Lattice Semiconductor Corporation - ORSO42G5-3BMN484C Datasheet

Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	10368
Total RAM Bits	113664
Number of I/O	204
Number of Gates	643000
Voltage - Supply	1.425V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	484-BBGA
Supplier Device Package	484-FPBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/lattice-semiconductor/orso42g5-3bmn484c

- Optional cell processing blocks included. Cell processing includes cell creation, extraction, idle cell insertion and deletion asynchronous from line rates. Four cell sizes supported:
  - 77 bytes per cell (75 bytes of data payload)
  - 81 bytes per cell (79 bytes of data payload)
  - 85 bytes per cell (83 bytes of data payload)
  - 93 bytes per cell (91 bytes of data payload)
- Automatic cell striping across either pairs of SERDES links or, for the ORSO82G5, all eight SERDES links.
- Addition of two 4K X 36 dual-port RAMs accessible by the programmable logic.

Programmable Features

- High-performance programmable logic:
  - 0.16 µm 7-level metal technology.
  - Internal performance of >250 MHz.
  - Over 400K usable system gates.
  - Meets multiple I/O interface standards.
  - 1.5V operation (30% less power than 1.8V operation) translates to greater performance.
- Traditional I/O selections:
  - LVTTL (3.3V) and LVCMOS (2.5V, and 1.8V) I/Os.
  - Per pin-selectable I/O clamping diodes provide 3.3V PCI compliance.
  - Individually programmable drive capability:
    24 mA sink/12 mA source, 12 mA sink/6 mA source, or 6 mA sink/3 mA source.
  - Two slew rates supported (fast and slew-limited).
  - Fast-capture input latch and input Flip-Flop (FF)/latch for reduced input setup time and zero hold time.
  - Fast open-drain drive capability.
  - Capability to register 3-state enable signal.
  - Off-chip clock drive capability.
  - Two-input function generator in output path.
- New programmable high-speed I/O:
  - Single-ended: GTL, GTL+, PECL, SSTL3/2 (class I and II), HSTL (Class I, III, IV), ZBT, and DDR.
  - Double-ended: LVDS, bused-LVDS, and LVPECL. Programmable (on/off), internal parallel termination (100 Ω) is also supported for these I/Os.
- New capability to (de)multiplex I/O signals:
  - New DDR on both input and output.
  - New 2x and 4x downlink and uplink capability per I/O.
- Enhanced twin-block Programmable Function Unit (PFU):
  - Eight 16-bit Look-Up Tables (LUTs) per PFU.
  - Nine user registers per PFU, one following each LUT, and organized to allow two nibbles to act independently, plus one extra for arithmetic operations.
  - New register control in each PFU has two independent programmable clocks, clock enables, local set/reset, and data selects.
  - New LUT structure allows flexible combinations of LUT4, LUT5, new LUT6, 4 → 1 MUX, new 8 → 1 MUX, and ripple mode arithmetic functions in the same PFU.
  - 32 x 4 RAM per PFU, configurable as single-port or dual-port. Create large, fast RAM/ROM blocks (128 x 8 in only eight PFUs) using the Supplemental Logic and Interconnect Cell (SLIC) decoders as bank drivers.
  - Soft-Wired LUTs (SWL) allow fast cascading of up to three levels of LUT logic in a single PFU through fast internal routing which reduces routing congestion and improves speed.
  - Flexible fast access to PFU inputs from routing.
  - Fast-carry logic and routing to all four adjacent PFUs for nibble-wide, byte-wide, or longer arithmetic functions, with the option to register the PFU carry-out.

Routing

The abundant routing resources of the Series 4 architecture are organized to route signals individually or as buses with related control signals. Both local and global signals utilize high-speed buffered and nonbuffered routes. One PLC segmented (x1), six PLC segmented (x6), and bused half chip (xHL) routes are patterned together to provide high connectivity with fast software routing times and high-speed system performance.

Eight fully distributed primary clocks are routed on a low-skew, high-speed distribution network and may be sourced from dedicated I/O pads, PLLs, PLC logic or the Embedded Core. Secondary and edge-clock routing is available for fast regional clock or control signal routing for both internal regions and on device edges. Secondary clock routing can also be sourced from any I/O pin, PLLs, PLC logic or the Embedded Core.

The improved routing resources offer great flexibility in moving signals to and from the logic core. This flexibility translates into an improved capability to route designs at the required speeds when the I/O signals have been locked to specific pins.

System Level Features

The Series 4 also provides system-level functionality by means of its MicroProcessor Interface, Embedded System Bus, block-port Embedded Block RAMs, universal Programmable Phase-Locked Loops, and the addition of highly tuned networking specific Phase-Locked Loops. These functional blocks allow for easy glueless system interfacing and the capability to adjust to varying conditions in today's high-speed networking systems.

MicroProcessor Interface

The MPI provides a glueless interface between the FPGA and *PowerPC* microprocessors. Programmable in 8-, 16-, and 32-bit interfaces with optional parity to the *Motorola® PowerPC* 860 bus, it can be used for configuration and readback, as well as for FPGA control and monitoring of FPGA status. All MPI transactions utilize the Series 4 Embedded System Bus at 66 MHz performance.

The MicroProcessor Interface (MPI) provides a system-level interface, using the system bus, to the FPGA user-defined logic following configuration, including access to the Embedded Block RAM and general logic. The MPI supports burst data read and write transfers, allowing short, uneven transmission of data through the interface by including data FIFOs. Transfer accesses can be single beat (1 x 4 bytes or less), 4-beat (4 x 4 bytes), 8-beat (8 x 2 bytes), or 16-beat (16 x 1 bytes).

System Bus

An on-chip, multimaster, 32-bit system bus with 4-bit parity facilitates communication among the MPI, configuration logic, FPGA control, and status registers, Embedded Block RAMs, as well as user logic. The Embedded System Bus offers arbiter, decoder, master, and slave elements. Master and slave elements are also available for the user-logic and a slave interface is used for control and status of the embedded backplane transceiver portion of the ORSO42G5 and ORSO82G5.

The system bus control registers can provide control to the FPGA such as signaling for reprogramming, reset functions, and PLL programming. Status registers monitor INIT, DONE, and system bus errors. An interrupt controller is integrated to provide up to eight possible interrupt resources. Bus clock generation can be sourced from the MicroProcessor Interface clock, configuration clock (for slave configuration modes), internal oscillator, user clock from routing, or from the port clock (for JTAG configuration modes).

Phase-Locked Loops

Up to eight PLLs are provided on each Series 4 device, with four user PLLs generally provided for FPSCs, including the ORSO42G5 and ORSO82G5. In the FPSCs, these PLLs can only be driven by the FPGA resources. Programmable PLLs can be used to manipulate the frequency, phase, and duty cycle of a clock signal. Each PPLL is capable of manipulating and conditioning clock outputs from 15 MHz to 420 MHz. Frequencies can be adjusted from 1/8x to 8x, the input clock frequency. Each programmable PLL provides two outputs that have different multiplication factors but can have the same phase relationships. Duty cycles and phase delays can be adjusted in

12.5% of the clock period increments. An automatic input buffer delay compensation mode is available for phase delay. Each PPLL provides two outputs that can have programmable (12.5% steps) phase differences.

Embedded Block RAM

New 512 x 18 block-port RAM blocks are embedded in the FPGA core to significantly increase the amount of memory and complement the distributed PFU memories. The EBRs include two write ports, two read ports, and two byte lane enables which provide four-port operation. Optional arbitration between the two write ports is available, as well as direct connection to the high-speed system bus.

Additional logic has been incorporated to allow significant flexibility for FIFO, constant multiply, and two-variable multiply functions. The user can configure FIFO blocks with flexible depths of 512K, 256K, and 1K including asynchronous and synchronous modes and programmable status and error flags. Multiplier capabilities allow a multiply of an 8-bit number with a 16-bit fixed coefficient or vice versa (24-bit output), or a multiply of two 8-bit numbers (16-bit output). On-the-fly coefficient modifications are available through the second read/write port.

