel Memory DP Asynchronous Timing

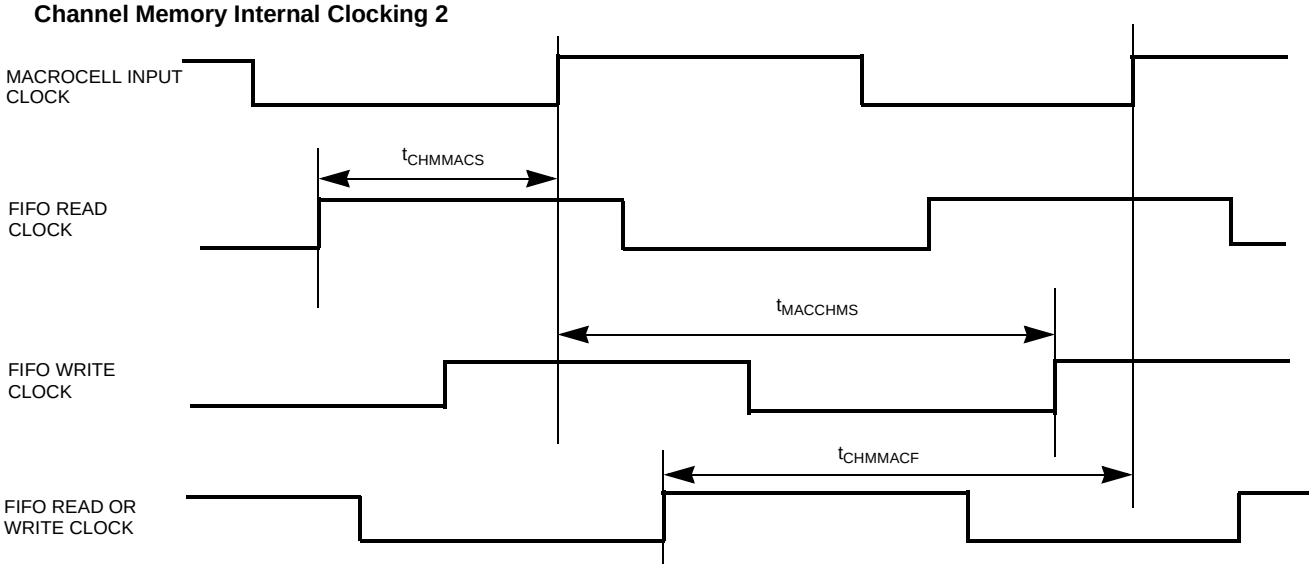


Channel Memory Internal Clocking

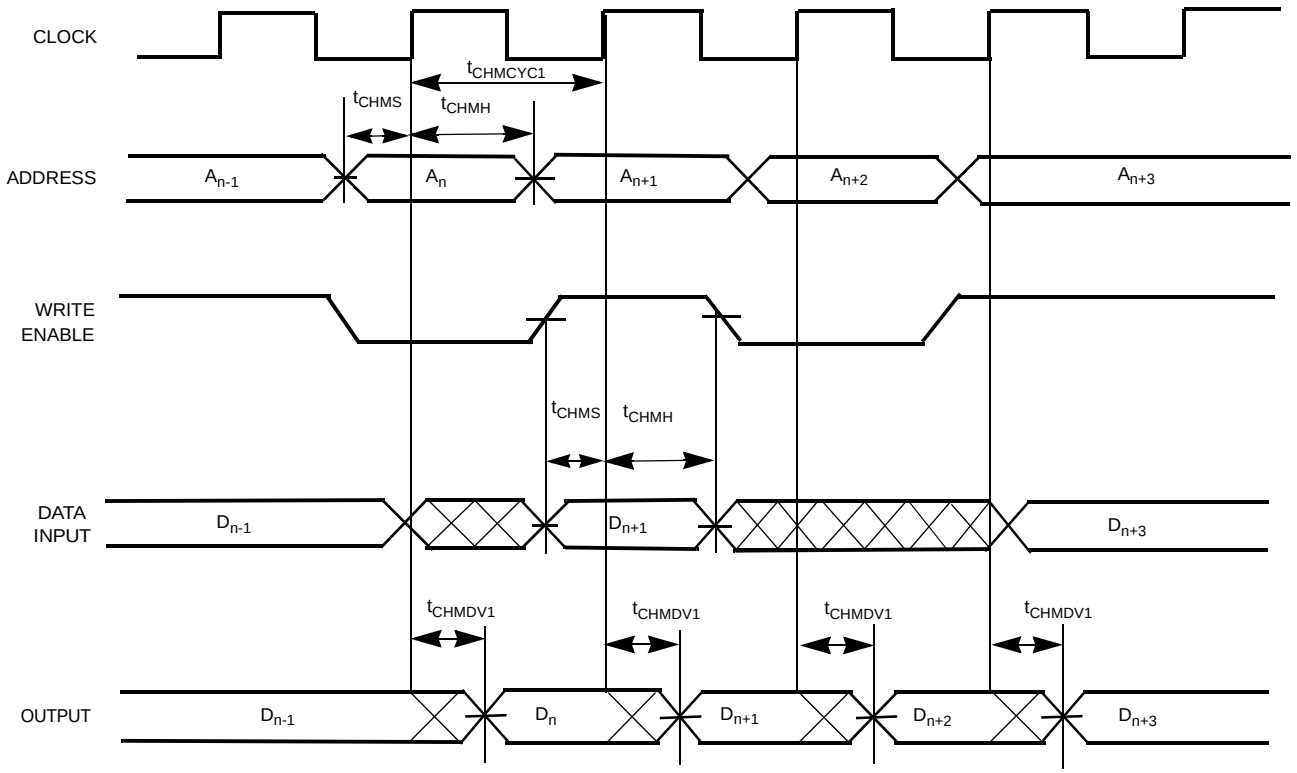


Switching Waveforms (continued)

Channel Memory Internal Clocking 2

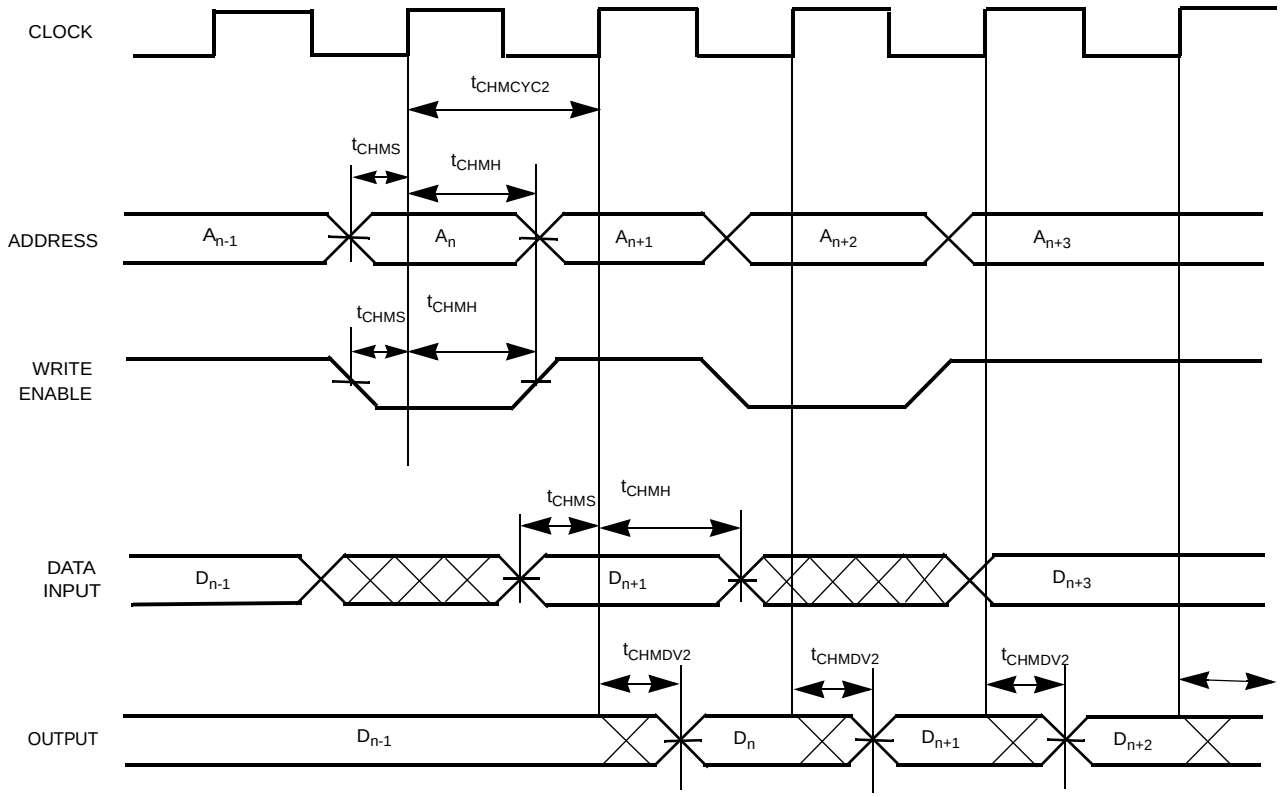


Channel Memory DP SRAM Flow Through R/W Timing

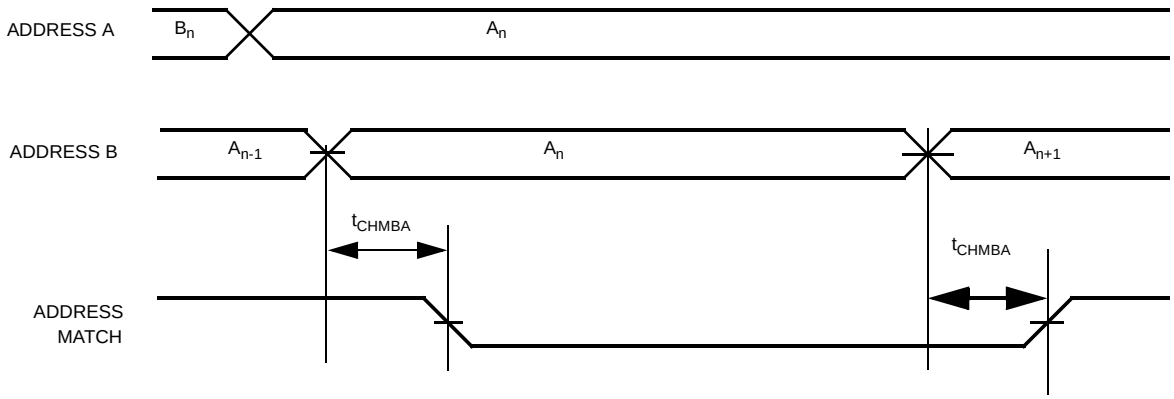


Switching Waveforms (continued)

