



XAPP486 (v1.1) June 21, 2010

## 7:1 Serialization in Spartan-3E/3A FPGAs at Speeds Up to 666 Mbps

### Summary

Spartan®-3E and Extended Spartan-3A devices are used in a wide variety of applications requiring 7:1 serialization at speeds up to 666 Mbps. This application note targets Spartan-3E/3A devices in applications that require 4-bit or 5-bit transmit data bus widths and operate at rates up to 666 Mbps per line with a forwarded clock at 1/7th the bit rate. This type of interface is commonly used in flat panel displays and automotive applications. Associated receiver designs are discussed in [XAPP485](#), *1:7 Deserialization in Spartan-3E/3A FPGAs at Speeds Up to 666 Mbps*.

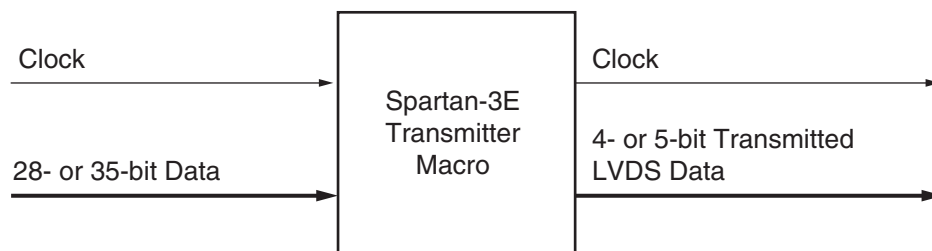
These designs are applicable to Spartan-3E/3A FPGAs and not to the original Spartan-3 device. The design files for this application note target the Spartan-3E family. The Extended Spartan-3A family also supports the same design approach. The maximum data rate for the Spartan-3A FPGA is 640 Mbps for the -4 speed grade and 700 Mbps for the -5 speed grade.

### Introduction

Two versions of the serializer design are available:

1. In the *Logic* version, the lower speed system clock and the higher speed transmitter clock are phase-aligned.
2. The *FIFO* version uses a block RAM based FIFO memory to ensure that there is no phase relationship requirement between the two clocks.

Both versions use both a transmission clock that is 3.5 times the system clock and double data rate (DDR) techniques to arrive at a serialization factor of seven. This is done both to keep the internal logic to a reasonable speed, and also to ensure that the clock generation falls into the range of the Digital Frequency Synthesizer (DFS) block of the Spartan-3E FPGA. The maximum data rate for the Spartan-3E FPGA is 622 Mbps for the -4 speed grade and 666 Mbps for the -5 speed grade. The limitation in both devices is the maximum speed of the DFS block in Stepping 1 silicon. Refer to [UG331](#), *Spartan-3 Generation FPGA User Guide*, for more information on the DFS block.



XAPP486\_01\_061010

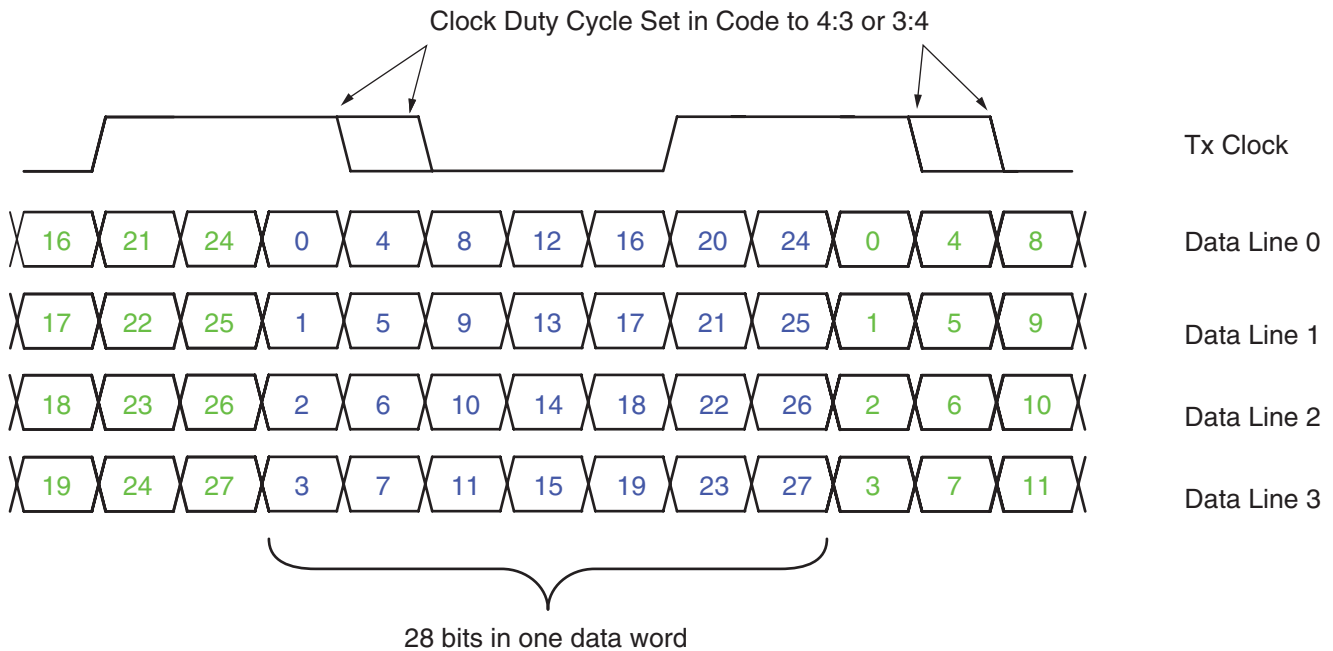
Figure 1: 7:1 Transmitter Module

## Output Pad Placement

The transmitter macro has been designed for use on the right hand side of the Spartan-3E silicon, and while the example pinouts supplied in the zip file ([xapp486.zip](#)) follow this rule, there is in fact no definitive requirement that it be placed there. The deciding factor in placing it elsewhere would be whether or not the existing internal floorplan of the macro could meet the designer's speed requirements or not in its new location.