Two 16 x 8-bit CAMs per embedded block can be implemented in single match, multiple match, and clear modes. The EBRs can also be preloaded at device configuration time.

Configuration

The FPGAs functionality is determined by internal configuration RAM. The FPGAs internal initialization/configuration circuitry loads the configuration data at powerup or under system control. The configuration data can reside externally in an EEPROM or any other storage media. Serial EEPROMs provide a simple, low pin-count method for configuring FPGAs.

The RAM is loaded by using one of several configuration modes. Supporting the traditional master/slave serial, master/slave parallel, and asynchronous peripheral modes, Series 4 also utilizes its MicroProcessor Interface and Embedded System Bus to perform both programming and readback. Daisy chaining of multiple devices and partial reconfiguration are also permitted.

Other configuration options include the initialization of the embedded-block RAM memories and FPSC memory as well as system bus options and bit stream error checking. Programming and readback through the JTAG *(IEEE* 1149.2*)* port is also available meeting In-System Programming (ISP) standards (*IEEE* 1532 Draft).

ORSO42G5 and ORSO82G5 Overview

The ORSO42G5 and ORSO82G5 FPSCs provide high-speed backplane transceivers combined with FPGA logic. The ORSO42G5 and ORSO82G5 devices are based on the 1.5V OR4E04 ORCA FPGA and have a 36 x 36 array of Programmable Logic Cells (PLCs). The embedded core, which contains the backplane transceivers, is attached to the right side of the device and is integrated directly into the FPGA array. A top level diagram of the basic chip configuration is shown in Figure 1.

ORSO42G5 and ORSO82G5 Embedded Core Detailed Description

The ORSO42G5 and ORSO82G5 have four and eight channels respectively, with a high-speed SERDES macro that performs clock data recovery, serializing and deserializing functions. There is also additional logic for SONET mode and cell mode data synchronization formatting and scrambling/descrambling. For all modes, the data paths can be characterized as the transmit path (FPGA to backplane) and receive path (backplane to FPGA); however the interface signal assignments between the FPGA logic and the core differ depending on the operating mode selected.

The three main operating modes in the ORSO42G5 and ORSO82G5 are:

- SERDES only mode
- SONET mode
- Cell mode
  - Two-link sub-mode
  - Eight-link sub-mode (ORSO82G5 only)

The SONET and cell modes each support sub-modes that can be selected by enabling or disabling certain functions through programmable register bits. Following the basic TX and RX architecture descriptions, the data formatting and logical implementations supporting each of the operational modes are described.

Top Level Description - Transmitter (TX) and Receiver (RX) Architectures

The next sections give a top level description of the transmitter and receive architectures. The high-speed transmit and receive serial data can operate at 0.6-2.7 Gbps depending on the state of the control bits from the system bus and the provided reference clock. For all of the architecture and clock distribution descriptions, however, the standard SONET STS-48 rate of 2,488.32 Mbits/s (i.e., REFCLK_[P:N] = 155.52 MHz for the full rate modes) is assumed.

Transmitter Architecture

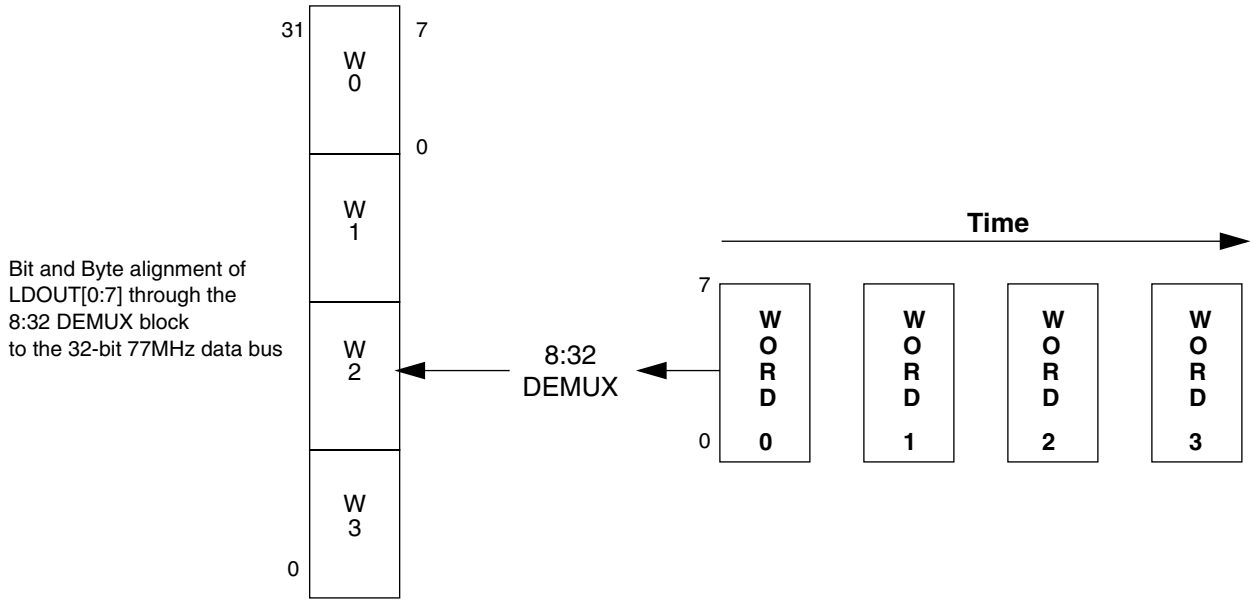
The transmitter section accepts parallel data for transmission from the FPGA logic, formats it for transmission and serializes the data. It also accepts the low-speed reference clock at the REFCLK input and uses this clock to synthesize the internal high-speed serial bit clock. The serialized transmitted data are available at the differential CML output pins to drive either an optical transmitters, coaxial media or a circuit board backplane.

The top level transmit architecture is shown in Figure 3. The main logical blocks in the transmit path are:

- Output Port Controllers (OPCs) which contain the cell processing logic.
- SONET processing logic.
- Transmit SERDES and 32:8 MUX.

Depending on the mode of operation, the FPGA to backplane data path may include or bypass the various logical blocks.

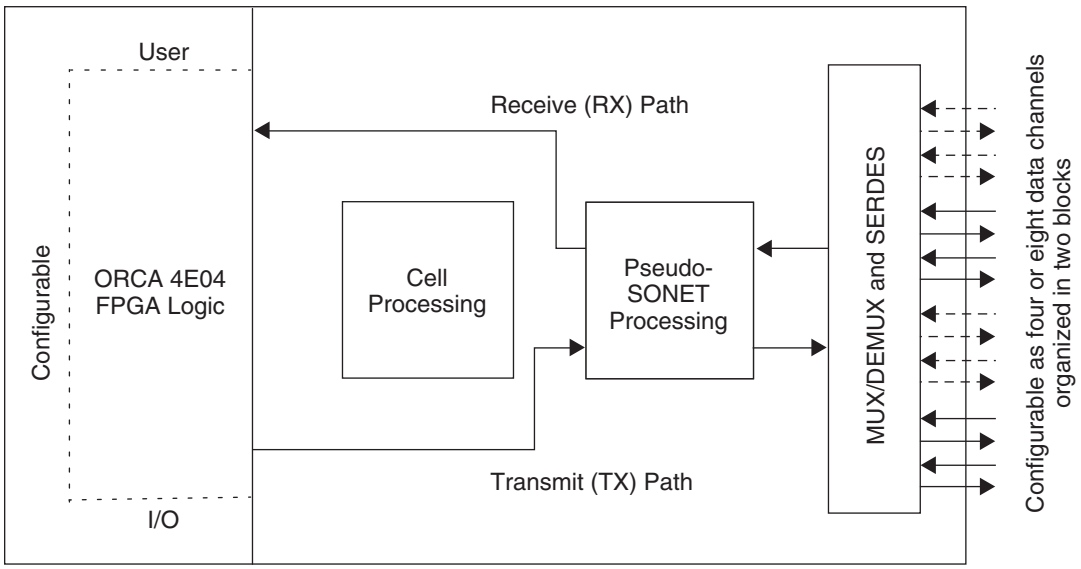
Figure 10. Bit and Byte Alignment for DEMUX Block