Channel Memory DP SRAM Pipeline R/W Timing

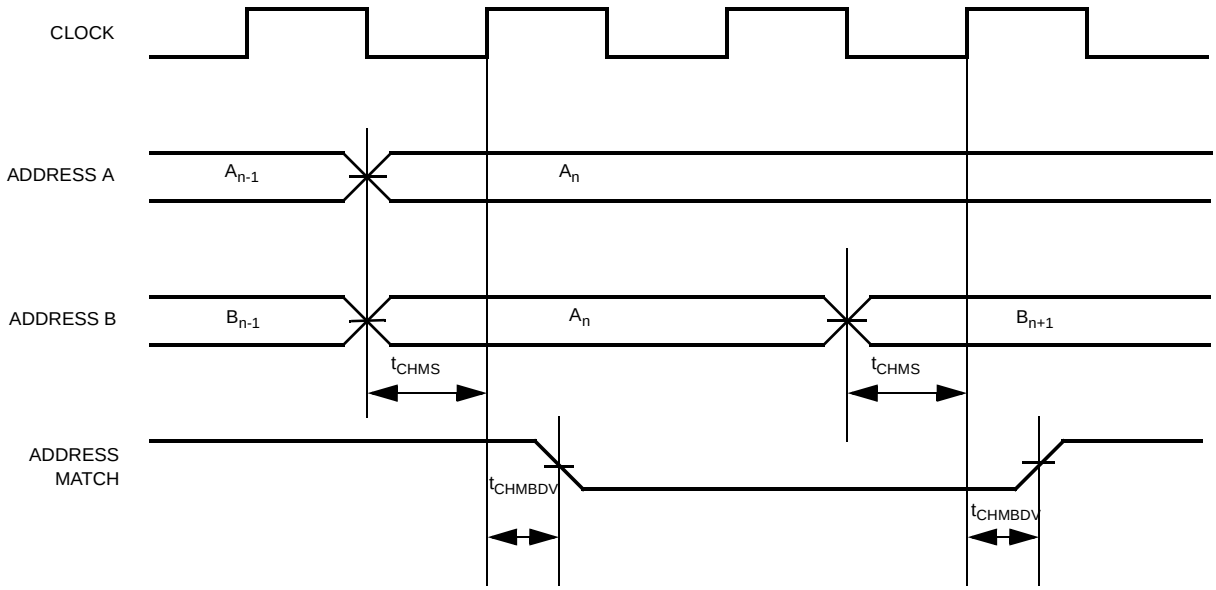


Dual-Port Asynchronous Address Match Busy Signal



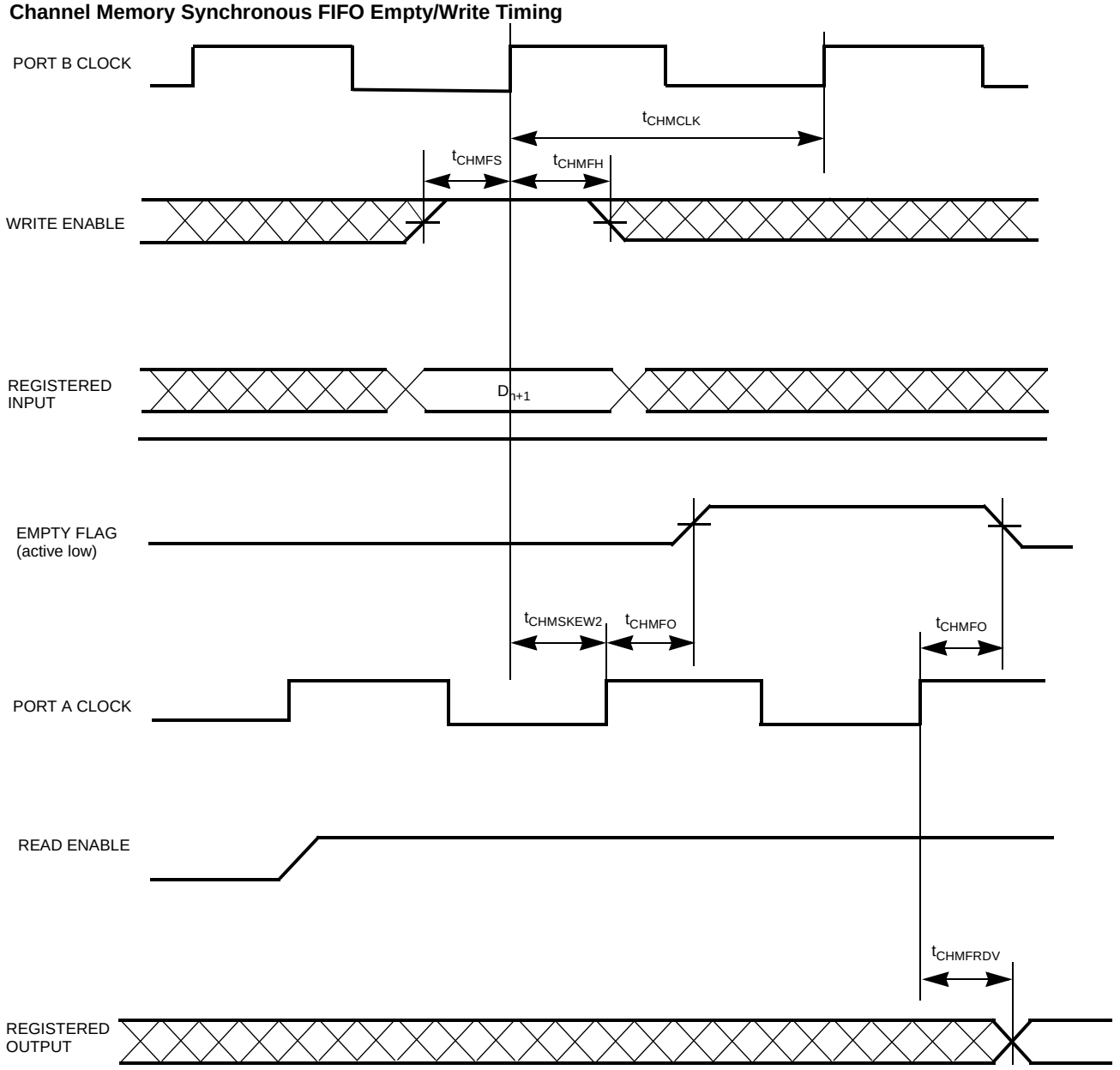
Switching Waveforms (continued)

Dual-Port Synchronous Address Match Busy Signal



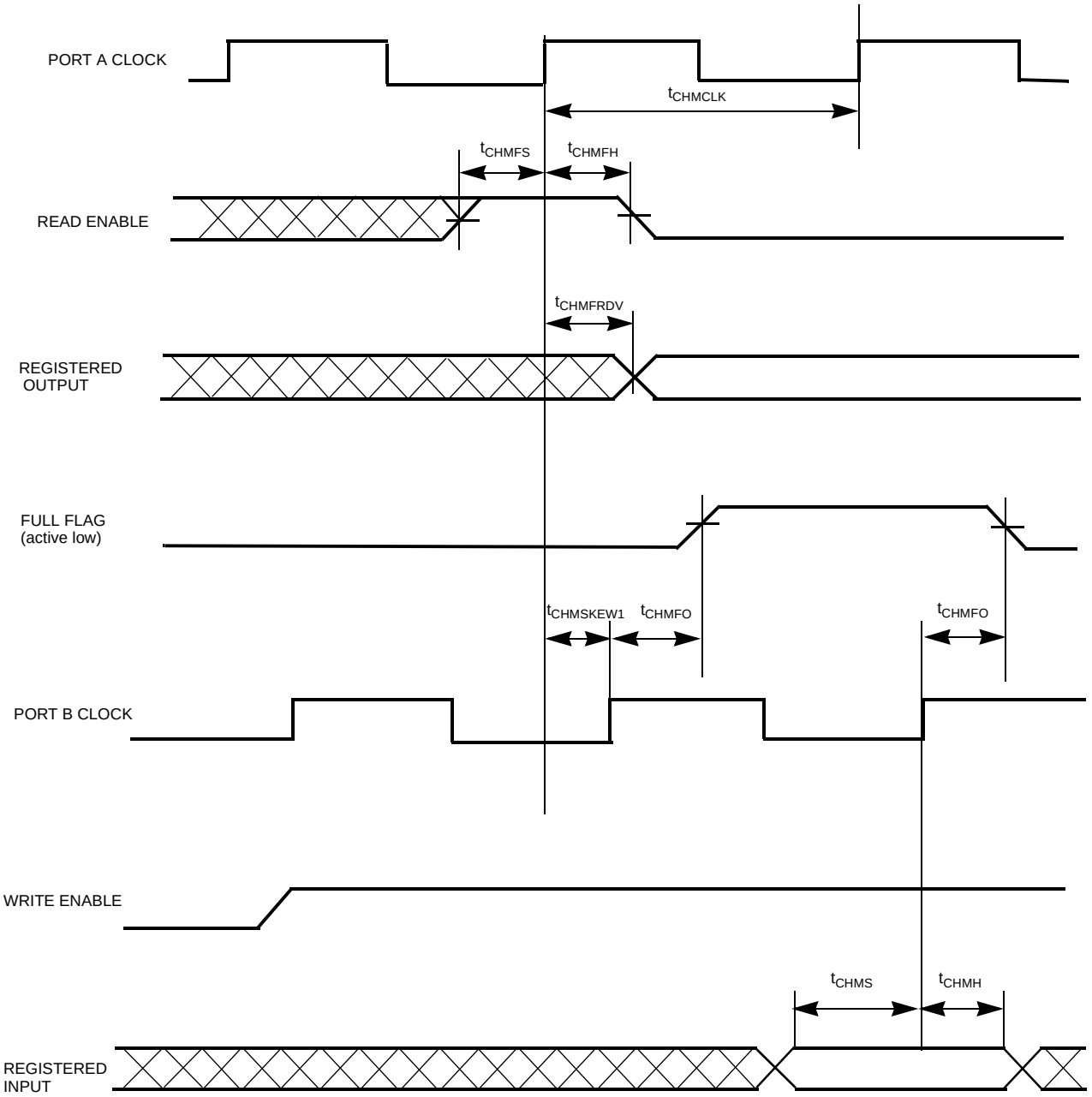
Switching Waveforms (continued)

Channel Memory Synchronous FIFO Empty/Write Timing



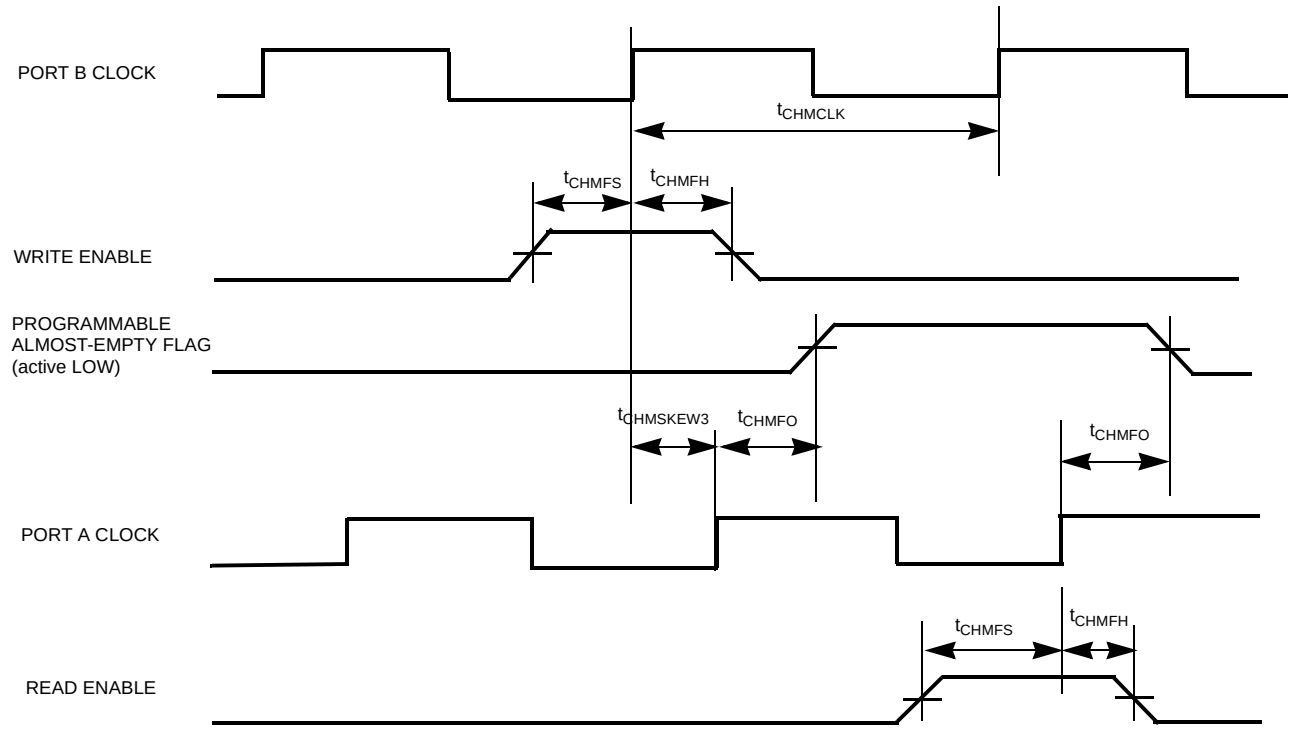
Switching Waveforms (continued)