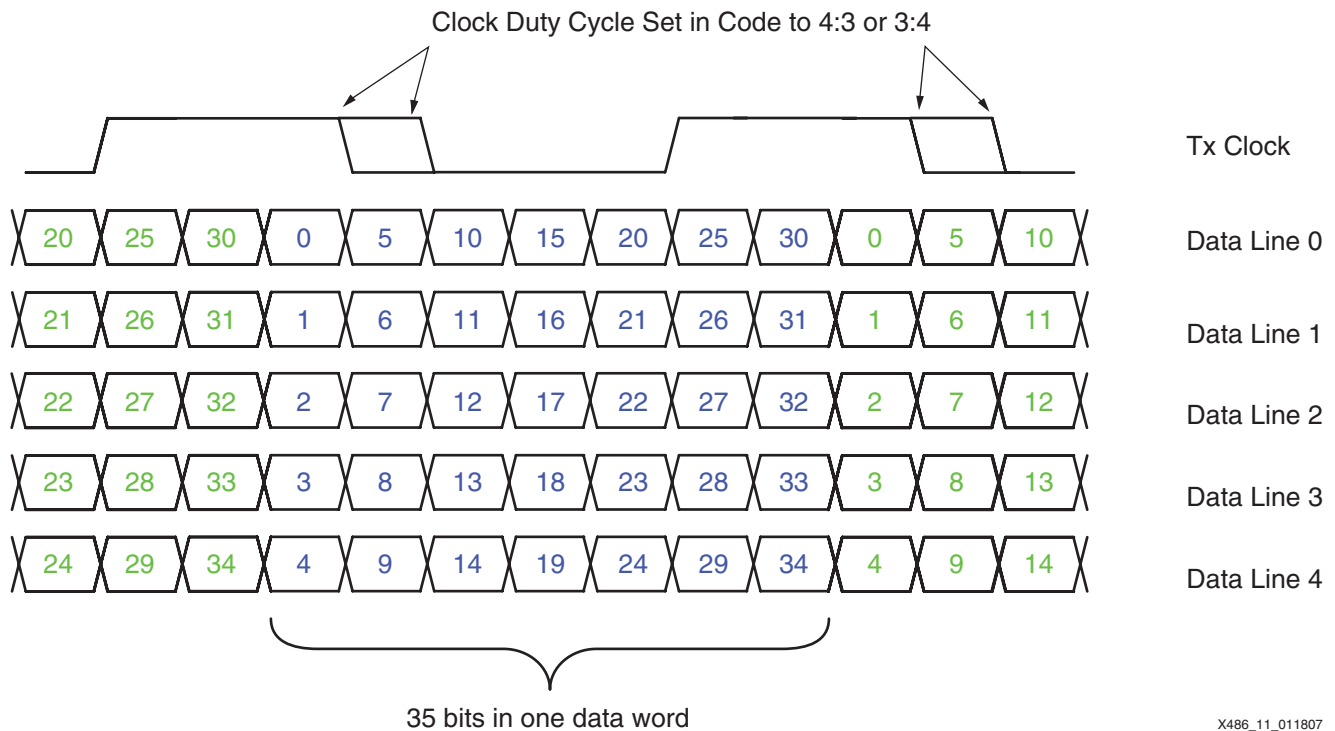
The transmitted data consists of either four or five data lines plus the forwarded clock, all of which are very closely aligned. The clock is either a 4:3 or a 3:4 duty cycle, selectable in the code. The output pads should be kept as close as possible to each other to minimize clock skew, and care should obviously be taken with the PCB design, such that the tracks for all signals are as close to an ideal 100 Ω impedance as possible, and matched in length as closely as possible.

Figure 2 and Figure 3 show the relationship of clock and data along with the location of each bit to be transmitted within the 7-bit long and 4- or 5-bit wide frame. If the application requires a different bit ordering, the scrambling is easily taken care of in the design code as described in "Design Files."



X486\_02\_011807

Figure 2: 4-Bit Transmit Data Formatting



**Figure 3: 5-Bit Transmit Data Formatting**

The DDR flip-flops inside the IOBs that are part of the Spartan-3E architecture require data to be presented synchronously to both clocks used for transmission.

## Clock Considerations

The internal system clock is multiplied up by 3.5 inside the DFS to generate the high-speed clock required by this application. There are two possibilities for clocking. Either one clock can be distributed from the DFS using one global buffer, and this clock is inverted where required, or two clocks, 180 degrees apart can be distributed using two global buffers from the CLKFX and CLKFX180 outputs of the DFS.