SONET Mode Operation – Detailed Description

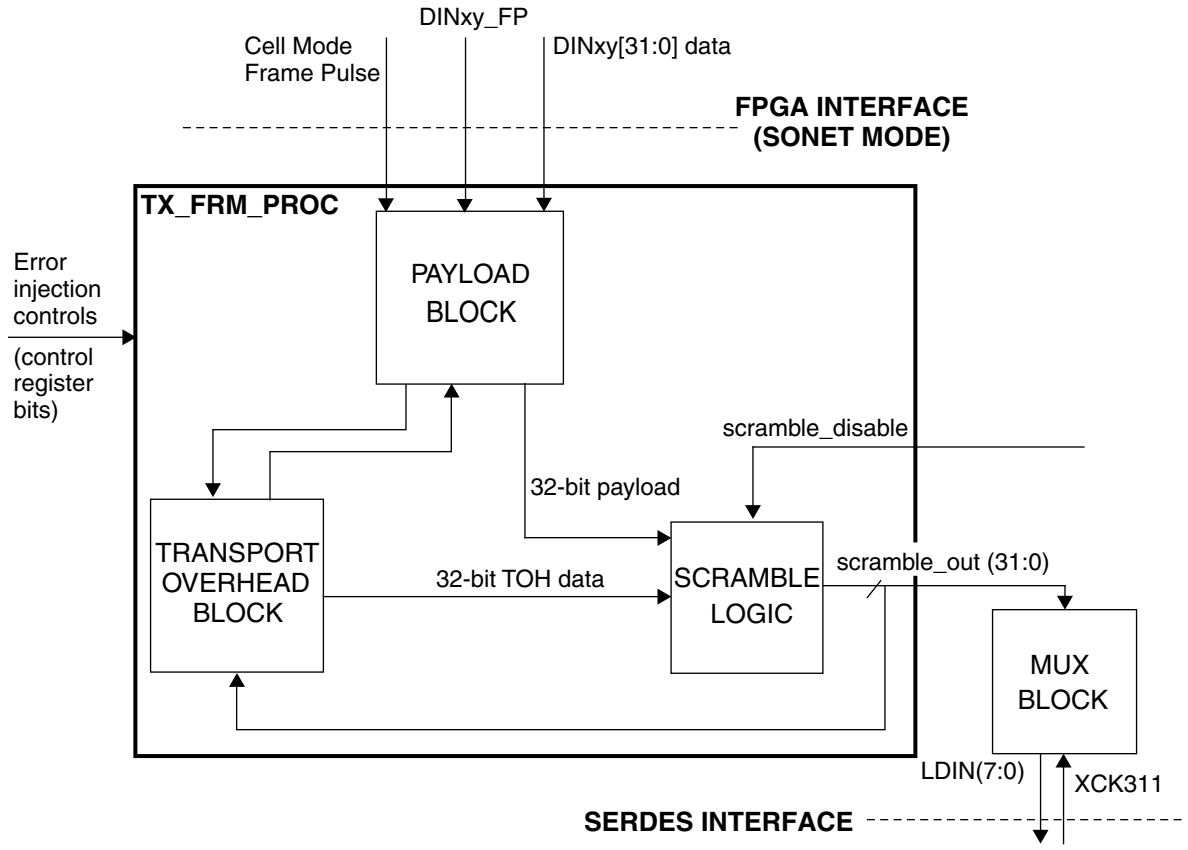
The following sections describe the data processing performed in the SONET logic blocks. The basic data flows in the SONET Mode are shown in Figure 11. At a top level, the descriptions are separated into processing in the transmit path (FPGA to serial link) and processing in the receive path (serial link to FPGA). In general, the descriptions in the next sections are written to describe SONET mode operation, although some of the "SONET logic blocks" are also used in cell mode operation. The various processing options are selected by setting bits in control registers and status information is written to status registers. Both types of registers can be written and/or read from the System Bus. Memory maps and descriptions for the registers are given in Table 21 through Table 36.

Figure 11. Basic Data Flows - SONET Mode



In the SONET mode, the transmit block receives 32-bit wide data from the FPGA (DINxx) on each of its channels along with a frame pulse (DINxx_FP) per channel and a transmit clock (TSYCLKxx). Typically this will represent a STS-48 stream on each link. The data are first passed through a TOH block which will generate all the timing pulses that are required to isolate individual overhead bytes (e.g., A1, A2, B1, D1-D3, etc.). The timing pulse gener-

Figure 16. TX Frame Processor (TFP) Block Diagram



Payload Sub-block

The Payload sub block is activated by the cell mode frame pulse (cell mode) or DINxx_FP from FPGA (SONET mode). A pulse on this signal indicates the start of a frame.

In SONET mode, only two types of data bytes are in each frame:

- TOH bytes
- SPE data bytes

There are N x 3 (N = 48) bytes of TOH per row and there are a total of 9 rows in a SONET frame. The rest of the bytes in each row are SPE data bytes in SONET mode.

TOH Sub-block

This block is responsible formatting the 144 (48 x 3) bytes of TOH at the beginning of each row of the transport frame. All TOH bytes may be transmitted transparently from the FPGA logic using the transparent mode. Alternately, some or all TOH bytes may be inserted by the TOH block (AUTO_SOH and AUTO_TOH mode). The TOH data is transferred across the FPGA/core interface as 32-bit words, hence 36 clock cycles (12 x 3) are needed to transfer a TOH row. TOH insertion is controlled by software register bits as shown in the Register Map tables.

frame (or A1A2 framing bytes). This frame pulse is used to synchronize multiple channels within an alignment group.

If a channel is not in any alignment group, the FIFO control logic will set the FIFO-write-address to the beginning of the FIFO, and will set the FIFO-read-address to the middle of the FIFO at the first assertion of frame pulse after reset or after the resync command.

The RX_FIFO_MIN register bits can be used to control the threshold for minimum unused buffer space in the alignment FIFOs between read and write pointers before OVFL status is flagged. The synchronization algorithm consists of a down counter which starts to count down by 1 from its initial value of 18 (decimal) when a frame pulse from any channel within an alignment group has been received. The OOS alarm indicates the FIFO is out-of sync and the channel skew exceeds that which can be handled by the FIFO. Once the frame pulse for all channels within the alignment group have been received, the count is decremented by 2 until 0 is reached. Data are then read from the FIFOs and output to the SPE generator before being sent to the FPGA.

For every alignment group, there is an OVFL and OOS status register bit. The OOS bit is flagged when the down counter in the synchronization algorithm has reached a value of 0 and frame pulse from all channels within an alignment group have not been received. The OVFL bit is flagged when the read address at the time of receiving a frame pulse, is less than the minimum threshold set by RX_FIFO_MIN. In the memory map section OOS is referred to as SYNC[2,4]_[A1,A2,B1,B2]_OOS, SYNC8_OOS. OVFL is referred to as SYNC[2,4]_[A1,A2,B1,B2]_OVFL, SYNC8_OVFL.

Receive Clocking for Multi-channel Alignment – ORSO82G5

There are a total of nine clocks for the receive path, from FPGA to the core. The four used in SONET mode are RSYSCLKA1 and RSYSCLKA2 (both for block A), and RSYSCLKB1 and RSYSCLKB2 (both for block B). The following diagrams show the recommended clock distribution approaches for SONET mode multi-channel alignment modes. Cell mode alignment is discussed in the section describing the Input Port Controller (IPC).)

