SmartFusion SoC FPGA Fabric

User's Guide

Microsemi

Table of Contents

1 – Device Architecture

FPGA Architecture

This section presents an overview of the Microsemi SmartFusion® customizable system-on-chip (cSoC) FPGA fabric, which provides a general description of the FPGA architecture and routing resources.

VersaTile

Based upon successful ProASIC®3 logic architecture, SmartFusion devices provide granularity comparable to gate arrays. The SmartFusion SoC FPGA fabric consists of a sea-of-VersaTiles architecture.

As illustrated in Figure 1-1, a logic VersaTile cell has four inputs and one output. Each VersaTile can be configured using the appropriate flash switch connections:

- Any 3-input combinatorial function
- Latch with clear or set
- D-flip-flop with clear or set
- Enable D-flip-flop with clear or set (on a 4th input)

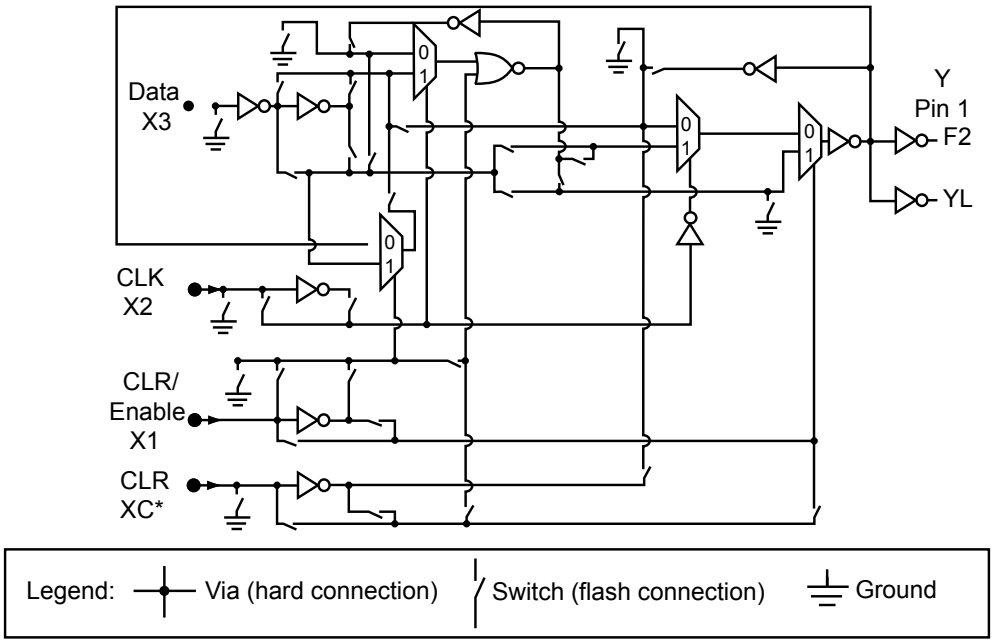
VersaTiles can flexibly map the combinatorial and sequential cells of a design. The inputs of the VersaTile can be inverted (allowing bubble pushing), and the output of the tile can connect to high-speed, very-long-line routing resources. VersaTiles and larger functions are connected with any of the four levels of routing hierarchy.

When the VersaTile is used as an enable D-flip-flop, the SET/CLR signal is supported by a fourth input, which can only be routed to the core cell over the VersaNet or global network.

The output of the VersaTile is F2 when the connection is to the ultra-fast local lines, or YL when the connection is to the efficient long-line or very-long-line resources (Figure 1-1).

For details on combinatorial and sequential modules available for use with the SmartFusion SoC FPGA fabric, refer to the *IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide*.

For module timing information, refer to the "SmartFusion DC and Switching Characteristics" chapter of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.



*Note: *This input can only be connected to the global clock distribution network.*

Figure 1-1 • **SmartFusion Core VersaTile**

Array Coordinates

During many place-and-route operations performed by the Designer software tool, it is possible to set constraints that require array coordinates. Table 1-1 is provided as a reference. The array coordinates are measured from the lower left (0, 0) corner. They can be used in region constraints for specific logic groups/blocks, designated by a wildcard, and can contain core cells, memories, and I/Os.

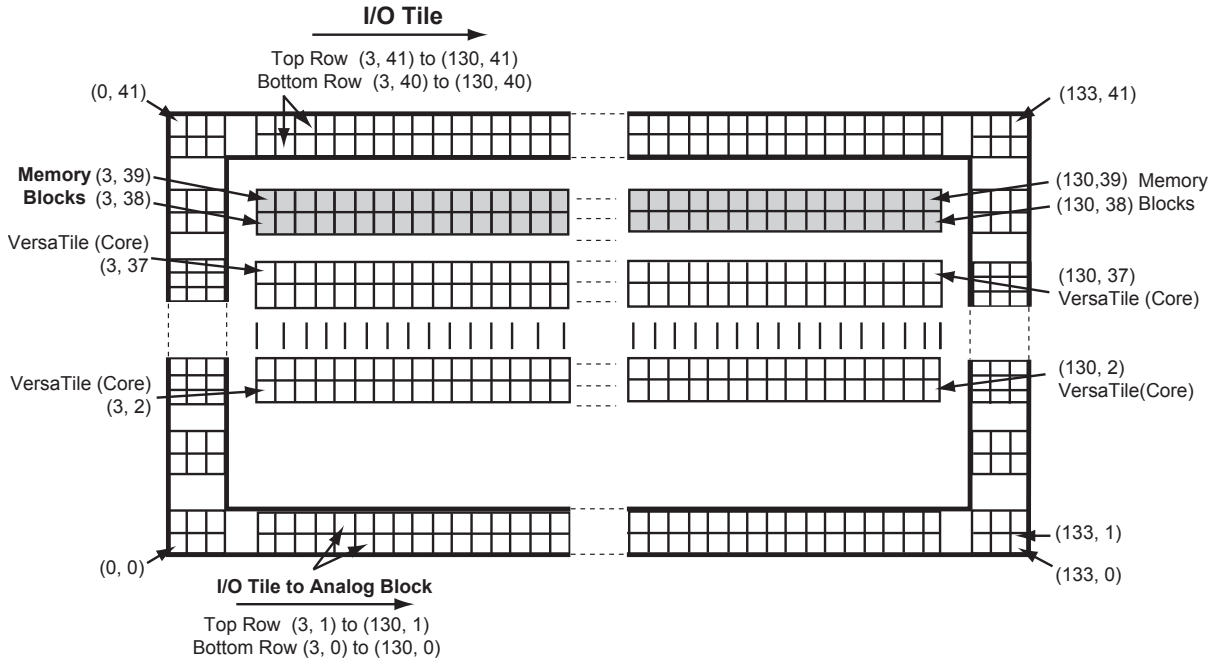
Table 1-1 provides array coordinates of core cells and memory blocks.

I/O and cell coordinates are used for placement constraints. Two coordinate systems are needed because there is not a one-to-one correspondence between I/O cells and edge core cells. In addition, the I/O coordinate system changes depending on the die/package combination. It is not listed in Table 1-1. The Designer ChipPlanner tool provides array coordinates of all I/O locations. I/O and cell coordinates are used for placement constraints. However, I/O placement is easier by package pin assignment.

Figure 1-2 illustrates the array coordinates of an A2F200 device. For more information on how to use array coordinates for region/placement constraints, see the *Designer User's Guide* or online help (available in the software) for SmartFusion software tools.

Table 1-1 • **Array Coordinates**

	VersaTiles				Memory Rows		All	
	Minimum		Maximum		Bottom	Top	Minimum	Maximum
Device	x	y	x	y	(x, y)	(x, y)	(x, y)	(x, y)
A2F060	–	–	–	–	–	–	–	–
A2F200	3	2	130	37	None	(3, 38)	(0, 0)	(133, 41)
A2F500	–	–	–	–	–	–	–	–



Note: The vertical I/O tile coordinates are not shown. West side coordinates are {(0, 2) to (2, 2)} to {(0, 41) to (2, 41)}; east side coordinates are {(131, 2) to (133, 2)} to {(131, 39) to (133, 39)}.

Figure 1-2 • **Example of Using FPGA Fabric Array Coordinates for A2F200 (does not show device feature locations)**

Routing Architecture

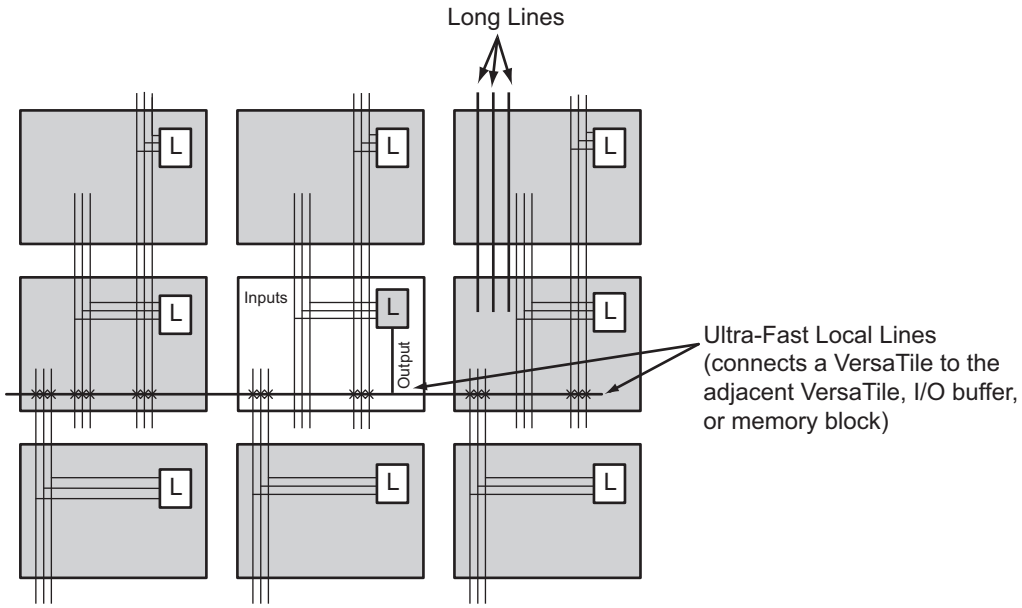
The routing structure of SmartFusion devices is designed to provide high performance through a flexible four-level hierarchy of routing resources: ultra-fast local resources; efficient long-line resources; high-speed very-long-line resources; and the high-performance VersaNet global networks.

The ultra-fast local resources are dedicated lines that allow the output of each VersaTile to connect directly to every input of the eight surrounding VersaTiles (Figure 1-3). The exception to this is that the SET/CLR input of a VersaTile configured as a D-flip-flop is driven only by the VersaNet global network.

The efficient long-line resources provide routing for longer distances and higher-fanout connections. These resources vary in length (spanning one, two, or four VersaTiles), run both vertically and horizontally, and cover the entire SmartFusion SoC FPGA fabric (Figure 1-4 on page 7). Each VersaTile can drive signals onto the efficient long-line resources, which can access every input of every VersaTile. Active buffers are inserted automatically by routing software to limit loading effects.

The high-speed very-long-line resources, which span the entire device with minimal delay, are used to route very long or high-fanout nets: a length of +/–12 VersaTiles in the vertical direction and a length of +/–16 VersaTiles in the horizontal direction from a given core VersaTile (Figure 1-5 on page 8). Very long lines in SmartFusion devices, like those in ProASIC3 devices, have been enhanced. This provides a significant performance boost for long-reach signals.

The high-performance VersaNet global networks are low-skew, high-fanout nets that are accessible from external pins or from internal logic (Figure 1-6 on page 10). These routes are typically used to distribute clocks, reset signals, and other high-fanout nets requiring minimum skew or least penalty of fanout on net delays. The VersaNet networks are implemented as clock trees, and signals can be introduced at any junction. These can be employed hierarchically, with signals accessing every input on all VersaTiles.



Note: Input to the core cell for the D-flip-flop set and reset is only available via the VersaNet global network connection.

Figure 1-3 • **Ultra-Fast Local Lines Connected to the Eight Nearest Neighbors**

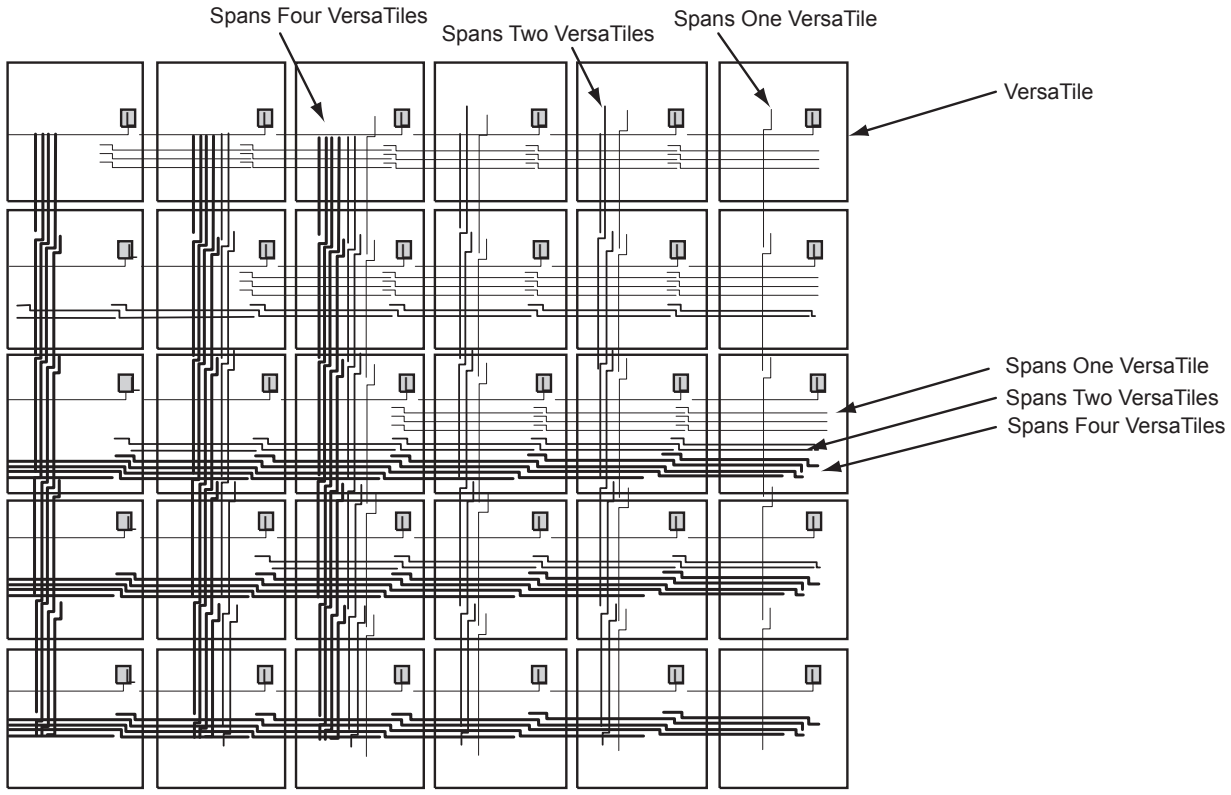


Figure 1-4 • **Efficient Long-Line Resources**

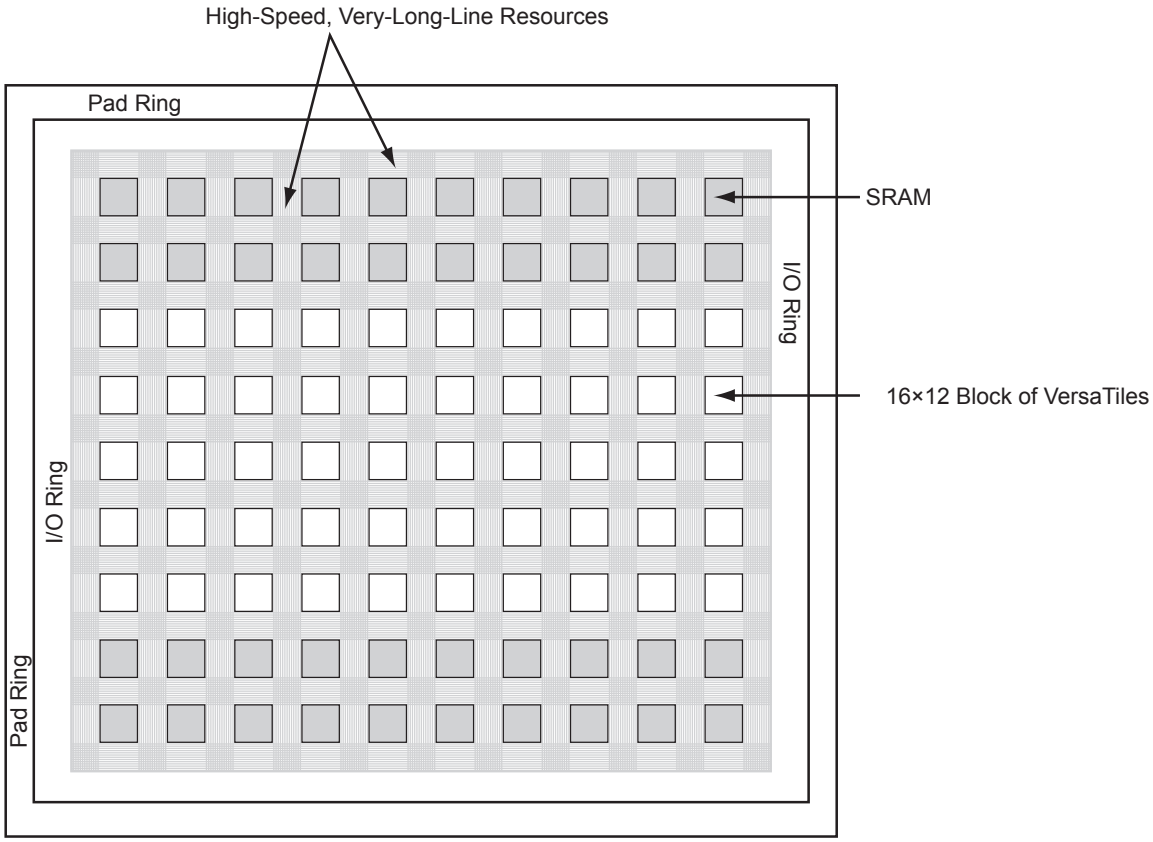


Figure 1-5 • **Very-Long-Line Resources**

Global Resources (VersaNets)

SmartFusion devices offer powerful and flexible control of circuit timing through the use of clock conditioning circuitry (CCC). The SmartFusion SoC FPGA fabric has six CCCs, which include access to at least one PLL core that is controlled by the microcontroller subsystem (MSS). The shared PLL is part of the microcontroller clock circuitry (MCC) and is configured by firmware running on the microcontroller. Users have the option of using Libero® System-on-Chip (SoC) MSS Configuration Graphical User Interface to define the MCC configuration or can use the Microsemi System Boot Firmware to achieve the same result. Alternatively, users can create custom firmware to setup the MCC Configuration Registers. For more information, refer to the appropriate clock circuitry description in the "PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators" section of the *SmartFusion Microcontroller Subsystem User's Guide*. The PLL includes a phase shifter (0°, 90°, 180°, 270°), and clock multipliers/dividers. Each CCC has all the circuitry needed for the selection and interconnection of inputs to the VersaNet global network. The east and west CCCs each have access to three VersaNet global lines on each side of the chip (six lines total). The CCCs at the four corners each have access to three quadrant global lines on each quadrant of the chip.

Advantages of the VersaNet Approach

One of the architectural benefits of SmartFusion is the set of powerful and low-delay VersaNet global networks. SmartFusion offers six chip (main) global networks that are distributed from the center of the FPGA array (Figure 1-6 on page 10). In addition, SmartFusion devices have three regional globals (quadrant globals) in each of the four chip quadrants. Each core VersaTile has access to nine global network resources: three quadrant and six chip (main) global networks. There are a total of 18 global networks on the device. Each of these networks contains spines and ribs that reach all VersaTiles in all quadrants (Figure 1-7 on page 11). This flexible VersaNet global network architecture allows users to create a total number of clock networks equal to the total number of global spines in the device. Details on the VersaNet networks are given in Table 1-2 on page 11. The flexibility of the SmartFusion VersaNet global network allows the designer to address several design requirements. User applications that are clock-resource-intensive can easily route external or gated internal clocks using VersaNet global routing networks. Designers can also drastically reduce delay penalties and minimize resource usage by mapping timing-critical, high-fanout nets to the VersaNet global network.