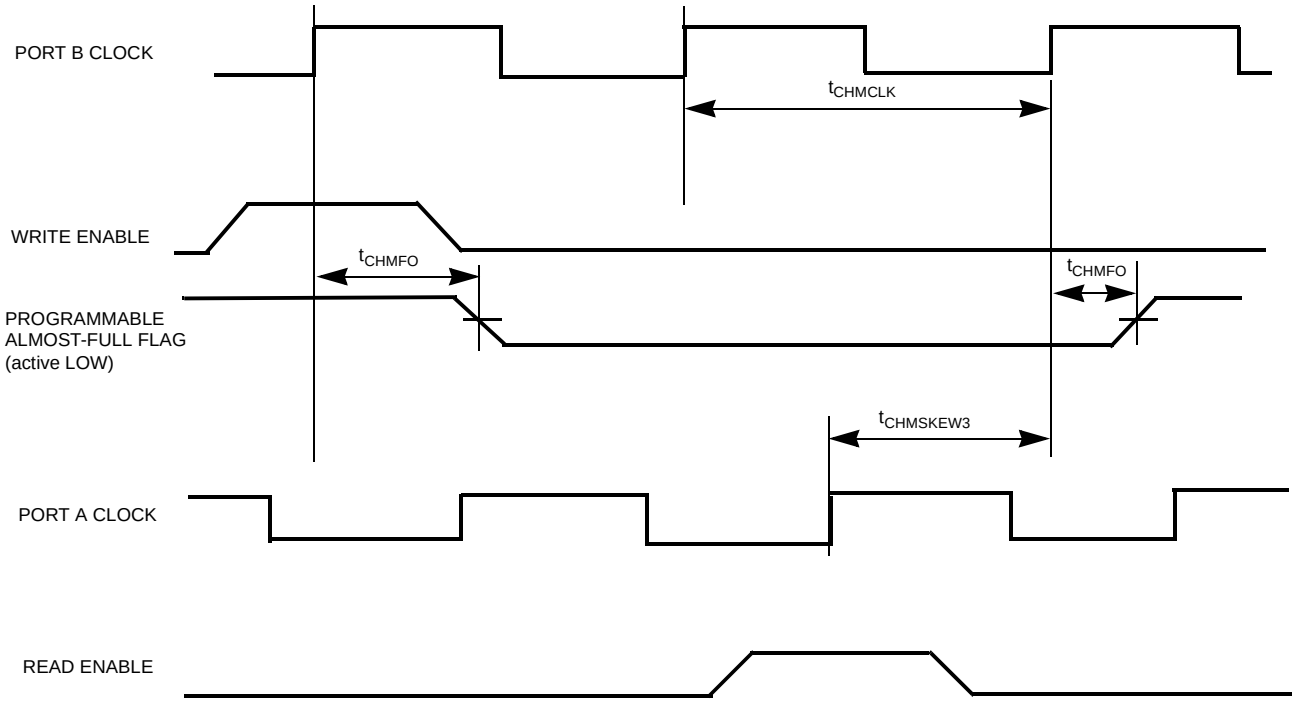
Channel Memory Synchronous FIFO Full/Read Timing



Switching Waveforms (continued)

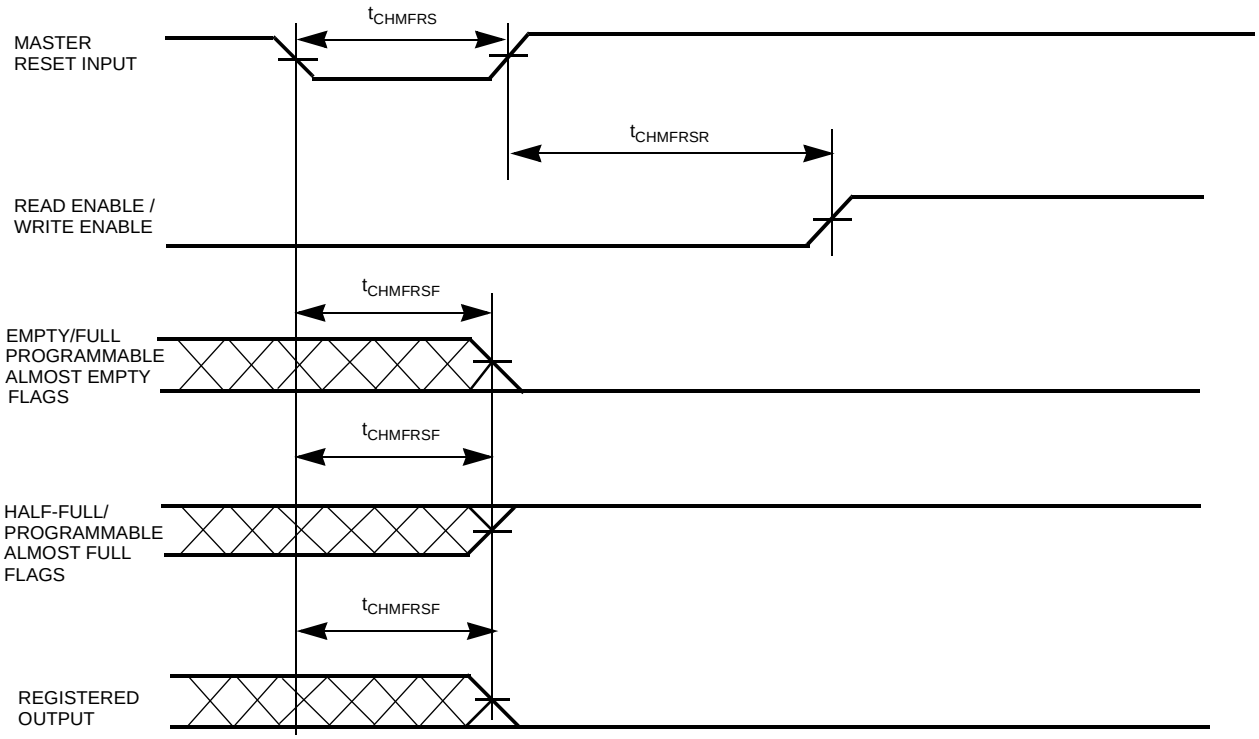
Channel Memory Synchronous FIFO Programmable Flag Timing





Switching Waveforms (continued)

Channel Memory Synchronous FIFO Master Reset Timing



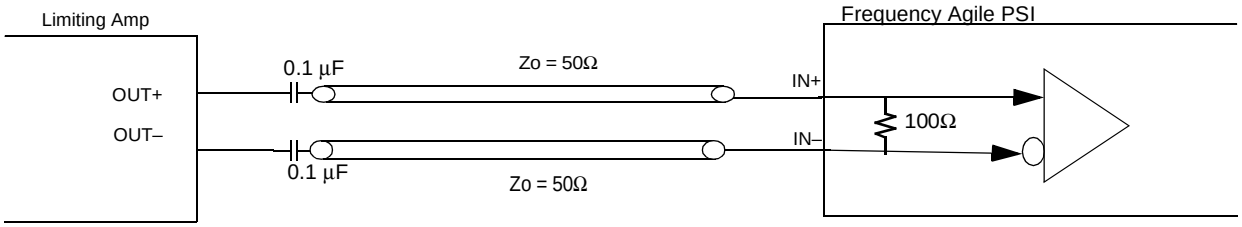


Figure 16. Serial Input Termination

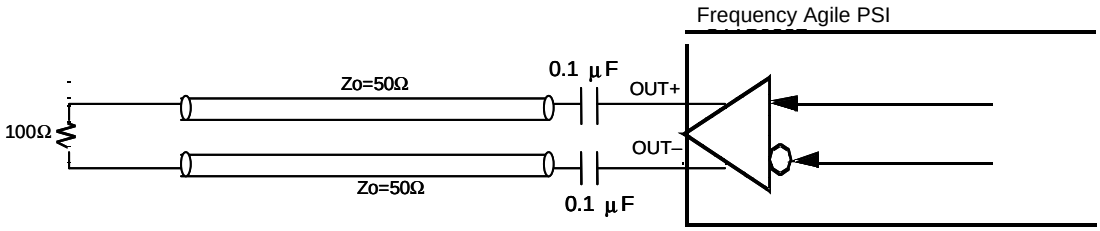


Figure 17. Serial Output Termination

Pin and Signal Description

Frequency Agile PSI

Name	Function	Signal Description
Standard Device Signals		
CCLK	Output	Configuration Clock for serial interface with the external boot PROM
Config_Done	Output	Flag indicating that configuration is complete
Data	Input	Pin to receive configuration data from the external boot PROM
GCLK0-3	Input	Global Input Clock signals 0 through 3
CCE	Output	Chip select for the external boot PROM
GCTL0-3	Input	Global Control signals 0 through 3
IO/V_{REF0}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 0
IO/V_{REF1}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 1
IO/V_{REF2}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 2
IO/V_{REF3}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 3
IO/V_{REF4}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 4
IO/V_{REF5}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 5
IO/V_{REF6}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 6
IO/V_{REF7}	Input/Output	Dual function pin: I/O or Reference Voltage for Bank 7
IO	Input/Output	Input or Output pin
IO6/Lock	Input/Output	Dual function pin: I/O in Bank 6 or PLL lock output signal
MSEL	Input	Mode Select Pin
Reconfig	Input	Pin to start configuration of PSI
Reset	Output	Reset signal to interface with the external boot PROM
TCLK	Input	JTAG Test Clock
TDI	Input	JTAG Test Data In
TDO	Output	JTAG Test Data Out
TMS	Input	JTAG Test Mode Select
Name	**I/O Characteristics**	**Signal Description**
Transmit Path Data Signals		
TXPERA TXPERB TXPERC TXPERD	LVTTL Output, changes relative to REFCLK↑ [36]	Transmit Path Error. Active HIGH. When BIST is enabled for the specific transmit channel, BIST progress is presented on these outputs. Once every 511 character times (plus a 16-character Word Sync Sequence when the receive channels are clocked by a common clock), the associated TXPERx signal will pulse HIGH for one transmit-character clock period to indicate a complete pass through the BIST sequence. These outputs also provide indication of a transmit Phase-Align Buffer underflow or overflow. When the transmit Phase-Align Buffers are enabled (TXCKSEL ≠ LOW, or TXCKSEL = LOW and TXRATE = HIGH), if an underflow or overflow condition is detected, TXPERx for the channel in error is asserted and remains asserted until either an atomic Word Sync Sequence is transmitted or TXRST is sampled LOW to re-center the transmit Phase-Align Buffers.

