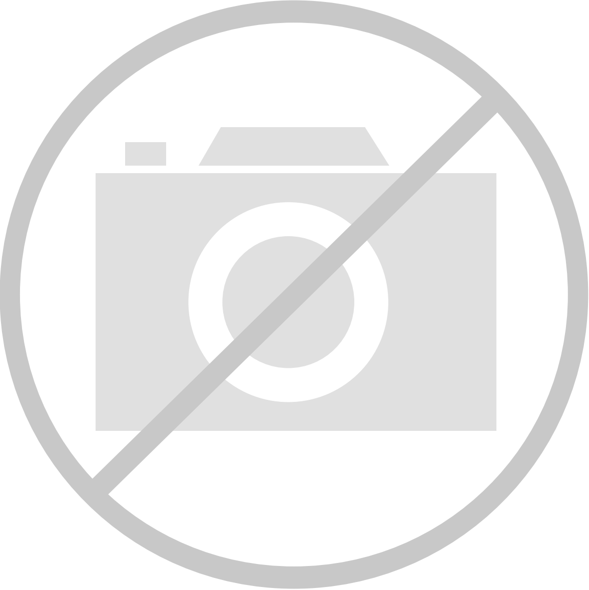


Welcome to E-XFL.COM

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	372
Number of Gates	643000
Voltage - Supply	1.425V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	680-BBGA
Supplier Device Package	680-FPBGA (35x35)
Purchase URL	https://www.e-xfl.com/product-detail/lattice-semiconductor/ort82g5-2fn680c

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 at rates up to 350 MHz (700 MHz effective rate).
  - New 2x and 4x downlink and uplink capability per I/O (i.e., 50 MHz internal to 200 MHz 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- 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.
- Abundant high-speed buffered and nonbuffered routing resources provide 2x average speed improvements over previous architectures.
- Hierarchical routing optimized for both local and global routing with dedicated routing resources. This results in faster routing times with predictable and efficient performance.
- SLIC provides eight 3-statable buffers, up to a 10-bit decoder, and PAL®-like AND-OR-Invert (AOI) in each programmable logic cell.
- New 200 MHz embedded block-port RAM blocks, two read ports, two write ports, and two sets of byte lane enables. Each embedded RAM block can be configured as:

Description

What is an FPSC?

FPSCs, or field-programmable system chips, are devices that combine field-programmable logic with ASIC or mask-programmed logic on a single device. FPSCs provide the time to market and the flexibility of FPGAs, the design effort savings of using soft Intellectual Property (IP) cores, and the speed, design density, and economy of ASICs.

FPSC Overview

Lattice's Series 4 FPSCs are created from Series 4 ORCA FPGAs. To create a Series 4 FPSC, several columns of Programmable Logic Cells (see FPGA Logic Overview section for FPGA logic details) are added to an embedded logic core. Other than replacing some FPGA gates with ASIC gates, at greater than 10:1 efficiency, none of the FPGA functionality is changed—all of the Series 4 FPGA capability is retained including: the Embedded Block RAMs, MicroProcessor Interface (MPI), boundary scan, etc. The columns of programmable logic are replaced at the right of the device, allowing pins from the replaced columns to be used as I/O pins for the embedded core. The remainder of the device pins retain their FPGA functionality.

FPSC Gate Counting

The total gate count for an FPSC is the sum of its embedded core (standard-cell/ASIC gates) and its FPGA gates. Because FPGA gates are generally expressed as a usable range with a nominal value, the total FPSC gate count is sometimes expressed in the same manner. Standard-cell ASIC gates are, however, 10 to 25 times more silicon-area efficient than FPGA gates. Therefore, an FPSC with an embedded function is gate equivalent to an FPGA with a much larger gate count.

FPGA/Embedded Core Interface

The interface between the FPGA logic and the embedded core has been enhanced to allow for a greater number of interface signals than on previous FPSC architectures. Compared to bringing embedded core signals off-chip, this on-chip interface is much faster and requires less power. All of the delays for the interface are precharacterized and accounted for in the Lattice ispLEVER™ System software.

Series 4 based FPSCs expand this interface by providing a link between the embedded block and the multi-master 32-bit system bus in the FPGA logic. This system bus allows the core easy access to many of the FPGA logic functions including the Embedded Block RAMs and the microprocessor interface.

Clock spines also can pass across the FPGA/embedded core boundary. This allows for fast, low-skew clocking between the FPGA and the embedded core. Many of the special signals from the FPGA, such as DONE and global set/reset, are also available to the embedded core, making it possible to fully integrate the embedded core with the FPGA as a system.

For even greater system flexibility, FPGA configuration RAMs are available for use by the embedded core. This allows for user-programmable options in the embedded core, in turn allowing for greater flexibility. Multiple embedded core configurations may be designed into a single device with user-programmable control over which configurations are implemented, as well as the capability to change core functionality simply by reconfiguring the device.

FPSC Design Kit

Development is facilitated by an FPSC design kit which, together with ispLEVER System software and third-party synthesis and simulation engines, provides all software and documentation required to design and verify an FPSC implementation. Included in the kit are the FPSC configuration manager, complied Verilog simulation models, HSPICE and/or IBIS models for I/O buffers, and complete online documentation. The kit's software coupled with the design environment, provides a seamless FPSC design environment. More information can be obtained by visiting the Lattice web site at www.latticesemi.com or contacting a local sales office.

System Bus

An on-chip, multimaster, 8-bit system bus with 1-bit parity facilitates communication among the MPI, configuration logic, FPGA control, status registers, Embedded Block RAMs, as well as user logic. Utilizing the AMBA specification Rev 2.0 AHB protocol, 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 device.

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. 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 clocks from 20 MHz to 200 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 multiple of an 8-bit number with a 16-bit fixed coefficient or vice versa (24-bit output), or a multiple 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.

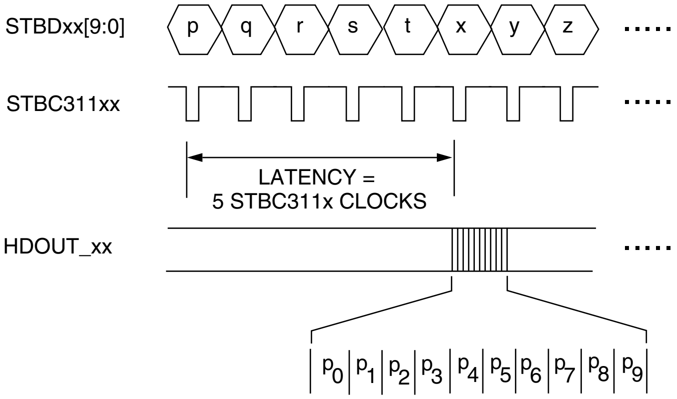
Configuration

The FPGAs functionality is determined by internal configuration RAM. The FPGAs internal initialization/configuration circuitry loads the configuration data at power up 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, the 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).

Figure 5. Transmit Path Timing - Single SERDES Channel



Each block also sends a clock to the FPGA logic. This clock, TCK78[A,B], is sourced from one of the four MUX blocks and has the same frequency as TSYS_CLK_xx, but arbitrary phase. Within each MUX block, the low frequency clock output is obtained by dividing by 4 the SERDES STBC311x clock which is used internally to synchronize the transmit data words. TCKSEL control bits select the channel to source TCK78[A:B].

The internal signals STBDxx[9:0] (where xx is represents AA...BD or AC, AD, BC, BD) from the MUX block carry unencoded character data and control bits. The 10th bit (STBDxx[9]) of each data lane into the SERDES is used to force a negative disparity present state.