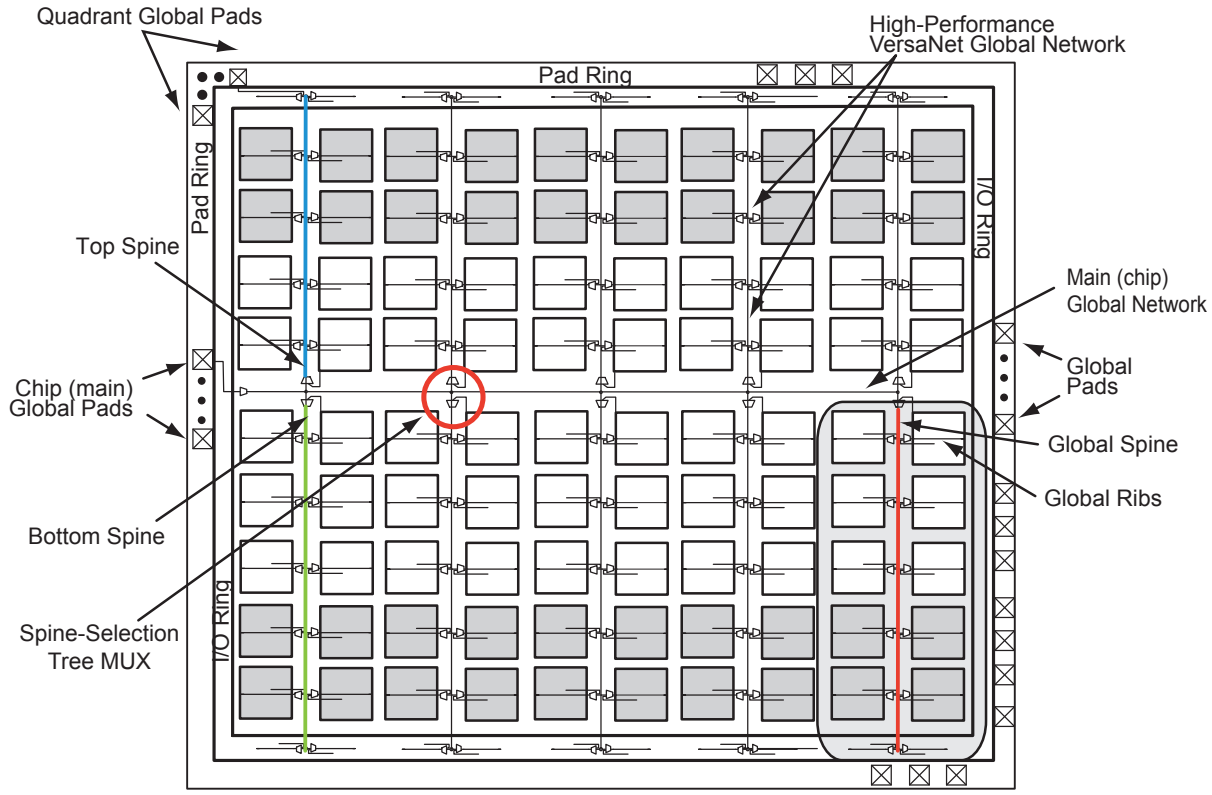


Figure 1-6 • **Overview of SmartFusion VersaNet Global Network**

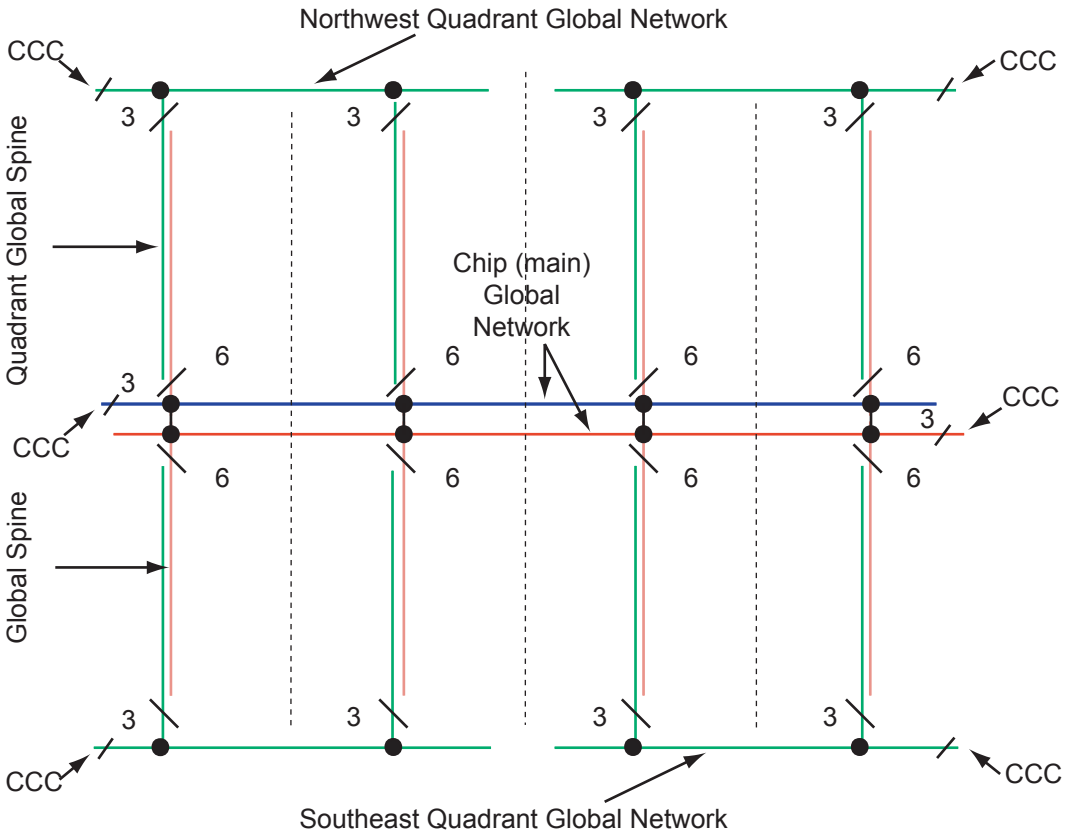


Figure 1-7 • **Global Network Architecture**

Table 1-2 • **Globals/Spines/Rows by Device**

	A2F060	A2F200	A2F500
Global VersaNets (trees)*/ Quadrant	–	9	–
VersaNet Spines/Tree	–	8	–
Total Spines	–	72	–
VersaTiles in Each Top Spine	–	384	–
VersaTiles in Each Bottom Spine	–	768	–
Total VersaTiles = 4 × (384 + 768)	–	4,608	–

Note: There are six chip (main) globals and three globals per quadrant.

VersaNet Global Networks and Spine Access

The SmartFusion architecture contains a total of 18 segmentable global networks that can access the VersaTiles, SRAM, and I/O tiles on the SmartFusion device. There are 6 chip (main) global networks that access the entire device and 12 quadrant networks (3 in each quadrant). Each device has a total of 18 globals. These VersaNet global networks offer fast, low-skew routing resources for high-fanout nets, including clock signals. In addition, these highly segmented global networks offer users the flexibility to create low-skew local networks using each available device spine for internal/external clocks or other high-fanout nets in SmartFusion devices. Optimal usage of these low-skew networks can result in significant improvement in design performance on SmartFusion devices.

The nine spines available in a vertical column reside in global networks with two separate regions of scope: the quadrant global network, which has three spines, and the chip (main) global network, which has six spines. Note that there are three quadrant spines in each quadrant of the device. There are four quadrant global network regions per device (Figure 1-7 on page 11).

The spines are the vertical branches of the global network tree, shown in Figure 1-6 on page 10. Each spine in a vertical column of a chip (main) global network is further divided into two spine segments: one in the top and one in the bottom half of the die.

Each spine and its associated ribs cover a certain area of the SmartFusion device (the "scope" of the spine; see Figure 1-6 on page 10). Each spine is accessed by the dedicated global network MUX tree architecture, which defines how a particular spine is driven—either by the signal on the global network from a CCC, for example, or another net defined by the user (Figure 1-8). Quadrant spines can be driven from user I/Os on the north and south sides of the die, via analog I/Os configured as direct digital inputs. The ability to drive spines in the quadrant global networks can have a significant effect on system performance for high-fanout inputs to a design.

Details of the chip (main) global network spine-selection MUX are presented in Figure 1-8. The spine drivers for each spine are located in the middle of the die.

Quadrant spines are driven from a north or south rib. Access to the top and bottom ribs is from the corner CCC or from the I/Os on the north and south sides of the device.

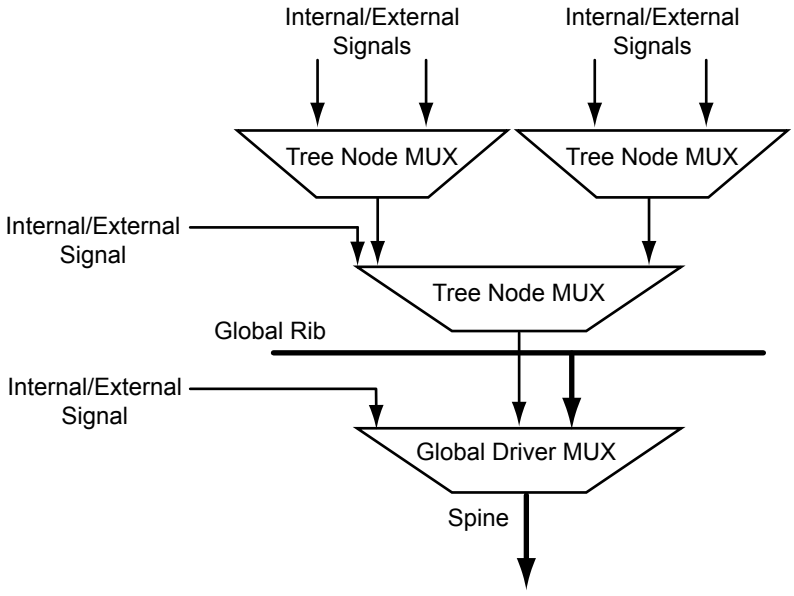


Figure 1-8 • **Spine-Selection MUX of Global Tree**

SmartFusion A2F200 VersaNet Topology

Figure 1-9 is an example of a global tree used for clock routing. The global tree presented in Figure 1-9 is driven by a CCC located on the west side of the device. It is used to drive all D-flip-flops in the device. For device-specific VersaNet timing characteristics, refer to the "DC and Switching Characteristics" chapter of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

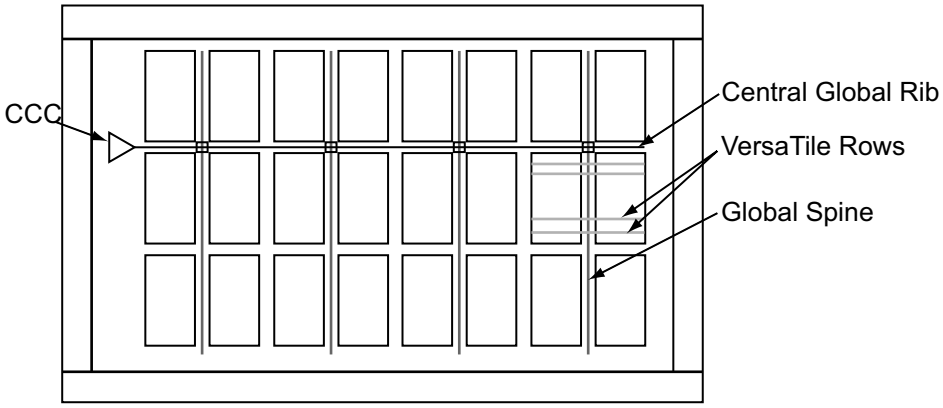


Figure 1-9 • **Example of Global Tree Used in an A2F200 Device for Clock Routing**

Clock Aggregation

Clock aggregation allows for multi-spine clock domains. A MUX tree provides the necessary flexibility to allow long lines or I/Os to build domains of one, two, or four global spines. Signal access to the clock aggregation system is achieved through long-line resources in the central rib, and also through local resources in the north and south ribs, allowing I/Os to feed directly into the clock system. As Figure 1-10 indicates, this access system is contiguous.

There is no break in the middle of the chip for north and south I/O VersaNet access. This is different from the quadrant clocks, located in these ribs, which only reach the middle of the rib. Refer to *Using Global Resources in Actel SmartFusion Devices*.

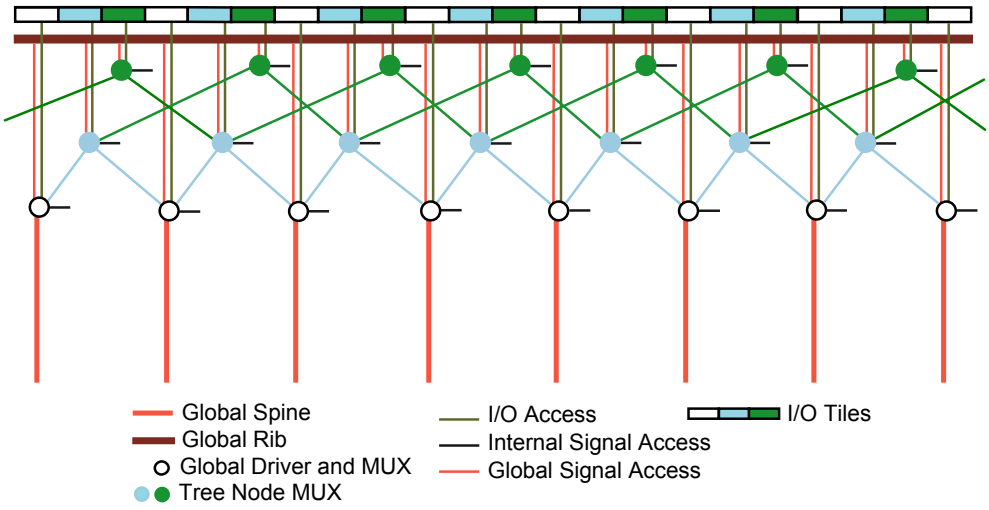


Figure 1-10 • **Clock Aggregation Tree Architecture**

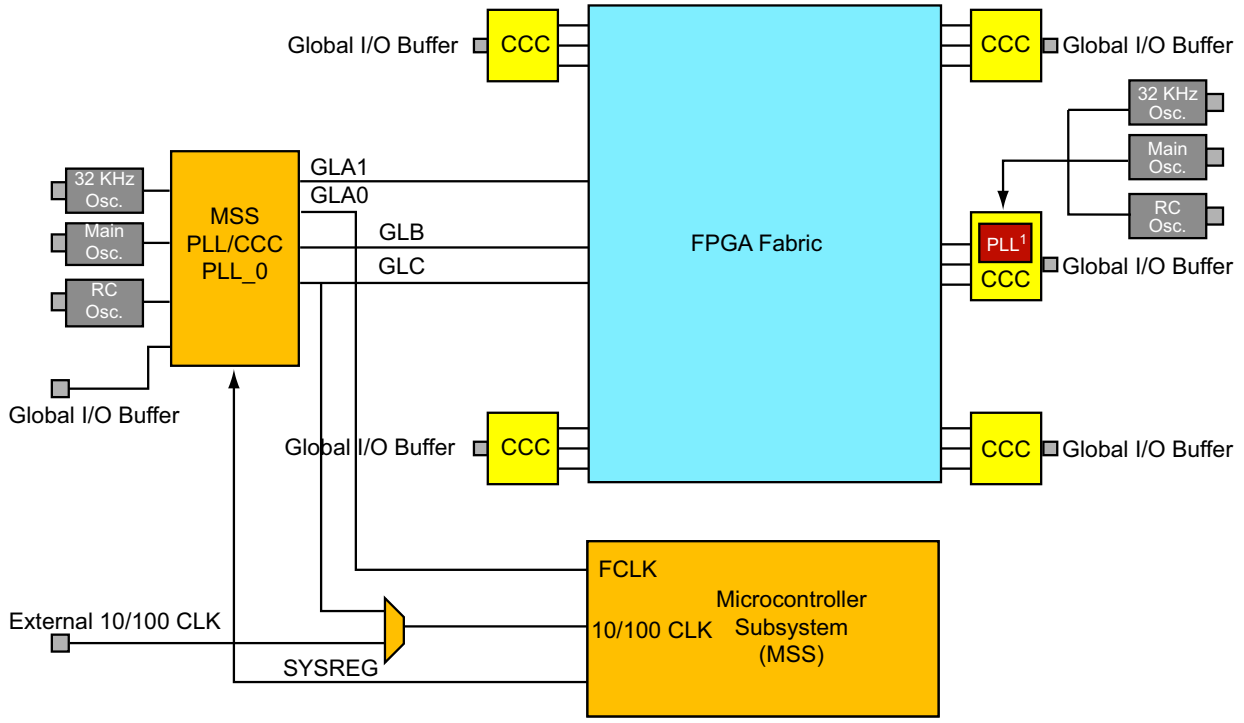
2 – Clocking Resources Available to the SmartFusion SoC FPGA Fabric

Introduction

This section describes the clocking resources available to the SmartFusion SoC FPGA fabric. Some of the resources are embedded within the SmartFusion microcontroller subsystem (MSS), but provide the FPGA fabric with access to internal and external clock signals.

The SmartFusion device family has a robust collection of clocking peripherals, some of which are shared between the SmartFusion SoC FPGA fabric and the microcontroller subsystem.

Figure 2-1 provides a top-level representation of the clocking resources available to the SmartFusion SoC FPGA fabric. As shown in Figure 2-1, there is an MSS clock conditioning circuit (CCC) that contains a PLL. This MSS CCC is primarily configured via firmware running on the ARM® Cortex™-M3 processor and is shared between the MSS and FPGA fabric. Users have the option of using Libero MSS configurator to configure the MSS CCC and Microsemi System Boot Firmware. Alternatively, users can create custom firmware to setup the MSS CCC Configuration Registers. For more information about configuring the MSS clocking resources, refer to the “PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators" section of the *SmartFusion Microcontroller Subsystem User’s Guide*. Additionally, there are five CCCs dedicated to the FPGA fabric. In the A2F200 device, the standard CCCs do not integrate a PLL. In contrast, the A2F500 device has an enhanced fabric CCC in the East (right) Middle CCC called the FAB_CCC in the Libero software. The A2F500 device in the FG484 and CS288 packages allows the use of the dedicated fabric PLL in the FAB_CCC. Although the A2F500 device in the FG256 also contains the enhanced FAB_CCC, it does not support the PLL feature and only allows clock frequency division, delayed clocks, and glitchless clock switching. For more information refer to the "Dedicated FPGA Fabric CCC with PLL (supported for A2F500)" section on page 21.



Notes:

1. *The FPGA fabric CCC with PLL is available only on A2F500 devices. The FG484 and CS288 packages feature the fabric PLL while the FG256 package does not. A2F500-FG256 users can still use the FAB_CCC with the input clock sources listed in this figure to perform clock division and/or delay, glitchless clock switching, and putting out the generated clock to the FPGA fabric.*
2. *The oscillator sources shown on both the MSS_CCC and FAB_CCC with PLL are shared, which means that the software configuration settings for both the MSS and FAB_CCC with PLL must be consistent with the physical configuration of the clock source. See the MSS Configurator and FAB_CCC Configurator Help Documents for more information about configuring the MSS_CCC and the FAB_CCC with PLL.*

Figure 2-1 • **SmartFusion Device Clocking Resources**

Table 2-1 provides a list view representation of the various clocking sources that can drive a clock signal into the SmartFusion SoC FPGA fabric grouped by the device resources used to route the input clock to the FPGA fabric.

***Table 2-1* • SmartFusion SoC FPGA Clocking Sources**

Input Clock Source	Clocking Resource Used to Route Clock to Fabric	Notes
100 MHz RC oscillator	MSS_CCC / FPGA Fabric CCC with PLL[1]	This is an internal RC oscillator.
32 KHz low-power crystal oscillator	MSS_CCC / FPGA Fabric CCC with PLL[1]	Requires external crystal.
32 KHz – 20 MHz oscillator	MSS_CCC FPGA Fabric CCC with PLL[1]	Requires external crystal or RC circuit
Derived clock routed from FPGA fabric	MSS_CCC	During Cortex-M3 boot at power-up, CLKC source is set to internal RC oscillator (100 MHz) and divided by 4 such that both GLA and GLC outputs are fixed at 25 MHz. GLB bypasses PLL and outputs clock sourced from global I/O buffer. After power-up, user firmware can reconfigure the CLKA, CLKB or CLKC sources to be sourced from the FPGA fabric. Refer to the PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators section in the *SmartFusion Microcontroller Subsystem User's Guide*.
	FPGA Fabric CCC	Microsemi primitive macros such as CLKDLY or CLKINT should be instantiated in user design.

Notes:

1. *The FPGA fabric CCC with PLL is available only on A2F500 devices. The FG484 and CS288 packages feature the fabric PLL while the FG256 package does not. A2F500-FG256 users can still use the FAB_CCC with the input clock sources listed in this table to perform clock division and/or delay, glitchless clock switching and output the generated clock to the FPGA fabric.*
2. *The microcontroller clock conditioning circuitry (MSS_CCC) can be configured using the Libero MSS configurator software tool or by user firmware running on the Cortex-M3 processor. Refer to the Libero SoC Online Help and the* PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators section in the *SmartFusion Microcontroller Subsystem User's Guide for more information. For electrical specifications and timing characteristics of the SmartFusion clocking resources, refer to the "DC and Switching Characteristics" chapter of the of the SmartFusion Customizable System-on-Chip (cSoC) datasheet.*

***Table 2-1* • SmartFusion SoC FPGA Clocking Sources**

External clock from global I/O pad	GLA1 of MSS_CCC	During Cortex-M3 boot at power-up, GLA1 output fixed at 25 MHz until user-defined configuration is applied. Refer to the PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators section in the *SmartFusion Microcontroller Subsystem User's Guide*.
	GLB of MSS_CCC	GLB can be used to output an FPGA fabric ONLY clock signal not tied to MSS clock frequency if configured to bypass the PLL and source the input clock from a CLKB global I/O Buffer. Refer to the PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators section in the *SmartFusion Microcontroller Subsystem User's Guide*.
	GLC of MSS_CCC	1) At power-up, during Cortex-M3 boot, CLKC is set to internal RC oscillator (100 MHz) and divided by 4 such that GLC output fixed at 25 MHz until user defined configuration is applied through firmware. 2) GLC optionally shared with FPGA fabric and MSS Ethernet MAC. Refer to the PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators section in the *SmartFusion Microcontroller Subsystem User's Guide*.
	FPGA Fabric CCC	Configured by Programming FPGA fabric

Notes:

1. *The FPGA fabric CCC with PLL is available only on A2F500 devices. The FG484 and CS288 packages feature the fabric PLL while the FG256 package does not. A2F500-FG256 users can still use the FAB_CCC with the input clock sources listed in this table to perform clock division and/or delay, glitchless clock switching and output the generated clock to the FPGA fabric.*
2. *The microcontroller clock conditioning circuitry (MSS_CCC) can be configured using the Libero MSS configurator software tool or by user firmware running on the Cortex-M3 processor. Refer to the Libero SoC Online Help and the* PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators section in the *SmartFusion Microcontroller Subsystem User's Guide for more information. For electrical specifications and timing characteristics of the SmartFusion clocking resources, refer to the "DC and Switching Characteristics" chapter of the of the SmartFusion Customizable System-on-Chip (cSoC) datasheet.*

Dedicated FPGA Clock Conditioning Circuits without PLL

In SmartFusion devices (except A2F060), there are five CCCs dedicated to the FPGA fabric, as shown in Figure 2-1 on page 15. The CCCs located in the corners of the device allow access to the quadrant global routing within the SmartFusion SoC FPGA fabric, while the CCCs located at the middle of the device drive chip-level global lines.

Each of the FPGA CCCs without PLL can implement up to three independent global buffers (with or without programmable delay). A global buffer can be placed in any of the three global locations (CLKA-GLA, CLKB-GLB, and CLKC-GLC) of a given CCC. The FPGA CCC block is configurable via flash configuration bits set in the programming bitstream.

Each global buffer can be driven from one of the following:

- 3 dedicated single-ended I/Os using a hardwired connection
- 2 dedicated differential I/Os using a hardwired connection
- An internal net from the FPGA fabric

Figure 2-2 provides a simplified block diagram of the physical implementation of the building blocks in each of the SmartFusion SoC FPGA CCCs without PLL.