The advantage of the second method is that duty cycle distortion in the global clock networks becomes unimportant because only rising edges are used. For this reason, this method is recommended for high-speed interfaces.

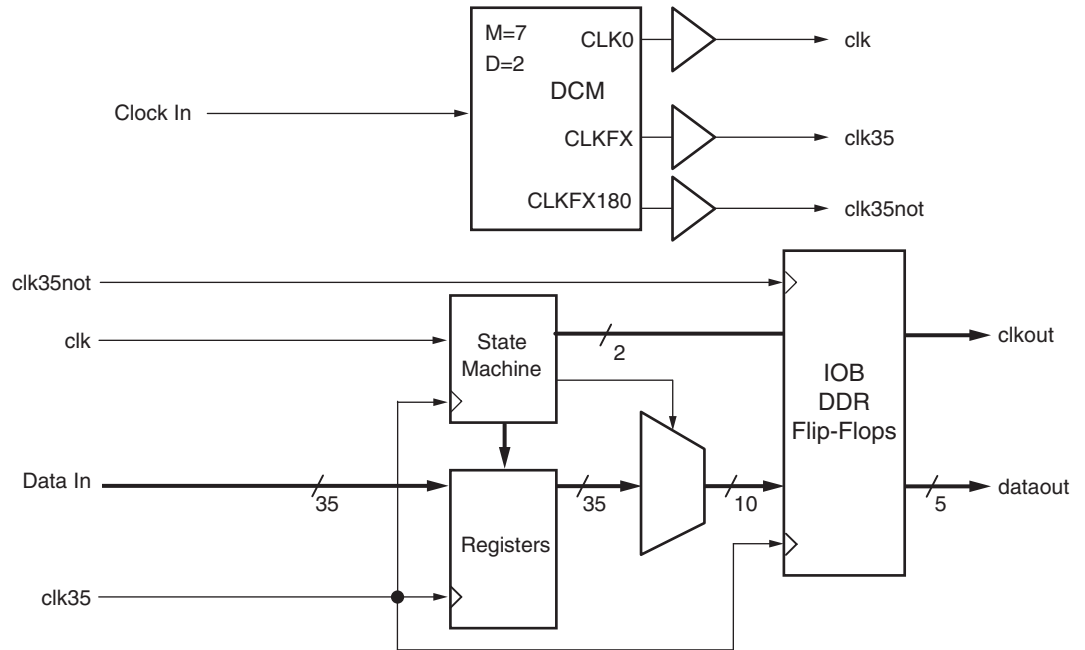
The DFS can work at speeds up to 311 MHz (622 Mbps per line) in the -4 speed grade, and 333 MHz (666 Mbps) in the -5 speed grade.

Very importantly, the DFS can also work at speeds down to 5 MHz (17.5 Mbps per line) without the requirement to reprogram the DFS and hence change the FPGA bitstream. This advantage is important in systems where the transmitted data stream has to change frequencies.

# Logic Description

## Logic Version

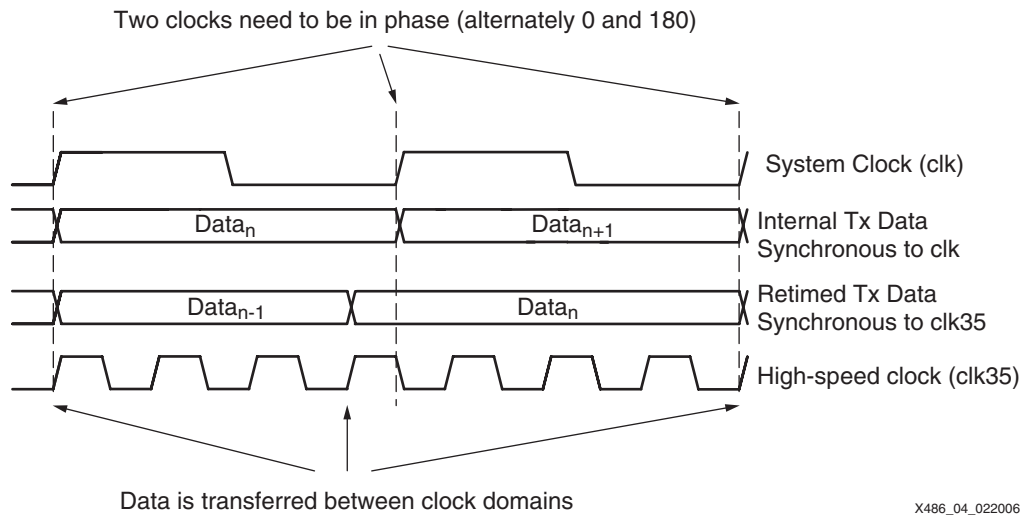
Figure 4 shows a block diagram for the Logic version of the transmitter. It is very simple to design, but tricky to make work at high frequencies. Data arrives synchronous to the clock, clk, in either 28-bit or 35-bit words depending on the output data width. A state machine compares the two incoming clocks, clk and clk35 (which are in phase), and when they coincide, data is moved from the low-speed to the high-speed domain. The logic to detect coincidence has to run at high speed (630 MHz for a 630 Mbps data link) and so placement inside the macro is extremely important.