Frequency Agile PSI

Name	Function	Signal Description
TXCTA[1:0] TXCTB[1:0] TXCTC[1:0] TXCTD[1:0]	LVTTL Input, synchronous, sampled by the selected TXCLKx↑ or REFCLK↑ [36]	Transmit Control. These inputs are captured on the rising edge of the transmit interface clock (selected by TXCKSEL) and are passed to the encoder or transmit shifter. They identify how the associated TXDx[7:0] characters are interpreted. When the encoder is bypassed, these inputs are interpreted as data bits. When the encoder is enabled, these inputs determine if the TXDx[7:0] character is encoded as Data, a Special Character code, or replaced with other Special Character codes. See *Table 8* for details. These signals are internal signals linking the SERDES block to the Logic block.
TXDA[7:0] TXDB[7:0] TXDC[7:0] TXDD[7:0]	LVTTL Input, synchronous, sampled by the selected TXCLKx↑ or REFCLK↑ [36]	Transmit Data Inputs. These inputs are captured on the rising edge of the transmit interface clock (selected by TXCKSEL) and passed to the encoder or transmit shifter. When the encoder is enabled (TXMODE[1:0] ≠ LL), TXDx[7:0] specify the specific data or command character to be sent. These signals are internal signals linking the SERDES block to the Logic block.
$\overline{\text{TXRST}}$	LVTTL Input, asynchronous, internal pull-up, sampled by TXCLKA↑ or REFCLK↑ [36]	Transmit Clock Phase Reset, active LOW. When LOW, the transmit Phase-Align Buffers are allowed to adjust their data-transfer timing (relative to the selected input clock) to allow clean transfer of data from the input register to the encoder or transmit shift register. When $\overline{\text{TXRST}}$ is deasserted (HIGH), the internal phase relationship between the associated TXCLKx and the internal character-rate clock is fixed and the device operates normally. When configured for half-rate REFCLK sampling of the transmit character stream (TNSXCKSEL = LOW and TXRATE = HIGH), assertion of $\overline{\text{TXRST}}$ is only used to clear phase align buffer faults caused by highly asymmetric REFCLK periods or REFCLKs with excessive cycle-to-cycle jitter. During this alignment period, one or more characters may be added to or lost from all the associated transmit paths as the transmit Phase-Align Buffers are adjusted. $\overline{\text{TXRST}}$ must be sampled LOW by a minimum of two consecutive rising edges of TXNSCLKA (or one REFCLK↑) to ensure the reset operation is initiated correctly on all channels. This input is not interpreted when both TXCKSEL and TXRATE are LOW.
Transmit Path Clock and Clock Control		
TXCKSEL	LVTTL Static Control Input	Transmit Clock Select. Selects the transmit clock source, used to write data into the transmit input register, for the transmit channel(s). When LOW, all four input registers are clocked by REFCLK↑ [36]. When HIGH, TXCLKA↑ is used to clock data into the input register of each channel.
TXCLKO	LVTTL Output	Transmit Clock Output. This output clock is synthesized by the transmit PLL and operates synchronous to the internal transmit character clock. It operates at either the same frequency as REFCLK, or at twice the frequency of REFCLK (as selected by TXRATE). TXCLKO is always equal to the transmit VCO bit-clock frequency ÷10. This output clock has no direct phase relationship to REFCLK or any recovered character clock.
TXRATE	LVTTL Input, Static Control input, internal pull-down	Transmit PLL Clock Rate Select. When TXRATE = HIGH, the Transmit PLL multiplies REFCLK by 20 to generate the serial bit-rate clock. When TXRATE = LOW, the transmit PLL multiples REFCLK by 10 to generate the serial bit-rate clock. See *Table 17* for a list of operating serial rates. When REFCLK is selected for clocking of the receive parallel interfaces (RXCKSEL = LOW), the TXRATE input also determines if the clock on the RXCLKA+ and RXCLKC+ outputs is a full or half-rate clock. When TXRATE = HIGH, these output clocks are half-rate clocks and follow the frequency and duty cycle of the REFCLK input. When TXRATE = LOW, these output clocks are full-rate clocks and follow the frequency and duty cycle of the REFCLK input.

Notes:

36. When REFCLK is configured for half-rate operation (TXRATE = HIGH), these inputs are sampled (or the outputs change) relative to both the rising and falling edges of REFCLK.
37. 3-Level select inputs are used for static configuration. They are ternary (not binary) inputs that make use of non-standard logic levels of LOW, MID, and HIGH. The LOW level is usually implemented by direct connection to V_{SS} (ground). The HIGH level is usually implemented by direct connection to V_{CC} (power). When not connected or allowed to float, a 3-Level select input will self-bias to the MID level.

Frequency Agile PSI

Name	Function	Signal Description
TXCLKA	LVTTL Clock Input, internal pull-down	Transmit path input clocks. These clocks must be frequency-coherent to TXCLKO±, but may be offset in phase. The internal operating phase of each input clock (relative to REFLCK) is adjusted when TXRST = LOW and locked when TXRST = HIGH.
Transmit Path Mode Control		
TXMODE[1:0]	3-Level Select[37] Static Control inputs	Transmit Operating Mode. These inputs are interpreted to select one of nine operating modes of the transmit path. See *Table 9* for a list of operating modes.
Receive Path Data Signals		
RXDA[7:0] RXDB[7:0] RXDC[7:0] RXDD[7:0]	LVTTL Output, synchronous to the selected RXCLKx↑ output or REFCLK↑[36] input	Parallel Data Output. These outputs change following the rising edge of the selected receive interface clock. These signals are internal signals linking the SERDES block to the Logic block.
RXSTA[2:0] RXSTB[2:0] RXSTC[2:0] RXSTD[2:0]	LVTTL Output, synchronous to the selected RXCLKx↑ output or REFCLK↑[36] input	Parallel Status Output. These outputs change following the rising edge of the selected receive interface clock. When the decoder is bypassed, RXSTx[1:0] become the two low-order bits of the 10-bit received character, while RXSTx[2] = HIGH indicates the presence of a COMMA character in the output register. These signals are internal signals linking the SERDES block to the Logic block.
RXRATE	LVTTL Input Static Control Input, internal pull-down	Receive Clock Rate Select. When LOW, the RXCLKx+ recovered clock outputs are complementary clocks operating at the recovered character rate. Data for the associated receive channels should be latched on the rising edge of RXCLKx+. When HIGH, the RXCLKx± recovered clock outputs are complementary clocks operating at half the character rate. Data for the associated receive channels should be latched alternately on the rising edge of RXCLKx+. When operated with REFCLK clocking of the received parallel data outputs (RXCKSEL = LOW), the RXRATE input is not interpreted.
Receive Path Clock and Clock Control		
RXCLKA RXCLKB RXCLKC RXCLKD	3-state, LVTTL Output clock or Static control input	Receive Character clock output or clock select input. When the receive Elasticity Buffers are disabled (RXCKSEL = MID), these true and complement clocks are the Receive interface clocks which are used to control timing of data output transfers. These clocks are output continuously at either the dual-character rate (1/20th the serial bit-rate) or character rate (1/10th the serial bit-rate) of the data being received, as selected by RXRATE. When configured such that all output data paths are clocked by REFCLK instead of a recovered clock (RXCKSEL = LOW), the RXCLKA+ and RXCLKC+ output drivers present a buffered form of REFCLK, and RXCLKB+ and RXCLKD+ are static control inputs used to select the master channel for bonding and status control. RXCLKA+ and RXCLKC+ are buffered forms of REFCLK that are slightly different in phase. This phase difference allows the user to select the optimal setup/hold timing for their specific interface. When dual-channel bonding is enabled and a recovered clock is used to present data (RXCKSEL = HIGH), RXCLKA+ drives the recovered clock from either receive channel A or receive channel B as selected by RXCLKB+, and RXCLKC+ drives the recovered clock from either receive channel C or receive channel D as selected by RXCLKD+. When quad-channel bonding is enabled and a recovered clock is used to present data (RXCKSEL = HIGH), RXCLKA+ and RXCLKC+ output the recovered clock from receive channel A, B, C, or D, as selected by RXCLKB+ and RXCLKD+.
RFEN	LVTTL input, asynchronous, internal pull-down	Reframe Enable for all channels. Active HIGH. When HIGH the framers in all four channels are enabled to frame per the presently enabled framing mode.

Frequency Agile PSI

Name	Function	Signal Description
RXMODE[1:0]	3-Level Select[37] Static Control Inputs	Receive Operating Mode. These inputs are interpreted to select one of nine operating modes of the receive path. See *Table 21* for details.
RXCKSEL	3-Level Select [37] Static Control Input	Receive Clock Mode. Selects the receive clock-source used to transfer data to the output registers. When LOW, all four output registers are clocked by REFCLK. RXCLKB+ and RXCLKD+ outputs are disabled (High-Z), and RXCLKA± and RXCLKC± present buffered and delayed forms of REFCLK. This clocking mode is required for channel bonding across multiple devices. When MID, each RXCLKx+ output follows the recovered clock for the respective channel, as selected by RXRATE. When HIGH, and channel bonding is enabled in dual-channel mode (RX modes 3 and 5), RXCLKA± outputs the recovered clock from either receive channel A or receive channel B as selected by RXCLKB+, and RXCLKC+ outputs the recovered clock from either receive channel C or receive channel D as selected by RXCLKD+. These output clocks may operate at the character-rate or half the character-rate as selected by RXRATE. When HIGH and channel bonding is enabled in quad channel mode (RX modes 6 and 8), or if the receive channels are operated in independent mode (RX modes 0 and 2), RXCLKA+ and RXCLKC+ output the recovered clock from receive channel A, B, C, or D, as selected by RXCLKB+ and RXCLKD+. This output clock may operate at the character-rate or half the character-rate as selected by RXRATE.
FRAMCHAR	3-Level Select[37] Static Control Input	Framing Character Select. Used to control the character or portion of a character used for character framing of the received data streams. When LOW, the framer looks for an 8-bit positive COMMA character in the data stream. When MID, the framer looks for both positive and negative disparity versions of the 8-bit COMMA character. When HIGH, the framer looks for both positive and negative disparity versions of the K28.5 character.
RFMODE	3-Level Select[37] Static Control Input	Reframe Mode Select. Used to control the type of character framing used to adjust the character boundaries (based on detection of one or more framing characters in the data stream). This signal operates in conjunction with the presently enabled channel bonding mode, and the type of framing character selected. When LOW, the low-latency framer is selected. This will frame on each occurrence of the selected framing character(s) in the received data stream. This mode of framing stretches the recovered clock for one or multiple cycles to align that clock with the recovered data. When MID, the Cypress-mode multi-byte parallel framer is selected. This requires a pair of the selected framing character(s), on identical 10-bit boundaries, within a span of 50 bits, before the character boundaries are adjusted. The recovered character clock remains in the same phasing regardless of character offset. When HIGH, the alternate mode multi-byte parallel framer is selected. This requires detection of the selected framing character(s) of the allowed disparities in the received data stream, on identical 10-bit boundaries, on four directly adjacent characters. The recovered character clock remains in the same phasing regardless of character offset.
DECMODE	3-Level Select[37] Static Control Input	Decoder Mode Select. This input selects the behavior of the decoder block. When LOW, the decoder is bypassed and raw 10-bit characters are passed to the output register. When MID, the Cypress decoder table for Special Code characters is used. When HIGH, the alternate decoder table for Special Code characters is used. See *Table 30* for a list of the Special Codes supported in both encoded modes.