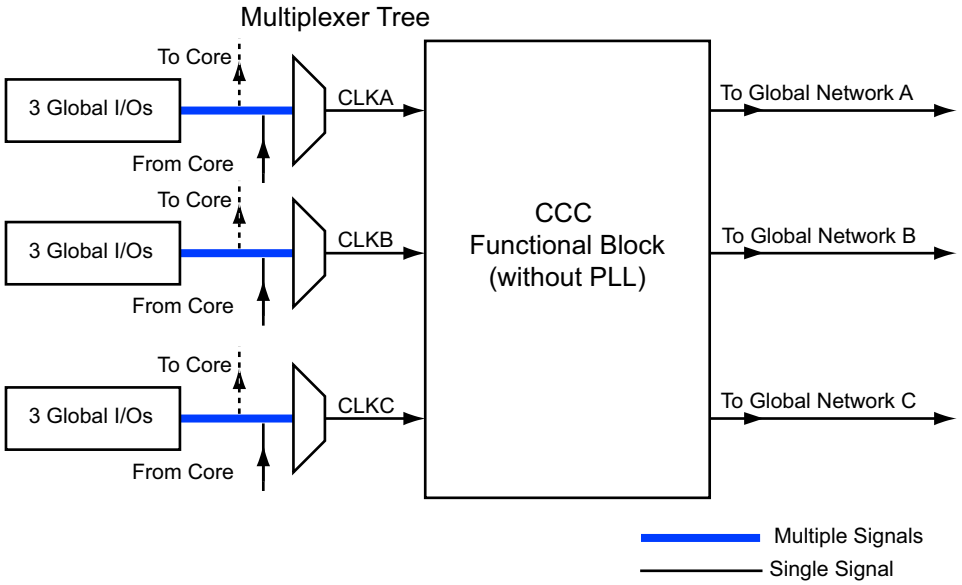


Figure 2-2 • **Overview of the FPGA CCCs Offered in SmartFusion cSoC**

Global Buffers with No Programmable Delays

The CLKBUF and CLKBUF_LVPECL/LVDS macros are composite macros that include an I/O macro driving a global buffer, hardwired together (Figure 2-3).

The CLKINT macro provides a global buffer function driven by the FPGA fabric.

The CLKBUF, CLKBUF_LVPECL/LVDS, and CLKINT macros are pass-through clock sources and do not use the PLL or provide any programmable delay functionality.

Many specific CLKBUF macros support the wide variety of single-ended and differential I/O standards supported by SmartFusion devices. The available CLKBUF macros are described in the *IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide*.

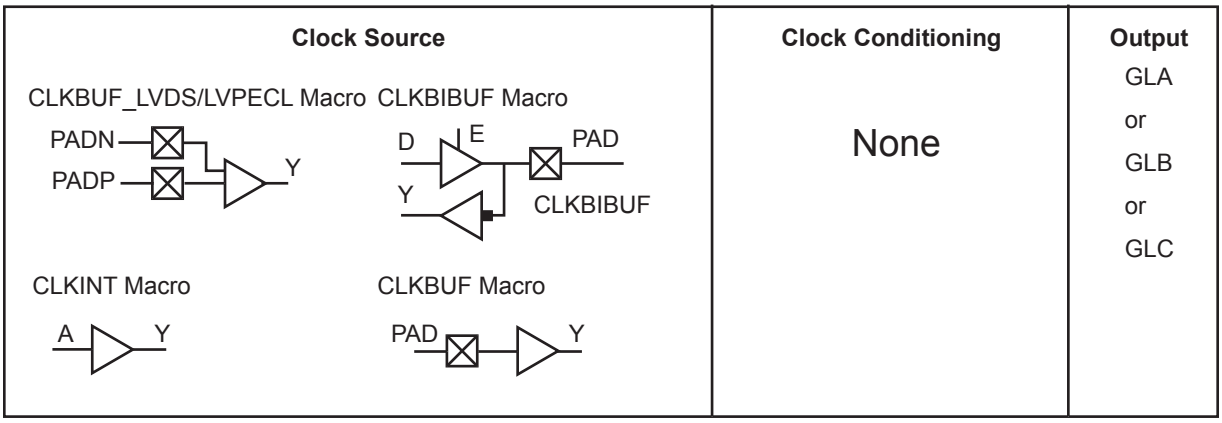


Figure 2-3 • **Global Buffers with No Programmable Delay**

Global Buffers with Programmable Delay

The CLKDLY macro is a pass-through clock source that does not use the PLL, but provides the ability to delay the clock input using a programmable delay (Figure 2-4). The CLKDLY macro takes the selected clock input and adds a user-defined delay. This macro generates an output clock phase shift from the input clock.

- The CLKDLY macro can be driven by an INBUF macro to create a composite macro, where the I/O macro drives the global buffer (with programmable delay) using a hardwired connection. In this case, the I/O must be placed in one of the dedicated global I/O locations.

  Many specific INBUF macros support the wide variety of single-ended and differential I/O standards supported by the SmartFusion family. The available INBUF macros are described in the *IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide*.
- The CLKDLY macro can be driven directly from the FPGA fabric.
- The CLKDLY macro can also be driven from an I/O that is routed through the FPGA regular routing fabric. In this case, users must instantiate a special macro, PLLINT, to differentiate from the hardwired I/O connection described earlier in the "Dedicated FPGA Clock Conditioning Circuits without PLL" section on page 18.

The visual CLKDLY configurator in the SmartGen tool of the Libero and Designer tools allows the user to select the desired amount of delay and configures the delay elements appropriately. SmartGen also allows the user to select the input clock source. SmartGen will automatically instantiate the special macro, PLLINT, when needed.

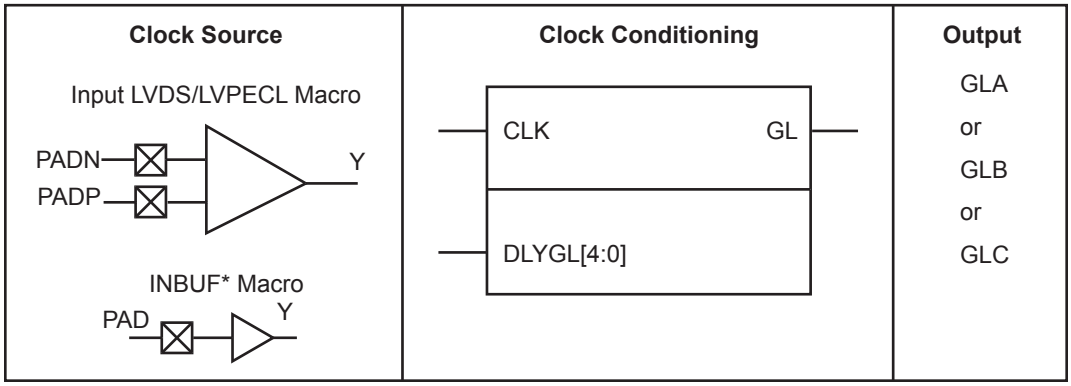


Figure 2-4 • **SmartFusion CCC Options: Global Buffers with Programmable Delay**

Dedicated FPGA Fabric CCC with PLL (supported for A2F500)

The SmartFusion A2F500 Device in the FG484 and CS288 packages supports a dedicated FPGA Fabric PLL in the East (Right) Middle Fabric CCC. This Fabric CCC with PLL can perform advanced clock conditioning in addition to the features provided by the standard Fabric CCCs without a PLL.

In the A2F500 device in the FG484 and CS288 packages, the fabric CCC with integrated PLL can be used to implement frequency division, frequency multiplication, phase shifting, and delay operations. In contrast, when using the A2F500 device in the FG256 package, the fabric CCC can only be used to implement frequency division, delay operations, and glitchless clock switching because the dedicated fabric PLL is not supported in the FG256 package.

The CCC with integrated PLL includes the following:

- One PLL core, which consists of a phase detector, a low-pass filter, and a four-phase voltage-controlled oscillator
- Three global mutliplexer blocks that steer signals from the global pads and the PLL core onto the global networks
- Six programmable delays and one fixed delay for time advance/delay adjustments
- Five programmable frequency divider blocks to provide frequency synthesis (automatically configured by the Libero Catalog's SmartGen macro builder tool)
- Three phase selectors
- One dynamic shift register that provides CCC dynamic reconfiguration capability

The CCC with integrated PLL is fully configurable. It is configured via static flash configuration bits in the array, set by the user in the programming bitstream. The Libero software contains a catalog of SmartGen Building Blocks which allows the user to instantiate and configure the PLL macro (called FAB_CCC) via a graphical configuration tool. The PLL can also be configured through an asynchronous, dedicated shift register which is dynamically accessible from the SmartFusion device. The dedicated shift register permits changes of parameters such as PLL divide ratios and delays during device operation. This dynamic configuration mode allows the user to reconfigure the PLL without the need for FPGA Fabric programming.

The dedicated fabric CCC with PLL block diagram is shown in Figure 2-5.

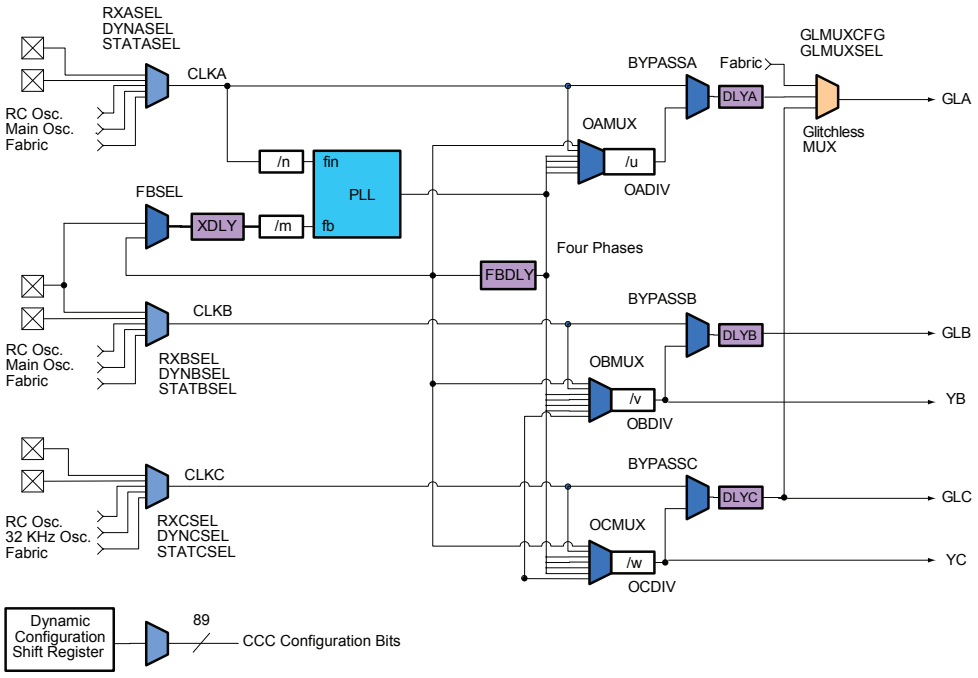
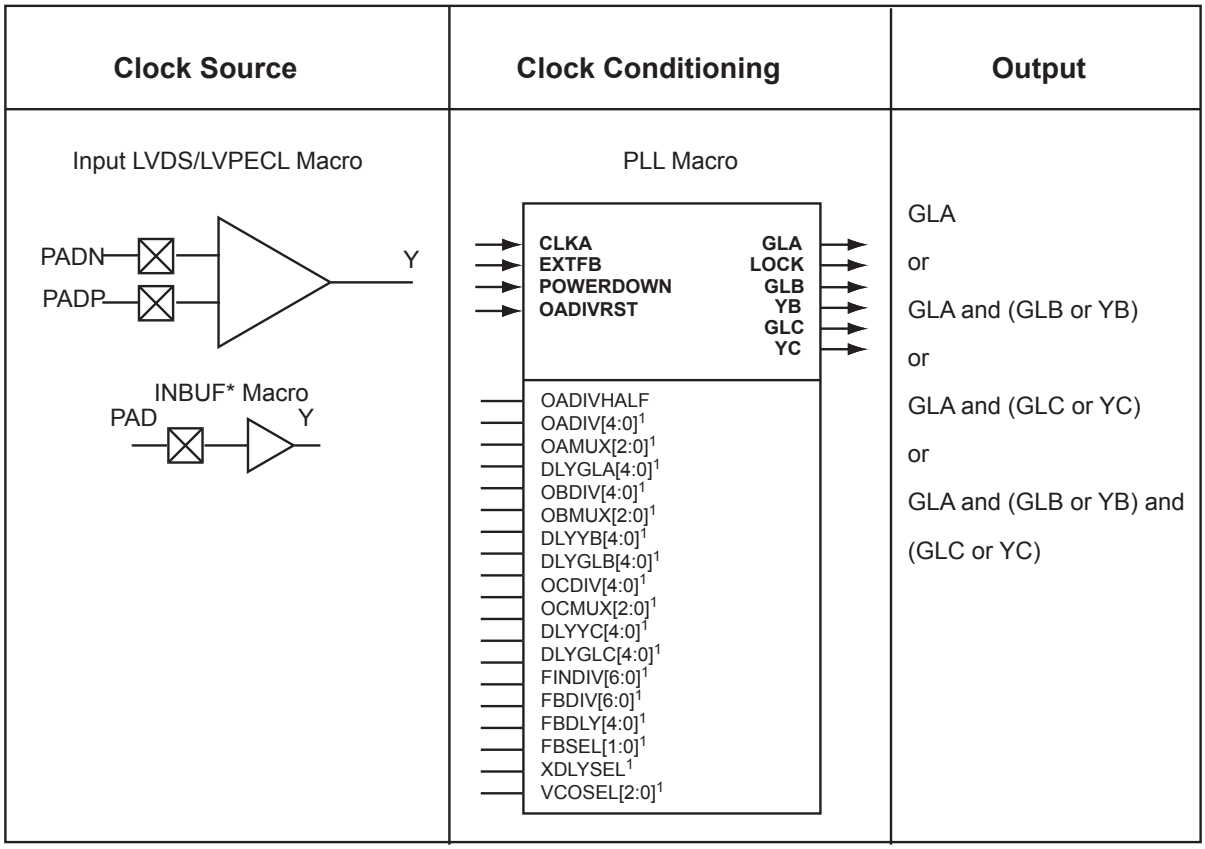


Figure 2-5 • **Block Diagram Fabric CCC with PLL**

For PLL electrical specifications, refer to the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

Global Buffers with PLL Function

The PLL functionality of the clock conditioning block is supported by the PLL macro, shown in Figure 2-6.



Notes:

1. *Refer to the IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide for more information.*
2. *For INBUF* driving a PLL macro or CLKDLY macro, the I/O will be hard-routed to the CCC; i.e., will be placed by software to a dedicated Global I/O.*
3. *Ports SDIN, SCLK, SSHIFT, SUPDATE, and MODE are also used when using the dynamic PLL configuration shift register.*

Figure 2-6 • **Overview of the FPGA Fabric CCC with PLL**

The PLL macro provides five derived clocks (three independent) from a single reference clock. The PLL macro also provides power-down input and lock output signals. The additional inputs shown on the macro are configuration settings, which are configured through the use of SmartGen. For manual setting of these bits refer to the *IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide* for details.

Figure 2-5 illustrates the various clock output options and delay elements.

PLL Macro Signal Descriptions

The PLL macro supports two inputs and up to six outputs. Table 2-2 gives a description of each signal.

***Table 2-2* • Input and Output Signals of the PLL Block**

Signal	Name	I/O	Description
CLKA	Reference Clock	Input	Reference clock input for PLL core; input clock for primary output clock, GLA
OADIVRST	Reset Signal for the Output Divider A	Input	OADIVRST can be used when you bypass the PLL core (i.e., OAMUX = 001). The purpose of the OADIVRST signals is to reset the output of the final clock divider to synchronize it with the input to that divider when the PLL is bypassed. The signal is active on a low to high transition. The signal must be low for at least one divider input. If PLL core is used, this signal is "don't care" and the internal circuitry will generate the reset signal for the synchronization purpose.
OADIVHALF	Output A Division by Half	Input	Active high. Division by half feature. This feature can only be used when users bypass the PLL core (i.e., OAMUX = 001) and the RC Oscillator (RCOSC) drives the CLKA input. This can be used to divide the 100 MHz RC oscillator by a factor of 1.5, 2.5, 3.5, 4.5... 14.5). Refer to Table 2-15 on page 38 for more information.
EXTFB	External Feedback	Input	Allows an external signal to be compared to a reference clock in the PLL core's phase detector.
PLLEN	PLL Enable	Input	Active high input that enables the PLL. When disabled, the PLL core sends 0 V signals on all of the outputs and is in power-down mode.
GLA	Primary Output	Output	Primary output clock to respective global/quadrant clock networks
GLB	Secondary 1 Output	Output	Secondary 1 output clock to respective global/quadrant clock networks
YB	Fabric 1 Output	Output	Fabric 1 output clock to local routing network
GLC	Secondary 2 Output	Output	Secondary 2 output clock to respective global/quadrant clock networks
YC	Fabric 2 Output	Output	Fabric 2 output clock to local routing network
LOCK	PLL Lock Indicator	Output	Active high signal indicating that steady-state lock has been achieved between CLKA and the PLL feedback signal

Input Clock

The following sources are available as PLL reference clocks:

- External I/O – The clock source can be any fabric I/O. The regular FPGA I/O is routed to the reference clock fabric interface pin.
- Hardwired I/O – The clock source is a dedicated global FPGA I/O.
- Hardwired I/O (LVPECL) – The clock source is a dedicated global FPGA I/O utilizing the LVPECL differential I/O standard.
- Hardwired I/O (LVDS) – The clock source is a dedicated global FPGA I/O utilizing the LVDS differential I/O standard.
- Internal logic – The clock source can be any FPGA fabric logic.
- On-chip RC Oscillator – The clock source is a dedicated 100 MHz on-chip RC oscillator available on the SmartFusion device. The on-chip RC oscillator output signal must be generated as an output of the SmartFusion MSS using the MSS clock configurator. Refer to the Libero MSS Clock Configurator Software Help document for details about exporting the on-chip RC oscillator output port.

- Main Crystal Oscillator – The source is an external crystal or an external RC circuit connected to the main crystal oscillator external pins. See the *SmartFusion Microcontroller Subsystem User's Guide* for details about how the external crystal must be connected to the SmartFusion device. This option is only available for CLKA and CLKB input clocks. The main crystal oscillator output must be generated as an output of the SmartFusion MSS using the MSS clock configurator. Refer to the Libero MSS Clock Configurator Software Help document for details about exporting the main crystal oscillator output port.
- Low Power 32 KHz Crystal Oscillator – The source is an external crystal circuit connected to the l low power 32KHz crystal oscillator external device pins. See the *SmartFusion Microcontroller Subsystem User's Guide* for details about how the external crystal must be connected to the SmartFusion device. This option is only available for the CLKC input clock.

The following lists the fabric PLL frequency requirements for the input reference clocks:

- The on-chip RC oscillator clock frequency is fixed to 100 MHz and cannot be changed.
- The main crystal oscillator (using external crystal) clock frequency must be between 1.5 MHz and 20 MHz when the PLL is used or between 32 KHz and 20 MHz when the PLL is bypassed. When using the main oscillator, use the same frequency as specified in the MSS_CCC configurator for the main crystal oscillator.
- The main crystal oscillator (external RC network configuration) clock frequency must be between 1.5 MHz and 4 MHz when the PLL is used or between 32 KHz and 4 MHz when the PLL is bypassed. Use the same frequency as the one specified in the MSS_CCC configurator for the main crystal oscillator.
- The input clock frequency must be 32 KHz when using the low power crystal oscillator reference clock source.
- For all other input sources, the source clock frequency must be between 1.5 MHz and 176 MHz when the PLL is used or between 32 KHz and 350 MHz when the PLL is bypassed.

Global Output Clocks

GLA (Primary), GLB (Secondary 1), and GLC (Secondary 2) are the outputs of Global Multiplexer 1, Global Multiplexer 2, and Global Multiplexer 3, respectively. These signals (GLx) can be used to drive the high-speed global and quadrant networks of the low power flash devices.

A global multiplexer block consists of the input routing for selecting the input signal for the GLx clock and the output multiplexer, as well as delay elements associated with that clock.

Core Output Clocks

YB and YC are known as Core Outputs and can be used to drive internal logic without using global network resources. This is especially helpful when global network resources must be conserved and utilized for other timing-critical paths.

YB and YC are identical to GLB and GLC, respectively, with the exception of a higher selectable final output delay. The SmartGen PLL Wizard will configure these outputs according to user specifications and can enable these signals with or without the enabling of Global Output Clocks.

The above signals can be enabled in the following output groupings in both internal and external feedback configurations of the static PLL:

- One output – GLA only
- Two outputs – GLA + (GLB and/or YB)
- Three outputs – GLA + (GLB and/or YB) + (GLC and/or YC)

PLL Feedback Clock

The PLL feedback source can be configured by the user. The feedback source can be internal or external, depending on the system requirements. The following is a list of the supported PLL feedback clock sources:

- Internal feedback from VCO (phase 0) without programmable delay. This is the default choice.
- Hardwired I/O. The clock source comes from a global FPGA I/O pin which has a dedicated path to drive the feedback clock.

PLL Macro Block Diagram

As illustrated, the PLL supports three distinct output frequencies from a given input clock. Two of these (GLB and GLC) can be routed to the B and C global network access, respectively, and/or routed to the device core (YB and YC).

There are five delay elements to support phase control on all five outputs (GLA, GLB, GLC, YB, and YC).

There are delay elements in the feedback loop that can be used to advance the clock relative to the reference clock.

The PLL macro reference clock can be driven in the following ways:

1. By an INBUF* macro to create a composite macro, where the I/O macro drives the global buffer (with programmable delay) using a hardwired connection. In this case, the I/O must be placed in one of the dedicated global I/O locations.
2. Directly from the FPGA core.
3. From an I/O that is routed through the FPGA regular routing fabric. In this case, users must instantiate a special macro, PLLINT, to differentiate from the hardwired I/O connection described earlier.
4. From an internal free running 100 MHz RC Oscillator.
5. From an external 32KHz to 20MHz Main Oscillator which can be externally driven by an RC Network or Crystal.
6. From an external 32KHz Low-Power Crystal Oscillator.

Refer to the SmartFusion MSS User's Guide CCC Chapter for a detailed description of the oscillator sources mentioned above.

During power-up, the PLL outputs will toggle around the maximum frequency of the voltage-controlled oscillator (VCO) gear selected. Toggle frequencies can range from 40 MHz to 250 MHz. This will continue as long as the clock input (CLKA) is constant (High or Low). This can be prevented by driving PLLEN signal Low.

The visual PLL configuration in SmartGen, a component of the Libero and Designer tools, will derive the necessary internal divider ratios based on the input frequency and desired output frequencies selected by the user.

SmartGen also allows the user to select the various delays and phase shift values necessary to adjust the phases between the reference clock (CLKA) and the derived clocks (GLA, GLB, GLC, YB, and YC). SmartGen allows the user to select the input clock source. SmartGen automatically instantiates the special macro, PLLINT, when needed.

Glitchless Clock Multiplexing

The dedicated fabric CCC with PLL also includes a glitchless multiplexor (no-glitch MUX or NGMUX) connected to the GLA output, which enables the device to switch between two asynchronous clock domains while avoiding unwanted narrow glitch pulses. For additional details about the NGMUX usage, refer to the "PLLs, Clock Conditioning Circuitry and On-Chip Crystal Oscillators" chapter of the *SmartFusion Microcontroller Subsystem User's Guide*. The following NGMUX configurations are available:

- GLA – No dynamic multiplexing is enabled and the GLA output is sourced from the CCC GLA signal only.
- GLA_GLC – Enables a glitchless switch between the CCC GLA signal and the CCC GLC signal. The output of the glitchless clock multiplexor is available on the GLA global network.
- GLA_GLMUXINT – Enables a glitchless switch between the CCC GLA signal and a signal from the FPGA fabric (on source GLMUXINT). The output of the glitchless clock multiplexor is available on the GLA global network.