X486\_03\_012607

Figure 4: Spartan-3E 1:7 Transmitter Logic Version (5-Bit Module)

Coincidence only actually occurs every two clock (or seven clk35) cycles because the high-speed clock is a 3.5x multiple of clk as shown in Figure 5. The state machine then schedules the retimed data for serialization and transmission via the DDR output registers discussed above.



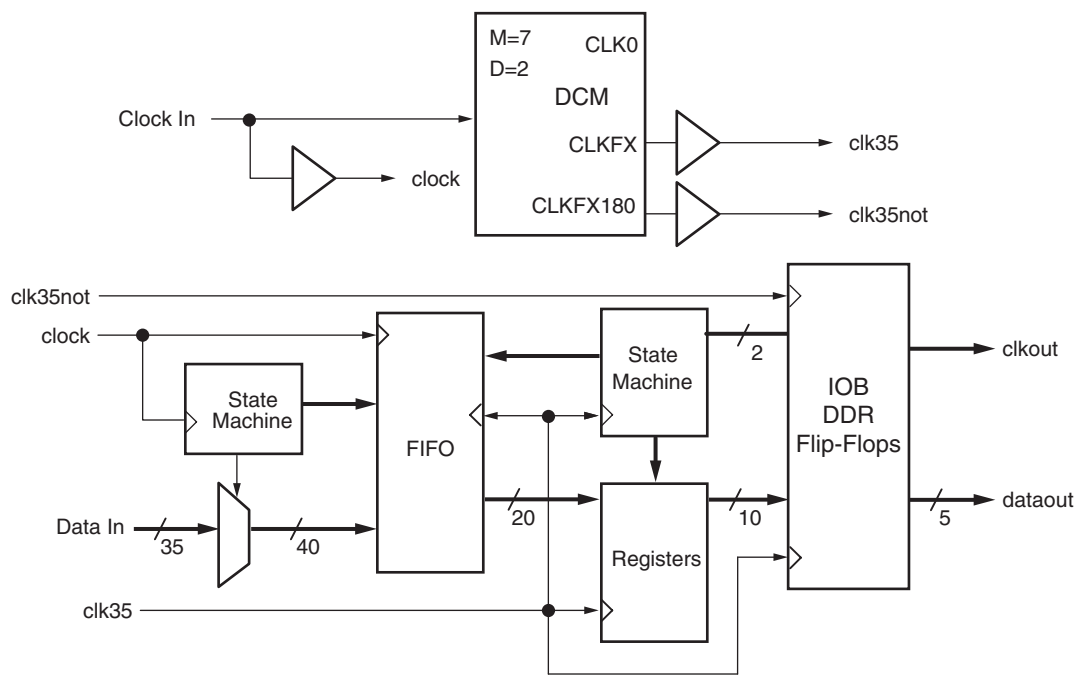
X486\_04\_022006

Figure 5: Internal Timing for Logic Version

The clock for forwarding is also generated as data in the output DDR registers, and as a result has a 4:3 duty cycle, although this ratio can be changed to 3:4 in the provided code as required.

## FIFO Version

The FIFO version of the code is somewhat more complex, but has the advantage of not requiring that the two clocks be in phase. Figure 6 shows the logic block diagram. Data still arrives 28 or 35 bits wide synchronous to *clk*, and is written into a block memory configured with either 32- or 40-bit inputs (40-bit inputs actually require two block memories), and either 16- or 20-bit outputs. When writing the memory, some of the data is *borrowed* from the next word to arrive, to pad out the incoming data to the RAM width. In this way, the RAM is written for seven out of every eight clocks. For the 35-bit internal data case, 8 x 35 bits = 280 bits are written to the RAM every 7 cycles as 7 x 40 bits = 280 bits, and the RAM is disabled every eighth cycle. This may be clearer from Table 1, where *datain* is the input data, and *dataind* is the data from one clock cycle previously.



X486\_05\_022006

Figure 6: Spartan-3E 1:7 Transmitter FIFO Version (5-Bit Module)

Table 1: Address/Data Sequence when Writing to FIFO

RAM Address Low Three Bits	WE	Data Written
..0	Active	datain[4:0], dataind[34:0]
..1	Active	datain[9:0], dataind[34:5]
..2	Active	datain[14:0], dataind[34:10]
..3	Active	datain[19:0], dataind[34:15]
..4	Active	datain[24:0], dataind[34:20]
..5	Active	datain[29:0], dataind[34:25]
..6	Active	datain[34:0], dataind[34:30]
..7	Inactive	XXXXXXXX

Data is then read out from the other port of the RAM synchronous to the high-speed clock, clk35. It is actually read every other cycle, so that the clock-to-out spec of the RAM is not exceeded. Data is read out 16 bits wide for the 4-bit external data case, and 20 bits wide for the 5-bit external data case.

Care must be taken to only read locations of the memory that contain valid data. Valid data is written into locations 0 to 6, which corresponds to locations 0 to 13 when reading because the read port is only half the width of the write port. The read address for the RAM therefore counts from 0 to 0xD, and then resets to 0. Thus 14 x 20 bits are read over 28 cycles of the high-speed clock, where 28 high speed clocks = 8 slow speed clocks and 14 x 20 = 280 bits read. So the input and output bandwidths are the same. From this point, data is simply serialized by a factor of 2 in logic, and a factor of 2 in the output DDR registers to reduce the 16 or 20 bits down to 4 or 5 bits.