8b/10b Encoder and 1:10 Multiplexer

The 8b/10b encoder encodes the incoming 8-bit data into a 10-bit format as described previously. The input signals to the block, STBDxx[7:0] are used for the 8-bit unencoded data. STBDxx[8] is used as the K_control input to indicate whether the 8 data bits need to be encoded as special characters (K_control = 1) or as data characters (K_control = 0). When STBDxx[9:0] = 1, a negative disparity present state is forced. When the encoder is bypassed STBDxx[9:0] serve as the data bits for the 10-bit unencoded data.

Within the definition of the 8b/10b transmission code, the bit positions of the 10-bit encoded transmission characters are labeled as a, b, c, d, e, i, f, g, h, and j in that order. Bit a corresponds to STBDxx[0], bit b to STBDxx[1], bit c to STBDxx[2], bit d to STBDxx[3], bit e to STBDxx[4], bit i to STBDxx[5], bit f to STBDxx[6], bit g to STBDxx[7], bit h to STBDxx[8], and bit j to STBDxx[9].

The 10-bit wide parallel data is converted to serial data by the 10:1 Multiplexer. The serial data are then sent to the CML output buffer and are transmitted serially with STBDxx[0] transmitted first and STBDxx[9] transmitted last.

CML Output Buffer

The transmitter's CML output buffer is terminated on-chip in 86 ohms to optimize the data eye as well as to reduce the number of discrete components required. The differential output swing reaches a maximum of 1.2 VPP in the normal amplitude mode. A half amplitude mode can be selected via configuration register bit HAMP_xx. Half amplitude mode can be used to reduce power dissipation when the transmission medium has minimal attenuation or for testing of the integrity (loss) of the physical medium.

A programmable preemphasis circuit is provided to boost the high frequencies in the transmit data signal to maximize the data eye opening at the far-end receiver. Preemphasis is particularly useful when the data are transmitted over backplanes or low-quality coax cables which have a frequency-dependent amplitude loss. For example, for FR4 material at 2.5 GHz, the attenuation compared to the 1.0 GHz value is about 3 dB. The attenuation is a result of skin effect loss of the PCB conductor and the dielectric loss of the PCB substrate. This attenuation causes intersymbol interface which results in the closing of the data eye opening at the receiver.

Multi-channel Alignment

The alignment FIFO allows the transfer of all data to the system clock. The Multi-Channel Alignment block (Figure 6) allows the system to be configured to allow the frame alignment of multiple slightly varying data streams. This optional alignment ensures that matching SERDES streams will arrive at the FPGA end in perfect data synchronization.

Each channel is provided with a 24 word x 36-bit FIFO. The FIFO can perform two tasks: (1) to change the clock domain from receive clock to a clock from the FPGA side, and (2) to align the receive data over 2, 4, or 8 channels. This FIFO allows a timing budget of ±230.4 ns that can be allocated to skew between the data lanes and for transfer to the system clock. The input to the FIFO consists of 36 bits of demultiplexed data, RALIGN_xx[3:0], RWD_xx[31:0], and RWBIT8_xx[3:0].

The four RALIGN_xx bits are control signals, and can be the alignment character detect signals indicating the presence of a comma character in Fibre Channel mode and the /A/ character in XAUI mode. The other 32 RWD_xx bits are the 8-bit data bytes from the 8b/10b decoder. The alignment character, if present, is the MSB of the data. The RWBIT8_xx indicates the presence of a Km.n control character in the receive data byte. Only RWBIT8_xx and RWD_xx inputs are stored in the FIFO. During alignment process, RALIGN[3]_xx is used to synchronize multiple channels.

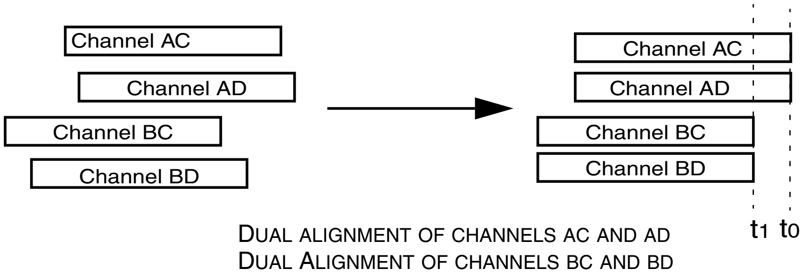
If a channel is not in any alignment group, it will set the FIFO-write-address to the beginning of the FIFO, and will set the FIFO-read-address to the middle of the FIFO, at the first assertion of RALIGN[3]_xx after reset or after the resync command.

The RX_FIFO_MIN_xx register bits can be used to control the threshold for minimum unused buffer space in the alignment FIFOs between read and write pointers before overflow (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 an alignment character from any channel within an alignment group has been received. Once all the alignment characters within the alignment group have been received, the count is decremented by 2 until 0 is reached. Data is then read from the FIFOs and output to the FPGA. This algorithm is not repeated after multi-channel alignment has been achieved; resynchronization must be forced by toggling the appropriate FMPU_RESYNC bit.

ORT42G5 Multi-channel Alignment

The ORT42G5 has a total of four channels. The incoming data of these channels can be synchronized in two ways or they can be independent of one other. Two channels, C and D, within either SERDES block can be aligned together to form a pair, as shown in Figure 11. Alternately, all four channels can be aligned together to form a communication channel with a bandwidth of 10 Gbps, as shown in Figure 12. Individual channels within an alignment group can be disabled (i.e., powered down) without disrupting other channels.

Figure 11. Dual Channel Alignment



Embedded Core/FPGA Interface

This block provides the data formatting and receive data and clock signal transfers between the Embedded Core and the FPGA Logic. There are also control and status registers in the FPGA portion of the chip which contain bits to control the receive logic and to record status. These are described in later sections of this data sheet and communicate with the core using the System Bus.

The demultiplexed, receive word outputs to the FPGA are shown in Figure 6. These are each 40 bits wide. There are eight of these interfaces, one for each SERDES channel. Each consist of four groups of 10-bit data or four groups of decoded information depending on setting of 8b10bR_xx control register bits.

Each 10-bit group of decoded information includes 8 bits of data and a 1 bit K_CTRL indicator derived from the received data and a tenth bit of status information. The function of the tenth bit varies from group to group and includes code violation, Out of Synchronization (OOS) indicators and the CH24_SYNC24_xx and CH248_SYNC_xx status bits. CH24_SYNC or CH248_SYNC_xx indicates the status of multi-channel alignment of channel xx and are high when the count for the multi-channel alignment block reaches zero regardless of whether or not multi-channel alignment is successful. The mapping of the 10-bit groups to the MRWD_xx[39:0] bits output to the FPGA logic is summarized in Table 8. The various functions of the bits that vary from channel to channel, i.e., bits 29 and 19, are also described in Table 9 and Table 10.