SONET Mode Twin Alignment

Figure 27 describes the clocking scheme for twin alignment in the ORSO82G5. In twin alignment, the valid channel pairs are AA,AB and AC,AD in block A and BA,BB and BC,BD in block B. The figure provides the clocking scheme for block A as an example. RSYSCLKA1 should be sourced from RWCKAA or RWCKAB. RSYSCLKA2 should be sourced from RWCKAC or RWCKAD. This clocking approach provides the required 0 ppm clock frequency matching for each pair and provides flexibility in applications where the two pairs in the block are received from asynchronous sources or operate at different rates.

Figure 27. Receive Clocking Diagram for Twin Alignment in Block A – ORSO82G5

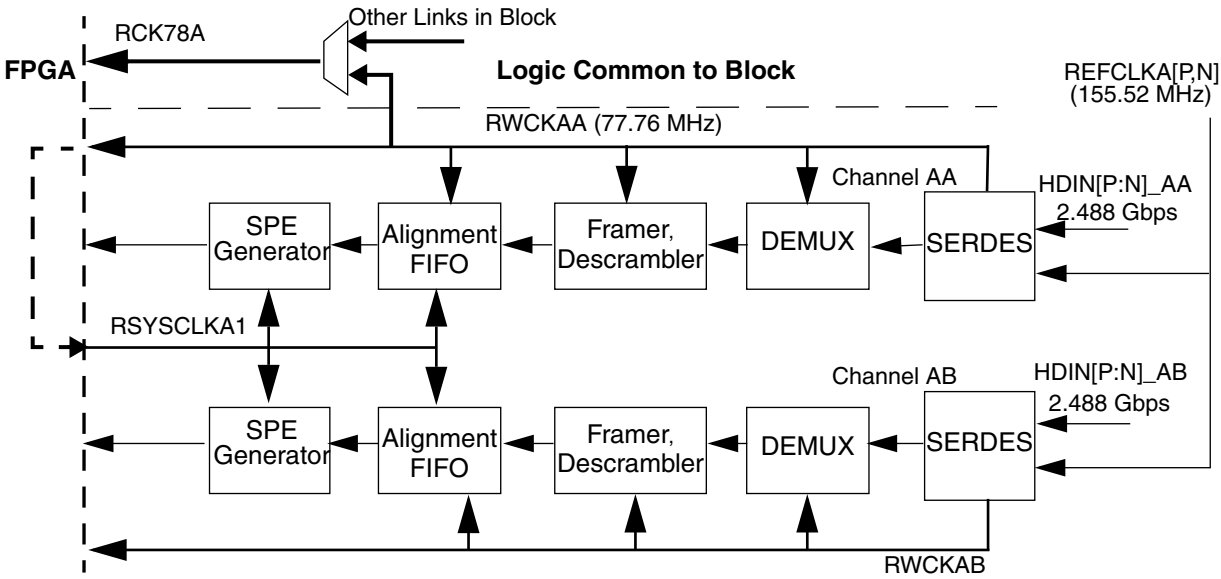
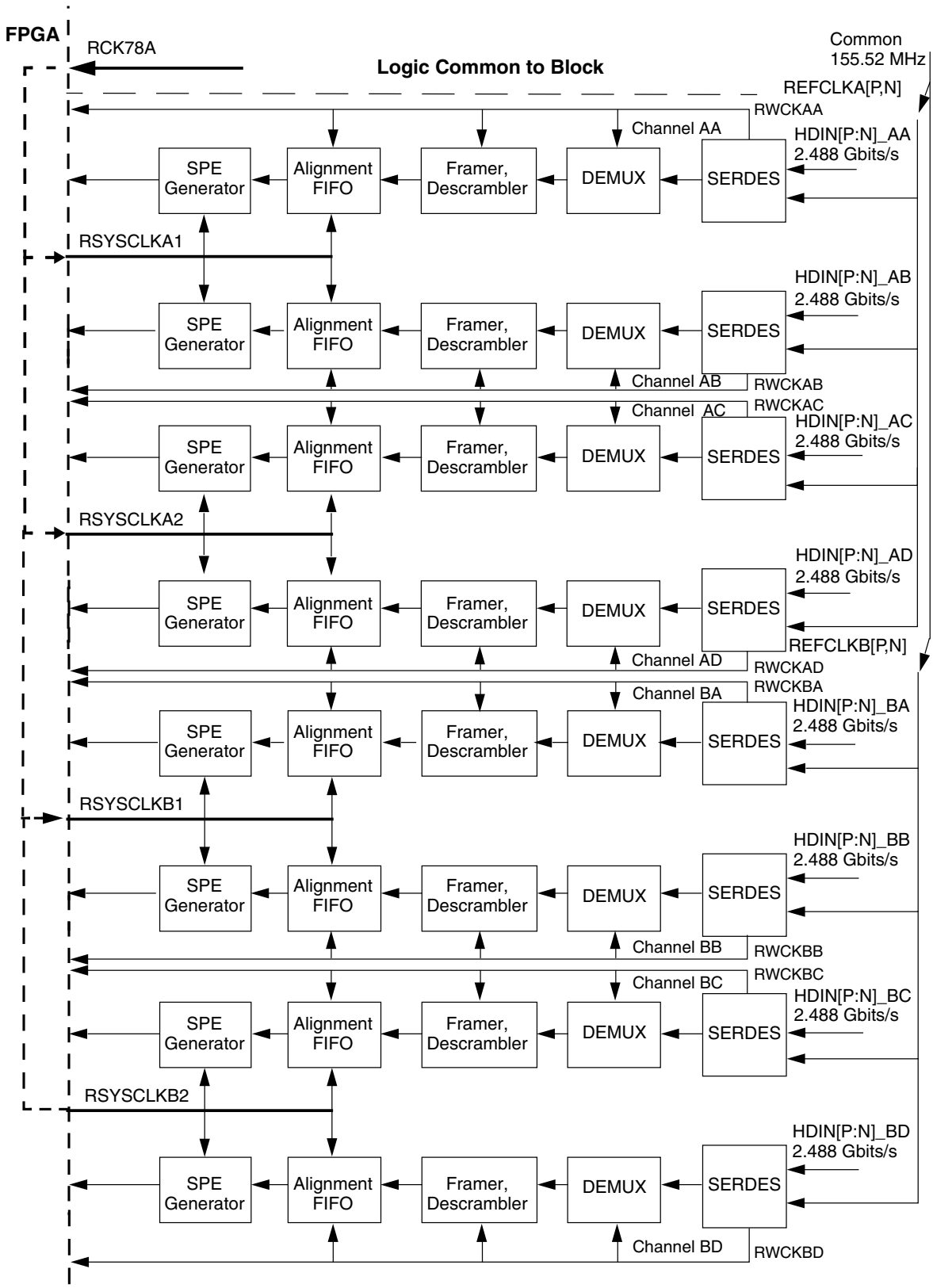


Figure 29. Receive Clocking Diagram for SONET Mode Eight-Channel Alignment – ORSO82G5



Link Header Detector

The Link Header detector determines when the next Link Header is coming in the frame and what the sequence number contained within it should be. If the value of the cell sequence counter is not equal to the expected value, then an error flag bit will set in the status registers and an error signal will be sent to the FPGA logic. The sequence counter will increment to the next sequence number. The sequence count value is NOT updated with the incorrect value, but is incremented each time a Link Header is received.

If excessive sequence errors are detected (three or more in a row), and the AUTO_REMOVE_[A:B] register bit is set, then the corresponding link will be treated as not valid. The RX_LINK_GOOD status bit will go low indicating the link is no longer receiving cells. If the AUTO_REMOVE_[A:B] register bit is not set, then the link is still valid when excessive sequence errors are detected. An OOF condition will also trigger the link to be removed from service if the AUTO_REMOVE_[A:B] register bit is set.