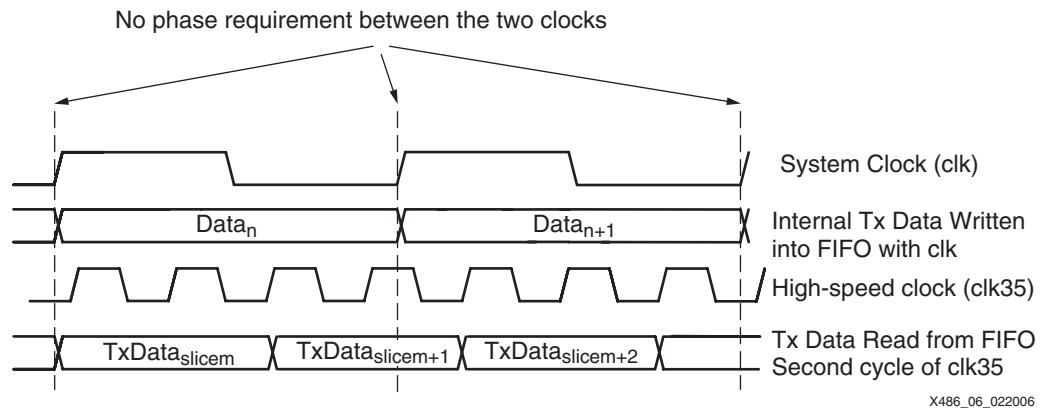


Figure 7: Internal Timing for FIFO Version

The clock for forwarding is generated in an identical fashion to the Logic based macro described above.

## Timing Analysis

The timing analysis for the transmitter consists of adding the various sources of timing errors and uncertainty. These include:

- All mismatch and silicon variations
  - Skew between the two global buffers distributing the high-speed clock and its complement. This number is included within the jitter figure specified below.
  - The package skew among all the data and clock lines.
  - The internal clock skew between the IOB flip-flops in the device. This number varies with the placement of the output lines in the package. If all the Xilinx placement guidelines described herein are followed, this number is less than 50 ps.
- Jitter and timing uncertainty is another important source that cuts into overall timing budget. This parameter is referred to as  $T_{J35}$ . Because this parameter strongly depends on the environment in which Xilinx chips are used, it is not possible to guarantee a worst-case number without knowing the environment. However, Xilinx has done extensive characterization with various amounts of noise, and expect this number for all Spartan-3E devices to be better than 400 ps plus 2% of the output clock period. The chip and environmental factors that contribute to this number include (but are not limited to):
  - The jitter introduced by phase shifting in the DFS unit when it multiplies the incoming clock by 3.5.
  - Incoming clock jitter, which is obviously dependent on the system in question. While the characterization number,  $T_{J35}$ , includes a reasonable amount (100 ps) of input clock jitter, it is clear that increasing input clock jitter adversely affects this parameter.

- Excessive switching activity in the fabric of FPGA can also contribute to chip jitter and timing uncertainty. Typical fabric switching activity is 12% for most applications. The Xilinx characterized number is based on 25% fabric switching activity.
- Switching I/Os at high drive strengths and the frequency of switching contribute to the additional timing uncertainty. The Xilinx characterization result includes the noise of 40 simultaneous switching outputs (SSOs) running at 80 MHz.
- The board design and chip package are also important factors. The Xilinx characterization number is based on a four-layer board and an FT256 package.

The example margin timing analysis is for a 600 Mbps design, where the DFS clock is 300 MHz.

$T_{J35}$	$400 + 0.02 * (10^6/300)$ ps	Xilinx number
+ clock skew	50 ps	Xilinx number
=	516 ps	Tx uncertainties

## Design Files

The design files, written in both Verilog and VHDL for both 4-bit and 5-bit versions of the transmitter interface, are available from the Xilinx web site ([xapp486.zip](http://xapp486.zip)). The files include source code, design examples, timing constraints (\*.UCF files) and example pinouts for many part/package combinations. If you have a requirement for a part/package combination that is not included, or any other questions, contact [spartan3e.serdes71@xilinx.com](mailto:spartan3e.serdes71@xilinx.com).