Table 8. Definition of Bits of MRWDxx[39:0]

	8b10bR=0	8b10bR=1	
Bit Index	NOCHALGN[A:B]=1 CV_SELxx=0	NOCHALGN[A:B]=1 CV_SELxx=1	NOCHALGN[A:B]=0 CV_SELxx=1
39	bit 9 of 10-bit data 3	CV_xx3, code violation, byte 3	See Table 9 and Table 10
38	bit 8 of 10-bit data 3	K_CTRL for byte 3	K_CTRL for byte 3
37	bit 7 of 10-bit data 3	bit 7 of byte3	bit 7 of byte3
36	bit 6 of 10-bit data 3	bit 6 of byte 3	bit 6 of byte 3
35	bit 5 of 10-bit data 3	bit 5 of byte 3	bit 5 of byte 3
34	bit 4 of 10-bit data 3	bit 4 of byte 3	bit 4 of byte 3
33	bit 3 of 10-bit data 3	bit 3 of byte 3	bit 3 of byte 3
32	bit 2 of 10-bit data 3	bit 2 of byte 3	bit 2 of byte 3
31	bit 1of 10-bit data 3	bit 1 of byte 3	bit 1 of byte 3
30	bit 0 of 10-bit data 3	bit 0 of byte 3	bit 0 of byte 3
29	bit 9 of 10-bit data 2	CV_xx2, code violation, byte 2	See Table 9 and Table 10
28	bit 8 of 10-bit data 2	K_CTRL for byte 2	K_CTRL for byte 2
27	bit 7 of 10-bit data 2	bit 7 of byte 2	bit 7 of byte 2
26	bit 6 of 10-bit data 2	bit 6 of byte 2	bit 6 of byte 2
25	bit 5 of 10-bit data 2	bit 5 of byte 2	bit 5 of byte 2
24	bit 4 of 10-bit data 2	bit 4 of byte 2	bit 4 of byte 2
23	bit 3 of 10-bit data 2	bit 3 of byte 2	bit 3 of byte 2
22	bit 2 of 10-bit data 2	bit 2 of byte 2	bit 2 of byte 2
21	bit 1 of 10-bit data 2	bit 1 of byte 2	bit 1 of byte 2
20	bit 0 of 10-bit data 2	bit 0 of byte 2	bit 0 of byte 2
19	bit 9 of 10-bit data 1	CV_xx1, code violation, byte 1	See Table 9 and Table 10
18	bit 8 of 10-bit data 1	K_CTRL for byte 1	K_CTRL for byte 1
17	bit 7 of 10-bit data 1	bit 7 of byte 1	bit 7 of byte 1
16	bit 6 of 10-bit data 1	bit 6 of byte 1	bit 6 of byte 1
15	bit 5 of 10-bit data 1	bit 5 of byte 1	bit 5 of byte 1
14	bit 4 of 10-bit data 1	bit 4 of byte 1	bit 4 of byte 1
13	bit 3 of 10-bit data 1	bit 3 of byte 1	bit 3 of byte 1

Table 12. Transceiver Embedded Core/FPGA Interface Signal Description for the ORT82G5

FPGA/Embedded Core Interface Signal Name xx=... line remain (xx = [AA, ..., BD]	Input (I) to or Output (O) from Core	Signal Description
Transmit Path Signals		
TWDxx[31:0]	I	Transmit data – channel xx.
TCOMMAxx[3:0]	I	Transmit comma character – channel xx.
TBIT9xx[3:0]	I	Transmit force negative disparity – channel xx
TSYS_CLK_xx	I	Transmit low-speed clock to the FPGA – channel xx
TCK78[A:B]	O	Transmit low-speed clock to the FPGA – SERDES Quad [A:B].
Receive Path Signals		
MRWDxx[39:0]	O	Receive data – Channel xx (see Table 8 and Table).
RWCKxx	O	Low-speed receive clock—Channel xx.
RCK78[A:B]	O	Receive low-speed clock to FPGA—SERDES Quad [A:B].
RSYS_CLK_A1	I	Low-speed receive FIFO clock for channels AA, AB
RSYS_CLK_A2	I	Low-speed receive FIFO clock for channels AC, AD
RSYS_CLK_B1	I	Low-speed receive FIFO clock for channels BA, BB
RSYS_CLK_B2	I	Low-speed receive FIFO clock for channels BC, BD
CV_SELxx	I	Enable detection of code violations in the incoming data
SYS_RST_N	I	Synchronous reset of the channel alignment blocks.

Reference Clocks and Internal Clock Distribution

Reference Clock Requirements

There are two pairs of reference clock inputs on the ORT42G5 and ORT82G5. The differential 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 38.

Note: In sections of this data sheet, the differential clocks are simply referred to as the reference clock as REFCLK_[A:B].

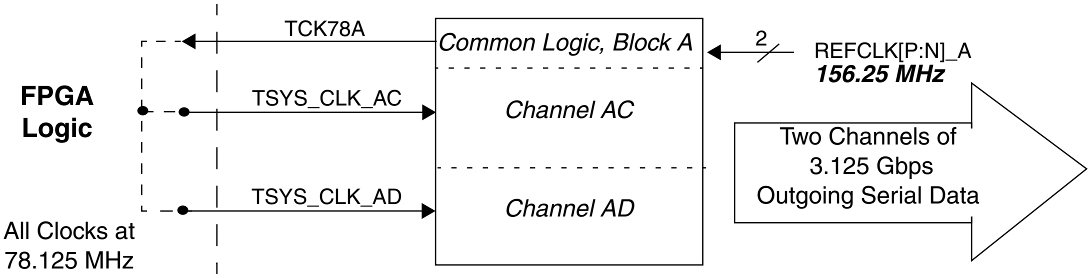
Synthesized and Recovered Clocks

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 ±350 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.

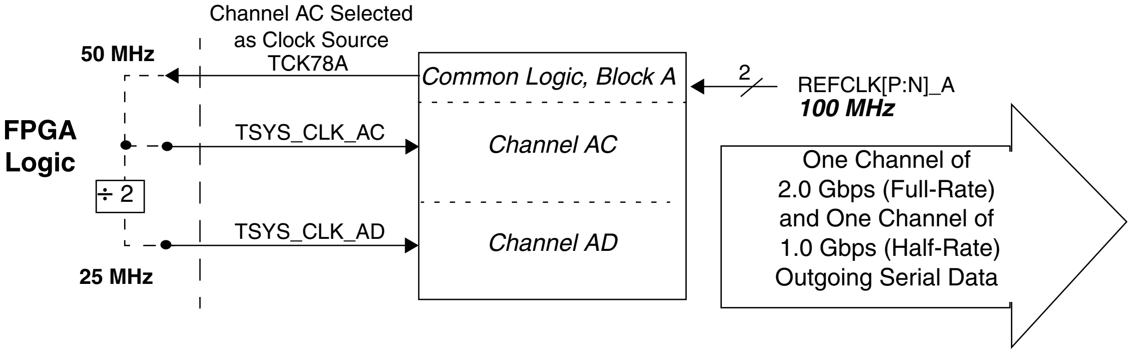
The high-speed transmit and receive serial data links can run at 0.6 to 3.7 Gbps, depending on the frequency of the reference clock and the state of the control bits from the internal transmit control register. The interface to the serializer/deserializer block runs at 1/10th the bit rate of the data lane. Additionally, the MUX/DEMUX logic converts the

Figure 18. Transmit Clocking for a Single Block (Similar Connections Would Be Used for Block B)



If the transmit line rate is mixed between half and full rate among the channels, then the scheme shown in Figure 19 can be used. The figure shows TSYS_CLK_AC being sourced by TCK78A and TSYS_CLK_AD being sourced by TCK78A/2 (the division is done in FPGA logic). Similar clocking would be used for Block B.

Figure 19. Mixed Rate Transmit Clocking for a Single Block (Similar Connections Would Be Used for Block B)


Channel AC Selected as Clock Source
50 MHz
TCK78A
FPGA Logic
÷ 2
25 MHz
TSYS_CLK_AC
TSYS_CLK_AD
Common Logic, Block A
Channel AC
Channel AD
2
REFCLK[P:N]_A
100 MHz
One Channel of 2.0 Gbps (Full-Rate) and One Channel of 1.0 Gbps (Half-Rate) Outgoing Serial Data

Receive Clock Source Selection and Recommended Clock Distribution

In the receive path, one clock per block of two channels, called RCK78[A:B], is sent to the FPGA logic. The control register bits RCKSEL[A:B] is used to select the clock source for these clocks. The selection of the source for RCK78[A:B] is controlled by this bit as shown in Table 15.