Frequency Agile PSI

Name	Function	Signal Description
Device Control Signals		
SPDSEL	3-Level Select[37], static configuration input	Serial Rate Select. This input specifies the operating bit-rate range of both transmit and receive PLLs. LOW = 200–400 MBd, MID = 400–800 MBd, HIGH = 800–1500 MBd.
REFCLK±	Differential LVPECL or single-ended LVTTL input clock	Reference Clock. This clock input is used as the timing reference for the transmit and receive PLLs. This input clock may also be selected to clock the transmit and receive parallel interfaces. For an LVCMOS or LVTTL input clock, connect REFCLK+ to the reference clock and leave REFCLK– open. For an LVPECL differential clock, both inputs must be connected. When TXCKSEL = LOW, REFCLK is used as the clock for the parallel transmit data (input) interface. When RXCKSEL = LOW, REFCLK is used as the clock for the parallel receive data (output) interface.
Analog I/O and Control		
OUTA1± OUTB1± OUTC1± OUTD1±	CML Differential Output	Primary Differential Serial Data Outputs. These PECL-compatible CML outputs (+3.3V referenced) are capable of driving terminated transmission lines or standard fiber-optic transmitter modules. These outputs must be AC-coupled for PECL-compatible connections.
OUTA2± OUTB2± OUTC2± OUTD2±	CML Differential Output	Secondary Differential Serial Data Outputs. These PECL-compatible CML outputs (+3.3V referenced) are capable of driving terminated transmission lines or standard fiber-optic transmitter modules. These outputs must be AC-coupled for PECL-compatible connections.
INA1± INB1± INC1± IND1±	LVPECL Differential Input	Primary Differential Serial Data Inputs. These inputs accept the serial data stream for deserialization and decoding. The INx1± serial streams are passed to the receiver Clock and Data Recovery (CDR) circuits to extract the data content when INSELx = HIGH.
INA2± INB2± INC2± IND2±	LVPECL Differential Input	Secondary Differential Serial Data Inputs. These inputs accept the serial data stream for deserialization and decoding. The INx2± serial streams are passed to the receiver Clock and Data Recovery (CDR) circuits to extract the data content when INSELx = LOW.
INSELA INSELB INSELC INSELD	LVTTL Input, asynchronous	Receive Input Selector. Determines which external serial bit stream is passed to the receiver Clock and Data Recovery circuit. When HIGH, the INx1± input is selected. When LOW, the INx2± input is selected.
SDASEL	3-Level Select[37], static configuration input	Signal Detect Amplitude Level Select. Allows selection of one of three predefined amplitude trip points for a valid signal indication, as listed in *Table 18*.
LPEN	LVTTL Input, asynchronous, internal pull-down	All-Port Loop-Back-Enable. Active HIGH. When asserted (HIGH), the transmit serial data from each channel is internally routed to the associated receiver Clock and Data Recovery (CDR) circuit. All serial drivers are forced to differential logic “1”. All serial data inputs are ignored.
OELE	LVTTL Input, asynchronous, internal pull-up	Serial Driver Output Enable Latch Enable. Active HIGH. When OELE = HIGH, the signals on the BOE[7:0] inputs directly control the OUTxy± differential drivers. When the BOE[x] input is HIGH, the associated OUTxy± differential driver is enabled. When the BOE[x] input is LOW, the associated OUTxy± differential driver is powered down. When OELE returns LOW, the last values present on BOE[7:0] are captured in the internal Output Enable latch. The specific mapping of BOE[7:0] signals to transmit output enables is listed in *Table 16*. When the latch is closed, if the device is reset ($\overline{\text{TRSTZ}}$ is sampled LOW), the latch is reset to enable all outputs.

Frequency Agile PSI

Name	Function	Signal Description
BISTLE	LVTTL Input, asynchronous, internal pull-up	Transmit and Receive BIST Latch Enable. Active HIGH. When BISTLE = HIGH, the signals on the BOE[7:0] inputs directly control the transmit and receive BIST enables. When BOE[x] input is LOW, the associated transmit or receive channel is configured to generate or compare the BIST sequence. When the BOE[x] input is HIGH, the associated transmit or receive channel is configured for normal data transmission or reception. When BISTLE returns LOW, the last values present on BOE[7:0] are captured in the internal BIST Enable latch. The specific mapping of BOE[7:0] signals to transmit and receive BIST enables is listed in *Table 16*. When the latch is closed, if the device is reset ($\overline{\text{TRSTZ}}$ is sampled LOW), the latch is reset to disable BIST on all transmit and receive channels.
RXLE	LVTTL Input, asynchronous, internal pull-up	Receive Channel Power-Control Latch Enable. Active HIGH. When RXLE = HIGH, the signals on the BOE[7:0] inputs directly control the power enables for the receive PLLs and analog logic. When the BOE[7:0] input is HIGH, the associated receive channel A through receive channel D PLL and analog logic are active. When the BOE[7:0] input is LOW, the associated receive channel A through receive channel D PLL and analog logic are placed in a non-functional power saving mode. When RXLE returns LOW, the last values present on BOE[7:0] are captured in the internal RX PLL Enable latch. The specific mapping of BOE[7:0] signals to the associated receive channel enables is listed in *Table 16*. When the latch is closed, if the device is reset ($\overline{\text{TRSTZ}}$ is sampled LOW), the latch is reset to enable all receive channels.
BOE[7:0]	LVTTL Input, asynchronous, internal pull-up	BIST, Serial Output, and Receive Channel Enables. These inputs are passed to and through the output enable latch when OELE is HIGH, and captured in this latch when OELE returns LOW. These inputs are passed to and through the BIST enable latch when BISTLE is HIGH, and captured in this latch when BISTLE returns LOW. These inputs are passed to and through the Receive Channel enable latch when RXLE is HIGH, and captured in this latch when RXLE returns LOW. The specific mapping of BOE[7:0] signals to transmit output enables is listed in Table 15
$\overline{\text{LFIA}}$ $\overline{\text{LFIB}}$ $\overline{\text{LFIC}}$ $\overline{\text{LFID}}$	LVTTL Output, synchronous to the selected RXCLKx↑ output or REFCLK↑[36] input, asynchronous to receive channel enable/disable	Link Fault Indication output. Active LOW. $\overline{\text{LFI}}$ is the logical OR of four internal conditions: 1. Received serial data frequency outside expected range 2. Analog amplitude below expected levels 3. Transition density lower than expected 4. Receive Channel disabled
Bonding Control		
BONDST[1:0]	Bidirectional Open Drain, internal pull-up	Bonding Status. These signals are only used when multiple devices are bonded together. They communicate the status of the present internal bonding and Elasticity Buffer management events to the slave devices. These outputs change with the same timing as the receive output data buses, but are connected only to all the slave PSI transceiver block devices. When $\overline{\text{MASTER}}$ = LOW, these are output signals and present the Elasticity Buffer status from the selected receive channel of the device configured as the master. Receive master channel selection is performed using the RXCKB+ and RXCKD+ inputs. These status outputs indicate one of four possible conditions, on a synchronous basis, to the slave devices. These condition are: 00—Word Sync Sequence received 01—Add one K28.5 immediately following the next framing character received 10—Delete next framing character received 11—Normal data These outputs are driven only when the device is configured as a master, all four channels are bonded together, and the receive parallel interface is clocked by REFCLK↑.

Frequency Agile PSI

Name	Function	Signal Description
$\overline{\text{MASTER}}$	LVTTL Input, static configuration input internal pull-down	Master Device Select. When LOW, the present device is configured as the master, and BONDST[1:0] are outputs. When MASTER = HIGH, BONDST[1:0] are inputs. $\overline{\text{MASTER}}$ is only interpreted when configured for quad channel bonding, and the receive parallel interface is clocked by REFCLK↑.
BOND_ALL	Bidirectional Open Drain, Internal pull-up	All Channels Bonded Indicator. Active HIGH, wired AND. When HIGH, all receive channels have detected valid framing. This output is driven only when all four channels are bonded together, and the receive parallel interface is clocked by REFCLK↑.
$\overline{\text{BOND\_INH}}$	LVTTL Input, static configuration input Internal pull-up	Parallel Bond Inhibit. Active LOW. When asserted (LOW), this signal inhibits the adjustment of character offsets in all receive channels if the Bonding Sequence has *not* been detected in all bonded channels. When HIGH, all channels that have detected the Bonding Sequence are allowed to align their Receive Elasticity Buffer pipelines. For any channels to bond, the selected master channel must be a member of the group. When multiple devices are used together, the $\overline{\text{BOND\_INH}}$ input on all parts must be configured the same.
JTAG Interface		
TMS	LVTTL Input, internal pull-up	Test Mode Select. Used to control access to the JTAG Test Modes. If maintained high for ≥5 TCLK cycles, the JTAG test controller is reset.
TCLK	LVTTL Input, internal pull-down	JTAG Test Clock
TDO	Three-State LVTTL Output	Test Data Out. JTAG data output buffer which is High-Z while JTAG test mode is not selected.
TDI	LVTTL Input, internal pull-up	Test Data In. JTAG data input port.
$\overline{\text{TRSTZ}}$	LVTTL Input, internal pull-up	Test Port and Device Reset. Active LOW. Initializes the JTAG controller and all state machines and counters in the device. When asserted (LOW), this input asynchronously resets the JTAG test access port controller. When sampled LOW by the rising edge of REFLCK, this input resets the internal state machines and sets the Elasticity Buffer pointers to a nominal offset. When the reset is removed ($\overline{\text{TRSTZ}}$ sampled HIGH by REFCLK↑), the status and data outputs will become deterministic in less than 16 REFCLK cycles. The BISTLE, OELE, and RXLE latches will be reset by $\overline{\text{TRSTZ}}$.
Power		
V_{CC}	Power	+3.3V Supply (operating voltage)
GND	Ground	Signal and Power Ground
V_{CCIO0}	Power	V_{CC} for I/O bank 0
V_{CCIO1}	Power	V_{CC} for I/O bank 1
V_{CCIO2}	Power	V_{CC} for I/O bank 2
V_{CCIO3}	Power	V_{CC} for I/O bank 3
V_{CCIO4}	Power	V_{CC} for I/O bank 4
V_{CCIO5}	Power	V_{CC} for I/O bank 5 - Preset @ 3.3V TTL
V_{CCIO6}	Power	V_{CC} for I/O bank 6 - Preset @ 3.3V TTL
V_{CCIO7}	Power	V_{CC} for I/O bank 7 - Preset @ 3.3V TTL
V_{CCJTAG}	Power	V_{CC} for JTAG pins
V_{CCCNFG}	Power	V_{CC} for Configuration port
V_{CCPLL}	Power	V_{CC} for logic PLL
V_{CCPRG}	Power	V_{CC} for the Self-Boot™ solution embedded boot PROM

X3.230 Codes and Notation Conventions

Information to be transmitted over a serial link is encoded eight bits at a time into a 10-bit Transmission Character and then sent serially, bit by bit. Information received over a serial link is collected ten bits at a time, and those Transmission Characters that are used for data (Data Characters) are decoded into the correct eight-bit codes. The 10-bit Transmission Code supports all 256 8-bit combinations. Some of the remaining Transmission Characters (Special Characters) are used for functions other than data transmission.

The primary rationale for use of a Transmission Code is to improve the transmission characteristics of a serial link. The encoding defined by the Transmission Code ensures that sufficient transitions are present in the serial bit stream to make clock recovery possible at the Receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some Special Characters of the Transmission Code selected by Fibre Channel Standard consist of a distinct and easily recognizable bit pattern (the Special Character COMMA) that assists a Receiver in achieving word alignment on the incoming bit stream.

Notation Conventions

The documentation for the 8B/10B Transmission Code uses letter notation for the bits in an 8-bit byte. Fibre Channel Standard notation uses a bit notation of A, B, C, D, E, F, G, H for the 8-bit byte for the raw 8-bit data, and the letters a, b, c, d, e, i, f, g, h, j for encoded 10-bit data. There is a correspondence between bit A and bit a, B and b, C and c, D and d, E and e, F and f, G and g, and H and h. Bits i and j are derived, respectively, from (A,B,C,D,E) and (F,G,H).

The bit labeled A in the description of the 8B/10B Transmission Code corresponds to bit 0 in the numbering scheme of the FC-2 specification, B corresponds to bit 1, as shown below.