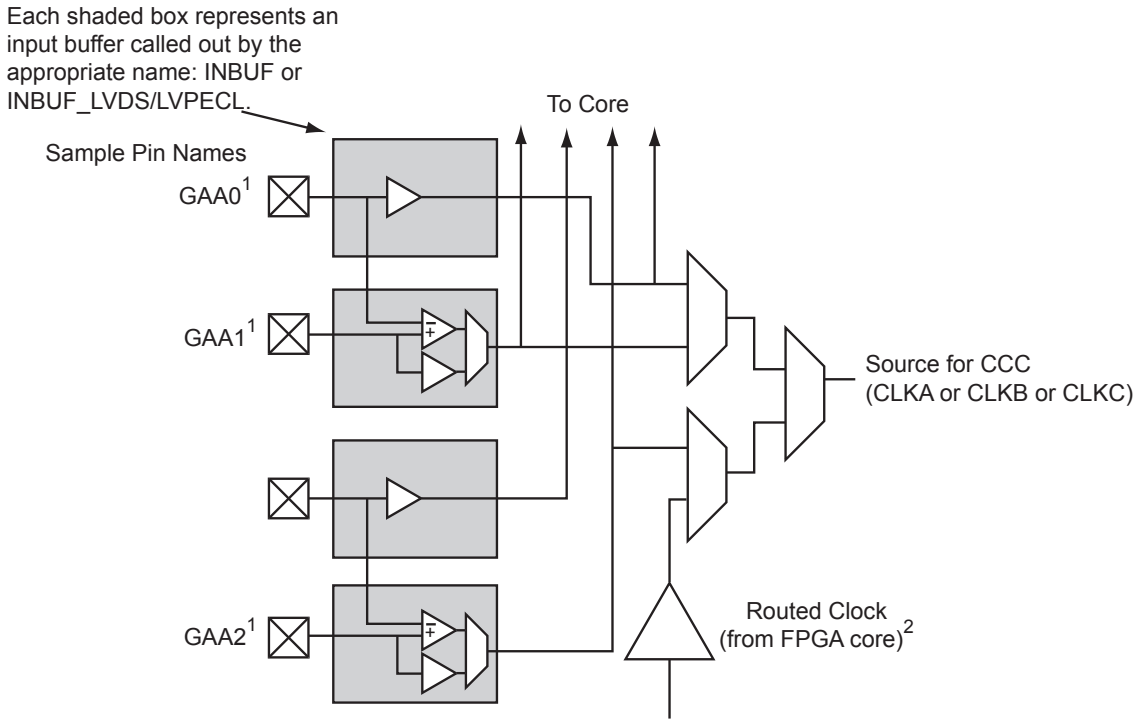
Dynamic Shift Register

The dedicated fabric CCC with PLL is dynamically reconfigurable from within the user design by using the CCC configuration dynamic shift register. When generating the FAB_CCC within the Libero SmartGen Graphical Configurator, the user must enable the Dynamic Shift Register. Once enabled, the dynamic configuration ports SDIN, SCLK, SSHIFT, SUPDATE, SDOUT and MODE will be exposed on the FAB_CCC and can be used to perform dynamic CCC reconfiguration. Refer to the "Dynamic PLL Configuration" section on page 30 for details about performing dynamic PLL configuration.

Global Input Selections

Each global buffer, as well as the PLL reference clock, can be driven from one of the following (Figure 2-7):

- 3 dedicated single-ended I/Os using a hardwired connection
- 2 dedicated differential I/Os using a hardwired connection
- The FPGA fabric



GAA[0:2]: GA represents global in the northwest corner of the device. A[0:2]: designates specific A clock source.

Notes:

1. *Represents the global input pins. Globals have direct access to the clock conditioning block and are not routed via the FPGA fabric. Refer to the "User Pins" section of the SmartFusion Customizable System-on-Chip (cSoC) datasheet for more information.*
2. *Instantiate the routed clock source input as follows:*
   a) Connect the output of a logic element to the clock input of the PLL, CLKDLY, or CLKINT macro.
   b) Do not place a clock source I/O (INBUF or INBUF_LVPECL/LVDS) in a relevant global pin location.

Figure 2-7 • **Clock Input Sources Including CLKBUF, CLKBUF_LVDS/LVPECL, and CLKINT**

If the single-ended I/O standard is selected, there is flexibility to choose one of the global input pads (the first, second, and fourth input). The other two global I/O locations are used as regular I/Os. If the differential I/O standard is chosen, the first and second inputs are considered to be paired, and the third input is paired with a regular I/O. Note, the global I/O pads do not need to feed the global network; they can also be used as regular I/O pads.

Device-Specific Layout

The SmartFusion SoC FPGA Fabric can support up to two kinds of CCCs: those with integrated PLLs and those without integrated PLLs. Table 2-3 summarizes the number of dedicated FPGA fabric CCCs available in various SmartFusion device and package combinations. Table 2-3 also lists the MSS CCC with PLL for reference, since the FPGA fabric can also be clocked by the MSS CCC.

***Table 2-3* • Number of CCCs by Device Size and Package**

Device	Package	Fabric CCC with PLL	Fabric CCC without PLL	MSS CCC with PLL
A2F060	All	0	0	1
A2F200	All	0	5	1
A2F500	FG256	0	5	1
A2F500	FG484, CS288	1	4	1

PLL Core Specifications

PLL core specifications can be found in the DC and Switching Characteristics chapter of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

Loop Bandwidth

Common design practice for systems with a low-noise input clock is to have PLLs with small loop bandwidths to reduce the effects of noise sources at the output. Table 2-4 shows the PLL loop bandwidth, providing a measure of the PLL's ability to track the input clock and jitter.

***Table 2-4* • –3 dB Frequency of the PLL**

	Minimum (T_a = +125°C, VCCA = 1.4 V)	Typical (T_a = +25°C, VCCA = 1.5 V)	Maximum (T_a = –55°C, VCCA = 1.6 V)
–3 dB Frequency	15 kHz	25 kHz	45 kHz

PLL Core Operating Principles

This section briefly describes the basic principles of PLL operation. The PLL core is composed of a phase detector (PD), a low-pass filter (LPF), and a four-phase voltage-controlled oscillator (VCO). Figure 2-8 illustrates a basic single-phase PLL core with a divider and delay in the feedback path.

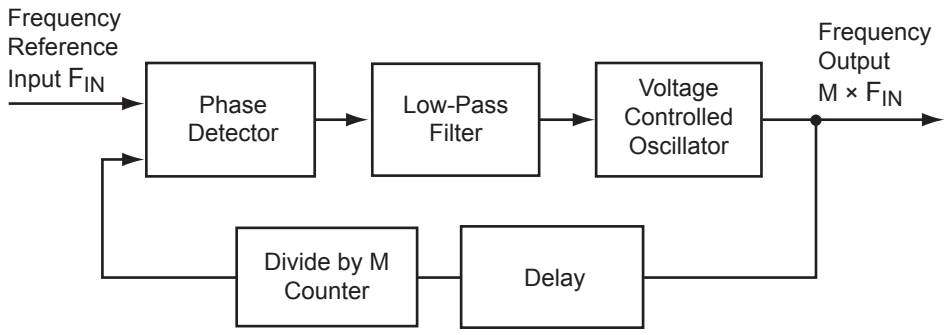


Figure 2-8 • **Simplified PLL Core with Feedback Divider and Delay**

The PLL is an electronic servo loop that phase-aligns the PD feedback signal with the reference input. To achieve this, the PLL dynamically adjusts the VCO output signal according to the average phase difference between the input and feedback signals.

The first element is the PD, which produces a voltage proportional to the phase difference between its inputs. A simple example of a digital phase detector is an Exclusive-OR gate. The second element, the LPF, extracts the average voltage from the phase detector and applies it to the VCO. This applied voltage alters the resonant frequency of the VCO, thus adjusting its output frequency.

Consider Figure 2-8 with the feedback path bypassing the divider and delay elements. If the LPF steadily applies a voltage to the VCO such that the output frequency is identical to the input frequency, this steady-state condition is known as lock. Note that the input and output phases are also identical. The PLL core sets a LOCK output signal HIGH to indicate this condition.

Should the input frequency increase slightly, the PD detects the frequency/phase difference between its reference and feedback input signals. Since the PD output is proportional to the phase difference, the change causes the output from the LPF to increase. This voltage change increases the resonant frequency of the VCO and increases the feedback frequency as a result. The PLL dynamically adjusts in this manner until the PD senses two phase-identical signals and steady-state lock is achieved. The opposite (decreasing PD output signal) occurs when the input frequency decreases.

Now suppose the feedback divider is inserted in the feedback path. As the division factor M (shown in Figure 2-9 on page 29) is increased, the average phase difference increases. The average phase difference will cause the VCO to increase its frequency until the output signal is phase-identical to the input after undergoing division. In other words, lock in both frequency and phase is achieved when the output frequency is M times the input. Thus, clock division in the feedback path results in multiplication at the output.

A similar argument can be made when the delay element is inserted into the feedback path. To achieve steady-state lock, the VCO output signal will be delayed by the input period *less* the feedback delay. For periodic signals, this is equivalent to time-advancing the output clock by the feedback delay.

Another key parameter of a PLL system is the acquisition time. Acquisition time is the amount of time it takes for the PLL to achieve lock (i.e., phase-align the feedback signal with the input reference clock). For example, suppose there is no voltage applied to the VCO, allowing it to operate at its free-running frequency. Should an input reference clock suddenly appear, a lock would be established within the maximum acquisition time.

Functional Description

This section provides detailed descriptions of PLL block functionality: clock dividers and multipliers, clock delay adjustment, phase adjustment, and dynamic PLL configuration.

Clock Dividers and Multipliers

The PLL block contains five programmable dividers. Figure 2-9 shows a simplified PLL block.

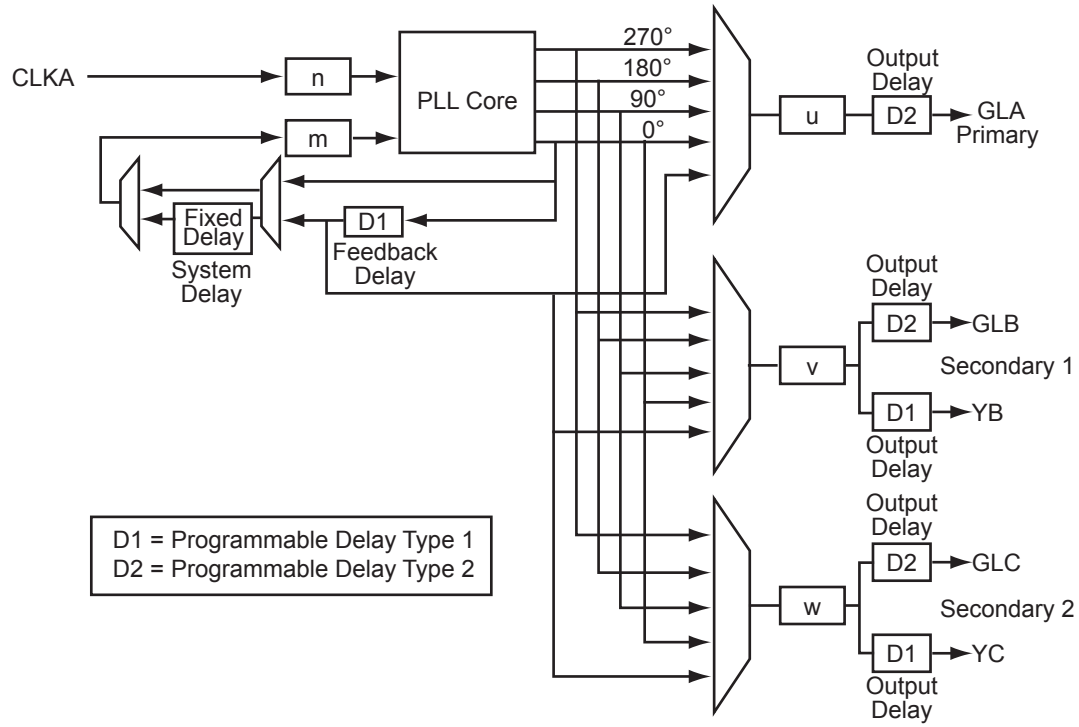


Figure 2-9 • **PLL Block Diagram**

Dividers *n* and *m* (the input divider and feedback divider, respectively) provide integer frequency division factors from 1 to 128. The output dividers *u*, *v*, and *w* provide integer division factors from 1 to 32. Frequency scaling of the reference clock CLKA is performed according to the following formulas:

$f_{GLA} = f_{CLKA} \times m / (n \times u)$ – GLA Primary PLL Output Clock

EQ 1

$f_{GLB} = f_{YB} = f_{CLKA} \times m / (n \times v)$ – GLB Secondary 1 PLL Output Clock(s)

EQ 2

$f_{GLC} = f_{YC} = f_{CLKA} \times m / (n \times w)$ – GLC Secondary 2 PLL Output Clock(s)

EQ 3

SmartGen provides a user-friendly method of generating the configured PLL netlist, which includes automatically setting the division factors to achieve the closest possible match to the requested frequencies. Since the five output clocks share the *n* and *m* dividers, the achievable output frequencies are interdependent and related according to the following formula:

$f_{GLA} = f_{GLB} \times (v / u) = f_{GLC} \times (w / u)$

EQ 4

Clock Delay Adjustment

There are a total of seven configurable delay elements implemented in the PLL architecture.

Two of the delays are located in the feedback path, entitled System Delay and Feedback Delay. System Delay provides a fixed delay of 2 ns (typical), and Feedback Delay provides selectable delay values from 0.6 ns to 5.56 ns in 160 ps increments (typical). For PLLs, delays in the feedback path will effectively advance the output signal from the PLL core with respect to the reference clock. Thus, the System and Feedback delays generate negative delay on the output clock. Additionally, each of these delays can be independently bypassed if necessary.

The remaining five delays perform traditional time delay and are located at each of the outputs of the PLL. Besides the fixed global driver delay of 0.755 ns for each of the global networks, the global multiplexer outputs (GLA, GLB, and GLC) each feature an additional selectable delay value from 0.025 ns to 0.76 ns in the first step, and then to 5.56 ns in 160 ps increments. The additional YB and YC signals have access to a selectable delay from 0.6 ns to 5.56 ns in 160 ps increments (typical). This is the same delay value as the CLKDLY macro. It is similar to CLKDLY, which bypasses the PLL core just to take advantage of the phase adjustment option with the delay value.

The following parameters must be taken into consideration to achieve minimum delay at the outputs (GLA, GLB, GLC, YB, and YC) relative to the reference clock: routing delays from the PLL core to CCC outputs, core outputs and global network output delays, and the feedback path delay. The feedback path delay acts as a time advance of the input clock and will offset any delays introduced beyond the PLL core output. The routing delays are determined from back-annotated simulation and are configuration-dependent.

Phase Adjustment

The output from the PLL core can be phase-adjusted with respect to the reference input clock, CLKA. The user can select a 0°, 90°, 180°, or 270° phase shift independently for each of the outputs YA, GLB/YB, and GLC/YC. Note that each of these phase-adjusted signals might also undergo further frequency division and/or time adjustment via the remaining dividers and delays located at the outputs of the PLL.

Dynamic PLL Configuration

The CCCs can be configured both statically and dynamically.

In addition to the ports available in the Static CCC, the Dynamic CCC has the dynamic shift register signals that enable dynamic reconfiguration of the CCC. With the Dynamic CCC, the ports CLKB and CLKC are also exposed. All three clocks (CLKA, CLKB, and CLKC) can be configured independently.

The CCC block is fully configurable. The following two sources can act as the CCC configuration bits.

Flash Configuration Bits

The flash configuration bits are the configuration bits associated with programmed flash switches. These bits are used when the CCC is in static configuration mode. Once the device is programmed, these bits cannot be modified. They provide the default operating state of the CCC.

Dynamic Shift Register Outputs

This source does not require core reprogramming and allows core-driven dynamic CCC reconfiguration. When the dynamic register drives the configuration bits, the user-defined core circuit takes full control over SDIN, SDOUT, SCLK, SSHIFT, and SUPDATE. The configuration bits can consequently be dynamically changed through shift and update operations in the serial register interface. Access to the logic core is accomplished via the dynamic bits in the specific tiles assigned to the PLLs.

Figure 2-10 illustrates a simplified block diagram of the MUX architecture in the CCCs.

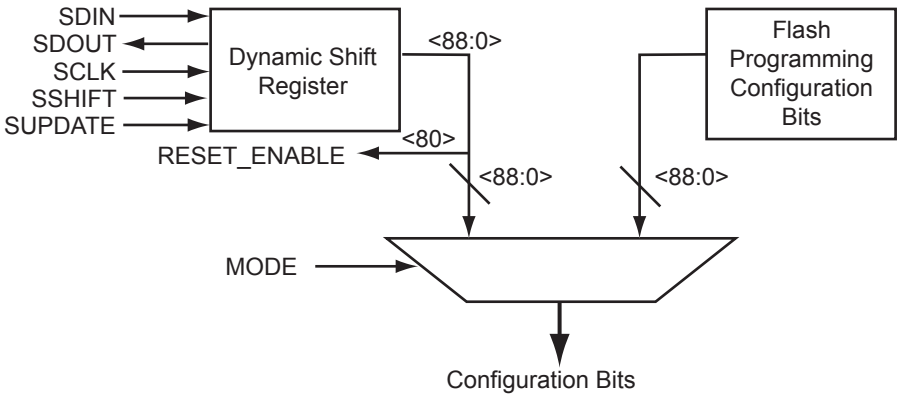


Figure 2-10 • **The CCC Configuration MUX Architecture**

The selection between the flash configuration bits and the bits from the configuration register is made using the MODE signal shown in Figure 2-10. If the MODE signal is logic HIGH, the dynamic shift register configuration bits are selected. There are 89 control bits to configure the different functions of the CCC.

Each group of control bits is assigned a specific location in the configuration shift register. For a list of the 89 configuration bits (C[88:0]) in the CCC and a description of each, refer to "PLL Configuration Bits Description" on page 33. The configuration register can be serially loaded with the new configuration data and programmed into the CCC using the following ports:

- SDIN: The configuration bits are serially loaded into a shift register through this port. The LSB of the configuration data bits should be loaded first.
- SDOUT: The shift register contents can be shifted out (LSB first) through this port using the shift operation.
- SCLK: This port should be driven by the shift clock.
- SSHIFT: The active-high shift enable signal should drive this port. The configuration data will be shifted into the shift register if this signal is HIGH. Once SSHIFT goes LOW, the data shifting will be halted.
- SUPDATE: The SUPDATE signal is used to configure the CCC with the new configuration bits when shifting is complete.

To access the configuration ports of the shift register (SDIN, SDOUT, SSHIFT, etc.), the user should instantiate the CCC macro in his design with appropriate ports. Microsemi recommends that users choose SmartGen to generate the CCC macros with the required ports for dynamic reconfiguration.

Users must familiarize themselves with the architecture of the CCC core and its input, output, and configuration ports to implement the desired delay and output frequency in the CCC structure. Figure 2-11 on page 32 shows a model of the CCC with configurable blocks and switches.

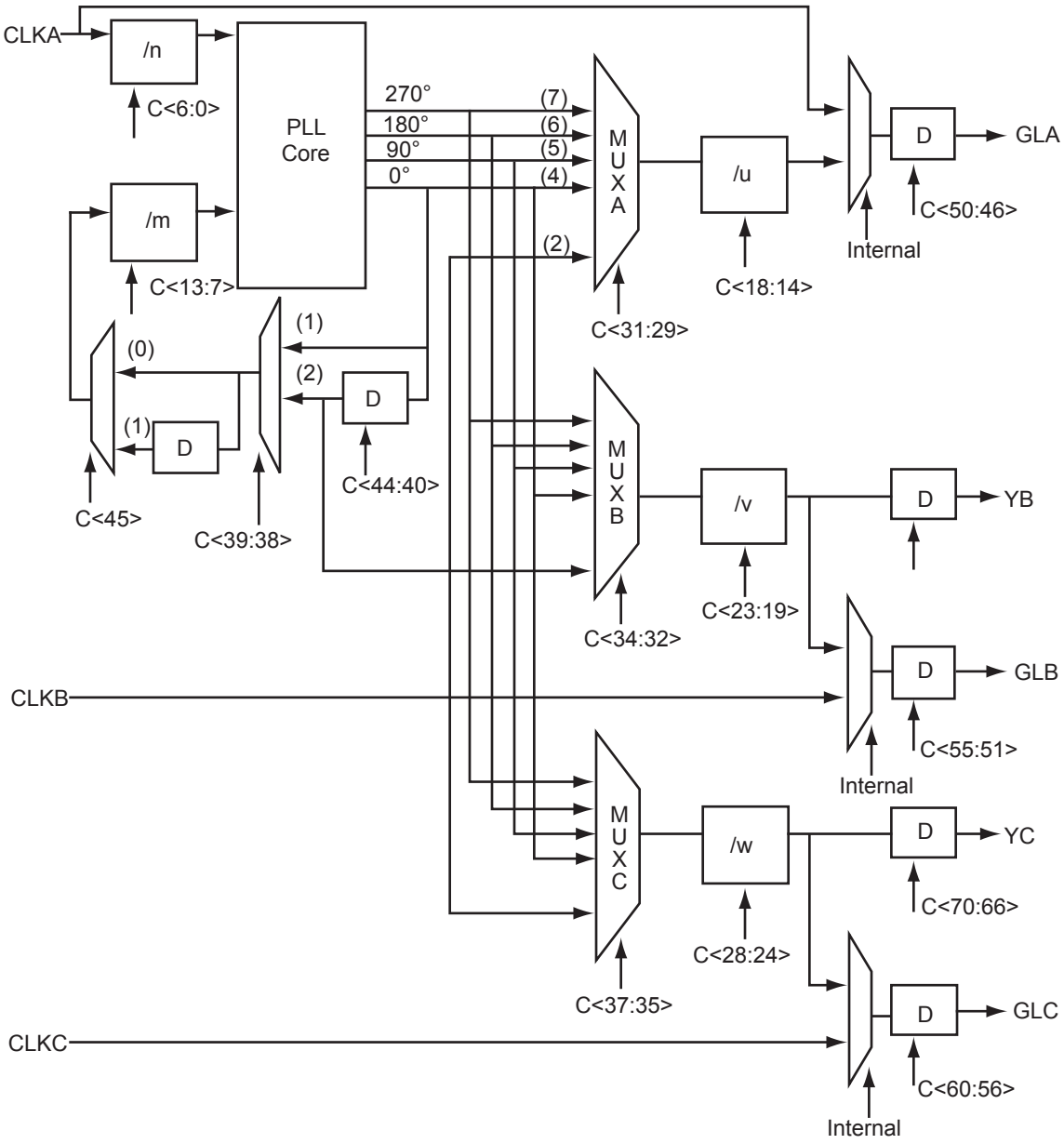


***Figure 2-11* • CCC Block Control Bits – Graphical Representation of Assignments**

Loading the Configuration Register

The most important part of CCC dynamic configuration is to load the shift register properly with the configuration bits. There are different ways to access and load the configuration shift register:

- JTAG interface
- Logic fabric
- Specific I/O tiles

JTAG Interface

The JTAG interface requires no additional I/O pins. The JTAG TAP controller is used to control the loading of the CCC configuration shift register.