At startup or after a link has been removed from service (indicated by RX_LINK_GOOD going low), a link can be rejoined into the group. This is performed via the per block REJOIN_[A:B] register bit. When rejoining a link the RX FIFO will begin receiving cells. To cleanly rejoin a link into a group there are two methods to insure the RX FIFO begins loading correctly. The first method is to use the fast framing mode during the rejoin process. The can be done by setting the FFRM_EN_xx for the links that need to be rejoined before setting the REJOIN_[A:B] bit.

The second method is to issue a block reset to clear the FIFO once all links that have been selected to be rejoined are rejoined. This is done by first setting the REJOIN_[A:B] bit. Once the RX_LINK_GOOD status bit is high for the selected channels the GSWRST_[A:B] for the block should be set and cleared to reset the block. This method will disturb traffic on all links in the block during the GSWRST_[A:B] reset time.

Once all of the links in a group are rejoined and the traffic is again flowing the REJOIN_[A:B] bit should be cleared. If this bit is not cleared, a link may drop out using the AUTO_REMOVE mode and the channel may be rejoined incorrectly, causing errors on the entire group.

Receive FIFO

The main clock domain transfer for the data path is handled by the receive FIFO. A 16 x 161 FIFO is used in cell mode. The FIFO is implemented as a dual-port memory which will support simultaneous reads and writes. The receive FIFO block is written to at 77.76 MHz and read at 156 MHz.

The receive FIFO can allow for inter-link skew of about 800 ns (16 x 160 = 2560 bits, 400 ps per bit gives 1024 ns). The 160 LSBs in the memory are received data and the 161st bit indicates the start of a new cell. The FIFO write control logic indicates to the IPC, the start of a new frame of data. This signal will only be active for the A1 word of a frame.

Once frame synchronization has occurred and the IPC has responded with a FIFO enable signal, data will be written into the memory. Only the payload (cells) is written to the FIFO. The TOH bytes are not written into the FIFO. The cell octets immediately following the A1A2 bytes will be always written to the top of the FIFO.

Once a full cell has been written to the memory, the write control logic will send a control signal to the IPC8 or IPC2 block which will start the process of reading data from the FIFO. The IPC will read one whole cell at a time from each of the 8 FIFO blocks, if configured for the eight-link cell mode (ORSO82G5 only) or from each of 2 FIFO blocks if configured for the two-link cell mode.

A FIFO occupancy counter generates a RX_FIFO_OVRUN indication to the register interface if it detects a FIFO overflow condition. The cell mode allows for alignment of all eight-links or alignment of two-links. Thus there will be two IPC blocks for two pairs of channels per block.

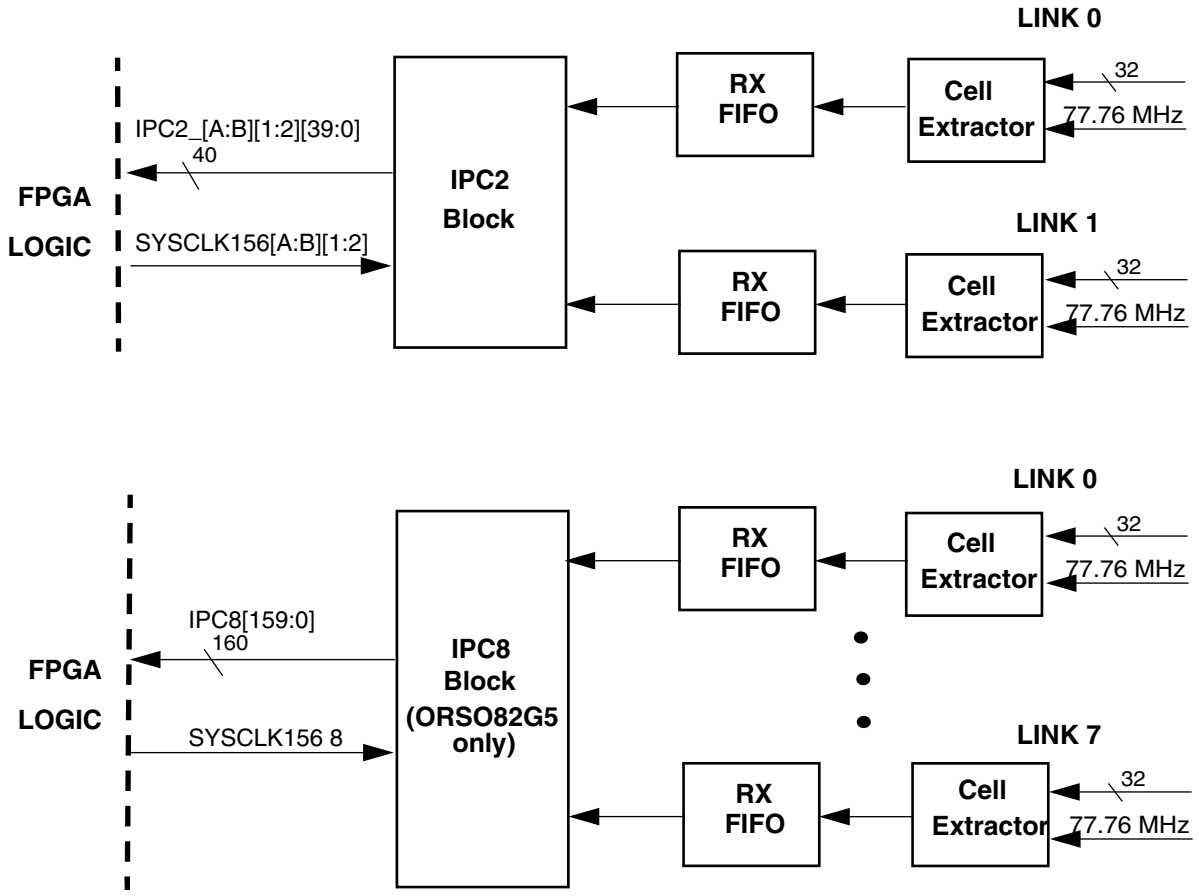
Input Port Controllers

The input port controllers (IPCs) are the block responsible for “directing traffic” for the receive traffic flow. The block diagrams for the 2-link and 8-link IPCs are shown in Figure 44. They provide the following essential functions.

- Determining when cell data can be read from the FIFOs of the individual links.

- Insuring group bundles are properly aligned.
- Scheduling reads from the RX FIFOs. Cells are read one at a time from the configured links.
- Parsing the cell data into payload data (along with selected header information). Cells which have errors that make them unusable (such as BIP or sequence number errors) are thrown away. This dropping of errored cells can be disabled through register bits CELL_BIP_INH_xx and CELL_SEQ_INX_xx.

Figure 44. IPC2 and IPC8 Block Diagrams



There are 5 IPC blocks in the embedded core. There is an IPC2 block for every channel pair:

- IPC2_A1 combines links from channels AA,AB (ORSO82G5 only)
- IPC2_A2 combines links from channels AC,AD
- IPC2_B1 combines links from channels BA,BB (ORSO82G5 only)
- IPC2_B2 combines links from channels BC,BD

The IPC8 block combines cells from all eight aligned links and transmits them to the FPGA logic (ORSO82G5 only).

Before an IPC can begin reading data from the Rx FIFOs and assembling cells, it must first align all FIFOs in a port bundle. This is accomplished by handshaking signals between the framer and the IPC. The framer indicates to the IPC that framing has been acquired. The framer does not start filling the FIFOs, however, until the next A1/A2 SONET signal.