FC-2 bit designation—	7	6	5	4	3	2	1	0
HOTLink D/Q designation—	7	6	5	4	3	2	1	0
8B/10B bit designation—	H	G	F	E	D	C	B	A

To clarify this correspondence, the following example shows the conversion from an FC-2 Valid Data Byte to a Transmission Character (using 8B/10B Transmission Code notation)

```
FC-2  45
      Bits: 7654 3210
            0100 0101
```

Converted to 8B/10B notation (note carefully that the order of bits is reversed):

```
Data Byte Name   D5.2
                 Bits:ABCDEFGH
                      10100  010
```

Translated to a transmission Character in the 8B/10B Transmission Code:

```
      Bits: abcdeifghj
            1010010101
```

Each valid Transmission Character of the 8B/10B Transmission Code has been given a name using the following convention: cxx.y, where c is used to show whether the Transmission Character is a Data Character (c is set to D, and SC/$\overline{D}$ = LOW) or a Special Character (c is set to K, and SC/$\overline{D}$ = HIGH). When c is set to D, xx is the decimal value of the binary number composed of the bits E, D, C, B, and A in that order, and the y is the decimal value of the binary number composed of the bits H, G, and F in that order. When c is set to K, xx and y are derived by comparing the encoded bit patterns of the Special Character to those patterns derived from encoded Valid Data bytes and selecting the names of the patterns most similar to the encoded bit patterns of the Special Character.

Under the above conventions, the Transmission Character used for the examples above, is referred to by the name D5.2. The Special Character K29.7 is so named because the first six bits (abcdei) of this character make up a bit pattern similar to that resulting from the encoding of the unencoded 11101 pattern (29), and because the second four bits (fghj) make up a bit pattern similar to that resulting from the encoding of the unencoded 111 pattern (7).

Note: This definition of the 10-bit Transmission Code is based on (and is in basic agreement with) the following references, which describe the same 10-bit transmission code.

A.X. Widmer and P.A. Franaszek. “A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code” *IBM Journal of Research and Development*, 27, No. 5: 440–451 (September, 1983).

U.S. Patent 4,486,739. Peter A. Franaszek and Albert X. Widmer. “Byte-Oriented DC Balanced (0.4) 8B/10B Partitioned Block Transmission Code” (December 4, 1984).

Fibre Channel Physical and Signaling Interface (ANS X3.230–1994 ANSI FC–PH Standard).

IBM Enterprise Systems Architecture/390 ESCON I/O Interface (document number SA22–7202).

8B/10B Transmission Code

The following information describes how the tables shall be used for both generating valid Transmission Characters (encoding) and checking the validity of received Transmission Characters (decoding). It also specifies the ordering rules to be followed when transmitting the bits within a character and the characters within the higher-level constructs specified by the standard.

Transmission Order

Within the definition of the 8B/10B Transmission Code, the bit positions of the Transmission Characters are labeled a, b, c, d, e, i, f, g, h, j. Bit “a” shall be transmitted first followed by bits b, c, d, e, i, f, g, h, and j in that order. (Note that bit i shall be transmitted between bit e and bit f, rather than in alphabetical order.)

Valid and Invalid Transmission Characters

The following tables define the valid Data Characters and valid Special Characters (K characters), respectively. The tables are used for both generating valid Transmission Characters (encoding) and checking the validity of received Transmission Characters (decoding). In the tables, each Valid-Data-byte or Special-Character-code entry has two columns that represent two (not necessarily different) Transmission Characters. The two columns correspond to the current value of the running disparity (“Current RD–” or “Current RD+”). Running disparity is a binary parameter with either the value negative (–) or the value positive (+).

After powering on, the Transmitter may assume either a positive or negative value for its initial running disparity. Upon transmission of any Transmission Character, the transmitter will select the proper version of the Transmission Character based on the current running disparity value, and the Transmit-

ter shall calculate a new value for its running disparity based on the contents of the transmitted character. Special Character codes C1.7 and C2.7 can be used to force the transmission of a specific Special Character with a specific running disparity as required for some special sequences in X3.230.

After powering on, the Receiver may assume either a positive or negative value for its initial running disparity. Upon reception of any Transmission Character, the Receiver shall decide whether the Transmission Character is valid or invalid according to the following rules and tables and shall calculate a new value for its Running Disparity based on the contents of the received character.

The following rules for running disparity shall be used to calculate the new running-disparity value for Transmission Characters that have been transmitted (Transmitter's running disparity) and that have been received (Receiver's running disparity).

Running disparity for a Transmission Character shall be calculated from sub-blocks, where the first six bits (abcdei) form one sub-block and the second four bits (fghj) form the other sub-block. Running disparity at the beginning of the 6-bit sub-block is the running disparity at the end of the previous Transmission Character. Running disparity at the beginning of the 4-bit sub-block is the running disparity at the end of the 6-bit sub-block. Running disparity at the end of the Transmission Character is the running disparity at the end of the 4-bit sub-block.

Running disparity for the sub-blocks shall be calculated as follows:

1. Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the 6-bit sub-block if the 6-bit sub-block is 000111, and it is positive at the end of the 4-bit sub-block if the 4-bit sub-block is 0011.
2. Running disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the 6-bit sub-block if the 6-bit sub-block is 111000, and it is negative at the end of the 4-bit sub-block if the 4-bit sub-block is 1100.
3. Otherwise, running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

Use of the Tables for Generating Transmission Characters

The appropriate entry in the table shall be found for the Valid Data byte or the Special Character byte for which a Transmission Character is to be generated (encoded). The current value of the Transmitter's running disparity shall be used to select the Transmission Character from its corresponding column. For each Transmission Character transmitted, a new value of the running disparity shall be calculated. This new value shall be used as the Transmitter's current running disparity for the next Valid Data byte or Special Character byte to be encoded and transmitted. *Table 27* shows naming notations and examples of valid transmission characters.

Use of the Tables for Checking the Validity of Received Transmission Characters

The column corresponding to the current value of the Receiver's running disparity shall be searched for the received Transmission Character. If the received Transmission Character is found in the proper column, then the Transmission Character is valid and the associated Data byte or Special Character code is determined (decoded). If the received Transmission Character is not found in that column, then the Transmission Character is invalid. This is called a code violation. Independent of the Transmission Character's validity, the received Transmission Character shall be used to calculate a new value of running disparity. The new value shall be used as the Receiver's current running disparity for the next received Transmission Character.

Table 27. Valid Transmission Characters

Data			
	D_{IN} or Q_{OUT}		
Byte Name	765	43210	Hex Value
D0.0	000	00000	00
D1.0	000	00001	01
D2.0	000	00010	02
. .	. .	. .	. .
D5.2	010	00010 1	45
. .	. .	. .	. .
D30.7	111	11110	FE
D31.7	111	11111	FF

Detection of a code violation does not necessarily show that the Transmission Character in which the code violation was detected is in error. Code violations may result from a prior error that altered the running disparity of the bit stream which did not result in a detectable error at the Transmission Character in which the error occurred. *Table 28* shows an example of this behavior.

Table 28. Code Violations Resulting from Prior Errors

	RD	Character	RD	Character	RD	Character	RD
Transmitted data character	–	D21.1	–	D10.2	–	D23.5	+
Transmitted bit stream	–	101010 1001	–	010101 0101	–	111010 1010	+
Bit stream after error	–	101010 1011	+	010101 0101	+	111010 1010	+
Decoded data character	–	D21.0	+	D10.2	+	Code Violation	+

Table 29. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000)

Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj
D0.0	000 00000	100111 0100	011000 1011
D1.0	000 00001	011101 0100	100010 1011
D2.0	000 00010	101101 0100	010010 1011
D3.0	000 00011	110001 1011	110001 0100
D4.0	000 00100	110101 0100	001010 1011
D5.0	000 00101	101001 1011	101001 0100
D6.0	000 00110	011001 1011	011001 0100
D7.0	000 00111	111000 1011	000111 0100
D8.0	000 01000	111001 0100	000110 1011
D9.0	000 01001	100101 1011	100101 0100
D10.0	000 01010	010101 1011	010101 0100
D11.0	000 01011	110100 1011	110100 0100
D12.0	000 01100	001101 1011	001101 0100
D13.0	000 01101	101100 1011	101100 0100
D14.0	000 01110	011100 1011	011100 0100
D15.0	000 01111	010111 0100	101000 1011
D16.0	000 10000	011011 0100	100100 1011
D17.0	000 10001	100011 1011	100011 0100
D18.0	000 10010	010011 1011	010011 0100
D19.0	000 10011	110010 1011	110010 0100
D20.0	000 10100	001011 1011	001011 0100
D21.0	000 10101	101010 1011	101010 0100
D22.0	000 10110	011010 1011	011010 0100
D23.0	000 10111	111010 0100	000101 1011
D24.0	000 11000	110011 0100	001100 1011
D25.0	000 11001	100110 1011	100110 0100
D26.0	000 11010	010110 1011	010110 0100
D27.0	000 11011	110110 0100	001001 1011
D28.0	000 11100	001110 1011	001110 0100
D29.0	000 11101	101110 0100	010001 1011
D30.0	000 11110	011110 0100	100001 1011
D31.0	000 11111	101011 0100	010100 1011

Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj
D0.1	001 00000	100111 1001	011000 1001
D1.1	001 00001	011101 1001	100010 1001
D2.1	001 00010	101101 1001	010010 1001
D3.1	001 00011	110001 1001	110001 1001
D4.1	001 00100	110101 1001	001010 1001
D5.1	001 00101	101001 1001	101001 1001
D6.1	001 00110	011001 1001	011001 1001
D7.1	001 00111	111000 1001	000111 1001
D8.1	001 01000	111001 1001	000110 1001
D9.1	001 01001	100101 1001	100101 1001
D10.1	001 01010	010101 1001	010101 1001
D11.1	001 01011	110100 1001	110100 1001
D12.1	001 01100	001101 1001	001101 1001
D13.1	001 01101	101100 1001	101100 1001
D14.1	001 01110	011100 1001	011100 1001
D15.1	001 01111	010111 1001	101000 1001
D16.1	001 10000	011011 1001	100100 1001
D17.1	001 10001	100011 1001	100011 1001
D18.1	001 10010	010011 1001	010011 1001
D19.1	001 10011	110010 1001	110010 1001
D20.1	001 10100	001011 1001	001011 1001
D21.1	001 10101	101010 1001	101010 1001
D22.1	001 10110	011010 1001	011010 1001
D23.1	001 10111	111010 1001	000101 1001
D24.1	001 11000	110011 1001	001100 1001
D25.1	001 11001	100110 1001	100110 1001
D26.1	001 11010	010110 1001	010110 1001
D27.1	001 11011	110110 1001	001001 1001
D28.1	001 11100	001110 1001	001110 1001
D29.1	001 11101	101110 1001	010001 1001
D30.1	001 11110	011110 1001	100001 1001
D31.1	001 11111	101011 1001	010100 1001

Table 29. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000) (continued)