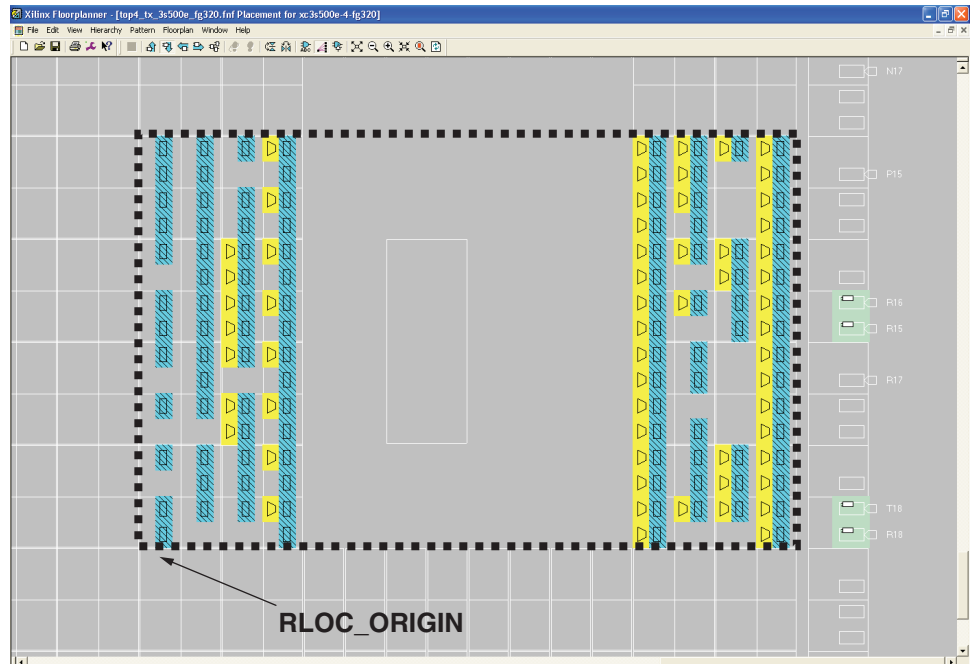
## Processing the Design

The design files have been tested with ISE® 9.1i and Synplify 8.4. For all ISE versions with both VHDL and Verilog, the following modification to the environment must be done:

- Ensure that ISE is keeping hierarchy – this is the default for Synplicity but not for ISE. Right-click on **synthesize-XST** to get the Properties window, and make sure “Keep Hierarchy” is set to **Yes**.

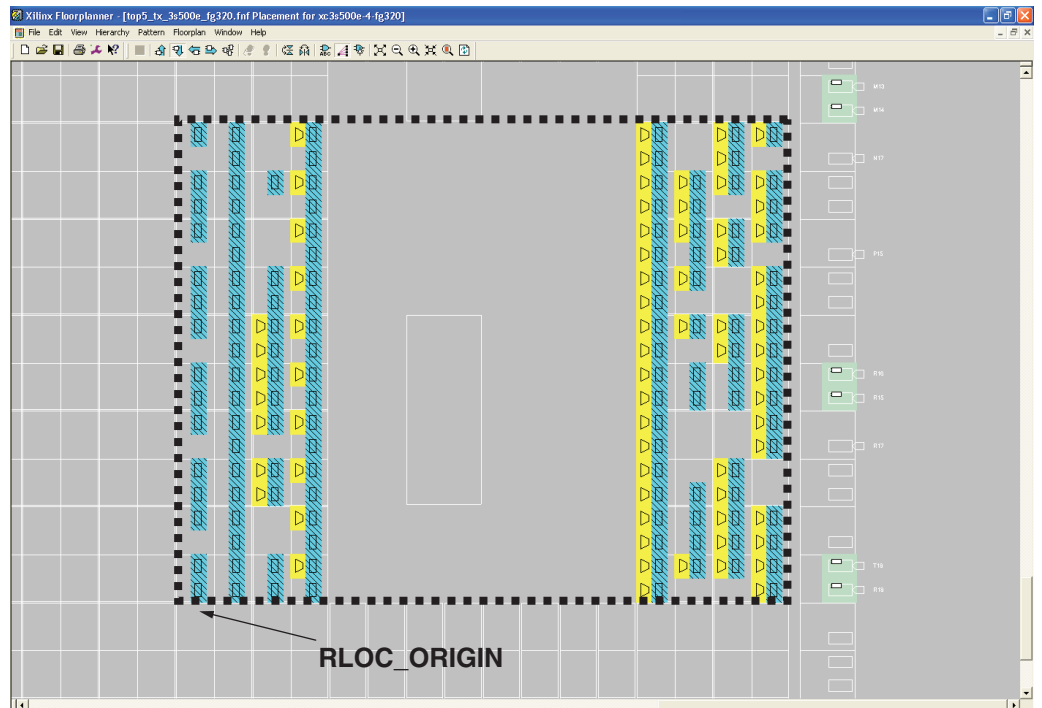
## Floorplanning the Design

The transmitter should be placed near the input pins located in bank 1 via the use of an RLOC\_ORIGIN statement in the design constraint file (\*.UCF). The 4-bit logic version of the transmitter module is actually a 4 wide x 4 high block of CLBs wrapped around a block RAM (see example in [Figure 8](#)), and the 5-bit version is slightly larger at 4 wide x 5 tall, again wrapped around the block memory (see example in [Figure 9](#)).