Low power flash devices provide a user interface macro between the JTAG pins and the device fabric logic. This macro is called UJTAG. A user should instantiate the UJTAG macro in his design to access the configuration register ports via the JTAG pins.

For more information on CCC dynamic reconfiguration using UJTAG, refer to the "UJTAG Applications in Microsemi's Low Power Flash Devices" section of the *Fusion FPGA Fabric User's Guide*.

Logic Fabric

If the logic fabric is employed, the user must design a module to provide the configuration data and control the shifting and updating of the CCC configuration shift register. In effect, this is a user-designed TAP controller, which requires additional chip resources.

Specific I/O Tiles

If specific I/O tiles are used for configuration, the user must provide the external equivalent of a TAP controller. This does not require additional fabric resources but does use pins.

Shifting the Configuration Data

To enter a new configuration, all 81 bits must shift in via SDIN. After all bits are shifted, SSHIFT must go LOW and SUPDATE HIGH to enable the new configuration. For simulation purposes, bits <71:73> and <77:80> are "don't care."

The SUPDATE signal must be Low during any clock cycle where SSHIFT is active. After SUPDATE is asserted, it must go back to the Low state until a new update is required.

PLL Configuration Bits Description

***Table 2-5* • Configuration Bit Descriptions for the CCC Blocks**

Config. Bits	Signal	Name	Description
<88:87>	GLMUXCFG [1:0]	NGMUX configuration	The configuration bits specify the input clocks to the NGMUX (refer to Table 2-14 on page 37).*
86	OCDIVHALF	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by the divider factor in Table 2-15 on page 38.
85	OBDIVHALF	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by a 0.5 factor (refer to Table 2-15 on page 38).
84	OADIVHALF	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by certain 0.5 factor (refer to Table 2-15 on page 38).
83	RXCSEL	CLKC input selection	Select the CLKC input clock source between RC oscillator and crystal oscillator (refer to Table 2-13 on page 37).*
82	RXBSEL	CLKB input selection	Select the CLKB input clock source between RC oscillator and crystal oscillator (refer to Table 2-13 on page 37).*

*Note: *This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the CCC_Configuration report by choosing **Tools** > **Report** > **CCC_Configuration**. The report contains the appropriate settings for these bits.*

***Table 2-5* • Configuration Bit Descriptions for the CCC Blocks (continued)**

Config. Bits	Signal	Name	Description
81	RXASEL	CLKA input selection	Select the CLKA input clock source between RC oscillator and crystal oscillator (refer to Table 2-13 on page 37).*
80	RESETEN	Reset Enable	Enables (active high) the synchronization of PLL output dividers after dynamic reconfiguration (SUPDATE). The Reset Enable signal is READ-ONLY and should not be modified via dynamic reconfiguration.
79	DYNCSEL	Clock Input C Dynamic Select	Configures clock input C to be sent to GLC for dynamic control.*
78	DYNBSEL	Clock Input B Dynamic Select	Configures clock input B to be sent to GLB for dynamic control.*
77	DYNASEL	Clock Input A Dynamic Select	Configures clock input A for dynamic PLL configuration.*
<76:74>	VCOSEL[2:0]	VCO Gear Control	Three-bit VCO Gear Control for four frequency ranges (refer to Table 2-16 on page 38 and Table 2-17 on page 38).
73	STATCSEL	MUX Select on Input C	MUX selection for clock input C*
72	STATBSEL	MUX Select on Input B	MUX selection for clock input B*
71	STATASEL	MUX Select on Input A	MUX selection for clock input A*
<70:66>	DLYC[4:0]	YC Output Delay	Sets the output delay value for YC.
<65:61>	DLYB[4:0]	YB Output Delay	Sets the output delay value for YB.
<60:56>	DLYGLC[4:0]	GLC Output Delay	Sets the output delay value for GLC.
<55:51>	DLYGLB[4:0]	GLB Output Delay	Sets the output delay value for GLB.
<50:46>	DLYGLA[4:0]	Primary Output Delay	Primary GLA output delay
45	XDLYSEL	System Delay Select	When selected, inserts System Delay in the feedback path in Figure 2-9 on page 29.
<44:40>	FBDLY[4:0]	Feedback Delay	Sets the feedback delay value for the feedback element in Figure 2-9 on page 29.
<39:38>	FBSEL[1:0]	Primary Feedback Delay Select	Controls the feedback MUX: no delay, or include programmable delay element.
<37:35>	OCMUX[2:0]	Secondary 2 Output Select	Selects from the VCO's four phase outputs for GLC/YC.
<34:32>	OBMUX[2:0]	Secondary 1 Output Select	Selects from the VCO's four phase outputs for GLB/YB.
<31:29>	OAMUX[2:0]	GLA Output Select	Selects from the VCO's four phase outputs for GLA.
<28:24>	OCDIV[4:0]	Secondary 2 Output Divider	Sets the divider value for the GLC/YC outputs. Also known as divider *w* in Figure 2-9 on page 29. The divider value will be OCDIV[4:0] + 1.
<23:19>	OBDIV[4:0]	Secondary 1 Output Divider	Sets the divider value for the GLB/YB outputs. Also known as divider *v* in Figure 2-9 on page 29. The divider value will be OBDIV[4:0] + 1.
<18:14>	OADIV[4:0]	Primary Output Divider	Sets the divider value for the GLA output. Also known as divider *u* in Figure 2-9 on page 29. The divider value will be OADIV[4:0] + 1.

*Note: *This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the CCC_Configuration report by choosing* ***Tools*** *>* ***Report*** *>* ***CCC_Configuration****. The report contains the appropriate settings for these bits.*

***Table 2-5* • Configuration Bit Descriptions for the CCC Blocks (continued)**

Config. Bits	Signal	Name	Description
<13:7>	FBDIV[6:0]	Feedback Divider	Sets the divider value for the PLL core feedback. Also known as divider *m* in Figure 2-9 on page 29. The divider value will be FBDIV[6:0] + 1.
<6:0>	FINDIV[6:0]	Input Divider	Input Clock Divider (/n). Sets the divider value for the input delay on CLKA. The divider value will be FINDIV[6:0] + 1.

*Note: *This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the CCC_Configuration report by choosing **Tools** > **Report** > **CCC_Configuration**. The report contains the appropriate settings for these bits.*

Table 2-6 to Table 2-12 on page 36 provide descriptions of the configuration data for the configuration bits.

***Table 2-6* • Input Clock Divider, FINDIV[6:0] (/n)**

FINDIV<6:0> State	Divisor	New Frequency Factor
0	1	1.00000
1	2	0.50000
⋮	⋮	⋮
127	128	0.0078125

***Table 2-7* • Feedback Clock Divider, FBDIV[6:0] (/m)**

FBDIV<6:0> State	Divisor	New Frequency Factor
0	1	1
1	2	2
⋮	⋮	⋮
127	128	128

***Table 2-8* • Output Frequency Dividers**
***A* Output Divider, OADIV <4:0> (/u);**
***B* Output Divider, OBDIV <4:0> (/v);**
***C* Output Divider, OCDIV <4:0> (/w)**

OADIV<4:0>; OBDIV<4:0>; CDIV<4:0> State	Divisor	New Frequency Factor
0	1	1.00000
1	2	0.50000
⋮	⋮	⋮
31	32	0.03125

***Table 2-9* • MUXA, MUXB, MUXC**

OAMUX<2:0>; OBMUX<2:0>; OCMUX<2:0> State	MUX Input Selected
0	None. Six-input MUX and PLL are bypassed. Clock passes only through global MUX and goes directly into HC ribs.
1	Not available
2	PLL feedback delay line output
3	Not used
4	PLL VCO 0° phase shift
5	PLL VCO 90° phase shift
6	PLL VCO 180° phase shift
7	PLL VCO 270° phase shift

***Table 2-10* • 2-Bit Feedback MUX**

FBSEL<1:0> State	MUX Input Selected
0	Ground. Used for power-down mode in power-down logic block.
1	PLL VCO 0° phase shift
2	PLL delayed VCO 0° phase shift
3	N/A

***Table 2-11* • Programmable Delay Selection for Feedback Delay and Secondary Core Output Delays**

FBDLY<4:0>; DLYYB<4:0>; DLYYC<4:0> State	Delay Value
0	Typical delay = 600 ps
1	Typical delay = 760 ps
2	Typical delay = 920 ps
⋮	⋮
31	Typical delay = 5.56 ns

***Table 2-12* • Programmable Delay Selection for Global Clock Output Delays**

DLYGLA<4:0>; DLYGLB<4:0>; DLYGLC<4:0> State	Delay Value
0	Typical delay = 225 ps
1	Typical delay = 760 ps
2	Typical delay = 920 ps
⋮	⋮
31	Typical delay = 5.56 ns

***Table 2-13* • SmartFusion FAB_CCC Dynamic Clock Source Selection**

RXASEL	DYNASEL	STATASEL	Source of CLKA
0	0	X	Hardwired Global I/O
0	1	0	Hardwired Global I/O
0	1	1	Fabric / External I/O
1	0	X	RC Oscillator
1	1	X	Main Oscillator
RXBSEL	**DYNBSEL**	**STATBSEL**	**Source of CLKB**
0	0	X	Hardwired Global I/O
0	1	0	Hardwired Global I/O
0	1	1	Fabric / External I/O
1	0	X	RC Oscillator
1	1	X	Main Oscillator
RXCSEL	**DYNCSEL**	**STATCSEL**	**Source of CLKC**
0	0	X	Hardwired Global I/O
0	1	0	Hardwired Global I/O
0	1	1	Fabric / External I/O
1	0	X	RC Oscillator
1	1	X	32 KHz Oscillator

Note: STATxCEL depends on results of Layout in Designer Software. To obtain STATxCEL bit, after completing Layout, select ***Tools*** *>****Report*** *>* ***CCC Configuration.***

***Table 2-14* • SmartFusion Dynamic CCC NGMUX Configuration**

GLMUXCFG<1:0>	NGMUX Select Signal	Supported Input Clocks to NGMUX
00	0	GLA
	1	GLC
01	0	GLA
	1	GLINT
OTHERS RESERVED	DO NOT USE	RESERVED

OxDIVHALF (x = <A, B, C))

If this bit is set to 1, it divides the output frequency of the output divider defined by OxDIV by 0.5 when the PLL is bypassed with the 100 MHz RC or 32 KHz low-power oscillator. If OxDIVHALF = 1 and OxDIV = 2, the OxDIV divisor is 3, so 3 ÷ 2 = 1.5. If the GLx/Yx input is sourced from the 100 MHz RC, the output of GLx/Yx will be 100 ÷ 1.5 = 66.67 MHz. This bit is only valid if the input to the GLx/Yx divider is not being sourced by the PLL. Table 2-15 lists the only supported values for OxDIVHALF and OxDIV. Other combinations of OxDIVHALF and OxDIV can lead to unpredictable results.

***Table 2-15* • SmartFusion Dynamic CCC Division by Half Configuration**

OADIVHALF / OBDIVHALF / OCDIVHALF	OADIV<4:0> / OBDIV<4:0> / OCDIV<4:0> (in decimal)	Divider Factor	Input Clock Frequency	Output Clock Frequency (MHz)
1	2	1.5	100 MHz RC Oscillator	66.7
	4	2.5		40.0
	6	3.5		28.6
	8	4.5		22.2
	10	5.5		18.2
	12	6.5		15.4
	14	7.5		13.3
	16	8.5		11.8
	18	9.5		10.5
	20	10.5		9.5
	22	11.5		8.7
	24	12.5		8.0
	26	13.5		7.4
	28	14.5		6.9
0	0–31	1–32	Other Clock Sources	Depends on other divider settings

***Table 2-16* • Configuration Bit <76:75> / VCOSEL<2:1> Selection for SmartFusion**

	VCOSEL[2:1]							
	00		01		10		11	
Voltage	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)
1.5 V ± 5%	24	43.75	33.75	87.5	67.5	175	135	350

***Table 2-17* • Configuration Bit <74> / VCOSEL<0> Selection for SmartFusion**

VCOSEL[0]	Description
0	Fast PLL lock acquisition time with high tracking jitter. Refer to the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet for specific value and definition.
1	Slow PLL lock acquisition time with low tracking jitter. Refer to the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet for specific value and definition.

3 – SRAM and FIFO

Introduction

This section provides an overview of the embedded SRAM and FIFO features of the FPGA fabric within the SmartFusion device. This section does not discuss the embedded memory elements (eSRAM) associated with the SmartFusion microcontroller subsystem (MSS). Refer to the appropriate datasheet chapters for a description of the MSS memory features and usage.

SRAM

The SmartFusion SoC FPGA fabric has SRAM blocks along the north side of the die. To meet the needs of high-performance designs, the memory blocks operate strictly in synchronous mode for both read and write operations. The read and write clocks can be independent, and each may operate at any desired frequency less than or equal to 350 MHz. The following configurations are available:

- 4k×1, 2k×2, 1k×4, 512×9 (dual-port RAM—two read, two write or one read, one write)
- 512×9, 256×18 (two-port RAM—one read and one write)
- Sync write, sync pipelined/nonpipelined read

The SmartFusion SRAM memory block includes dedicated FIFO control logic to generate internal addresses and external flag logic (FULL, EMPTY, AFULL, AEMPTY).

During RAM operation, addresses are sourced by the user logic, and the FIFO controller is ignored. In FIFO mode, the internal addresses are generated by the FIFO controller and routed to the RAM array by internal MUXes. Refer to Figure 3-1 on page 40 for more information about the implementation of the embedded FIFO controller.

The SmartFusion architecture enables the read and write data widths of RAMs to be organized independently, allowing for bus conversion. This is done with the WW (write width) and RW (read width) pins. The different depth × width or D×W configurations are 256×18, 512×9, 1k×4, 2k×2, and 4k×1. For example, the write size can be set to 256×18 and the read size to 512×9.

Both the write and read widths for the RAM blocks can be specified independently with the WW (write width) and RW (read width) pins. The different D×W configurations are 256×18, 512×9, 1k×4, 2k×2, and 4k×1.

Refer to the allowable RW and WW values supported for each of the RAM macro types in Table 3-1 on page 42.

When a width of one, two, or four is selected, the ninth bit is unused. For example, when writing 9-bit values and reading 4-bit values, only the first four bits and the second four bits of each 9-bit value are addressable for read operations. The ninth bit is not accessible.

For the FPGA core embedded SRAM timing characteristics, refer to the "DC and Switching Characteristics" chapter of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

Conversely, when writing 4-bit values and reading 9-bit values, the ninth bit of a read operation will be undefined. The RAM blocks employ little-endian byte order for read and write operations.

Figure 3-1 shows a SmartFusion RAM block with an embedded FIFO controller.

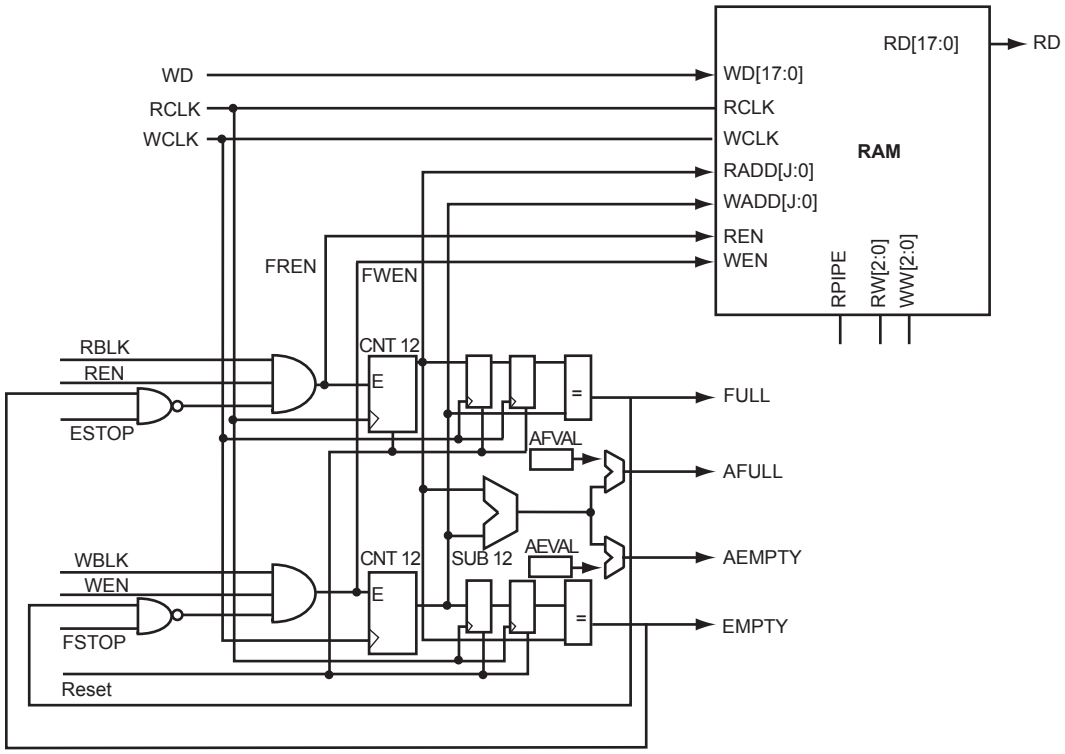


Figure 3-1 • **SmartFusion RAM Block with Embedded FIFO Controller**

RAM4K9 Description

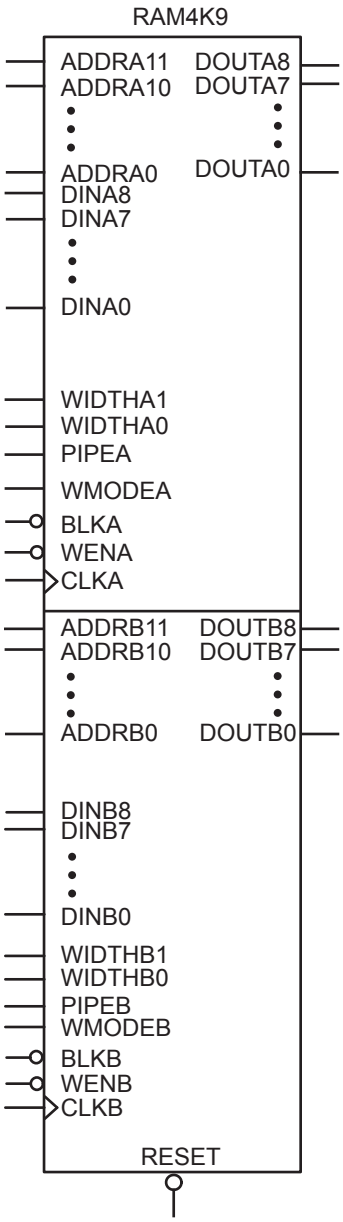


Figure 3-2 • **RAM4K9**

The following signals are used to configure the RAM4K9 memory element:

WIDTHA and WIDTHB

These signals enable the RAM to be configured in one of four allowable aspect ratios (Table 3-1).

***Table 3-1* • Allowable Aspect Ratio Settings for WIDTHA[1:0]**

WIDTHA1, WIDTHA0	WIDTHB1, WIDTHB0	D×W
00	00	4k×1
01	01	2k×2
10	10	1k×4
11	11	512×9

Note: The aspect ratio settings are constant and cannot be changed on the fly.

BLKA and BLKB

These signals are active low and will enable the respective ports when asserted. When a BLKx signal is deasserted, the associated port's outputs hold the previous value.

WENA and WENB

These signals switch the RAM between read and write mode for the respective ports. A LOW on these signals indicates a write operation, and a HIGH indicates a read.

CLKA and CLKB

These are the clock signals for the synchronous read and write operations. These can be driven independently or with the same driver.

PIPEA and PIPEB

These signals are used to specify pipelined read on the output. A Low on PIPEA or PIPEB indicates a nonpipelined read, and the data appears on the corresponding output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the corresponding output in the next clock cycle.

WMODEA and WMODEB

These signals are used to configure the behavior of the output when the RAM is in write mode. A LOW on these signals makes the output retain data from the previous read. A HIGH indicates pass-through behavior, wherein the data being written will appear immediately on the output. This signal is overridden when the RAM is being read.

RESET

This active low signal resets the output to zero, disables reads and writes from the SRAM block, and clears the data hold registers when asserted. It does not reset the contents of the memory.

ADDRA and ADDRB

These are used as read or write addresses, and they are 12 bits wide. When a depth of less than 4 k is specified, the unused high-order bits must be grounded (Table 3-2).

Table 3-2 • Address Pins Unused/Used for Various Supported Bus Widths

D×W	ADDRx	
	Unused	Used
4k×1	None	[11:0]
2k×2	[11]	[10:0]
1k×4	[11:10]	[9:0]
512×9	[11:9]	[8:0]

Note: The "x" in ADDRx implies A or B.

DINA and DINB

These are the input data signals, and they are nine bits wide. Not all nine bits are valid in all configurations. When a data width less than nine is specified, unused high-order signals must be grounded (Table 3-3).

DOUTA and DOUTB

These are the nine bit output data signals. Not all nine bits are valid in all configurations. As with DINA and DINB, high-order bits may not be used (Table 3-3). The output data on unused pins is undefined.

Table 3-3 • Unused/Used Input and Output Data Pins for Various Supported Bus Widths

D×W	DINx/DOUTx	
	Unused	Used
4k×1	[8:1]	[0]
2k×2	[8:2]	[1:0]
1k×4	[8:4]	[3:0]
512×9	None	[8:0]

Note: The "x" in DINx and DOUTx implies A or B.

RAM512X18 Description

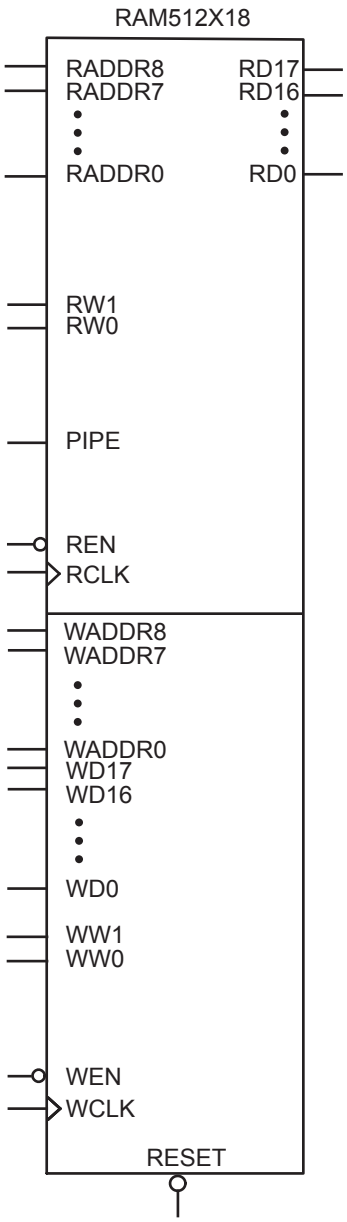


Figure 3-3 • **RAM512X18**

RAM512X18 exhibits slightly different behavior from RAM4K9, as it has dedicated read and write ports.