Data Byte Name	Bits	Current RD–	Current RD+	Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj		HGF EDCBA	abcdei fghj	abcdei fghj
D0.2	010 00000	100111 0101	011000 0101	D0.3	011 00000	100111 0011	011000 1100
D1.2	010 00001	011101 0101	100010 0101	D1.3	011 00001	011101 0011	100010 1100
D2.2	010 00010	101101 0101	010010 0101	D2.3	011 00010	101101 0011	010010 1100
D3.2	010 00011	110001 0101	110001 0101	D3.3	011 00011	110001 1100	110001 0011
D4.2	010 00100	110101 0101	001010 0101	D4.3	011 00100	110101 0011	001010 1100
D5.2	010 00101	101001 0101	101001 0101	D5.3	011 00101	101001 1100	101001 0011
D6.2	010 00110	011001 0101	011001 0101	D6.3	011 00110	011001 1100	011001 0011
D7.2	010 00111	111000 0101	000111 0101	D7.3	011 00111	111000 1100	000111 0011
D8.2	010 01000	111001 0101	000110 0101	D8.3	011 01000	111001 0011	000110 1100
D9.2	010 01001	100101 0101	100101 0101	D9.3	011 01001	100101 1100	100101 0011
D10.2	010 01010	010101 0101	010101 0101	D10.3	011 01010	010101 1100	010101 0011
D11.2	010 01011	110100 0101	110100 0101	D11.3	011 01011	110100 1100	110100 0011
D12.2	010 01100	001101 0101	001101 0101	D12.3	011 01100	001101 1100	001101 0011
D13.2	010 01101	101100 0101	101100 0101	D13.3	011 01101	101100 1100	101100 0011
D14.2	010 01110	011100 0101	011100 0101	D14.3	011 01110	011100 1100	011100 0011
D15.2	010 01111	010111 0101	101000 0101	D15.3	011 01111	010111 0011	101000 1100
D16.2	010 10000	011011 0101	100100 0101	D16.3	011 10000	011011 0011	100100 1100
D17.2	010 10001	100011 0101	100011 0101	D17.3	011 10001	100011 1100	100011 0011
D18.2	010 10010	010011 0101	010011 0101	D18.3	011 10010	010011 1100	010011 0011
D19.2	010 10011	110010 0101	110010 0101	D19.3	011 10011	110010 1100	110010 0011
D20.2	010 10100	001011 0101	001011 0101	D20.3	011 10100	001011 1100	001011 0011
D21.2	010 10101	101010 0101	101010 0101	D21.3	011 10101	101010 1100	101010 0011
D22.2	010 10110	011010 0101	011010 0101	D22.3	011 10110	011010 1100	011010 0011
D23.2	010 10111	111010 0101	000101 0101	D23.3	011 10111	111010 0011	000101 1100
D24.2	010 11000	110011 0101	001100 0101	D24.3	011 11000	110011 0011	001100 1100
D25.2	010 11001	100110 0101	100110 0101	D25.3	011 11001	100110 1100	100110 0011
D26.2	010 11010	010110 0101	010110 0101	D26.3	011 11010	010110 1100	010110 0011
D27.2	010 11011	110110 0101	001001 0101	D27.3	011 11011	110110 0011	001001 1100
D28.2	010 11100	001110 0101	001110 0101	D28.3	011 11100	001110 1100	001110 0011
D29.2	010 11101	101110 0101	010001 0101	D29.3	011 11101	101110 0011	010001 1100
D30.2	010 11110	011110 0101	100001 0101	D30.3	011 11110	011110 0011	100001 1100
D31.2	010 11111	101011 0101	010100 0101	D31.3	011 11111	101011 0011	010100 1100

Table 29. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000) (continued)

Data Byte Name	Bits	Current RD–	Current RD+	Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj		HGF EDCBA	abcdei fghj	abcdei fghj
D0.4	100 00000	100111 0010	011000 1101	D0.5	101 00000	100111 1010	011000 1010
D1.4	100 00001	011101 0010	100010 1101	D1.5	101 00001	011101 1010	100010 1010
D2.4	100 00010	101101 0010	010010 1101	D2.5	101 00010	101101 1010	010010 1010
D3.4	100 00011	110001 1101	110001 0010	D3.5	101 00011	110001 1010	110001 1010
D4.4	100 00100	110101 0010	001010 1101	D4.5	101 00100	110101 1010	001010 1010
D5.4	100 00101	101001 1101	101001 0010	D5.5	101 00101	101001 1010	101001 1010
D6.4	100 00110	011001 1101	011001 0010	D6.5	101 00110	011001 1010	011001 1010
D7.4	100 00111	111000 1101	000111 0010	D7.5	101 00111	111000 1010	000111 1010
D8.4	100 01000	111001 0010	000110 1101	D8.5	101 01000	111001 1010	000110 1010
D9.4	100 01001	100101 1101	100101 0010	D9.5	101 01001	100101 1010	100101 1010
D10.4	100 01010	010101 1101	010101 0010	D10.5	101 01010	010101 1010	010101 1010
D11.4	100 01011	110100 1101	110100 0010	D11.5	101 01011	110100 1010	110100 1010
D12.4	100 01100	001101 1101	001101 0010	D12.5	101 01100	001101 1010	001101 1010
D13.4	100 01101	101100 1101	101100 0010	D13.5	101 01101	101100 1010	101100 1010
D14.4	100 01110	011100 1101	011100 0010	D14.5	101 01110	011100 1010	011100 1010
D15.4	100 01111	010111 0010	101000 1101	D15.5	101 01111	010111 1010	101000 1010
D16.4	100 10000	011011 0010	100100 1101	D16.5	101 10000	011011 1010	100100 1010
D17.4	100 10001	100011 1101	100011 0010	D17.5	101 10001	100011 1010	100011 1010
D18.4	100 10010	010011 1101	010011 0010	D18.5	101 10010	010011 1010	010011 1010
D19.4	100 10011	110010 1101	110010 0010	D19.5	101 10011	110010 1010	110010 1010
D20.4	100 10100	001011 1101	001011 0010	D20.5	101 10100	001011 1010	001011 1010
D21.4	100 10101	101010 1101	101010 0010	D21.5	101 10101	101010 1010	101010 1010
D22.4	100 10110	011010 1101	011010 0010	D22.5	101 10110	011010 1010	011010 1010
D23.4	100 10111	111010 0010	000101 1101	D23.5	101 10111	111010 1010	000101 1010
D24.4	100 11000	110011 0010	001100 1101	D24.5	101 11000	110011 1010	001100 1010
D25.4	100 11001	100110 1101	100110 0010	D25.5	101 11001	100110 1010	100110 1010
D26.4	100 11010	010110 1101	010110 0010	D26.5	101 11010	010110 1010	010110 1010
D27.4	100 11011	110110 0010	001001 1101	D27.5	101 11011	110110 1010	001001 1010
D28.4	100 11100	001110 1101	001110 0010	D28.5	101 11100	001110 1010	001110 1010
D29.4	100 11101	101110 0010	010001 1101	D29.5	101 11101	101110 1010	010001 1010
D30.4	100 11110	011110 0010	100001 1101	D30.5	101 11110	011110 1010	100001 1010
D31.4	100 11111	101011 0010	010100 1101	D31.5	101 11111	101011 1010	010100 1010
D0.6	110 00000	100111 0110	011000 0110	D0.7	111 00000	100111 0001	011000 1110
D1.6	110 00001	011101 0110	100010 0110	D1.7	111 00001	011101 0001	100010 1110
D2.6	110 00010	101101 0110	010010 0110	D2.7	111 00010	101101 0001	010010 1110

Table 29. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000) (continued)

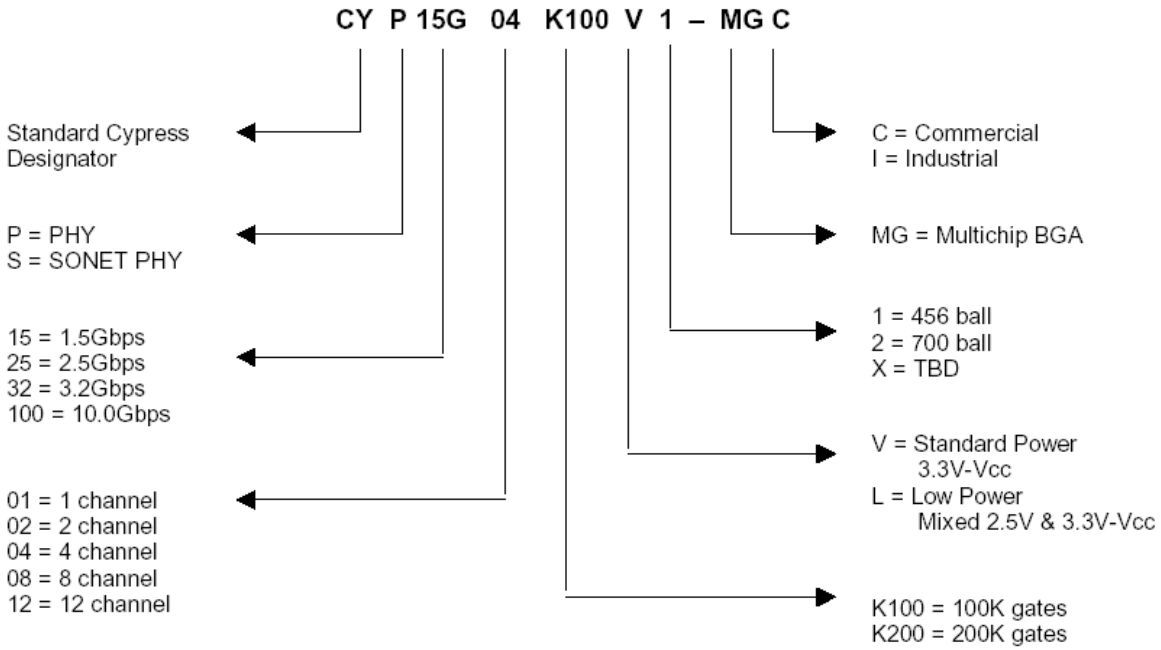
Data Byte Name	Bits	Current RD–	Current RD+	Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj		HGF EDCBA	abcdei fghj	abcdei fghj
D3.6	110 00011	110001 0110	110001 0110	D3.7	111 00011	110001 1110	110001 0001
D4.6	110 00100	110101 0110	001010 0110	D4.7	111 00100	110101 0001	001010 1110
D5.6	110 00101	101001 0110	101001 0110	D5.7	111 00101	101001 1110	101001 0001
D6.6	110 00110	011001 0110	011001 0110	D6.7	111 00110	011001 1110	011001 0001
D7.6	110 00111	111000 0110	000111 0110	D7.7	111 00111	111000 1110	000111 0001
D8.6	110 01000	111001 0110	000110 0110	D8.7	111 01000	111001 0001	000110 1110
D9.6	110 01001	100101 0110	100101 0110	D9.7	111 01001	100101 1110	100101 0001
D10.6	110 01010	010101 0110	010101 0110	D10.7	111 01010	010101 1110	010101 0001
D11.6	110 01011	110100 0110	110100 0110	D11.7	111 01011	110100 1110	110100 1000
D12.6	110 01100	001101 0110	001101 0110	D12.7	111 01100	001101 1110	001101 0001
D13.6	110 01101	101100 0110	101100 0110	D13.7	111 01101	101100 1110	101100 1000
D14.6	110 01110	011100 0110	011100 0110	D14.7	111 01110	011100 1110	011100 1000
D15.6	110 01111	010111 0110	101000 0110	D15.7	111 01111	010111 0001	101000 1110
D16.6	110 10000	011011 0110	100100 0110	D16.7	111 10000	011011 0001	100100 1110
D17.6	110 10001	100011 0110	100011 0110	D17.7	111 10001	100011 0111	100011 0001
D18.6	110 10010	010011 0110	010011 0110	D18.7	111 10010	010011 0111	010011 0001
D19.6	110 10011	110010 0110	110010 0110	D19.7	111 10011	110010 1110	110010 0001
D20.6	110 10100	001011 0110	001011 0110	D20.7	111 10100	001011 0111	001011 0001
D21.6	110 10101	101010 0110	101010 0110	D21.7	111 10101	101010 1110	101010 0001
D22.6	110 10110	011010 0110	011010 0110	D22.7	111 10110	011010 1110	011010 0001
D23.6	110 10111	111010 0110	000101 0110	D23.7	111 10111	111010 0001	000101 1110
D24.6	110 11000	110011 0110	001100 0110	D24.7	111 11000	110011 0001	001100 1110
D25.6	110 11001	100110 0110	100110 0110	D25.7	111 11001	100110 1110	100110 0001
D26.6	110 11010	010110 0110	010110 0110	D26.7	111 11010	010110 1110	010110 0001
D27.6	110 11011	110110 0110	001001 0110	D27.7	111 11011	110110 0001	001001 1110
D28.6	110 11100	001110 0110	001110 0110	D28.7	111 11100	001110 1110	001110 0001
D29.6	110 11101	101110 0110	010001 0110	D29.7	111 11101	101110 0001	010001 1110
D30.6	110 11110	011110 0110	100001 0110	D30.7	111 11110	011110 0001	100001 1110
D31.6	110 11111	101011 0110	010100 0110	D31.7	111 11111	101011 0001	010100 1110