Reference Clock Requirements

There are two pairs of reference clock inputs in the ORSO42G5 and ORSO82G5 devices. Each reference clock is distributed to all channels in a block. Each channel has a differential buffer to isolate the clock from the other channels. The input clock is preferably a differential signal; however, the device can operate with a single-ended input. The input reference clock directly impacts the transmit data eye, so the clock should have low jitter. In particular, jitter components in the DC to 5 MHz range should be minimized. The required electrical characteristics for the reference clock are given in Table 46.

Synthesized and Recovered Clocks

The SERDES Embedded Core block contains its own dedicated PLLs for transmit and receive clock generation. The user provides a reference clock of the appropriate frequency, as described in the previous section. The transmitter PLL uses the REFCLK_[A,B] inputs to synthesize the internal high-speed serial bit clocks. The receiver PLLs extract the clock from the serial input data and retime the data with the recovered clock.

The receive PLL for each channel has two modes of operation - lock to reference and lock to data with retiming. When no data or invalid data is present on the HDINP_xx and HDINN_xx pins, the receive VCO will not lock to data and its frequency can drift outside of the nominal ±100 ppm range. Under this condition, the receive PLL will lock to REFCLK_[A,B] for a fixed time interval and then will attempt to lock to receive data. The process of attempting to lock to data, then locking to clock will repeat until valid input data exists. There is also a control register bit per channel to force the receive PLL to always lock to the reference clock.

The high-speed transmit and receive serial data links can run at 0.6 to 2.7 Gbps, depending on the frequency of the reference clock and the state of the control bits from the internal transmit control register. The interface to the serializer/deserializer block runs at 1/8th the bit rate of the data lane. Additionally, the MUX/DEMUX logic converts the data rate and bit-width so the FPGA core can run at 1/4th this frequency which gives a range of 18.8 to 84.4 MHz for the data in and out of the FPGA in SONET mode. In cell mode, there is a clock domain transfer to a 2x clock domain, which gives a range of 37.5 to 168.8 MHz for the data in and out of the FPGA.

Internal Clock Signals at the FPGA/Core Interface

There are several clock signals defined at the FPGA/Embedded Core interface in addition to the external reference clock for each SERDES block. All of the ORSO42G5 and ORSO82G5 clock signals are shown in Figure 47 and are described following the figure.

Table 26. Per-Channel Control Register Descriptions – ORSO42G5 (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30823 - AC 30833 - AD 30923 - BC 30933 - BD	[0]	BYPASS_ALGN_FIFO_xx	00	Bypass alignment FIFO =1 in RX path in SONET mode. DOUT_xx data is clocked off RWCKxx.	SONET
	[1]	SERDES_ONLY_MODE_xx		SERDES-Only Mode. SERDES_ONLY_MODE =1 bypasses all the SONET and cell functions. Data is sent directly from the SERDES to the FPGA logic in the RX path and is passed directly from the FPGA logic to the SERDES in the TX path.	SERDES Only
	[2]	FORCE_B1_ERR_xx		Force B1 Error. FORCE_B1_ERR =1 forces a Section B1 error (Bit Interleaved Parity Error) in the TX path. Valid only when the corresponding AUTO_TOH_xx bit or AUTO_B1_xx bit is set to one.	Both
	[3]	FORCE_BIP8_ERR_xx		Force BIP8 Error, FORCE_BIP8_ERR =1 forces cell BIP8 error in the TX path.	Cell
	[4]	FORCE_A1A2_ERR_xx		Force A1A2 Error, FORCE_A1A2_ERR =1 forces an error in A1A2 bytes (framing error) in the TX path. Valid only when the corresponding AUTO_TOH_xx bit or AUTO_A1A2_xx bit is set to one.	Both
	[5]	FORCE_EX_SEQ_ERR_xx		Force Excessive Sequence Errors. FORCE_EXP_SEQ_ERR_xx = 1 forces sequence errors in the TX path by inverting the link header byte sequence number.	Cell
	[6]	FORCE_SEQ_ERR_xx		Force Sequence Error, FORCE_SEQ_ERR =1 forces one sequence error in the TX path. After this bit is set, the next Link Header byte's sequence number is inverted. The Link Header after the errored Link Header byte will have the correct sequence number	Cell
	[7]	FORCE_RDI_xx		Force Remote Defect Indication, FORCE_RDI =1 forces RDI errors in the TX path. The K2 byte in TOH is set to "00000110. Valid only when AUTO_TOH_xx bit is set to 1.	Cell

Table 28. Common Control Register Descriptions – ORSO42G5

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30A00	[0:1]	RCKSELB	00	“10” - Channel BC source for clock RCK78B “11” - Channel BD source for clock RCK78B	Both
	[2:3]	TCKSELB		“10” - Channel BC source for clock TCK78B “11” - Channel BD source for clock TCK78B	Both
	[4:5]	RCKSELA		“10” - Channel AC source for clock RCK78A “11” - Channel AD source for clock RCK78A	Both
	[6:7]	TCKSELA		“10” - Channel AC source for clock TCK78A “11” - Channel AD source for clock TCK78A	Both
30A01	[0:2]	CELL_SIZE	00	Cell Size, Three bits to set cell size. “000” - Cell size is 75 bytes, “001” - Cell size is 79 bytes, “010” - Cell size is 83 bytes, “011” - Cell size is 91 bytes These are the only supported cell sizes.	Cell
	[3:7]	RX_FIFO_MIN		Set Minimum threshold value for alignment FIFO in SONET mode. When the read address for the FIFO is below this value at the time when write address is zero, it indicates that the FIFO is near overflow. This event will go high only once during a frame when a framing byte has been detected by the aligner. The default threshold value is “00000”.	SONET
30A02	0	TX_DISABLE_ON_RDI	00	Transmitter Disable on RDI (Detection), If TX_DISABLE_ON_RDI = 1 - No cell data is transmitted on a link in which a RDI has been detected by the corresponding link’s receiver. If this bit is set to 0, cell data will be transmitted on a link irrespective of detection of a RDI.	Cell
	[1:7]	RSVD		Reserved	—

Table 28. Common Control Register Descriptions – ORSO42G5 (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30A05	[0:2]	ERRCNT_CH	00	Error Count Channel Select, Control bits to select which channel's Section B1 error and Cell BIP error counts are recorded by the BIP_ERR_CNT and CELL_BIP_ERR_CNT registers. "010" - Channel AC, "011" - Channel AD, "110" - Channel BC, "111" - Channel BD	Both
	[3]	RSVD		Reserved	—
	[4]	CELL_MODE_A2		Cell Mode Enable, CELL_MODE_A2 = 1 enables cell mode for the channel group AC and AD.	Cell
	[5]	RSVD		Reserved	—
	[6]	CELL_MODE_B2		Cell Mode Enable, CELL_MODE_B2 = 1 enables cell mode for the channel group BC and BD.	Cell
	[7]	RSVD		Reserved	—
30A06	[0:4]	RSVD	00	Reserved	—
	[5:6]	RESET_PHASE		Reset Phase, Two bits to select delay phase for delaying the soft reset bit SOFT_RESET with respect to the synchronizing clock. Four delay phases can be selected through the values "00", "01", "10" and "11".	Both
	[7]	SOFT_RESET		Soft Reset, SOFT_RESET=1 resets the embedded core flip flops except for the software registers. This bit does not affect the state of the registers inside the SERDES blocks.	Both
30A07	[0:6]	RSVD	00	Reserved	—
	[7]	TX_CFG_DONE		Transmitter Configuration Done, Edge sensitive bit to indicate that all TX configuration bits are set. After all register bits have been set for Transmit direction, write a 0 and then a 1 to this bit.	Cell
30A08	[0]	ALARM_STATUS_BD	00	Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel BD. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[1]	ALARM_STATUS_BC		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel BC. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[2:3]	RSVD		Reserved	—
	[4]	ALARM_STATUS_AD		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel AD. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[5]	ALARM_STATUS_AC		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel AC. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[6:7]	RSVD		Reserved	—