X486\_07\_022006

Figure 8: 4-Bit Spartan-3E Transmitter Macro



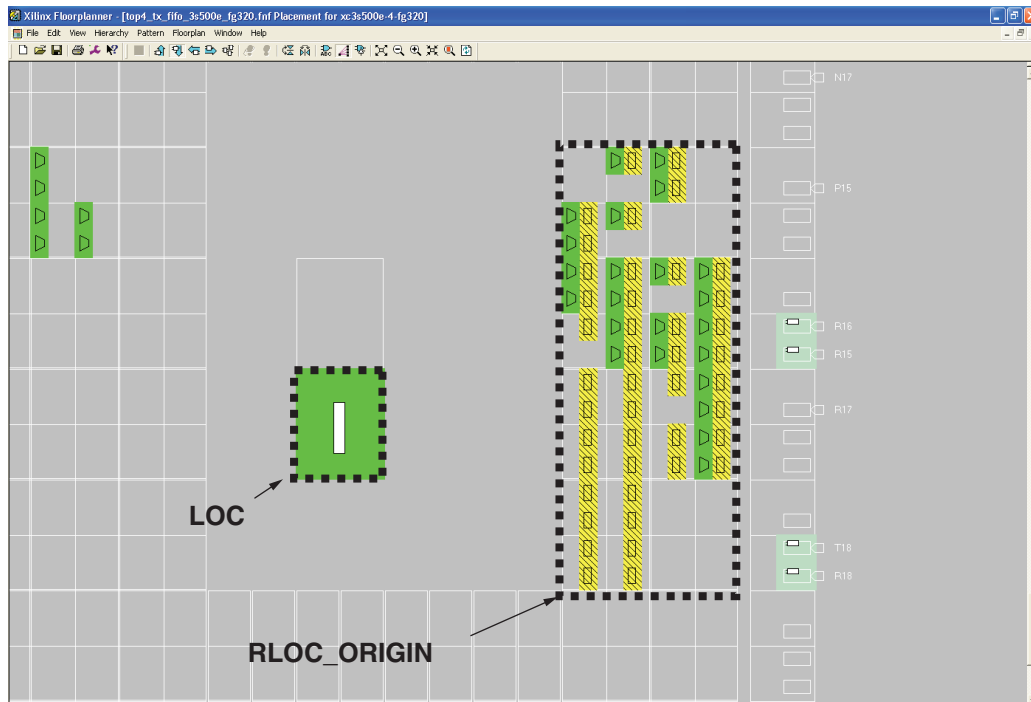
X486\_08\_022006

Figure 9: 5-Bit Spartan-3E Transmitter Macro

The smallest family device (XC3S100E) has no block memory on the right of the device to wrap around. So for this device, there is a switch in the HDL code that generates a macro where all the logic is squeezed together, leaving no room for the non-existent block RAM.

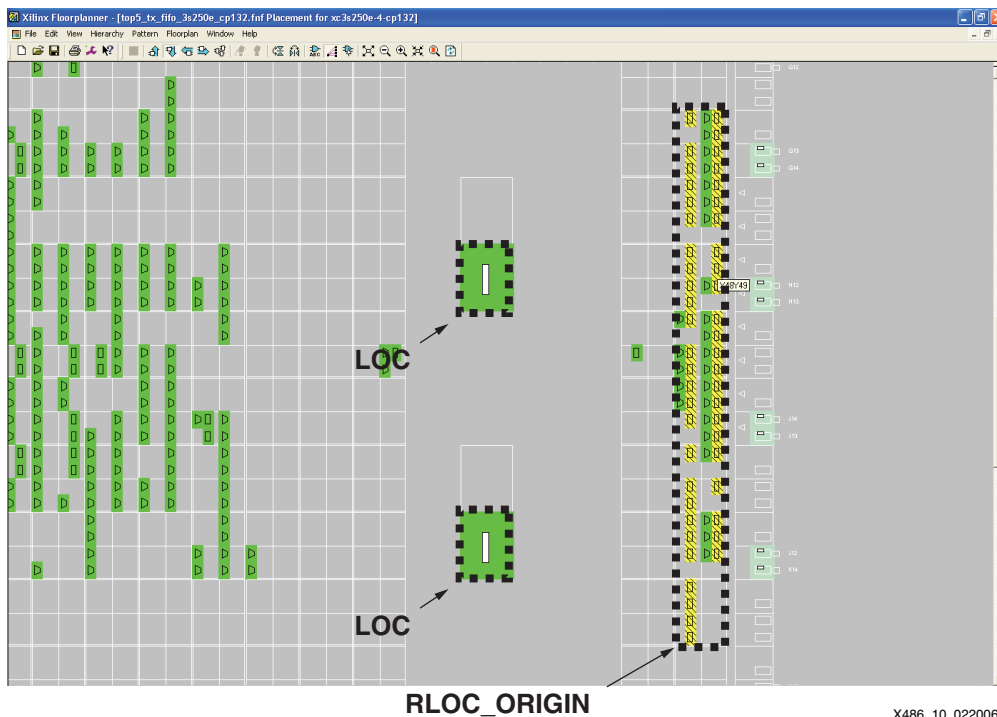


Figure 10 and Figure 11 shows the floorplans for the 4-bit and 5-bit FIFO versions of the macro, respectively. These designs require an RLOC\_ORIGIN statement in the UCF (both versions), a LOC statement for the one block RAM (4-bit version only), and two LOC statements for the two block RAMs (5-bit version only).



X486\_09\_022006

Figure 10: 4-Bit Spartan-3E Transmitter with FIFO Macro



X486\_10\_022006

Figure 11: 5-Bit Spartan-3E Transmitter with FIFO Macro

The DCM blocks have dedicated output connections to the global buffer inputs and global buffer multiplexers on the same edge of the device, either top or bottom. They are an integral part of the FPGA's global clocking infrastructure. (See Figure 12.) To ensure the dedicated connection is used, constraining the DCM is required. Additional information can be found in the DCM Locations and Clock Distribution Network Interface section of [UG331, Spartan-3 Generation FPGA User Guide](#).