WW and RW

These signals enable the RAM to be configured in one of the two allowable aspect ratios (Table 3-4).

***Table 3-4* • Aspect Ratio Settings for WW[1:0]**

WW[1:0]	RW[1:0]	D×W
01	01	512×9
10	10	256×18
00, 11	00, 11	Reserved

WD and RD

These are the input and output data signals, and they are 18 bits wide. When a 512×9 aspect ratio is used for write, WD[17:9] are unused and must be grounded. If this aspect ratio is used for read, then RD[17:9] are undefined.

WADDR and RADDR

These are read and write addresses, and they are nine bits wide. When the 256×18 aspect ratio is used for write or read, WADDR[8] or RADDR[8] are unused and must be grounded.

WCLK and RCLK

These signals are the write and read clocks, respectively. They are both active high.

WEN and REN

These signals are the write and read enables, respectively. They are both active low by default. These signals can be configured as active high.

RESET

This active low signal resets the output to zero, disables reads and/or writes from the SRAM block, and clears the data hold registers when asserted. It does not reset the contents of the memory array.

PIPE

This signal is used to specify pipelined read on the output. A Low on PIPE indicates a nonpipelined read, and the data appears on the output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the output in the next clock cycle.

Clocking

The dual-port SRAM blocks are only clocked on the rising edge. SmartGen allows falling-edge-triggered clocks by adding inverters to the netlist, hence achieving dual-port SRAM blocks that are clocked on either edge (rising or falling). For dual-port SRAM, each port can be clocked on either edge or by separate clocks.

SmartFusion devices support inversion (bubble pushing) throughout the FPGA architecture, including the clock input to the SRAM modules. Inversions added to the SRAM clock pin on the design schematic or in the HDL code will be automatically accounted for during design compile without incurring additional delay in the clock path.

The two-port SRAM can be clocked on the rising edge or falling edge of WCLK and RCLK.

If negative-edge RAM and FIFO clocking is selected for memory macros, clock edge inversion management (bubble pushing) is automatically used within the SmartFusion development tools, without performance penalty.

Modes of Operation

There are two read modes and one write mode:

- Read Nonpipelined (synchronous—1 clock edge): In the standard read mode, new data is driven onto the RD bus in the same clock cycle following RA and REN valid. The read address is registered on the read port clock active edge, and data appears at RD after the RAM access time. Setting PIPE to OFF enables this mode.
- Read Pipelined (synchronous—2 clock edges): The pipelined mode incurs an additional clock delay from the address to the data but enables operation at a much higher frequency. The read address is registered on the read port active clock edge, and the read data is registered and appears at RD after the second read clock edge. Setting PIPE to ON enables this mode.
- Write (synchronous—1 clock edge): On the write clock active edge, the write data is written into the SRAM at the write address when WEN is HIGH. The setup times of the write address, write enables, and write data are minimal with respect to the write clock. Write and read transfers are described with timing requirements in the "FPGA Fabric SRAM and FIFO Characteristics” chapter of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

RAM Initialization

Each SRAM block can be individually initialized on power-up by means of the JTAG port using the UJTAG macro. Refer to the "JTAG 1532 Characteristics” section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet. The shift register for a target block can be selected and loaded with the proper bit configuration to enable serial loading. The 4,608 bits of data can be loaded in a single operation.

FIFO

FIFO4K18 Description

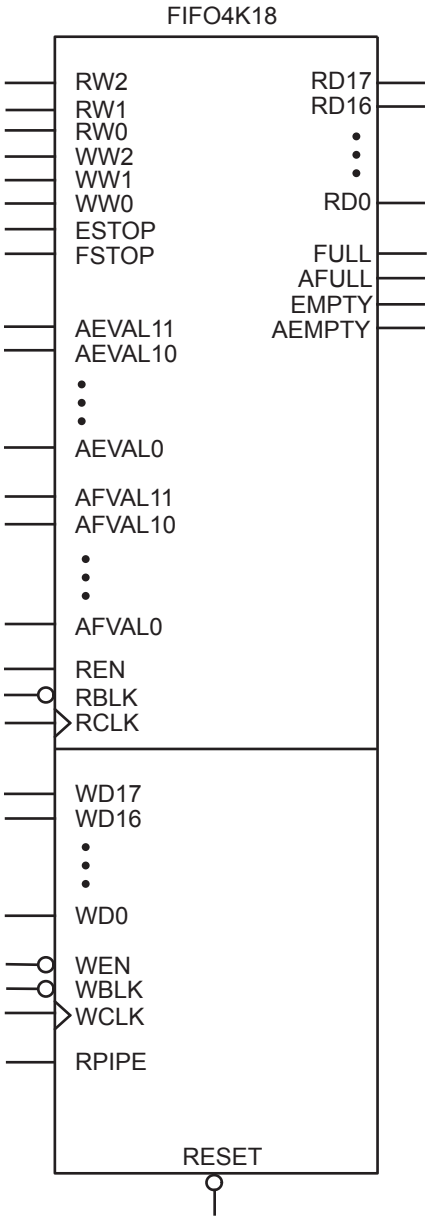


Figure 3-4 • **FIFO4KX18**

The following signals are used to configure the FIFO4K18 memory element:

WW and RW

These signals enable the FIFO to be configured in one of the five allowable aspect ratios (Table 3-5).

***Table 3-5* • Aspect Ratio Settings for WW[2:0]**

WW2, WW1, WW0	RW2, RW1, RW0	D×W
000	000	4k×1
001	001	2k×2
010	010	1k×4
011	011	512×9
100	100	256×18
101, 110, 111	101, 110, 111	Reserved

WBLK and RBLK

These signals are active low and will enable the respective ports when LOW. When the RBLK signal is HIGH, the corresponding port's outputs hold the previous value.

WEN and REN

Read and write enables. WEN is active low and REN is active high by default. These signals can be configured as active high or low.

WCLK and RCLK

These are the clock signals for the synchronous read and write operations. These can be driven independently or with the same driver.

RPIPE

This signal is used to specify pipelined read on the output. A LOW on RPIPE indicates a nonpipelined read, and the data appears on the output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the output in the next clock cycle.

RESET

This active low signal resets the output to zero when asserted. It resets the FIFO counters. It also sets all the RD pins LOW, the FULL and AFULL pins LOW, and the EMPTY and AEMPTY pins HIGH.

WD

This is the input data bus and is 18 bits wide. Not all 18 bits are valid in all configurations. When a data width less than 18 is specified, unused most significant bits (MSBs) must be grounded (Table 3-6).

***Table 3-6* • Input Data Signal Usage for Different Aspect Ratios**

D×W	WD/RD Unused
4k×1	WD[17:1], RD[17:1]
2k×2	WD[17:2], RD[17:2]
1k×4	WD[17:4], RD[17:4]
512×9	WD[17:9], RD[17:9]
256×18	–

RD

This is the output data bus and is 18 bits wide. Not all 18 bits are valid in all configurations. As with the WD bus, the MSBs become unusable if the data width is less than 18. The output data on unused pins is undefined (Table 3-6 on page 48.)

ESTOP, FSTOP

ESTOP is used to stop the FIFO read counter from further counting once the FIFO is empty (i.e., the EMPTY flag goes HIGH). A HIGH on this signal inhibits the counting.

FSTOP is used to stop the FIFO write counter from further counting once the FIFO is full (i.e., the FULL flag goes HIGH). A HIGH on this signal inhibits the counting.

For more information on these signals, refer to the "ESTOP and FSTOP Usage" section.

FULL, EMPTY

When the FIFO is full, no more data can be written and the FULL flag asserts HIGH. The FULL flag is synchronous to WCLK to inhibit writing immediately upon detection of a full condition and to prevent overflows. Since the write address is compared to a resynchronized (and thus time-delayed) version of the read address, the FULL flag will remain asserted until two WCLK active edges after a read operation eliminates the full condition.

When the FIFO is empty and no more data can be read, the EMPTY flag asserts HIGH. The EMPTY flag is synchronous to RCLK to inhibit reading immediately upon detection of an empty condition and to prevent underflows. Since the read address is compared to a resynchronized (and thus time-delayed) version of the write address, the EMPTY flag will remain asserted until two RCLK active edges after a write operation removes the empty condition.

For more information on these signals, refer to the "FIFO Flag Usage Considerations" section.

AFULL, AEMPTY

These are programmable flags and will be asserted on the threshold specified by AFVAL and AEVAL, respectively.

When the number of words stored in the FIFO reaches the amount specified by AEVAL while reading, the AEMPTY output will go HIGH. Likewise, when the number of words stored in the FIFO reaches the amount specified by AFVAL while writing, the AFULL output will go HIGH.

AFVAL, AEVAL

The AEVAL and AFVAL pins are used to specify the almost-empty and almost-full threshold values, respectively. They are 12-bit signals. For more information on these signals, refer to the "FIFO Flag Usage Considerations" section.

ESTOP and FSTOP Usage

The ESTOP pin is used to stop the read counter from counting any further once the FIFO is empty (i.e., the EMPTY flag goes HIGH). Likewise, the FSTOP pin is used to stop the write counter from counting any further once the FIFO is full (i.e., the FULL flag goes HIGH).

The FIFO counters in the SmartFusion device start the count at 0, reach the maximum depth for the configuration (511 for a 512×9 configuration, for example), and then restart at 0. An example application for the ESTOP, where the read counter keeps counting, would be writing to the FIFO once and reading the same content over and over without doing another write.

FIFO Flag Usage Considerations

The AEVAL and AFVAL pins are used to specify the 12-bit AEMPTY and AFULL threshold values, respectively. The FIFO contains separate 12-bit write address (WADDR) and read address (RADDR) counters. WADDR is incremented every time a write operation is performed, and RADDR is incremented every time a read operation is performed. Whenever the difference between WADDR and RADDR is greater than or equal to AFVAL, the AFULL output is asserted. Likewise, whenever the difference between WADDR and RADDR is less than or equal to AEVAL, the AEMPTY output is asserted. To handle different read and write aspect ratios, AFVAL and AEVAL are expressed in terms of total data bits instead of total data words. When users specify AFVAL and AEVAL in terms of read or write words, the SmartGen tool translates them into bit addresses and configures these signals automatically. SmartGen configures the AFULL flag to assert when the write address exceeds the read address by at least a predefined value. In a 2k×8 FIFO, for example, a value of 1,500 for AFVAL means that the AFULL flag will be asserted after a write when the difference between the write address and the read address reaches 1,500 (there have been at least 1,500 more writes than reads). It will stay asserted until the difference between the write and read addresses drops below 1,500.

The AEMPTY flag is asserted when the difference between the write address and the read address is less than a predefined value. In the example above, a value of 200 for AEVAL means that the AEMPTY flag will be asserted when a read causes the difference between the write address and the read address to drop to 200. It will stay asserted until that difference rises above 200. Note that the FIFO can be configured with different read and write widths; in this case, the AFVAL setting is based on the number of write data entries and the AEVAL setting is based on the number of read data entries. For aspect ratios of 512×9 and 256×18, only 4,096 bits can be addressed by the 12 bits of AFVAL and AEVAL. The number of words must be multiplied by 8 and 16, instead of 9 and 18. The SmartGen tool automatically uses the proper values. To avoid half-words being written or read, which could happen if different read and write aspect ratios are specified, the FIFO will assert FULL or EMPTY as soon as at least a minimum of one word cannot be written or read. For example, if a two-bit word is written and a four-bit word is being read, the FIFO will remain in the empty state when the first word is written. This occurs even if the FIFO is not completely empty, because in this case, a complete word cannot be read. The same is applicable in the full state. If a four-bit word is written and a two-bit word is read, the FIFO is full and one word is read. The FULL flag will remain asserted because a complete word cannot be written at this point.

For the FPGA core embedded FIFO timing characteristics, refer to the DC and AC specifications section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

4 – SmartFusion SoC FPGA User I/Os

Introduction

This section provides an overview of the digital user I/Os dedicated to the SmartFusion SoC FPGA fabric. The SmartFusion device also contains I/Os which are dedicated to the microcontroller subsystem (MSS) and I/Os which are part of the analog system. Additionally, some of the FPGA fabric I/Os can be configured to act as an external memory controller (EMC) interface to the MSS. This section discusses only the FPGA fabric I/Os; refer to the appropriate sections of the SmartFusion datasheet for a description of the MSS and analog I/Os. Refer to the "DC and Switching Characteristics" section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet for detailed electrical and timing data pertaining to the FPGA User I/Os.

SmartFusion devices feature a flexible I/O structure that supports a range of mixed voltages (1.5 V, 1.8 V, 2.5 V, and 3.3 V) through a bank-selectable voltage. Additionally, the FPGA user I/Os provide programmable slew rates, drive strengths, weak pull-up, and weak pull-down circuits. 5 V input tolerance can be achieved with some minimal external circuitry, as described in the "5 V Input Tolerance" section on page 59. Furthermore, 3.3 V LVTTL outputs are directly compatible with 5 V TTL inputs used on external devices.

All I/Os are in a known state during power-up and any power sequence is allowed. When the Digital FPGA user I/Os are not implemented in the user FPGA design, they are tristated.

The FPGA user I/O pin functions as an input, output, a tristate, or a bi-directional buffer. Input and output signal levels are compatible with the I/O standard selected. Unused FPGA I/O pins are disabled by Libero SoC software and are configured as follows:

- Output buffer is disabled (with tristate value of high impedance).
- Input buffer is disabled (with tristate value of high impedance).
- Weak pull-up is programmed.

During power-up, the used I/O pins are tristated with no pull-up or pull-down resistors until I/O is enabled (There is a delay after voltage stabilizes, and different I/O banks power up sequentially to avoid a surge of ICCI). The delay varies depending on the ramp rate of the supply voltages.

The I/Os remain tristated during the power-up until the last supply (being either VCCIBx or VCC) is powered to its functional activation voltage level. When the last supply reaches the functional voltage level, the outputs of the active I/O bank exits the tristate mode and drive the logic at the input of the output buffer. Similarly, the input buffers of the active I/O bank pass the external logic to the FPGA fabric once the last supply reaches its functional voltage level. The behavior of FPGA user I/Os is independent of the VCC and VCCIBx power-up sequence or the state of other voltage supplies of the FPGA (VPUMP and VJTAG). Similarly, during the power-down, FPGA I/Os in each bank are tristated once the first supply reaches its brownout deactivation voltage.

Some of these FPGA I/O pins are also multiplexed with integrated peripherals in the MSS (Ethernet MAC and external memory controller). Unused MSS I/Os are neither weakly pulled-up nor weakly pulled-down. The Schmitt trigger is disabled. Used MSS I/Os have the reset values as defined in Table 19-25 IOMUX_n_CR in the *SmartFusion Microcontroller Subsystem User's Guide*.

By default, during programming, FPGA and MSS I/Os become tristated and weakly pulled up to VCCI. With the VCCI and VCC supplies continuously powered up, when the device transitions from programming to operating mode, the I/Os are instantly configured to the desired user configuration. During programming, the FPGA and MSS I/Os (used or unused), except for the analog pins (AC, AT and AV), state can be specified using the "Specify I/O states during programming" option available in Flashpro software. Refer to *SmartFusion Datasheet* for more information.

Refer to *AC380: Internal Power-on Reset and Post Programming Reset Circuit for Flash-Based FPGAs Application Note* for more details on the I/Os behavior at power-up, power-down and during programming.

SmartFusion SoC FPGA fabric I/Os do not inherently support hot-insertion or cold-sparing, since the I/O VCCFPGAIOBx clamp diode is always connected to VCCFPGAIOBx. Refer to the "Cold-Sparing" section on page 58 and the "Electrostatic Discharge (ESD) Protection" section on page 59 for more information.

In addition, the registers available in the SmartFusion SoC FPGA digital I/O tile can be used to support high-performance registered inputs and outputs, with register enable if desired, as described later in this section. The registers can also be used to support the JESD-79C DDR standard within the I/O structure, as described in the "Double Data Rate (DDR) Support" section on page 57.

For I/O and global pin naming and assignments to specific I/O banks, refer to the "Pin Descriptions" section in the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

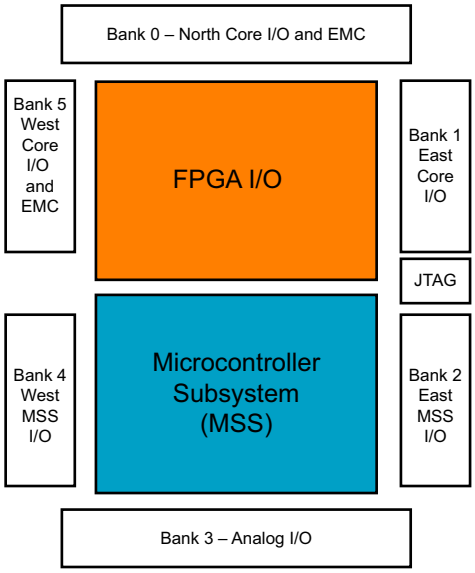


Figure 4-1 • **SmartFusion A2F200 I/O Bank Location and Naming**

I/O Banks and I/O Standards Compatibility

The SmartFusion SoC FPGA digital I/Os are grouped into I/O voltage banks. Figure 4-1 on page 52 shows the I/O bank configuration for the A2F200 device. Each I/O bank has dedicated I/O supply (VCCFPGAIOBx) and ground voltages (GNDQ for input buffers and GND for output buffers). The dedicated voltage supplies mean that only I/Os with compatible standards can be assigned to the same I/O voltage bank. I/O standards are compatible if their VCCFPGAIOBx values are identical. Refer to Table 4-1 for a list of the I/O standards supported on the FPGA fabric I/Os. Additionally, Table 4-2 lists compatible I/O standards for a given VCCFPGAIOBx voltage.

Table 4-1 • **I/O Standards Supported on FPGA I/O Banks**

I/O Bank	Single-Ended I/O Standards	Differential I/O Standards
Bank 0 – North core Bank 1 – East core Bank 5 – West core and EMC	LVTTL / LVCMOS 3.3 V LVCMOS 2.5 V / 1.8 V / 1.5 V LVCMOS 2.5/5.0 V[1] 3.3 V PCI / 3.3 V PCI-X	LVPECL LVDS (extendable to B-LVDS and M-LVDS)

Note: Bank 2 and Bank 4 belong to the MSS I/Os; Bank 3 is analog I/Os.

1. *LVCMOS 2.5/5.0 standard is similar to LVCMOS 2.5 V, with the exception that it can support up to 3.3 V on the input side (2.5 V output drive).*

***Table 4-2* • SmartFusion VCCFPGAIOBx Voltages and Compatible Standards**

VCCFPGAIOBx (Typical)	Compatible Standards
3.3 V	LVTTL/LVCMOS 3.3 V, PCI 3.3 V / PCI-X 3.3 V, LVPECL
2.5 V	LVCMOS 2.5 V, LVCMOS 2.5 / 5.0 V, LVDS
1.8 V	LVCMOS 1.8 V
1.5 V	LVCMOS 1.5 V

Features Supported on All Digital FPGA User I/Os

Table 4-3 lists all features supported by the transmitter and receiver for single-ended and differential I/Os.

***Table 4-3* • SmartFusion SoC FPGA Fabric I/O Features**

Feature	Description
Single-ended transmitter features	• Weak pull-up and pull-down • Two slew rates (Low, High) • Five drive strengths (2 mA, 4 mA, 8 mA, 16 mA, and 24 mA) • Skew between output buffer enable/disable time: 1.2 ns delay (rising edge) and 0 ns delay (falling edge); see Table 4-7 on page 64 for more information • LVTTL/LVCMOS 3.3 V outputs compatible with external 5 V TTL inputs (see "5 V Output Tolerance" section on page 64) • High performance (Table 4-5 on page 54)
Single-ended receiver features	• 5 V tolerant with use of minimal external circuitry • ESD protection • High performance (Table 4-5 on page 54) • Separate ground planes, GND/GNDQ for input and output buffers to avoid output-induced noise in the input circuitry
CMOS-style LVDS or LVPECL transmitter	• Two I/Os and external resistors are used to provide a CMOS-style LVDS or LVPECL transmitter solution (refer to the "Differential I/O Characteristics" section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet). • LVDS transmitter solution can also support bus LVDS (B-LVDS) and multipoint LVDS (M-LVDS). Refer to the "Differential I/O Characteristics" section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet. • Weak pull-up and pull-down • Fast slew rate
LVDS/LVPECL differential receiver	• ESD protection • High performance (Table 4-5 on page 54) • Separate input buffer ground and power planes to avoid output-induced noise in the input circuitry

Multiplexed I/Os Used for FPGA Fabric and External Memory Controller (EMC) Interface

Some of the SmartFusion SoC FPGA fabric user I/Os are shared between the FPGA fabric and the microcontroller subsystem's (MSS) external memory controller (EMC). These multiplexed I/O pins can be configured to act as an external memory controller (EMC), enabling the ARM Cortex-M3 processor to seamlessly interface to standard external memory chips via the internal EMC.

When the EMC is used, those I/O pins are no longer available to act as FPGA fabric I/Os. This means that user I/O signals must be placed on other available FPGA fabric I/Os. When the EMC is used, a total of 50 FPGA fabric I/Os will be dedicated to the EMC and are no longer available to user FPGA I/O signals. The EMC signals implement external memory control and data signals, including a memory clock, read/write enables, chip selects, memory address and data buses. The multiplexed I/Os are located on North I/O Bank 0 (for EMC address and control signals) and West I/O Bank 5 (for EMC Data signals) as shown in Figure 4-1 on page 52. To determine the specific pins which are used for the EMC, refer to the "User Pins" section in the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet and the pin assignment table for the specific device package in use.

The pins will be identified in the format shown below:

EMC_xxxx / IOuxwBy

When the EMC shares a global User I/O pin, the format matches the example below:

EMC_CLK / Gmn / IOuxwBy

Note that the I/O configuration is set when the SmartFusion SoC FPGA is programmed. The default configuration is FPGA fabric routed to EMC I/O (refer to the EMC_MUX_CR table in the *SmartFusion Microcontroller Subsystem User's Guide*). I/Os configured as EMC I/Os have their configuration fixed, as shown in Table 4-4. and are not affected by system reboot.