Table 30. Valid Special Character Codes and Sequences (TXCTx = special character code or RXSTx[2:0] = 001)[38, 39]

S.C. Code Name	S.C. Byte Name						Current RD– abcdei fghj	Current RD+ abcdei fghj
	Cypress			Alternate				
	S.C. Byte Name[40]		Bits HGF EDCBA	S.C. Byte Name[40]		Bits HGF EDCBA		
K28.0	C0.0	(C00)	000 00000	C28.0	(C1C)	000 11100	001111 0100	110000 1011
K28.1 [41]	C1.0	(C01)	000 00001	C28.1	(C3C)	001 11100	001111 1001	110000 0110
K28.2 [41]	C2.0	(C02)	000 00010	C28.2	(C5C)	010 11100	001111 0101	110000 1010
K28.3	C3.0	(C03)	000 00011	C28.3	(C7C)	011 11100	001111 0011	110000 1100
K28.4 [41]	C4.0	(C04)	000 00100	C28.4	(C9C)	100 11100	001111 0010	110000 1101
K28.5 [41, 42]	C5.0	(C05)	000 00101	C28.5	(CBC)	101 11100	001111 1010	110000 0101
K28.6 [41]	C6.0	(C06)	000 00110	C28.6	(CDC)	110 11100	001111 0110	110000 1001
K28.7 [41, 43]	C7.0	(C07)	000 00111	C28.7	(CFC)	111 11100	001111 1000	110000 0111
K23.7	C8.0	(C08)	000 01000	C23.7	(CF7)	111 10111	111010 1000	000101 0111
K27.7	C9.0	(C09)	000 01001	C27.7	(CFB)	111 11011	110110 1000	001001 0111
K29.7	C10.0	(C0A)	000 01010	C29.7	(CFD)	111 11101	101110 1000	010001 0111
K30.7	C11.0	(C0B)	000 01011	C30.7	(CFE)	111 11110	011110 1000	100001 0111
End of Frame Sequence								
EOFxx	C2.1	(C22)	001 00010	C2.1	(C22)	001 00010	–K28.5,Dn.xxx0[44]	+K28.5,Dn.xxx1[44]
Code Rule Violation and SVS Tx Pattern								
Exception [43, 45]	C0.7	(CE0)	111 00000	C0.7	(CE0)	111 00000 [49]	100111 1000	011000 0111
–K28.5[46]	C1.7	(CE1)	111 00001	C1.7	(CE1)	111 00001 [49]	001111 1010	001111 1010
+K28.5 [47]	C2.7	(CE2)	111 00010	C2.7	(CE2)	111 00010 [49]	110000 0101	110000 0101
Running Disparity Violation Pattern								
Exception [48]	C4.7	(CE4)	111 00100	C4.7	(CE4)	111 00100 [49]	110111 0101	001000 1010

Notes:

38. All codes not shown are reserved.
39. Notation for Special Character Code Name is consistent with Fibre Channel and ESCON naming conventions. Special Character Code Name is intended to describe binary information present on I/O pins. Common usage for the name can either be in the form used for describing Data patterns (i.e., C0.0 through C31.7), or in hex notation (i.e., Cnn where nn = the specified value between 00 and FF).
40. Both the Cypress and alternate encodings may be used for data transmission to generate specific Special Character Codes. The decoding process for received characters generates Cypress codes or Alternate codes as selected by the RXMODE[1:0] configuration inputs.
41. These characters are used for control of ESCON interfaces. They can be sent as embedded commands or other markers when not operating using ESCON protocols.
42. The K28.5 character is used for framing operations by the receiver. It is also the pad or fill character transmitted to maintain the serial link when no user data is available.
43. Care must be taken when using this Special Character code. When a C7.0 is followed by a D11.x or D20.x, or when an SVS (C0.7) is followed by a D11.x, an alias K28.5 sync character is created. These sequences can cause erroneous framing and should be avoided while RFEN = HIGH.
44. C2.1 = Transmit either –K28.5+ or +K28.5– as determined by Current RD and modify the Transmission Character that follows, by setting its least significant bit to 1 or 0. If Current RD at the start of the following character is plus (+) the LSB is set to 0, and if Current RD is minus (–) the LSB becomes 1. This modification allows construction of X3.230 “EOF” frame delimiters wherein the second data byte is determined by the Current RD.
For example, to send “EOFdt” the controller could issue the sequence C2.1–D21.4– D21.4–D21.4, and the HOTLink Transmitter will send either K28.5–D21.4–D21.4–D21.4 or K28.5–D21.5– D21.4–D21.4 based on Current RD. Likewise to send “EOFdti” the controller could issue the sequence C2.1–D10.4–D21.4–D21.4, and the HOTLink Transmitter will send either K28.5–D10.4–D21.4– D21.4 or K28.5–D10.5–D21.4– D21.4 based on Current RD.
The receiver will never output this Special Character, since K28.5 is decoded as C5.0, C1.7, or C2.7, and the subsequent bytes are decoded as data.
45. C0.7 = Transmit a deliberate code rule violation. The code chosen for this function follows the normal Running Disparity rules. Transmission of this Special Character has the same effect as asserting TXSVS = HIGH. The receiver will only output this Special Character if the Transmission Character being decoded is not found in the tables.
46. C1.7 = Transmit Negative K28.5 (–K28.5+) disregarding Current RD. The receiver will only output this Special Character if K28.5 is received with the wrong running disparity. The receiver will output C1.7 if –K28.5 is received with RD+, otherwise K28.5 is decoded as C5.0 or C2.7.
47. C2.7 = Transmit Positive K28.5 (+K28.5–) disregarding Current RD. The receiver will only output this Special Character if K28.5 is received with the wrong running disparity. The receiver will output C2.7 if +K28.5 is received with RD–, otherwise K28.5 is decoded as C5.0 or C1.7.
48. C4.7 = Transmit a deliberate code rule violation to indicate a Running Disparity violation. The receiver will only output this Special Character if the Transmission Character being decoded is found in the tables, but Running Disparity does not match. This might indicate that an error occurred in a prior byte.
49. Supported only for data transmission. The receive status for these conditions will be reported by specific combinations of receive status bits.

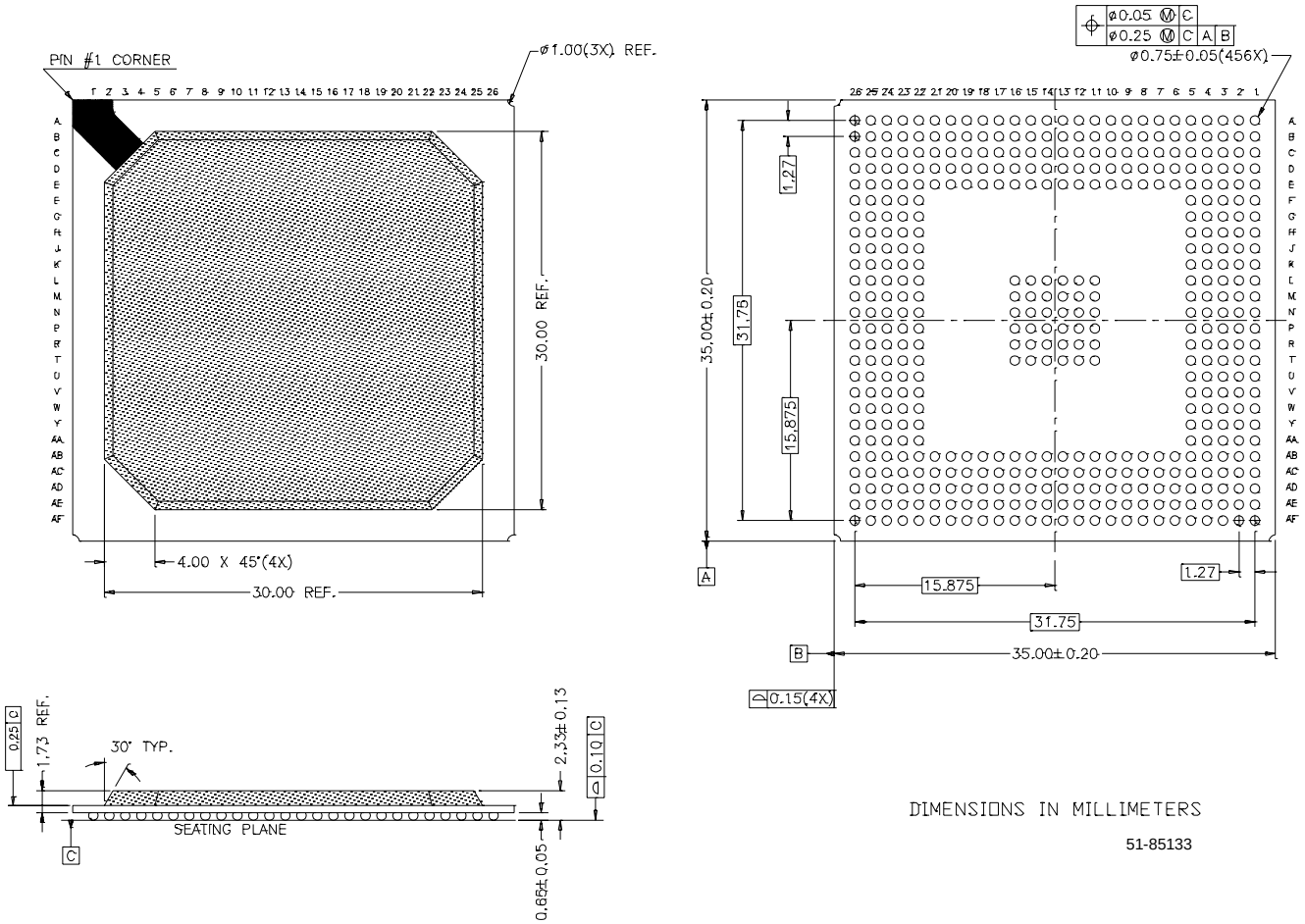


Ordering Information

Device	Channels & Link Speed	Ordering Code	Package Name	Package Type	Operating Range
25G01K100	1 x 2.5 Gbps	CYP25G01K100V1-MGC	456MGC	456-Ball Ball Grid Array	Commercial
	1 x 2.5 Gbps	CYS25G01K100V1-MGC	456MGC	456-Ball Ball Grid Array	Commercial
15G04K100	4 x 0.2 - 1.5 Gbps	CYP15G04K100V1-MGC	456MGC	456-Ball Ball Grid Array	Commercial
15G04K200	4 x 0.2 - 1.5 Gbps	CYP15G04K200V2-MGC	700MGC	700-Ball Ball Grid Array	Industrial
15G08K200	8 x 0.2 - 1.5 Gbps	CYP15G08K200V2-MGC	700MGC	700-Ball Ball Grid Array	Commercial

Package Diagrams

456-Lead Ball Grid Array (35 x 35 x 2.33 mm) BG456



Document Title: CYP15G04K100V1-MGC/CYP15G04K200V2-MGC Programmable Serial Interface (TM) (Frequency Agile Devices) Document Number: 38-02044				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	112077	02/14/02	MHW	New Data Sheet