Table 35. Per-Channel Status Register Descriptions – ORSO82G5 (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
3080B - AA 3081B - AB 3082B -AC 3083B - AD 3090B - BA 3091B - BB 3092B - BC 3093B - BD	[0]	RSVD	00	Reserved	—
	[1]	STAT_OOF_xx		STAT_OOF_xx = 1 same as OOF_xx except that a 1 on this bit can NOT cause an interrupt	Both
	[2]	STAT_EX_SEQ_ERR_xx		STAT_EX_SEQ_ERR_xx = 1 same as EX_SEQ_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[3]	STAT_SEQ_ERR_xx		STAT_SEQ_ERR_xx = 1 same as SEQ_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[4]	STAT_CELL_BIP_ERR_xx		STAT_CELL_BIP_ERR_xx = 1 same as CELL_BIP_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[5]	STAT_B1_ERR_xx		STAT_B1_ERR_xx = 1 same as B1_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Both
	[6]	STAT_RX_FIFO_OVRUN_xx		STAT_RX_FIFO_OVRUN_xx = 1 same as RX_FIFO_OVRUN_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[7]	STAT_RDI_xx		STAT_RDI_xx = 1 same as RDI_xx except that a 1 on this bit can NOT cause an interrupt	Both
3080C - AA 3081C - AB 3082C - AC 3083C - AD 3090C - BA 3091C - BB 3092C - BC 3093C - BD	[0:7]	LINK_NUM_RX_xx	00	Link number received that is stored in the F1 byte position of the SONET TOH	Both
3080E - AA 3081E - AB 3082E - AC 3083E AD 3090E - BA 3091E - BB 3092E- BC 3093E - BD	[0:5]	RSVD	00	Reserved	—
	[6]	CH248_SYNC_xx		CH248_SYNC_xx = 1 indicates that A1A2 bytes have been detected by link xx for multi-channel alignment	SONET
	[7]	RX_LINK_GOOD_xx		RX_LINK_GOOD_xx = 1 indicates that frame has been acquired and the link has been stable. Goes high at an A1A2 boundary but can go low at any point in a frame due to excessive errors	Cell

Table 47. Pin Descriptions (Continued)

Symbol	I/O	Description
P[TBLR]CLK[1:0][TC]	I	Pins dedicated for the primary clock. Input pins on the middle of each side with differential pairing.
	I/O	After configuration these pins are user programmable I/O, if not used for clock inputs.
TDI, TCK, TMS	I	If boundary-scan is used, these pins are test data in, test clock, and test mode select inputs. If boundary-scan is not selected, all boundary-scan functions are inhibited once configuration is complete. Even if boundary-scan is not used, either TCK or TMS must be held at logic 1 during configuration. Each pin has a pull-up enabled during configuration.
	I/O	After configuration, these pins are user-programmable I/O if boundary scan is not used.[1]
RDY/$\overline{\text{BUSY}}$/RCLK	O	During configuration in asynchronous peripheral mode, RDY/RCLK indicates another byte can be written to the FPGA. If a read operation is done when the device is selected, the same status is also available on D7 in asynchronous peripheral mode. During the master parallel configuration mode, RCLK is a read output signal to an external memory. This output is not normally used.
	I/O	After configuration this pin is a user-programmable I/O pin.[1]
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
	I/O	After configuration, this pin is a user-programmable I/O pin.[1]
$\overline{\text{LDC}}$	O	$\overline{\text{Low During Configuration}}$ is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
	I/O	After configuration, this pin is a user-programmable I/O pin.[1]
$\overline{\text{INIT}}$	I/O	$\overline{\text{INIT}}$ is a bidirectional signal before and during configuration. During configuration, a pull-up is enabled, but an external pull-up resistor is recommended. As an active-low open-drain output, $\overline{\text{INIT}}$ is held low during power stabilization and internal clearing of memory. As an active-low input, $\overline{\text{INIT}}$ holds the FPGA in the wait-state before the start of configuration. After configuration, this pin is a user-programmable I/O pin.[1]
$\overline{\text{CS0}}$, CS1	I	$\overline{\text{CS0}}$ and CS1 are used in the asynchronous peripheral, slave parallel, and microprocessor configuration modes. The FPGA is selected when $\overline{\text{CS0}}$ is low and CS1 is high. During configuration, a pull-up is enabled.
	I/O	After configuration, if MPI is not used, these pins are user-programmable I/O pins.[1]
$\overline{\text{RD}}$/$\overline{\text{MPI\_STRB}}$	I	$\overline{\text{RD}}$ is used in the asynchronous peripheral configuration mode. A low on $\overline{\text{RD}}$ changes D[7:3] into a status output. $\overline{\text{WR}}$ and $\overline{\text{RD}}$ should not be used simultaneously. If they are, the write strobe overrides. This pin is also used as the MPI data transfer strobe. As a status indication, a high indicates ready, and a low indicates busy.
$\overline{\text{WR}}$/MPI_RW	I	$\overline{\text{WR}}$ is used in asynchronous peripheral mode. A low on $\overline{\text{WR}}$ transfers data on D[7:0] to the FPGA. In MPI mode, a high on MPI_RW allows a read from the data bus, while a low causes a write transfer to the FPGA.
	I/O	After configuration, if the MPI is not used, $\overline{\text{WR}}$/MPI_RW is a user-programmable I/O pin.[1]
PPC_A[14:31]	I	During MPI mode the PPC_A[14:31] are used as the address bus driven by the PowerPC bus master utilizing the least-significant bits of the PowerPC 32-bit address.
$\overline{\text{MPI\_BURST}}$	I	$\overline{\text{MPI\_BURST}}$ is driven low to indicate a burst transfer is in progress in MPI mode. Driven high indicates that the current transfer is not a burst.
$\overline{\text{MPI\_BDIP}}$	I	$\overline{\text{MPI\_BDIP}}$ is driven by the PowerPC processor in MPI mode. Assertion of this pin indicates that the second beat in front of the current one is requested by the master. Negated before the burst transfer ends to abort the burst data phase.
MPI_TSZ[0:1]	I	MPI_TSZ[0:1] signals are driven by the bus master in MPI mode to indicate the data transfer size for the transaction. Set 01 for byte, 10 for half-word, and 00 for word.
A[21:0]	O	During master parallel mode A[21:0] address the configuration EPROMs up to 4M bytes.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.[1]

Table 51. ORSO42G5 484-pin PBGAM Pinout (Continued)

484-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	484-PBGAM
G16	-	-	VDD15	VDD15	-	-
H13	-	-	VSS	VSS	-	-
AB10	5 (BC)	2	IO	PB19A	-	L37T
AA10	5 (BC)	2	IO	PB19B	-	L37C
W10	5 (BC)	2	IO	PB19C	PBCK0T	L38T
Y10	5 (BC)	2	IO	PB19D	PBCK0C	L38C
V10	5 (BC)	2	IO	PB20A	-	-
U13	5 (BC)	-	VDDIO5	VDDIO5	-	-
AB11	5 (BC)	2	IO	PB20C	VREF_5_02	L39T
AA11	5 (BC)	2	IO	PB20D	-	L39C
U10	5 (BC)	2	IO	PB21A	-	-
H6	-	-	VDD15	VDD15	-	-
Y11	5 (BC)	3	IO	PB21C	-	L40T
W11	5 (BC)	3	IO	PB21D	VREF_5_03	L40C
U11	5 (BC)	3	IO	PB22A	-	-
J7	-	-	VSS	VSS	-	-
AB12	5 (BC)	3	IO	PB22C	-	L41T
AA12	5 (BC)	3	IO	PB22D	-	L41C
U12	5 (BC)	3	IO	PB23A	-	-
Y12	5 (BC)	3	IO	PB23C	PBCK1T	L42T
W12	5 (BC)	3	IO	PB23D	PBCK1C	L42C
V11	5 (BC)	3	IO	PB24A	-	-
J8	-	-	VSS	VSS	-	-
AB13	5 (BC)	4	IO	PB24C	-	L43T
AA13	5 (BC)	4	IO	PB24D	-	L43C
V12	5 (BC)	4	IO	PB25A	-	-
U14	5 (BC)	-	VDDIO5	VDDIO5	-	-
AB14	5 (BC)	4	IO	PB25C	-	L44T
AA14	5 (BC)	4	IO	PB25D	VREF_5_04	L44C
J9	-	-	VSS	VSS	-	-
Y13	5 (BC)	5	IO	PB26C	-	L45T
W13	5 (BC)	5	IO	PB26D	VREF_5_05	L45C
U15	5 (BC)	-	VDDIO5	VDDIO5	-	-
AB15	5 (BC)	5	IO	PB27C	-	L46T
AA15	5 (BC)	5	IO	PB27D	-	L46C
AB16	5 (BC)	6	IO	PB28C	-	L47T
AA16	5 (BC)	6	IO	PB28D	VREF_5_06	L47C
H14	-	-	VDD15	VDD15	-	-
Y14	5 (BC)	6	IO	PB29C	-	L48T
W14	5 (BC)	6	IO	PB29D	-	L48C
J10	-	-	VSS	VSS	-	-
AB17	5 (BC)	7	IO	PB30C	-	L49T
AA17	5 (BC)	7	IO	PB30D	-	L49C
U16	5 (BC)	-	VDDIO5	VDDIO5	-	-