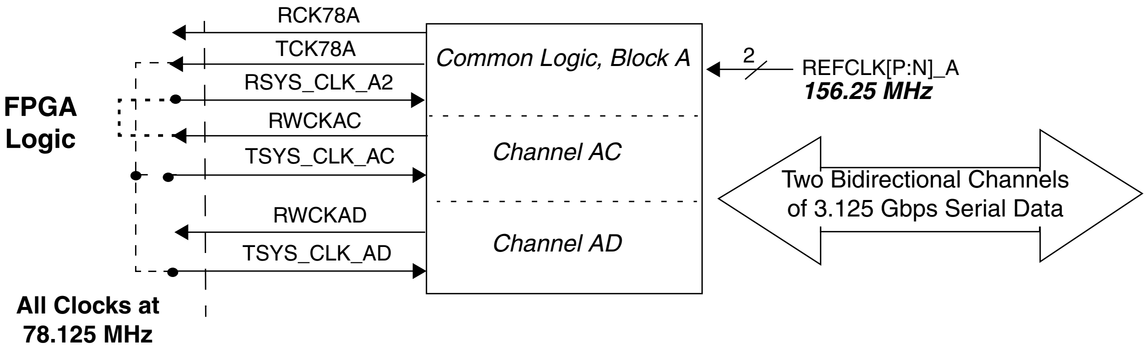
Table 15. RCK78[A:B] Source Selection

RCKSEL[A:B]	Clock Source
0	Channel C
1	Channel D

In the receive channel alignment bypass mode the data and recovered clocks for the four channels are independent. The data for each channel are synchronized to the recovered clock from that channel.

Figure 21 shows the recommended receive clocking for a single block.

Figure 22. Receive Clocking for a Dual Alignment in a Single Block (Similar Connections Would Be Used for Block B)



For quad alignment, either RCK78A or RCK78B can be used to source RSYS_CLK_[A:B]2 as shown in Figure 23.

Figure 23. Clocking for Quad Alignment

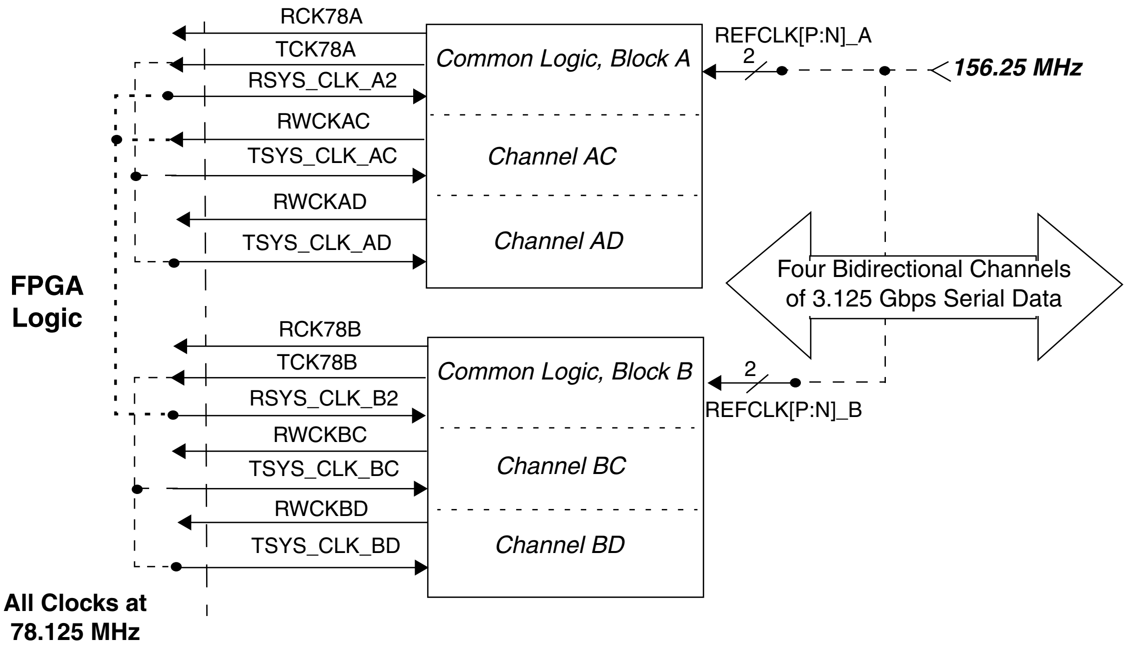


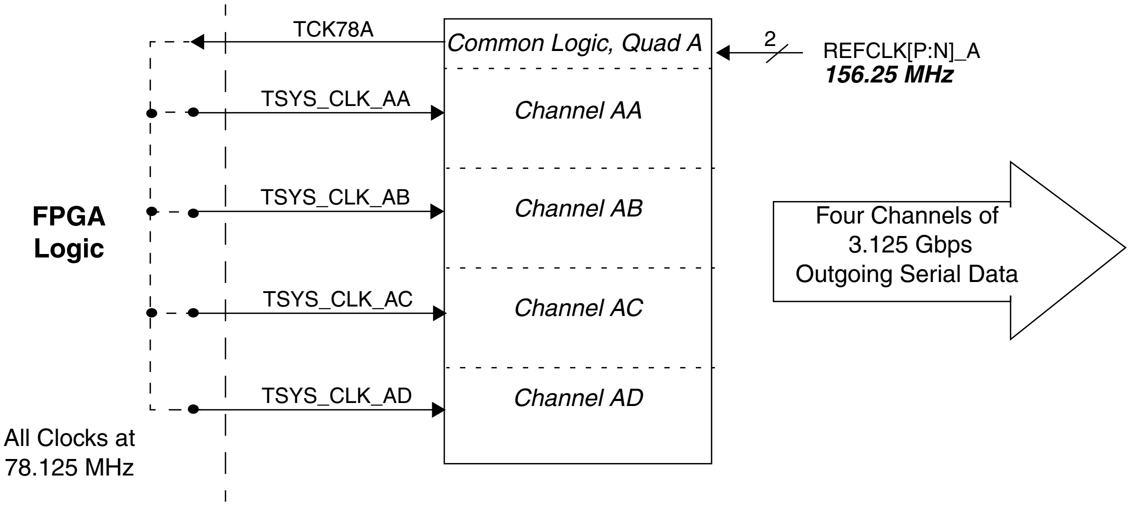
Table 17. TCK78[A:B] Source Selection

TCKSEL0	TCKSEL1	Clock Source
0	0	Channel A
1	0	Channel B
0	1	Channel C
1	1	Channel D