***Table 4-4* • I/O Standard and Attribute Fixed Configuration of EMC I/Os**

I/O Standard	SLEW (output only)	OUT_DRIVE (output only)	SKEW (TRIBUF and BIBUF macros with OE)	RES_PULL	COMBINE_REGISTER
LVTTL/LVCMOS 3.3 V	High	12 mA	Off	None	–

***Table 4-5* • Maximum I/O Frequency for Single-Ended and Differential I/Os (maximum drive strength and high slew rate selected)**

Specification	Performance Up To
LVTTL/LVCMOS 3.3 V	200 MHz
LVCMOS 2.5 V	250 MHz
LVCMOS 1.8 V	200 MHz
LVCMOS 1.5 V	130 MHz
PCI	200 MHz
PCI-X	200 MHz
LVDS	350 MHz
LVPECL	300 MHz

Shared MSS I/O Status During MSS Reset

This section summarizes what happens to fabric shared MSS I/Os during MSS resets. The default configurations and reset states for each fabric shared MSS I/O are covered.

When MSS Reset (MSS_SYSTEM_RESET_N) is asserted:

- All MSS I/O pins will revert back to the MSS default configuration.
- All MSS I/O pins will remain in the default configurations until the MSS reset is deasserted and system boot code is executed.
- The default configurations of fabric shared MSS I/Os are as follows:

  GPIO

  – Default configuration is GPIO routed to MSSIOBUF.
  – Schmitt trigger and pull-up/-down are disabled.
  – Output driver is disabled.

  EMC

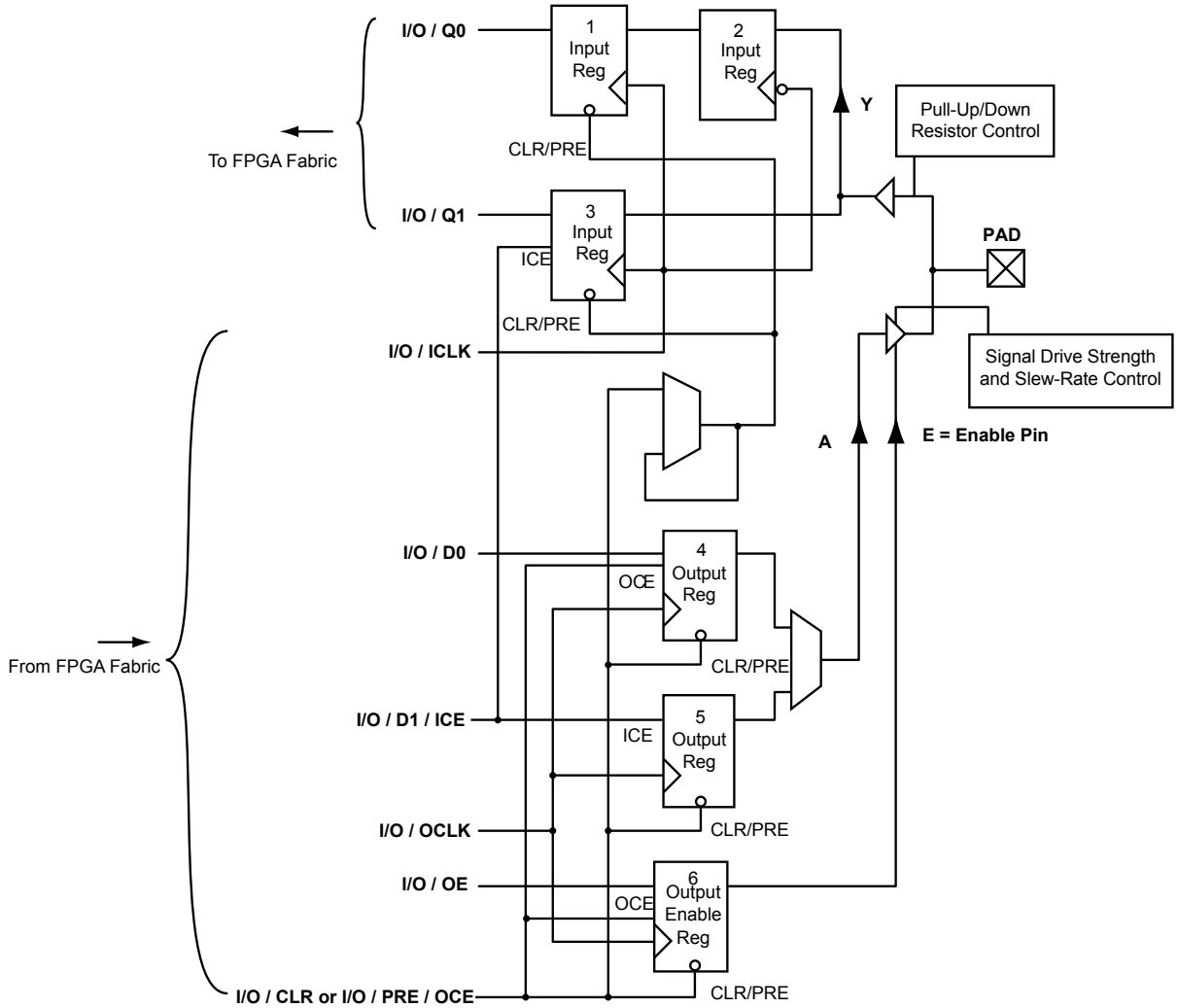
  – Default configuration is FPGA fabric routed to EMC I/O.
  – When configured as fabric I/O, pull-up/-downs (and other I/O attributes) are configured via flash cells based on user configuration.
  – When configured as EMC I/Os, the I/O configuration is fixed, as shown in Table 4-4 on page 54.

I/O Registers

Each I/O module contains several input, output, and enable registers. Refer to Figure 4-2 for a simplified representation of the I/O block.

The number of input registers is selected by a set of switches (not shown in Figure 4-2) between registers to implement single or differential data transmission to and from the FPGA fabric. The Designer software sets these switches for the user.

A common CLR/PRE signal is used by all I/O registers when I/O register combining is used. Input register 2 does not have a CLR/PRE pin, because this register is used for DDR implementation. The I/O register combining must satisfy specific design rules described in the Designer software documentation and online help.



Note: SmartFusion I/Os have registers to support DDR functionality (see the "Double Data Rate (DDR) Support" section on page 57 for more information).

Figure 4-2 • **I/O Block Logical Representation**

Double Data Rate (DDR) Support

SmartFusion I/Os support 350 MHz DDR inputs and outputs. In DDR mode, new data is present on every edge of the clock signal. Clock and data lines have identical bandwidths and signal integrity requirements, making it very efficient for implementing very high-speed systems.

DDR interfaces can be implemented using LVDS and LVPECL I/O standards. In addition, high-speed DDR interfaces can be implemented using LVDS I/O.

Input Support for DDR

The basic structure to support a DDR input is shown in Figure 4-3. Three input registers are used to capture incoming data, which is presented to the core on each rising edge of the I/O register clock.

Each FPGA fabric I/O tile on SmartFusion devices supports DDR inputs.

Output Support for DDR

The basic DDR output structure is shown in Figure 4-4 on page 58. New data is presented to the output every half clock cycle. Note: DDR macros and I/O registers do not require additional routing. The Designer Compile tool automatically recognizes the DDR macro and maps its registers to the I/O register area at the edge of the chip. The routing delay from the I/O registers to the I/O buffers is already taken into account in the DDR macro.

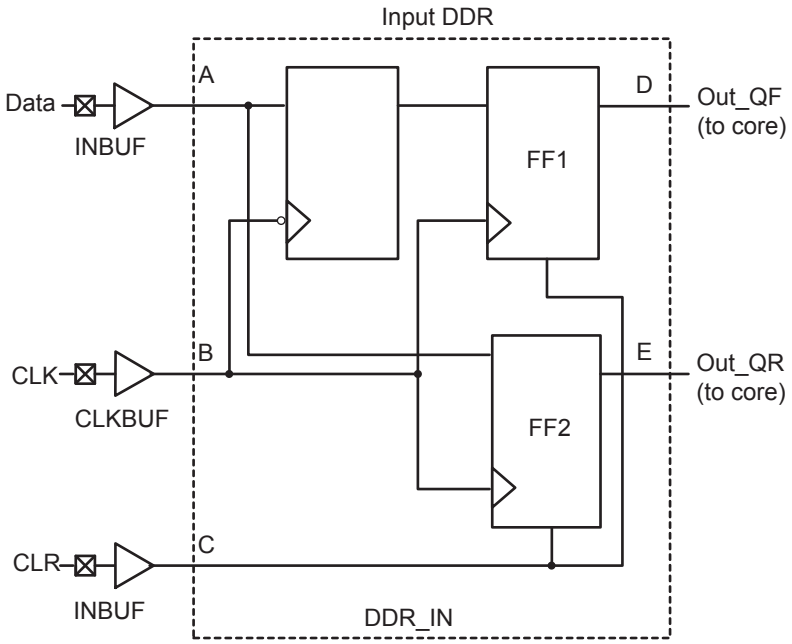


Figure 4-3 • **DDR Input Register Support in SmartFusion Devices**

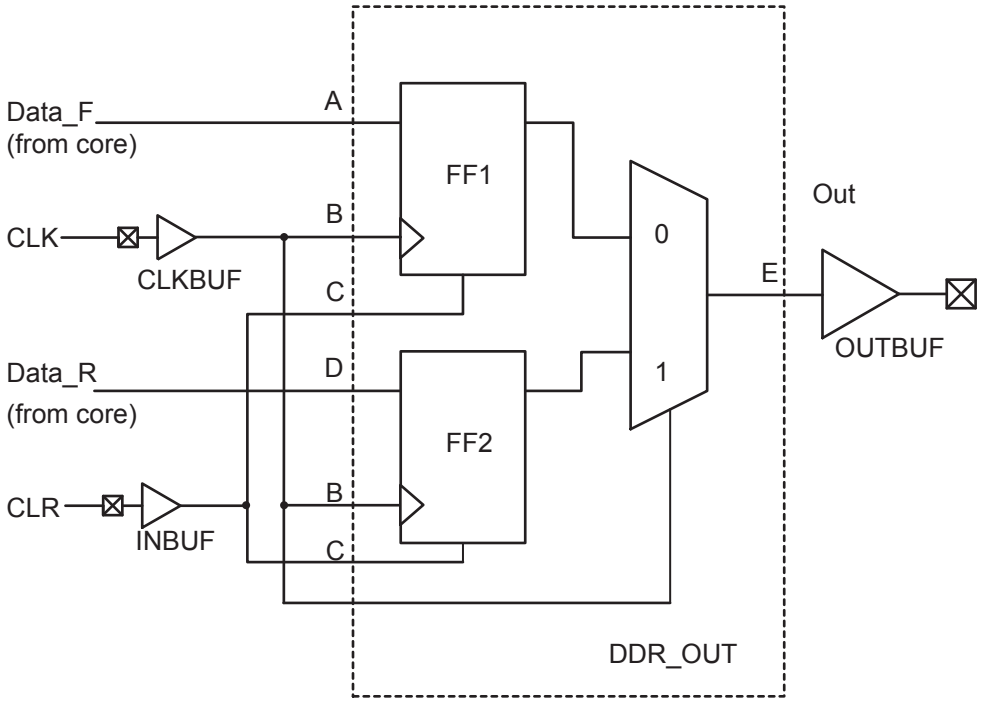


Figure 4-4 • **DDR Output Support in SmartFusion Devices**

Cold-Sparing

Cold-sparing means that a subsystem with no power applied (usually a circuit board) is electrically connected to the system that is in operation. This means that all input buffers of the subsystem must present very high input impedance with no power applied so as not to disturb the operating portion of the system.

While it is true that the SmartFusion SoC FPGA fabric I/Os are tristated when there is no power applied to the device, there is an I/O VCCFPGAIOBx clamp diode that is always connected to VCCFPGAIOBx. In normal DC operating conditions, the I/O pad voltage will not exceed the VCCFPGAIOBx supply voltage and the clamp diode will not be forward-biased. However, in a cold-sparing scenario when VCCFPGAIOBx is not powered (0 V) and a DC voltage is placed on the I/O pad (>0 V), this will forward bias the clamp diode and result in a potentially high current powering up the VCCFPGAIOBx plane. This situation can potentially damage the I/O pin, since it exceeds the recommended DC operating conditions.

If cold-sparing is required, it can be accomplished either by using an external bus switch to isolate the device I/Os from the rest of the system or by driving each FPGA fabric I/O pin to 0 V.

Hot-Swap

Hot-swapping (also called hot-plugging) is the operation of hot insertion (or hot removal) of a card in (or from) a powered-up system.

SmartFusion SoC FPGA fabric I/Os do not inherently support hot-swap since the VCCFPGAIOBx clamp diode is always enabled, as described in the "Cold-Sparing" section.

However, if external circuitry such as an external bus switch is used to isolate the device I/Os from the rest of the system, a cold-swap or a hot-swap while in reset might be possible if certain considerations are made.

A cold-swap implies that a system and card with a SmartFusion device are powered down before the card is inserted into the system. Card insertion is followed by the system power-up while the card supplies remain off. In this scenario, if the SmartFusion fabric I/Os are isolated from the card and system

via external circuitry such as a bus switch, this will ensure that the fabric I/Os are not driven while the VCCFPGAIOBx is not powered.

A hot-swap while in reset implies that the card containing the SmartFusion device is inserted into a powered system which is holding the card buses in reset until the card power supplies are at their nominal operating levels and are stable. Again, if the card designer ensures that the SmartFusion SoC FPGA fabric I/Os are not driven before the SmartFusion VCCFPGAIOBx power supply is applied, this type of operation is possible. This implies fabric I/O isolation via a bus switch and an appropriate card connection power-up sequence, such as grounds, then power supplies, then I/Os, etc.

Electrostatic Discharge (ESD) Protection

SmartFusion devices are tested per JEDEC Standard JESD22-A114-B.

SmartFusion devices contain clamp diodes at every FPGA fabric I/O, global, and power pad. Clamp diodes protect device pads against damage from ESD as well as from excessive voltage transients.

Each I/O has two clamp diodes:

- One diode has its positive (P) side connected to the pad and its negative (N) side connected to VCCFPGAIOBx. This is also referred to as the power clamp or the VCCFPGAIOBx clamp diode.
- The second diode has its P side connected to GND and its N side connected to the pad. This is also referred to as the GND clamp.

During operation, these diodes are normally biased in the OFF state, except when transient voltage is significantly above VCCFPGAIOBx or below GND levels.

On SmartFusion SoC FPGA fabric I/Os, each power and ground clamp diode is always connected to VCCFPGAIOBx and GND, respectively.

5 V Input Tolerance

I/Os can support 5 V input tolerance when LVTTL 3.3 V, LVCMOS 3.3 V, LVCMOS 2.5 V / 5 V, and LVCMOS 2.5 V configurations are used (see Table 4-6 on page 63 for more details). There are four recommended solutions (see Figure 4-5 on page 60 to Figure 4-8 on page 62 for details of board and macro setups) to achieve 5 V receiver tolerance. All the solutions meet a common requirement of limiting the voltage at the input to 3.6 V or less. In fact, the I/O absolute maximum voltage rating is 3.6 V, and any voltage above 3.6 V may cause long-term gate oxide failures.

Solution 1

The board-level design needs to ensure that the reflected waveform at the pad does not exceed the limits provided in the Overshoot and Undershoot Limits table in the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet. This is a long-term reliability requirement.

This scheme will also work for a 3.3 V PCI / PCI-X configuration, but the internal diode should not be used for clamping, and the voltage must be limited by the two external resistors, as described below. Relying on the diode clamping would create an excessive pad DC voltage of 3.3 V + 0.7 V = 4 V.

The following are some examples of possible resistor values (based on a simplified simulation model with no line effects and 10 Ω transmitter output resistance, where Rtx_out_high = (VCCFPGAIOBx – V_{OH}) / I_{OH}, Rtx_out_low = V_{OL} / I_{OL}).

Example 1 (high speed, high current):

Rtx_out_high = Rtx_out_low = 10 Ω

Rext1 = 36 Ω (±5%), P(r1)min = 0.069 Ω

Rext2 = 82 Ω (±5%), P(r2)min = 0.158 Ω

Imax_tx = 5.5 V / (82 × 0.95 + 36 × 0.95 + 10) = 45.04 mA

t_{RISE} = t_{FALL} = 0.85 ns at C_pad_load = 10 pF (includes up to 25% safety margin)

t_{RISE} = t_{FALL} = 4 ns at C_pad_load = 50 pF (includes up to 25% safety margin)

Example 2 (low–medium speed, medium current):

Rtx_out_high = Rtx_out_low = 10 Ω

Rext1 = 220 Ω (±5%), P(r1)min = 0.018 Ω

Rext2 = 390 Ω (±5%), P(r2)min = 0.032 Ω

Imax_tx = 5.5 V / (220 × 0.95 + 390 × 0.95 + 10) = 9.17 mA

t_{RISE} = t_{FALL} = 4 ns at C_pad_load = 10 pF (includes up to 25% safety margin)

t_{RISE} = t_{FALL} = 20 ns at C_pad_load = 50 pF (includes up to 25% safety margin)

Other values of resistors are also allowed as long as the resistors are sized appropriately to limit the voltage at the receiving end to 2.5 V < Vin(rx) < 3.6 V when the transmitter sends a logic 1. This range of Vin_dc(rx) must be assured for any combination of transmitter supply (5 V ± 0.5 V), transmitter output resistance, and board resistor tolerances.

Temporary overshoots are allowed according to the Overshoot and Undershoot Limits table in the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

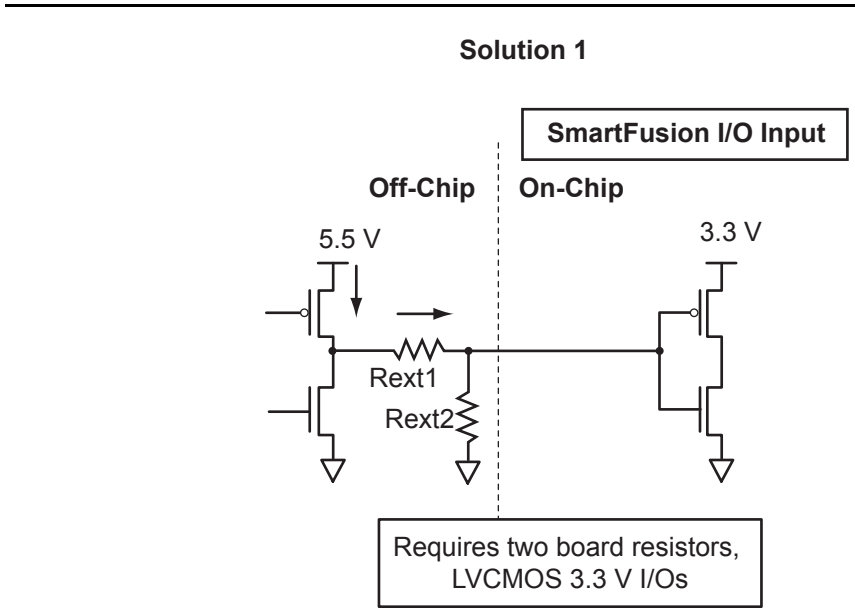


Figure 4-5 • **Solution 1**

Solution 2

The board-level design must ensure that the reflected waveform at the pad does not exceed limits provided in the Overshoot and Undershoot Limits table in the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet. This is a long-term reliability requirement.

This scheme will also work for a 3.3 V PCI/PCI-X configuration, but the internal diode should be turned off and not used for clamping, and the voltage must be limited by the external resistors and Zener, as shown in Figure 4-6. Relying on the diode clamping would create an excessive pad DC voltage of 3.3 V + 0.7 V = 4 V.

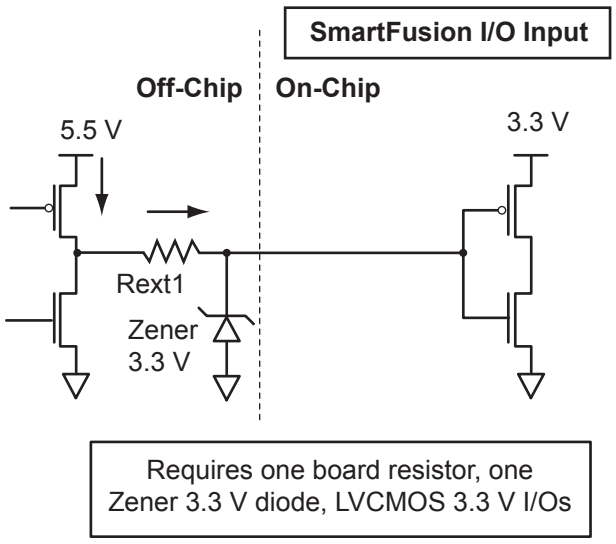


Figure 4-6 • **Solution 2**

Solution 3

The board-level design must ensure that the reflected waveform at the pad does not exceed limits provided in the Overshoot and Undershoot Limits table in the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet. This is a long-term reliability requirement.

This scheme will also work for a 3.3 V PCI/PCIX configuration, but the internal diode should not be used for clamping, and the voltage must be limited by the bus switch, as shown in Figure 4-7. Relying on the diode clamping would create an excessive pad DC voltage of 3.3 V + 0.7 V = 4 V.