Table 51. ORSO42G5 484-pin PBGAM Pinout (Continued)

484-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	484-PBGAM
K13	-	-	VSS	VSS	-	-
B21	1 (TC)	9	IO	PT31D	-	L55C
A21	1 (TC)	9	IO	PT31C	VREF_1_09	L55T
E13	1 (TC)	-	VDDIO1	VDDIO1	-	-
D14	1 (TC)	9	IO	PT30D	-	L56C
C14	1 (TC)	9	IO	PT30C	-	L56T
K14	-	-	VSS	VSS	-	-
B20	1 (TC)	9	IO	PT29D	-	L57C
A20	1 (TC)	9	IO	PT29C	-	L57T
N7	-	-	VDD15	VDD15	-	-
B19	1 (TC)	1	IO	PT28D	-	L58C
A19	1 (TC)	1	IO	PT28C	-	L58T
L8	-	-	VSS	VSS	-	-
D13	1 (TC)	1	IO	PT27D	VREF_1_01	L59C
C13	1 (TC)	1	IO	PT27C	-	L59T
E18	1 (TC)	-	VDDIO1	VDDIO1	-	-
A18	1 (TC)	1	IO	PT27B	-	L60C
B18	1 (TC)	1	IO	PT27A	-	L60T
A17	1 (TC)	2	IO	PT26D	-	L61C
B17	1 (TC)	2	IO	PT26C	VREF_1_02	L61T
L9	-	-	VSS	VSS	-	-
D12	1 (TC)	2	IO	PT25D	-	L62C
C12	1 (TC)	2	IO	PT25C	-	L62T
E19	1 (TC)	-	VDDIO1	VDDIO1	-	-
A16	1 (TC)	3	IO	PT24D	-	L63C
B16	1 (TC)	3	IO	PT24C	VREF_1_03	L63T
A15	1 (TC)	3	IO	PT23D	-	L64C
B15	1 (TC)	3	IO	PT23C	-	L64T
F16	1 (TC)	-	VDDIO1	VDDIO1	-	-
E11	1 (TC)	3	IO	PT22D	-	-
L10	-	-	VSS	VSS	-	-
D11	1 (TC)	4	IO	PT21D	-	L65C
C11	1 (TC)	4	IO	PT21C	-	L65T
A14	1 (TC)	4	IO	PT20D	-	L66C
B14	1 (TC)	4	IO	PT20C	-	L66T
A13	1 (TC)	4	IO	PT19D	-	L67C
B13	1 (TC)	4	IO	PT19C	VREF_1_04	L67T
G14	1 (TC)	-	VDDIO1	VDDIO1	-	-
L11	-	-	VSS	VSS	-	-
N15	-	-	VDD15	VDD15	-	-
D10	1 (TC)	5	IO	PT18D	PTCK1C	L68C
C10	1 (TC)	5	IO	PT18C	PTCK1T	L68T
A12	1 (TC)	5	IO	PT17D	PTCK0C	L69C
B12	1 (TC)	5	IO	PT17C	PTCK0T	L69T

Table 52. ORSO82G5 680-Pin PBGAM (fpBGA) Pinout (Continued)

680-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	680-PBGAM
D7	0 (TL)	—	VDDIO0	VDDIO0	—	—
C14	0 (TL)	1	IO	PT13B	—	L2C_A0
B14	0 (TL)	1	IO	PT13A	VREF_0_01	L2T_A0
A14	0 (TL)	1	IO	PT12D	M0	L3C_A0
A13	0 (TL)	1	IO	PT12C	M1	L3T_A0
AA20	—	—	Vss	Vss	—	—
E12	0 (TL)	2	IO	PT12B	MPI_CLK	L4C_A0
E13	0 (TL)	2	IO	PT12A	A21/MPI_BURST_N	L4T_A0
C13	0 (TL)	2	IO	PT11D	M2	L5C_A0
C12	0 (TL)	2	IO	PT11C	M3	L5T_A0
B12	0 (TL)	2	IO	PT11B	VREF_0_02	L6C_A0
A12	0 (TL)	2	IO	PT11A	MPI_TEA_N	L6T_A0
D12	0 (TL)	3	IO	PT10D	—	L7C_D0
C11	0 (TL)	3	IO	PT10C	—	L7T_D0
B11	0 (TL)	3	IO	PT10B	—	—
A11	0 (TL)	3	IO	PT9D	VREF_0_03	L8C_A0
A10	0 (TL)	3	IO	PT9C	—	L8T_A0
AA21	—	—	Vss	Vss	—	—
B10	0 (TL)	3	IO	PT9B	—	—
E11	0 (TL)	3	IO	PT8D	D0	L9C_D0
D10	0 (TL)	3	IO	PT8C	TMS	L9T_D0
C10	0 (TL)	3	IO	PT8B	—	—
A9	0 (TL)	4	IO	PT7D	A20/MPI_BDIP_N	L10C_A0
B9	0 (TL)	4	IO	PT7C	A19/MPI_TSZ1	L10T_A0
AA22	—	—	Vss	Vss	—	—
E10	0 (TL)	4	IO	PT7B	—	—
A8	0 (TL)	4	IO	PT6D	A18/MPI_TSZ0	L11C_A0
B8	0 (TL)	4	IO	PT6C	D3	L11T_A0
D9	0 (TL)	4	IO	PT6B	VREF_0_04	L12C_D0
C8	0 (TL)	4	IO	PT6A	—	L12T_D0
E9	0 (TL)	5	IO	PT5D	D1	L13C_D0
D8	0 (TL)	5	IO	PT5C	D2	L13T_D0
AB13	—	—	Vss	Vss	—	—
A7	0 (TL)	5	IO	PT5B	—	L14C_A0
A6	0 (TL)	5	IO	PT5A	VREF_0_05	L14T_A0
C7	0 (TL)	5	IO	PT4D	TDI	L15C_D0
B6	0 (TL)	5	IO	PT4C	TCK	L15T_D0
E8	0 (TL)	5	IO	PT4B	—	L16C_A0
E7	0 (TL)	5	IO	PT4A	—	L16T_A0
A5	0 (TL)	6	IO	PT3D	—	L17C_A0
B5	0 (TL)	6	IO	PT3C	VREF_0_06	L17T_A0
AB14	—	—	Vss	Vss	—	—
C6	0 (TL)	6	IO	PT3B	—	L18C_A0
D6	0 (TL)	6	IO	PT3A	—	L18T_A0