Recommended Transmit Clock Distribution for the ORT82G5

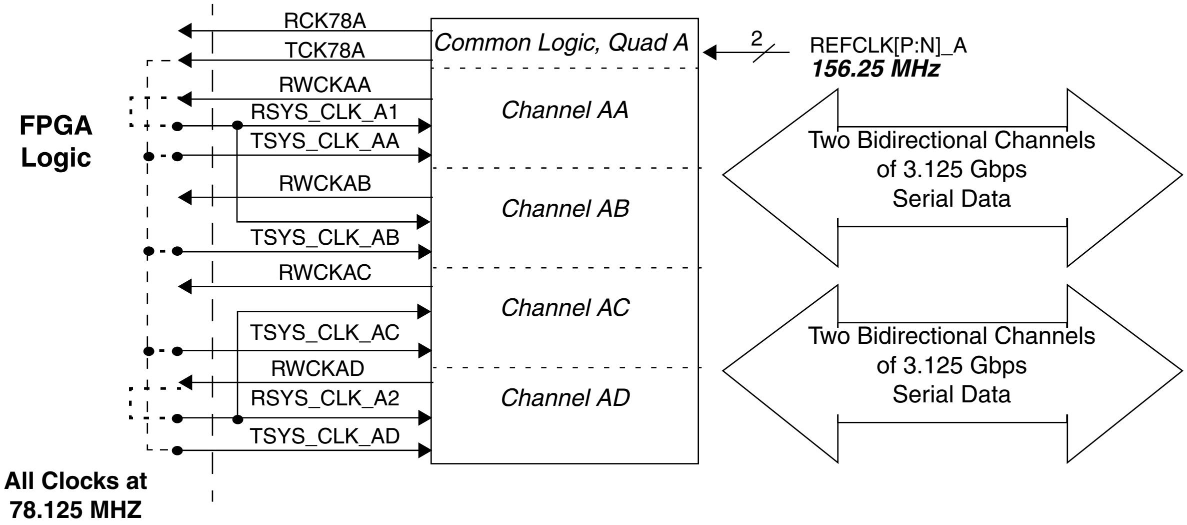
As an example of the recommended clock distribution approach, TSYS_CLK_A[A:D] can be sourced by TCK78A as shown in Figure 25 if the transmit line rate are common for all four channels in a quad. Similar clocking would be used for Quad B.

Figure 25. Transmit Clocking for a Single Block (Similar Connections Would Be Used for Block B)



If the transmit line rate is mixed between half and full rate among the channels, then the scheme shown in Figure 26 can be used. The figure shows TSYS_CLK_AA and TSYS_CLK_AB being sourced by TCK78A and TSYS_CLK_AC and TSYS_CLK_AD being sourced by TCK78A/2 (the division is done in FPGA logic). Similar clocking would be used for Quad B.

Figure 29. Receive Clocking for a Dual Alignment in a Single Quad (Similar Connections Would Be Used for Quad B)



For receive quad alignment, RSYS_CLK_[A1,B1] and RSYS_CLK_[A2,B2] can be tied together as shown for quad A and B in Figure 30. In receive eight-channel alignment, either RCK78A or RCK78B can be used to source RSYS_CLK_[A1,A2] and RSYS_CLK_[B1,B2] as shown in Figure 31.

Figure 30. Clocking for Quad Alignment in a Single Quad (Similar Connections Would Be Used for Quad B)

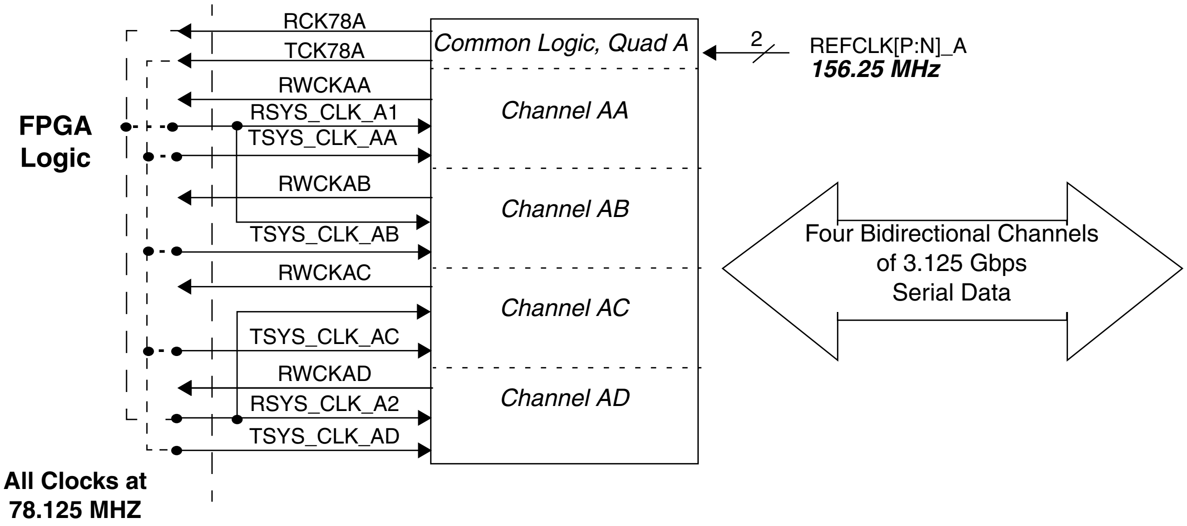


Table 28. ORT42G5 Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
Control Registers (Read/Write), xx=[AC, AD, BC or BD]				
30800 - Ax 30900 - Bx	[0]	—	00	Reserved for future use
	[1]	—		Reserved for future use
	[2]	ENBYSYNC_xC		ENBYSYNC_xC= 1 Enables Receiver Byte Synchronization for Channel xC. ENBYSYNC_xC = 0 on device reset.
	[3]	ENBYSYNC_xD		ENBYSYNC_xD = 1 Enables Receiver Byte Synchronization for Channel xA. ENBYSYNC_xD = 0 on device reset.
	[4]	—		Reserved for future use
	[5]	—		Reserved for future use
	[6]	LCKREFN_xC		LCKREFN_xC = 0 Locks the receiver PLL to reference clock for Channel xC. LCKREFN_xC =1 = Locks the receiver to data for Channel xx. NOTE: When LCKREFN_xx = 0, the corresponding LKI_xx bit is also 0. LCKREFN_xC = 0 on device reset.
	[7]	LCKREFN_xD		LCKREFN_xD = 0 Locks the receiver PLL to reference clock for Channel xD. LCKREFN_xD =1 = Locks the receiver to data for Channel xA. NOTE: When LCKREFN_xx = 0, the corresponding LKI_xx bit is also 0. LCKREFN_xD = 0 on device reset.
30801 - Ax 30901 - Bx	[0]	—	00	Reserved for future use
	[1]	—		Reserved for future use
	[2]	LOOPENB_xC		Enable Loopback Mode for Channel xC. When LOOPEN_xC=1, the transmitter high-speed output is looped back to the receiver high-speed input. This mode is similar to high-speed loopback mode enabled by TESTMODE_xx except that LOOPEN_xx disables the high-speed serial output. LOOPEN_xC=0 on device reset.
	[3]	LOOPENB_xD		Enable Loopback Mode for Channel xD. When LOOPEN_xD=1, the transmitter high-speed output is looped back to the receiver high-speed input. This mode is similar to high-speed loopback mode enabled by TESTMODE_xx except that LOOPEN_xx disables the high-speed serial output. LOOPEN_xD=0 on device reset.
	[4]	—		Reserved for future use
	[5]	—		Reserved for future use
	[6]	NOWDALIGN_xC		Word Align Disable Bit. When NOWDALIGN_xC=1, receiver word alignment is disabled for Channel xC. NOWDALIGN_xC=0 on device reset.
	[7]	NOWDALIGN_xD		Word Align Disable Bit. When NOWDALIGN_xD=1, receiver word alignment is disabled for Channel xD. NOWDALIGN_xD=0 on device reset.