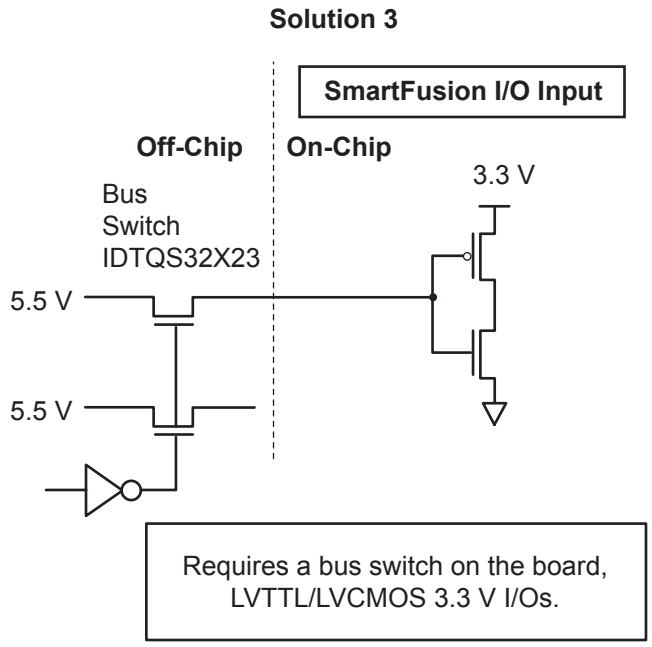


Figure 4-7 • **Solution 3**

Solution 4

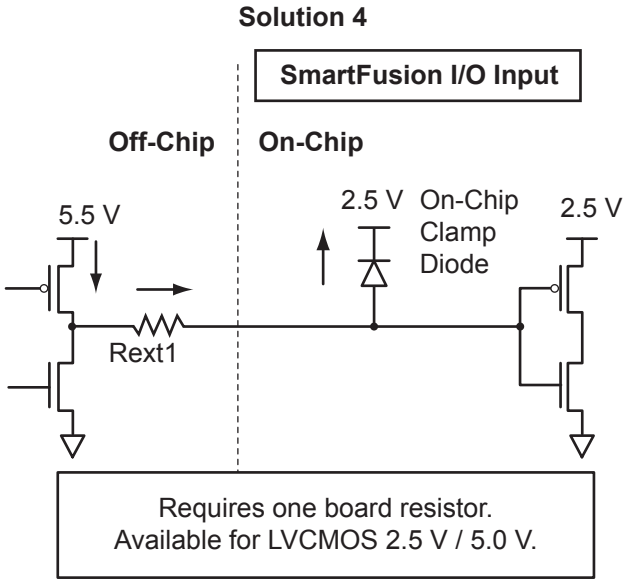


Figure 4-8 • **Solution 4**

***Table 4-6* • Comparison Table for 5 V Compliant Receiver Scheme**

Scheme	Board Components	Speed	Current Limitations
1	Two resistors	Low to high[1]	Limited by transmitter's drive strength
2	Resistor and Zener 3.3 V	Medium	Limited by transmitter's drive strength
3	Bus switch	High	N/A
4	Minimum resistor value[2] R = 47 Ω at T_J = 70°C R = 150 Ω at T_J = 85°C R = 420 Ω at T_J = 100°C	Medium	Maximum diode current at 100% duty cycle, signal constantly at 1. 52.7 mA at T_J =70°C / 10-year lifetime 16.5 mA at T_J = 85°C / 10-year lifetime 5.9 mA at T_J = 100°C / 10-year lifetime For duty cycles other than 100%, the currents can be increased by a factor = 1 / (duty cycle). Example: 20% duty cycle at 70°C Maximum current = (1 / 0.2) × 52.7 mA = 5 × 52.7 mA = 263.5 mA

Notes:

1. *Speed and current consumption increase as the board resistance values decrease.*
2. *Resistor values must ensure I/O diode long-term reliability.*

5 V Output Tolerance

SmartFusion I/Os must be set to 3.3 V LVTTL or 3.3 V LVCMOS mode to reliably drive 5 V TTL receivers. It is also critical that there be NO external I/O pull-up resistor to 5 V, since this resistor would pull the I/O pad voltage beyond the 3.6 V absolute maximum value and consequently cause damage to the I/O.

When set to 3.3 V LVTTL or 3.3 V LVCMOS mode, SmartFusion I/Os can directly drive signals into 5 V TTL receivers. In fact, V_{OL} = 0.4 V and V_{OH} = 2.4 V in both 3.3 V LVTTL and 3.3 V LVCMOS modes exceed the V_{IL} = 0.8 V and V_{IH} = 2 V level requirements of 5 V TTL receivers. Therefore, logic level 1 and logic level 0 will be correctly recognized by 5 V TTL receivers.

Table 4-7 • **Summary of SmartFusion Hot-Insertion, Cold-Sparing and 5 V Input Tolerance Capabilities**

I/O Assignment	Clamp Diode	Hot-Insertion	Cold-Sparing	5 V Input Tolerance	Input Buffer	Output Buffer
3.3 V LVTTL/LVCMOS	Yes	No	No	Yes[1]	Enabled/Disabled	
3.3 V PCI / PCI-X	Yes	No	No	Yes[1]	Enabled/Disabled	
LVCMOS 2.5 V	Yes	No	No	No	Enabled/Disabled	
LVCMOS 2.5 / 5.0 V	Yes	No	No	Yes[2]	Enabled/Disabled	
LVCMOS 1.8 V	Yes	No	No	No	Enabled/Disabled	
LVCMOS 1.5 V	Yes	No	No	No	Enabled/Disabled	
LVDS and LVPECL[3]	Yes	No	No	No	Enabled/Disabled	

Notes:

1. *Can be implemented with an external IDT bus switch, resistor divider, or Zener with resistor.*
2. *Can be implemented with an external resistor and an internal clamp diode.*
3. *Bidirectional LVPECL buffers are not supported. I/Os can be configured as either input buffers or output buffers.*

Simultaneously Switching Outputs and PCB Layout

- Simultaneously switching outputs (SSOs) can impact the signal integrity of adjacent signals that are not part of the SSO bus. Both inductive and capacitive coupling parasitics of bond wires inside packages and of traces on PCBs will transfer noise from SSO buses onto signals adjacent to those buses. Additionally, SSOs can produce ground bounce noise and VCCFPGAIOBx dip noise. These two noise types are caused by rapidly changing currents through GND and VCCFPGAIOBx package pin inductances during switching activities:
- Ground bounce noise voltage = L(GND) × di/dt
- VCCFPGAIOBx dip noise voltage = L(VCCFPGAIOBx) × di/dt

Any group of four or more output pins switching on the same clock edge is considered an SSO bus. The shielding should be done both on the board and inside the package unless otherwise described.

In-package shielding can be achieved in several ways; the required shielding will vary depending on whether pins next to the SSO bus are LVTTL/LVCMOS inputs, LVTTL/LVCMOS outputs, or LVDS/LVPECL inputs and outputs. Board traces in the vicinity of the SSO bus must be adequately shielded from mutual coupling and inductive noise that can be generated by the SSO bus. Also, noise generated by the SSO bus needs to be reduced inside the package.

PCBs perform an important function in feeding stable supply voltages to the IC and, at the same time, maintaining signal integrity between devices.

Key issues that need to be considered are as follows:

- Power and ground plane design and decoupling network design
- Transmission line reflections and terminations

Selectable Skew between Output Buffer Enable/Disable Time

The configurable skew block is used to delay the output buffer assertion (enable) without affecting deassertion (disable) time.

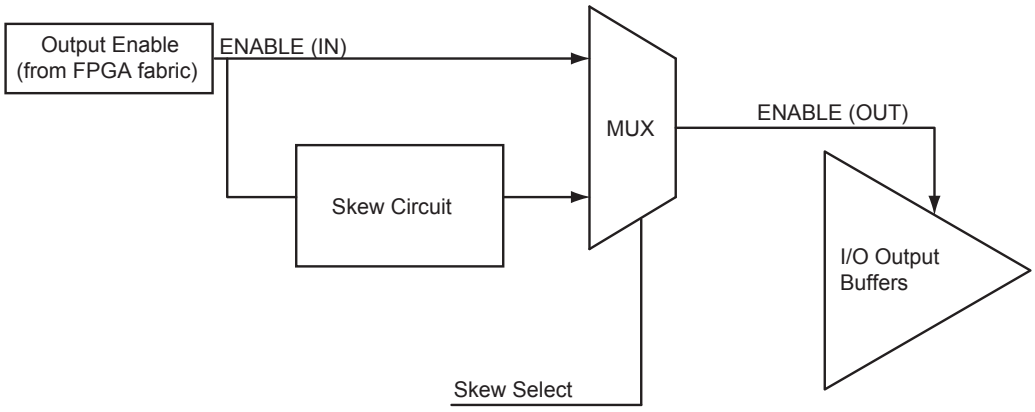


Figure 4-9 • **Block Diagram of Output Enable Path**

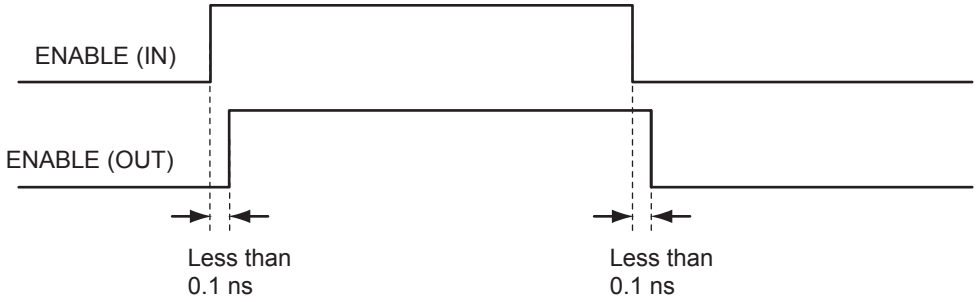


Figure 4-10 • **Timing Diagram (option 1: bypasses skew circuit)**

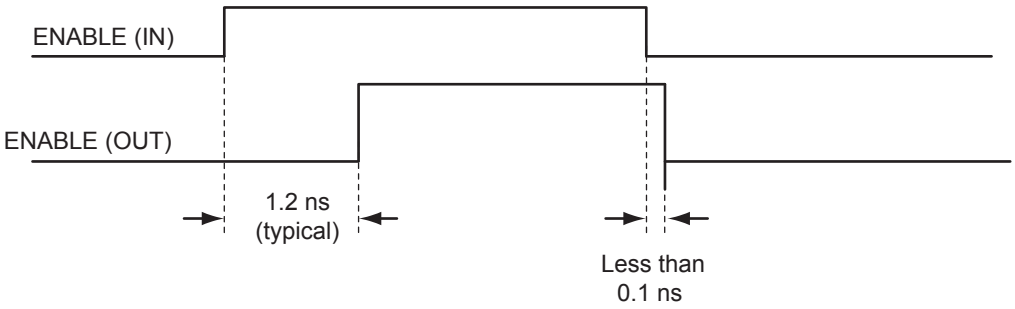


Figure 4-11 • **Timing Diagram (option 2: enables skew circuit)**

At the system level, the enable skew circuit can be used in applications where transmission activities on bidirectional data lines need to be coordinated. When selected, this circuit provides a timing margin that can prevent bus contention and subsequent data loss or transmitter overstressed due to transmitter-to-transmitter current shorts. Figure 4-12 presents an example of the skew circuit implementation in a bidirectional communication system. Figure 4-13 shows how bus contention is created, and Figure 4-14 on page 67 shows how it can be avoided with the skew circuit.

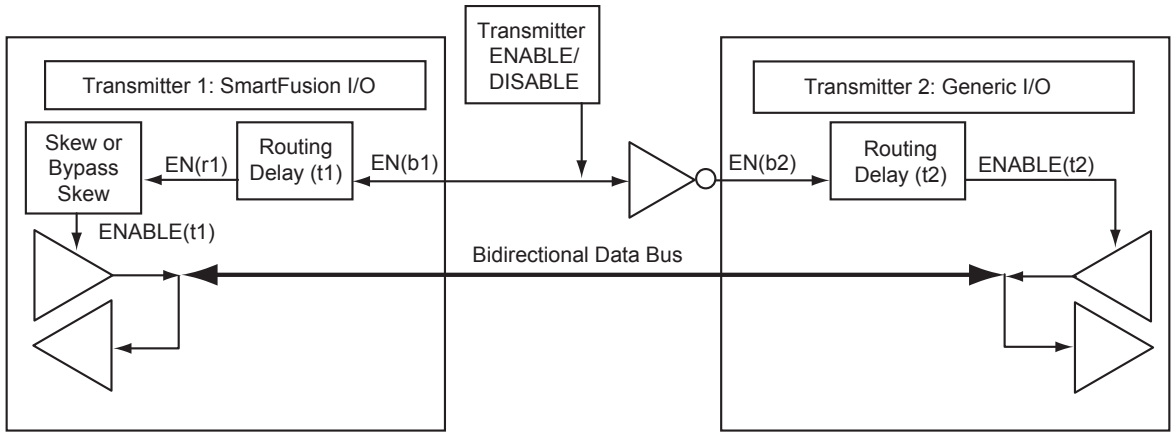


***Figure 4-12* • Example of Implementation of Skew Circuits in Bidirectional Transmission Systems Using SmartFusion Devices**

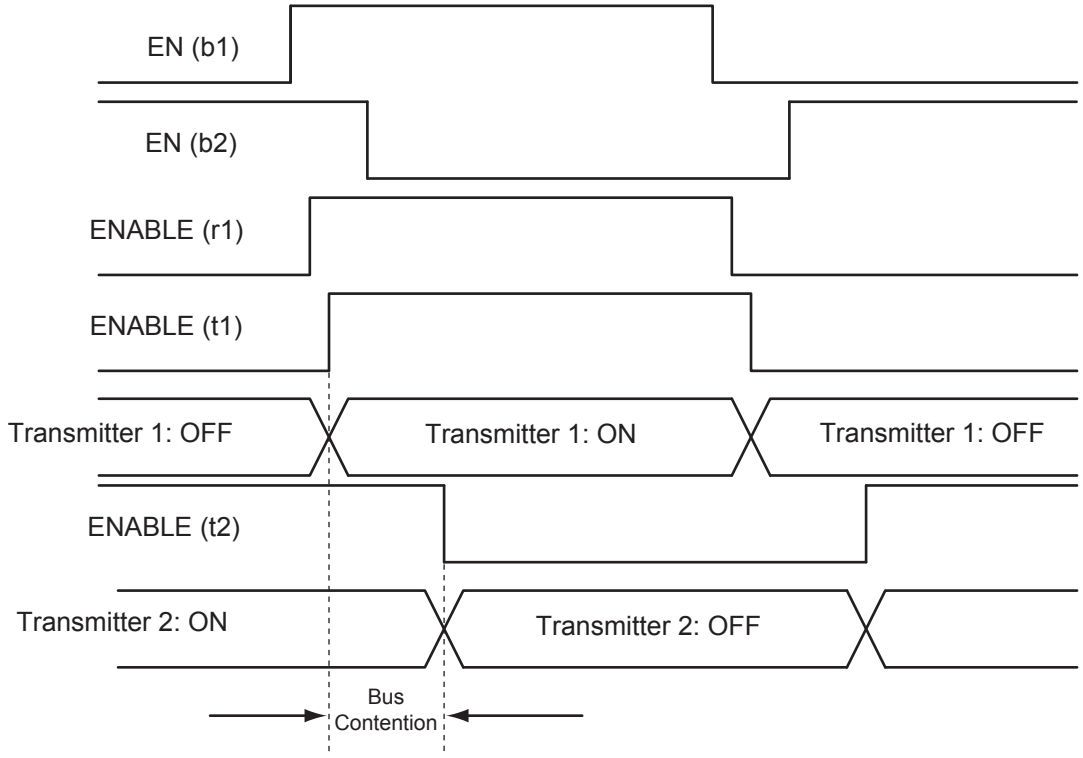


***Figure 4-13* • Timing Diagram (bypasses skew circuit)**

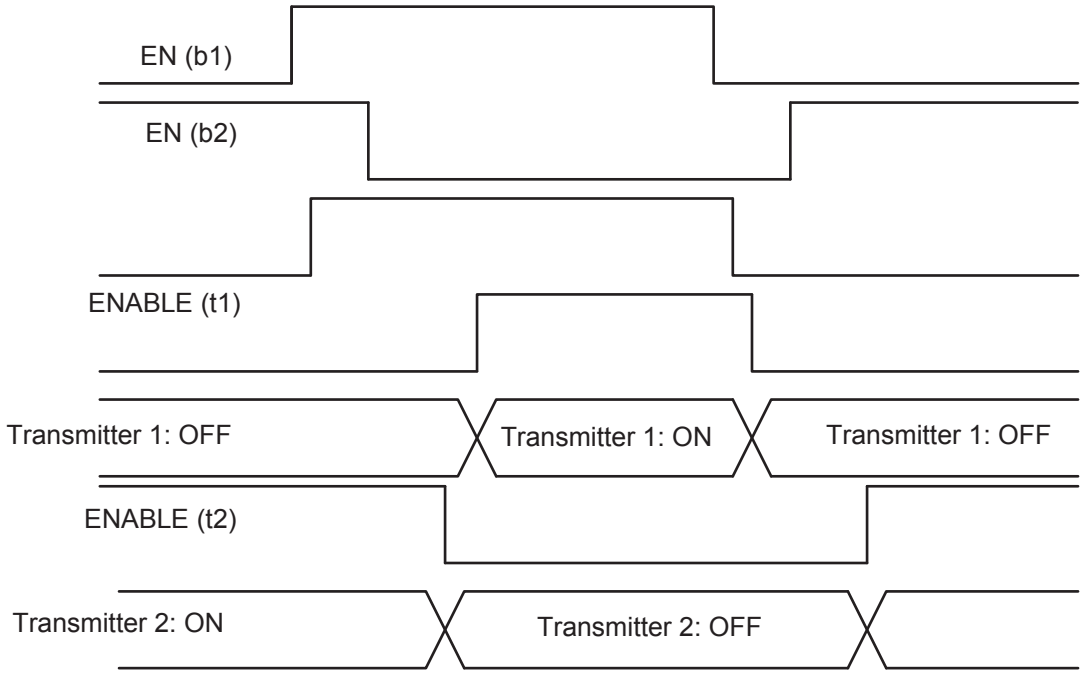


***Figure 4-14* • Timing Diagram (with skew circuit selected)**

Weak Pull-Up and Weak Pull-Down Resistors

SmartFusion devices support optional weak pull-up and pull-down resistors for each I/O pin. When the I/O is pulled up, it is connected to the VCCFPGAIOBx of its corresponding I/O bank. When it is pulled down, it is connected to GND. Refer to the "DC and Switching Characteristics" section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet for more information.

Slew Rate Control and Drive Strength

SmartFusion devices support output slew rate control: high and low. The high slew rate option is recommended to minimize the propagation delay. This high-speed option may introduce noise into the system if appropriate signal integrity measures are not adopted. Selecting a low slew rate reduces this kind of noise but adds some delays in the system. Low slew rate is recommended when bus transients are expected. Drive strength should also be selected according to the design requirements and noise immunity of the system.

The output slew rate and multiple drive strength controls are available in LVTTL/LVCMOS 3.3 V, LVCMOS 2.5 V, LVCMOS 2.5 V / 5.0 V input, LVCMOS 1.8 V, and LVCMOS 1.5 V. All differential I/O standards have a high output slew rate by default. Furthermore, 3.3 V PCI / PCI-X I/O standards have a low output slew rate by default.

For SmartFusion SoC FPGA fabric I/O slew rate and drive strength specifications, refer to Table 4-8 on page 68. Note that fabric I/Os being used as EMC I/Os will have a fixed I/O standard and attribute

configuration, as shown on Table 4-4 on page 54. Low slew and low drive strength are recommended in the case of a large SSO bus toggling at high frequency.

***Table 4-8* • SmartFusion SoC FPGA Fabric I/O SLEW and OUT_DRIVE Settings**

I/O Standard	OUT_DRIVE (mA)							SLEW	
	2	4	6	8	12	16	24		
LVTTL/LVCMOS 3.3 V	?	?	?	?	?	?	?	High	Low
LVCMOS 2.5 V	?	?	?	?	?	?	?	High	Low
LVCMOS 2.5 / 5.0 V	?	?	?	?	?	?	?	High	Low
LVCMOS 1.8 V	?	?	?	?	?	?	–	High	Low
LVCMOS 1.5 V	?	?	?	?	?	?	–	High	Low
LVDS, LVPECL	–	–	–	–	–	–	?	High	
3.3 V PCI / PCI-X	Fixed per 3.3 V PCI / PCI-X Spec							Low	

I/O Software Support

In the SmartFusion development software, default settings have been defined for the various I/O standards supported. Changes can be made to the default settings via the use of attributes. Note that not all I/O attributes are applicable for all I/O standards. Table 4-9 lists the valid I/O attributes that can be manipulated by the user for each I/O standard. Table 4-10 on page 68 lists the default software I/O attribute configuration.

***Table 4-9* • SmartFusion SoC FPGA Fabric I/O Configurable I/O Attributes**

I/O Standard	SLEW (output only)	OUT_DRIVE (output only)	SKEW (TRIBUF and BIBUF macros with OE)	RES_PULL	OUT_LOAD (output only)	COMBINE_REGISTER
LVTTL/LVCMOS 3.3 V	?	?	?	?	?	?
LVCMOS 2.5 V	?	?	?	?	?	?
LVCMOS 2.5 / 5.0 V	?	?	?	?	?	?
LVCMOS 1.8 V	?	?	?	?	?	?
LVCMOS 1.5 V	?	?	?	?	?	?
LVDS	–	–	?	–	–	?
LVPECL	–	–	–	–	–	?
3.3 V PCI / PCI-X	–	–	?	–	?	?

***Table 4-10* • FPGA Fabric I/O Default Attribute Configuration in Software**

I/O Standard	SLEW (output only)	OUT_DRIVE (output only)	SKEW (TRIBUF and BIBUF macros with OE)	RES_PULL	OUT_LOAD (output only)	COMBINE_REGISTER
LVTTL/LVCMOS 3.3 V	High	12 mA	Off	None	35 pF	–
LVCMOS 2.5 V	High	12 mA	Off	None	35 pF	–
LVCMOS 2.5 / 5.0 V	High	12 mA	Off	None	35 pF	–

***Table 4-10* • FPGA Fabric I/O Default Attribute Configuration in Software**

I/O Standard	SLEW (output only)	OUT_DRIVE (output only)	SKEW (TRIBUF and BIBUF macros with OE)	RES_PULL	OUT_LOAD (output only)	COMBINE_REGISTER
LVCMOS 1.8 V	High	12 mA	Off	None	35 pF	–
LVCMOS 1.5 V	High	12 mA	Off	None	35 pF	–
LVDS	High	24 mA	Off	None	–	–
LVPECL	High	24 mA	Off	None	–	–
3.3 V PCI / PCI-X	Low	PCI Setting	Off	None	10 pF	–

A – List of Changes

The following table lists critical changes that were made in each revision of the SmartFusion SoC FPGA Fabric User's Guide.

Revision	Changes	Page
Revision 2 (June 2014)	Updated the "Introduction" section for FPGA user I/O pin functions in 4-SmartFusion FPGA User I/Os (SAR43379).	51
	Added a footnote stating the LVCMOS 2.5/5.0 and LVCMOS 2.5 V similarity and their exception under Table 4-1 (SAR42429).	52
	Updated the 5 V Input Tolerance column for LVCMOS 2.5 V row in Table 4-7 (SAR42026).	64
Revision 1 (December 2011)	The "Clocking Resources Available to the SmartFusion SoC FPGA Fabric" section was revised significantly, with a considerable amount of information added to the chapter (SAR 25141).	14
	The "Multiplexed I/Os Used for FPGA Fabric and External Memory Controller (EMC) Interface" section was revised to clarify that the default configuration for FPGA user I/Os shared with EMC is FPGA fabric routed to EMC I/O (EMC_MUX_CR). The configuration is hardwired and is not affected by system reboot (SAR 30665).	54
	The "Shared MSS I/O Status During MSS Reset" section is new (SAR 31130).	55
Revision 0 (March 2010)	A note was added to Table 1-1 • Array Coordinates, accounting for Banks 2, 3, and 4.	5
	OUT_LOAD was removed from Table 4-4 • I/O Standard and Attribute Fixed Configuration of EMC I/Os.	54
	The "Simultaneously Switching Outputs and PCB Layout" section was revised.	64
Draft B (December 2009)	The "VersaTile" section was revised.	3
	EMI was changed to EMC in Figure 4-1 • SmartFusion A2F200 I/O Bank Location and Naming.	52

B – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 650.318.8044

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Index

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and security, aerospace, and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs, and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

50200249-2/6.14