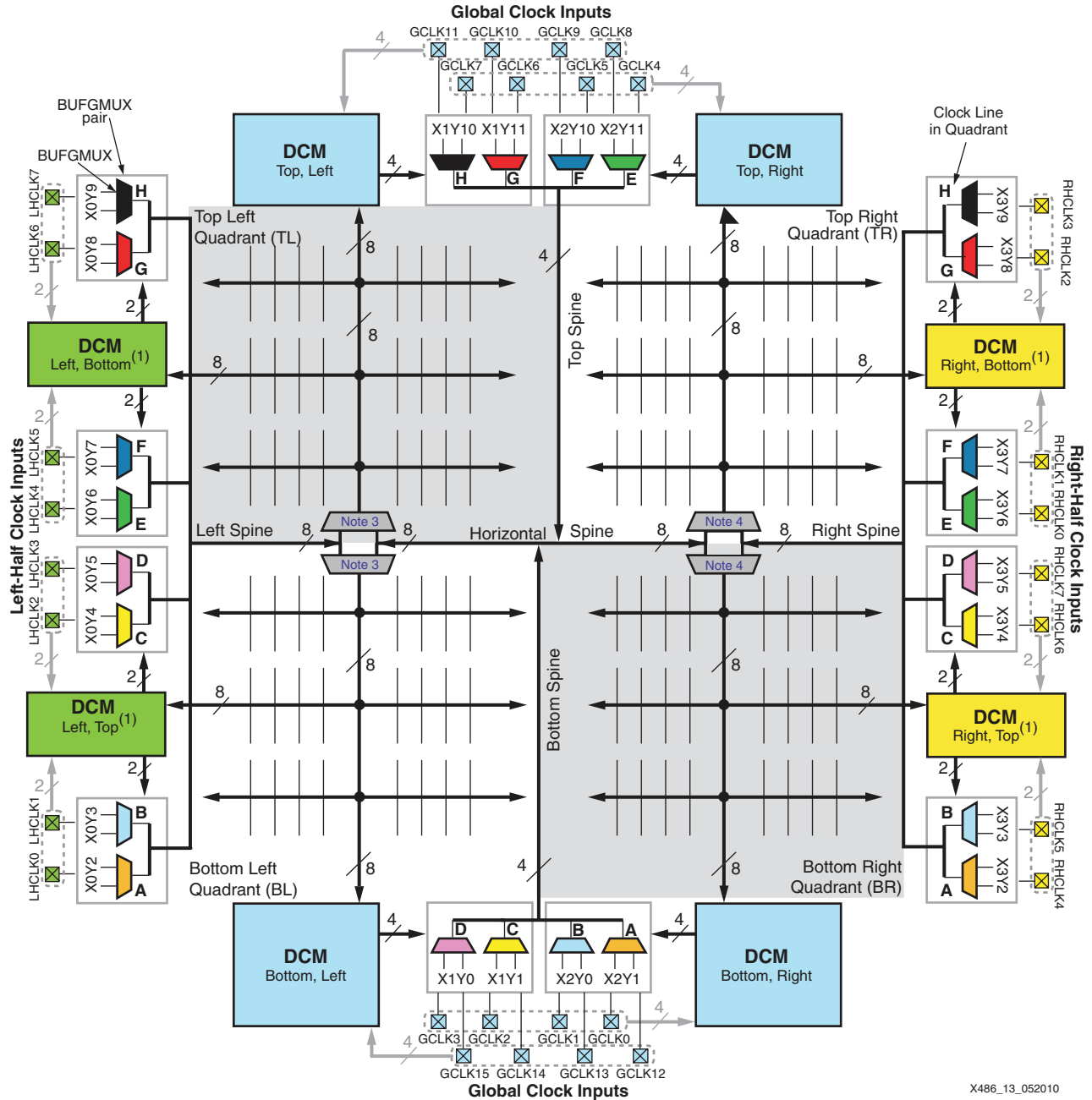


Figure 12: Spartan-3E and Extended Spartan-3A Family Internal Quadrant-Based Clock Structure

**Note:** This diagram also appears in the Global Clock Resources section of the *Spartan-3 Generation FPGA User Guide*. For more detailed information, refer to the notes and cross-references following the figure in the user guide.

Additionally, when using the DCM to generate high-speed clocks to drive the double data rate output flip-flop element, ODDR2, a specific BUFGMUX is recommended for both CLKFX and

CLKFX180 to minimize period jitter. See Table 3-17 Recommended DCM/BUFG Connections in [UG331](#), *Spartan-3 Generation FPGA User Guide*.

Incorrect DCM and BUFG placement may result in incorrect phase alignment. An example of the required .ucf constraints would be:

```
inst "dcm_rxclka" LOC = "DCM_X0Y1";
inst "rxclk35_bufg" LOC = "BUFGMUX_X0Y6";
inst "rxclk35not_bufg" LOC = "BUFGMUX_X0Y9";
```

## Conclusion

Spartan-3E/3A devices can be used in a wide variety of applications requiring 7:1 data serialization and clock forwarding at speeds up to 666 Mbps, depending on the speed grade used (see the summary in [Table 2](#)).

*Table 2: Speed Based on Family and Speed Grade*

	Spartan-3E	Spartan-3A
-4	622 Mbps	640 Mbps
-5	666 Mbps	700 Mbps

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/09/07	1.0	Initial Xilinx release.
06/21/10	1.1	Added Spartan-3A FPGA to document title and <a href="#">“Summary.”</a> Updated <a href="#">Figure 1</a> . Added new <a href="#">Figure 12</a> and associated text regarding use of proper DCM constraints and BUFG placement.

## Notice of Disclaimer

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.