Table 30. ORT82G5 Memory Map

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
SERDES Alarm Registers (Read Only), xx=[AA,...,BD]				
30000 - AA 30010 - AB 30020 - AC 30030 - AD 30100 - BA 30110 - BB 30120 - BC 30130 - BD	[0]	Reserved	00	Reserved
	[1]	LKI_xx		Receive PLL Lock Indication, Channel xx. LKI_xx = 1 indicates the receive PLL is locked.
	[2]	Reserved		Reserved
	[3]	Reserved		Reserved
	[4:7]	Not used		Not used
SERDES Alarm Mask Registers (Read/Write), xx=[AA,...,BD]				
30001 - AA 30011 - AB 30021 - AC 30031 - AD 30101 - BA 30111 - BB 30121 - BC 30131 - BD	[0]	Reserved	FF	Reserved, must be set to 1. Set to 1 on device reset.
	[1]	MLKI_xx		Mask Receive PLL Lock Indication, Channel xx.
	[2]	Reserved		Reserved, must be set to 1. Set to 1 on device reset.
	[3]	Reserved		Reserved, must be set to 1. Set to 1 on device reset.
	[4]	Reserved		Reserved, must be set to 1. Set to 1 on device reset.
	[5]	Reserved		Reserved, must be set to 1. Set to 1 on device reset.
	[6]	Reserved		Reserved, must be set to 1. Set to 1 on device reset.
	[7]	Reserved		Reserved, must be set to 1. Set to 1 on device reset.
SERDES Common Transmit and Receive Channel Configuration Registers (Read/Write), xx=[AA,...,BD]				
30002 - AA 30012 - AB 30022 - AC 30032 - AD 30102 - BA 30112 - BB 30122 - BC 30132 - BD	[0]	TXHR_xx	00	Transmit Half Rate Selection Bit, Channel xx. When TXHR_xx = 1, HDOUT_xx's baud rate = (REFCLK[A:B]*10) and TCK78[A:B] =(REFCLK[A:B]/4); when TXHR_xx=0, HDOUT_xx's baud rate = (REFCLK[A:B]*20) and TCK78[A:B]=(REFCLK[A:B]/2). TXHR_xx = 0 on device reset.
	[1]	PWRDNT_xx		Transmit Powerdown Control Bit, Channel xx. When PWRDNT_xx = 1, sections of the transmit hardware are powered down to conserve power. PWRDNT_xx = 0 on device reset.

	[2]	PE0_xx		Transmit Preemphasis Selection Bit 0, Channel xx. PE0_xx and PE1_xx select one of three preemphasis settings for the transmit section.
				PEO_xx=PE1_xx = 0, Preemphasis is 0%
				PEO_xx=1, PE1_xx = 0 or PEO_xx=0, PE1_xx = 1, Preemphasis is 12.5%
				PEO_xx=PE1_xx = 1, Preemphasis is 25%.
				PEO_xx=PE1_xx = 0 on device reset.

	[3]	PE1_xx		
	[4]	HAMP_xx		Transmit Half Amplitude Selection Bit, Channel xx. When HAMP_xx = 1, the transmit output buffer voltage swing is limited to half its normal amplitude. Otherwise, the transmit output buffer maintains its full voltage swing. HAMP_xx = 0 on device reset.
	[5]	Reserved		Reserved. Must be set to 0. Set to 0 on device reset.
	[6]	Reserved		Reserved
	[7]	8b10bT_xx		Transmit 8b/10b Encoder Enable Bit, Channel xx. When 8b10bT_xx = 1, the 8b/10b encoder in the transmit path is enabled. Otherwise, the data is passed unencoded. 8b10bT_xx = 0 on device reset.

Table 30. ORT82G5 Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
30003 - AA 30013 - AB 30023 - AC 30033 - AD 30103 - BA 30113 - BB 30123 - BC 30133 - BD	[0]	RXHR_xx	20	Receive Half Rate Selection Bit, Channel xx. When RXHR_xx =1, HDIN_xx's baud rate = (REFCLK[A:B]*10) and RCK78[A:B]=(REFCLK[A:B]/4); when RXHR_xx=0, HDIN_xx's baud rate = (REFCLK[A:B]*20) and RCK78[A:B]=(REFCLK/2). RXHR_xx = 0 on device reset.
	[1]	PWRDNR_xx		Receiver Power Down Control Bit, Channel xx. When PWRDNR_xx = 1, sections of the receive hardware are powered down to conserve power. PWRDNR_xx = 0 on device reset.
	[2]	Reserved		Reserved. Set to 1 on device reset.
	[3]	8b10bR_xx		Receive 8b/10b Decoder Enable Bit, Channel xx. When 8b10bR = 1, the 8b/10b decoder in the receive path is enabled. Otherwise, the data is passed undecoded. 8b10bR_xx = 0 on device reset.
	[4]	LINKSM_xx		Link State Machine Enable Bit, Channel xx. When LINKSM_xx = 1, the receiver Fiber Channel link state machine is enabled. Otherwise, the Fibre Channel link state machine is disabled. Note: LINKSM_xx is ignored when XAUI_MODE_xx=1. LINKSM_xx = 0 on device reset.
	[5:7]	Not used		Not used.
SERDES Common Transmit and Receive Channel Configuration Registers (Read/Write), xx=[AA,...,BD]				
30004 - AA 30014 - AB 30024 - AC 30034 - AD 30104 - BA 30114 - BB 30124 - BC 30134 - BD	[0]	Reserved	See bit descrip.	Reserved, must be set to 0. Set to 0 on device reset.
	[1]	MASK_xx		Transmit and Receive Alarm Mask Bit, Channel xx. When MASK_xx = 1, the transmit and receive alarms of a channel are prevented from generating an interrupt (i.e., they are masked or disabled). The MASK_xx bit overrides the individual alarm mask bits in the Alarm Mask Registers. MASK_xx = 1 on device reset.
	[2]	SWRST_xx		Transmit and Receive Software Reset Bit, Channel xx. When SWRST_ss = 1, this bit provides the same function as the hardware reset, except that all configuration register settings are unaltered. This is not a self-clearing bit. Once set, this bit must be manually set and cleared. SWRST = 0 on device reset.
	[3:6]	Not used		Not used. 0 on reset.
	[7]	TESTEN_xx		Transmit and Receive Test Enable Bit, Channel xx. When TESTEN_xx = 1, the transmit and receive sections are placed in test mode. The TestMode_[A:B][4:0] bits in the Global Control Registers specify the particular test, and must also be set. Note: When the global test enable bit GTESTEN_[A:B] = 0, the individual channel test enable bits are used to selectively place a channel in test or normal mode. When GTESTEN_[A:B] = 1, all channels are set to test mode regardless of their TESTEN setting. TESTEN_xx = 0 on device reset.

Table 30. ORT82G5 Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
30805 - Ax 30905 - Bx	[0]xA [1]xB [2]xC [3]xD	DEMUXWAS_xx	00	Status of Word Alignment. When DEMUX_WAS_xx=1, word alignment is achieved for Channel xx. DEMUX_WAS_xx=0 on device reset.
	[4]xA [5]xB [6]xC [7]xD	CH248_SYNC_xx		Status of Channel Alignment. When CH248_SYNC_xx=1, multi-channel alignment is achieved for Channel xx. CH248_SYNC_xx=0 on device reset.
30814 - Ax 30914 - Bx	[0] xA & AB [1] xC & xD	SYNC2_[A:B][1:2] OVFL	00	Multi-Channel Overflow Status. When SYNC2_[A:B][1:2]OVFL=1, dual-channel synchronization FIFO overflow has occurred. SYNC2_[A:B][1:2]OVFL=0 on device reset.
	[2]	SYNC4_ [A:B]OVFL		Multi-Channel Overflow Status. When SYNC4_[A:B]OVFL=1, quad-channel synchronization FIFO overflow has occurred. SYNC4_[A:B]OVFL=0 on device reset.
	[3] xA & AB [4] xC & xD	SYNC2_[A:B][1:2] OOS		Multi-Channel Out-Of-Sync Status. When SYNC2_[A:B][1:2] OOS=1, dual-channel synchronization has failed. SYNC2_[A:B][1:2] OOS=0 on device reset.
	[5]	SYNC4_[A:B]_OO S		Multi-Channel Out-Of-Sync Status. When SYNC4_[A:B]_OOS=1, quad-channel synchronization has failed. SYNC4_[A:B]_OOS=0 on device reset.
	[6:7]	Reserved for future use.		
Common Control Registers (Read/Write)				
30A00	[0:1]	TCKSELA	00	Transmit Clock Select. Controls source of 78 MHz TCK78 for SERDES quad A 00 = Channel AA 10 = Channel AB 01 = Channel AC 11 = Channel AD
	[2:3	RCKSELA		Receive Clock Select. Controls source of 78 MHz RCK78 for SEDRES quad A 00 = Channel AA 10 = Channel AB 01 = Channel AC 11 = Channel AD
	[4:5]	TCKSELB		Transmit Clock Select. Controls source of 78 MHz TCK78 for SERDES quad B 00 = Channel BA 10 = Channel BB 01 = Channel BC 11 = Channel BD
	[6:7]	RCKSELB		Receive Clock Select. Controls source of 78 MHz RCK78 for SERDES quad B 00 = Channel BA 10 = Channel BB 01 = Channel BC 11 = Channel BD
30A01	[0:4]	—	00	Reserved for future use
	[5:7]	RX_FIFO_MIN		LSb's for the threshold for low address in RX_FIFOs. RX_FIFO_MIN, Bit 5 is LSb. Useful values for RX_FIFO_MIN [0:4] are 0 to 17(decimal).

Table 41. FPSC Function Pin Descriptions (Continued)

Symbol	I/O	Description
HDOUTP_AB (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad A, channel B.
HDOUTN_AC	O	High-speed CML transmit data output – SERDES quad A, channel C.
HDOUTP_AC	O	High-speed CML transmit data output – SERDES quad A, channel C.
HDOUTN_AD	O	High-speed CML transmit data output – SERDES quad A, channel D.
HDOUTP_AD	O	High-speed CML transmit data output – SERDES quad A, channel D.
HDOUTN_BA (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad B, channel A.
HDOUTP_BA (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad B, channel A.
HDOUTN_BB (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad B, channel B.
HDOUTP_BB (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad B, channel B.
HDOUTN_BC	O	High-speed CML transmit data output – SERDES quad B, channel C.
HDOUTP_BC	O	High-speed CML transmit data output – SERDES quad B, channel C.
HDOUTN_BD	O	High-speed CML transmit data output – SERDES quad B, channel D.
HDOUTP_BD	O	High-speed CML transmit data output – SERDES quad B, channel D.
Power and Ground		
VDDIB_AA (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_AB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_AC	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_AD	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BA (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BC	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BD	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDOB_AA (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_AB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_AC	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_AD	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BA (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BC	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BD	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDGB_A	—	1.5V guard band power supply.
VDDGB_B	—	1.5V guard band power supply.
VDD_ANA	—	1.5V power supply for SERDES analog receive and transmit circuitry.

1. Should be externally connected on board to 3.3V pull-up resistor.

Table 44. ORT42G5 484-pin PBGAM (fpBGA) Pinout

484-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	484-PBGAM
E4	-	-	O	PRD_DATA	RD_DATA/TDO	-
C20	-	-	VDD15	VDD15	-	-
D3	-	-	I	PRESET_N	RESET_N	-
F5	-	-	I	PRD_CFG_N	RD_CFG_N	-
F4	-	-	I	PPRGRM_N	PRGRM_N	-
C2	0 (TL)	7	IO	PL2D	PLL_CK0C/HPPLL	L1C
C1	0 (TL)	7	IO	PL2C	PLL_CK0T/HPPLL	L1T
F3	0 (TL)	7	IO	PL3C	VREF_0_07	-
A1	-	-	VSS	VSS	-	-
D2	0 (TL)	7	IO	PL4D	D5	L2C
D1	0 (TL)	7	IO	PL4C	D6	L2T
E7	0 (TL)	-	VDDIO0	VDDIO0	-	-
E2	0 (TL)	8	IO	PL5D	HDC	L3C
E1	0 (TL)	8	IO	PL5C	LDC_N	L3T
G3	0 (TL)	8	IO	PL5A	-	-
G4	0 (TL)	9	IO	PL6C	D7	-
F2	0 (TL)	9	IO	PL7D	VREF_0_09	L4C
F1	0 (TL)	9	IO	PL7C	A17/PPC_A31	L4T
G2	0 (TL)	9	IO	PL8D	CS0_N	L5C
G1	0 (TL)	9	IO	PL8C	CS1	L5T
E8	0 (TL)	-	VDDIO0	VDDIO0	-	-
A2	-	-	VSS	VSS	-	-
H1	0 (TL)	10	IO	PL10D	INIT_N	L6C
H2	0 (TL)	10	IO	PL10C	DOUT	L6T
E5	-	-	VDD15	VDD15	-	-
H4	0 (TL)	10	IO	PL11D	VREF_0_10	-
H3	0 (TL)	10	IO	PL11C	A16/PPC_A30	-
J1	7 (CL)	1	IO	PL12D	A15/PPC_A29	L7C
J2	7 (CL)	1	IO	PL12C	A14/PPC_A28	L7T
J4	7 (CL)	1	IO	PL13C	D4	-
G7	-	-	VSS	VSS	-	-
J3	7 (CL)	2	IO	PL14D	RDY/BUSY_N/RCLK	-
K6	7 (CL)	-	VDDIO7	VDDIO7	-	-
K1	7 (CL)	2	IO	PL15D	A13/PPC_A27	L8C
K2	7 (CL)	2	IO	PL15C	A12/PPC_A26	L8T
K3	7 (CL)	3	IO	PL16C	-	-
K4	7 (CL)	3	IO	PL17D	A11/PPC_A25	-
G8	-	-	VSS	VSS	-	-
F8	-	-	VDD15	VDD15	-	-
K5	7 (CL)	4	IO	PL19D	RD_N/MPI_STRB_N	-
L1	7 (CL)	4	IO	PL20D	PLCK0C	L9C
L2	7 (CL)	4	IO	PL20C	PLCK0T	L9T
L6	7 (CL)	-	VDDIO7	VDDIO7	-	-
F9	-	-	VDD15	VDD15	-	-

Table 44. ORT42G5 484-pin PBGAM (fpBGA) Pinout (Continued)

484-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	484-PBGAM
V4	-	-	IO	LVDS_R	LVDS_R	-
V3	-	-	VDD33	VDD33	-	-
F17	-	-	VDD15	VDD15	-	-
W3	6 (BL)	5	IO	PB2A	DP2	-
AA2	6 (BL)	5	IO	PB2C	PLL_CK6T/PPLL	L23T
AB2	6 (BL)	5	IO	PB2D	PLL_CK6C/PPLL	L23C
AA3	6 (BL)	5	IO	PB4A	VREF_6_05	L24T
AB3	6 (BL)	5	IO	PB4B	DP3	L24C
T5	6 (BL)	-	VDDIO6	VDDIO6	-	-
H7	-	-	VSS	VSS	-	-
Y4	6 (BL)	6	IO	PB5C	VREF_6_06	L25T
W4	6 (BL)	6	IO	PB5D	D14	L25C
T8	6 (BL)	-	VDDIO6	VDDIO6	-	-
AA4	6 (BL)	7	IO	PB6C	D15	L26T
AB4	6 (BL)	7	IO	PB6D	D16	L26C
H8	-	-	VSS	VSS	-	-
W5	6 (BL)	7	IO	PB7C	D17	L27T
Y5	6 (BL)	7	IO	PB7D	D18	L27C
T9	6 (BL)	-	VDDIO6	VDDIO6	-	-
AA5	6 (BL)	7	IO	PB8C	VREF_6_07	L28T
AB5	6 (BL)	7	IO	PB8D	D19	L28C
H9	-	-	VSS	VSS	-	-
V6	6 (BL)	8	IO	PB9C	D20	-
G6	-	-	VDD15	VDD15	-	-
W6	6 (BL)	8	IO	PB10C	VREF_6_08	L29T
Y6	6 (BL)	8	IO	PB10D	D22	L29C
H10	-	-	VSS	VSS	-	-
AA6	6 (BL)	9	IO	PB11C	D23	L30T
AB6	6 (BL)	9	IO	PB11D	D24	L30C
U6	6 (BL)	-	VDDIO6	VDDIO6	-	-
W7	6 (BL)	9	IO	PB12C	VREF_6_09	L31T
Y7	6 (BL)	9	IO	PB12D	D25	L31C
H11	-	-	VSS	VSS	-	-
V7	6 (BL)	10	IO	PB14A	-	-
U7	6 (BL)	-	VDDIO6	VDDIO6	-	-
AA7	6 (BL)	10	IO	PB14C	VREF_6_10	L32T
AB7	6 (BL)	10	IO	PB14D	D28	L32C
V8	6 (BL)	11	IO	PB15A	-	-
H12	-	-	VSS	VSS	-	-
W8	6 (BL)	11	IO	PB15C	D29	L33T
Y8	6 (BL)	11	IO	PB15D	D30	L33C
U8	6 (BL)	11	IO	PB16A	-	-
AA8	6 (BL)	11	IO	PB16C	VREF_6_11	L34T
AB8	6 (BL)	11	IO	PB16D	D31	L34C

Table 44. ORT42G5 484-pin PBGAM (fpBGA) Pinout (Continued)

484-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	484-PBGAM
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
P6	-	-	VDD15	VDD15	-	-
A11	1 (TC)	5	IO	PT16D	VREF_1_05	L70C
B11	1 (TC)	5	IO	PT16C	-	L70T
L12	-	-	VSS	VSS	-	-
D9	1 (TC)	6	IO	PT15D	-	L71C
C9	1 (TC)	6	IO	PT15C	-	L71T
G15	1 (TC)	-	VDDIO1	VDDIO1	-	-
B10	1 (TC)	6	IO	PT14D	-	L72C
A10	1 (TC)	6	IO	PT14C	VREF_1_06	L72T
B9	0 (TL)	1	IO	PT13D	MPI_RTRY_N	L73C
A9	0 (TL)	1	IO	PT13C	MPI_ACK_N	L73T
D8	0 (TL)	1	IO	PT12D	M0	L74C
C8	0 (TL)	1	IO	PT12C	M1	L74T
A22	-	-	VSS	VSS	-	-
B8	0 (TL)	2	IO	PT12B	MPI_CLK	L75C
A8	0 (TL)	2	IO	PT12A	A21/MPI_BURST_N	L75T
C7	0 (TL)	2	IO	PT11D	M2	L76C
D7	0 (TL)	2	IO	PT11C	M3	L76T
E9	0 (TL)	-	VDDIO0	VDDIO0	-	-
E6	0 (TL)	2	IO	PT11A	MPI_TEA_N	-
F6	-	-	VDD15	VDD15	-	-
B7	0 (TL)	3	IO	PT9D	VREF_0_03	L77C
A7	0 (TL)	3	IO	PT9C	-	L77T
A6	0 (TL)	3	IO	PT8D	D0	L78C
B6	0 (TL)	3	IO	PT8C	TMS	L78T
C6	0 (TL)	4	IO	PT7D	A20/MPI_BDIP_N	L79C
D6	0 (TL)	4	IO	PT7C	A19/MPI_TSZ1	L79T
B1	-	-	VSS	VSS	-	-
A5	0 (TL)	4	IO	PT6D	A18/MPI_TSZ0	L80C
B5	0 (TL)	4	IO	PT6C	D3	L80T
C5	0 (TL)	5	IO	PT5D	D1	L81C
D5	0 (TL)	5	IO	PT5C	D2	L81T
B2	-	-	VSS	VSS	-	-
A4	0 (TL)	5	IO	PT4D	TDI	L82C
B4	0 (TL)	5	IO	PT4C	TCK	L82T
E10	0 (TL)	-	VDDIO0	VDDIO0	-	-
B22	-	-	VSS	VSS	-	-
C4	0 (TL)	6	IO	PT2D	PLL_CK1C/PPLL	L83C

Table 45. ORT82G5 680-Pin PBGAM (fpBGA) Pinout (Continued)

680-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	680-PBGAM
B24	1 (TC)	3	IO	PT23B	—	—
D20	1 (TC)	3	IO	PT22D	—	L15C_A0
D19	1 (TC)	3	IO	PT22C	—	L15T_A0
N14	—	—	Vss	Vss	—	—
E19	1 (TC)	3	IO	PT22B	—	L16C_A0
E18	1 (TC)	3	IO	PT22A	—	L16T_A0
C21	1 (TC)	4	IO	PT21D	—	L17C_A0
C20	1 (TC)	4	IO	PT21C	—	L17T_A0
A25	1 (TC)	4	IO	PT21B	—	L18C_A0
A24	1 (TC)	4	IO	PT21A	—	L18T_A0
B23	1 (TC)	4	IO	PT20D	—	L19C_A0
A23	1 (TC)	4	IO	PT20C	—	L19T_A0
N15	—	—	Vss	Vss	—	—
E17	1 (TC)	4	IO	PT20B	—	L20C_A0
E16	1 (TC)	4	IO	PT20A	—	L20T_A0
B22	1 (TC)	4	IO	PT19D	—	L21C_A0
B21	1 (TC)	4	IO	PT19C	VREF_1_04	L21T_A0
C18	1 (TC)	4	IO	PT19B	—	L22C_A0
C19	1 (TC)	4	IO	PT19A	—	L22T_A0
N20	—	—	Vss	Vss	—	—
A22	1 (TC)	5	IO	PT18D	PTCK1C	L23C_A0
A21	1 (TC)	5	IO	PT18C	PTCK1T	L23T_A0
N21	—	—	Vss	Vss	—	—
D17	1 (TC)	5	IO	PT18B	—	L24C_A0
D18	1 (TC)	5	IO	PT18A	—	L24T_A0
B20	1 (TC)	5	IO	PT17D	PTCK0C	L25C_A0
B19	1 (TC)	5	IO	PT17C	PTCK0T	L25T_A0
A20	1 (TC)	5	IO	PT17B	—	L26C_A0
A19	1 (TC)	5	IO	PT17A	—	L26T_A0
A18	1 (TC)	5	IO	PT16D	VREF_1_05	L27C_A0
B18	1 (TC)	5	IO	PT16C	—	L27T_A0
Y21	—	—	Vss	Vss	—	—
C17	1 (TC)	5	IO	PT16B	—	L28C_D0
D16	1 (TC)	5	IO	PT16A	—	L28T_D0
A17	1 (TC)	6	IO	PT15D	—	L29C_D0
B16	1 (TC)	6	IO	PT15C	—	L29T_D0
E15	1 (TC)	6	IO	PT15B	—	L30C_A0
E14	1 (TC)	6	IO	PT15A	—	L30T_A0
A16	1 (TC)	6	IO	PT14D	—	L31C_A0
A15	1 (TC)	6	IO	PT14C	VREF_1_06	L31T_A0
Y22	—	—	Vss	Vss	—	—
D14	1 (TC)	6	IO	PT14B	—	—
C16	0 (TL)	1	IO	PT13D	MPI_RTRY_N	L1C_A0
C15	0 (TL)	1	IO	PT13C	MPI_ACK_N	L1T_A0