Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

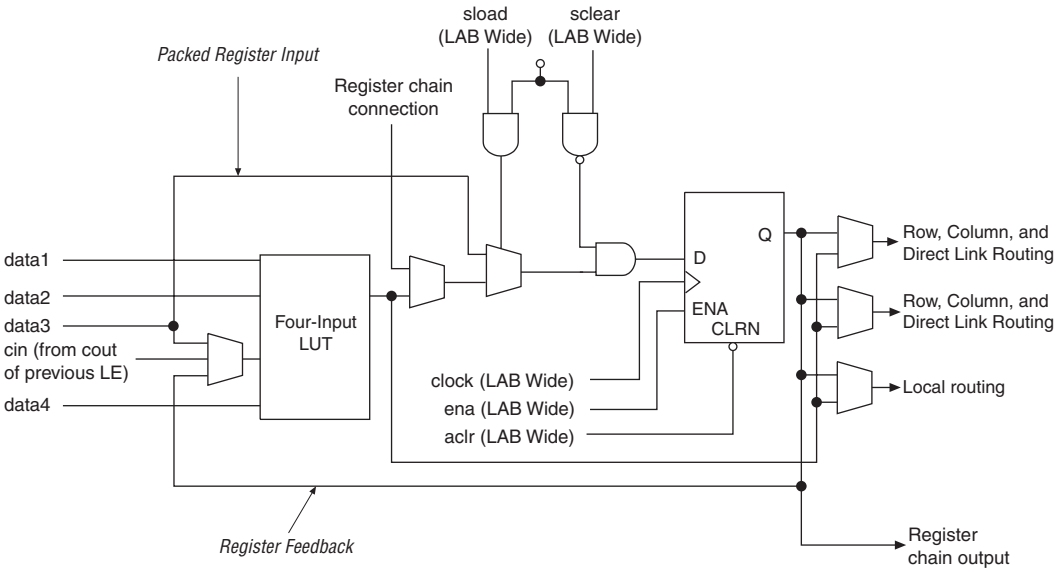
Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling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

Applications of Embedded - FPGAs

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

Details	
Product Status	Active
Number of LABs/CLBs	288
Number of Logic Elements/Cells	4608
Total RAM Bits	119808
Number of I/O	89
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	144-LQFP
Supplier Device Package	144-TQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep2c5t144c6n

Figure 2–3. LE in Normal Mode



Arithmetic Mode

The arithmetic mode is ideal for implementing adders, counters, accumulators, and comparators. An LE in arithmetic mode implements a 2-bit full adder and basic carry chain (see Figure 2–4). LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output. Register feedback and register packing are supported when LEs are used in arithmetic mode.

C16 column interconnects span a length of 16 LABs and provide the fastest resource for long column connections between LABs, M4K memory blocks, embedded multipliers, and IOEs. C16 column interconnects drive to other row and column interconnects at every fourth LAB. C16 column interconnects drive LAB local interconnects via C4 and R4 interconnects and do not drive LAB local interconnects directly. C16 interconnects can drive R24, R4, C16, and C4 interconnects.

Device Routing

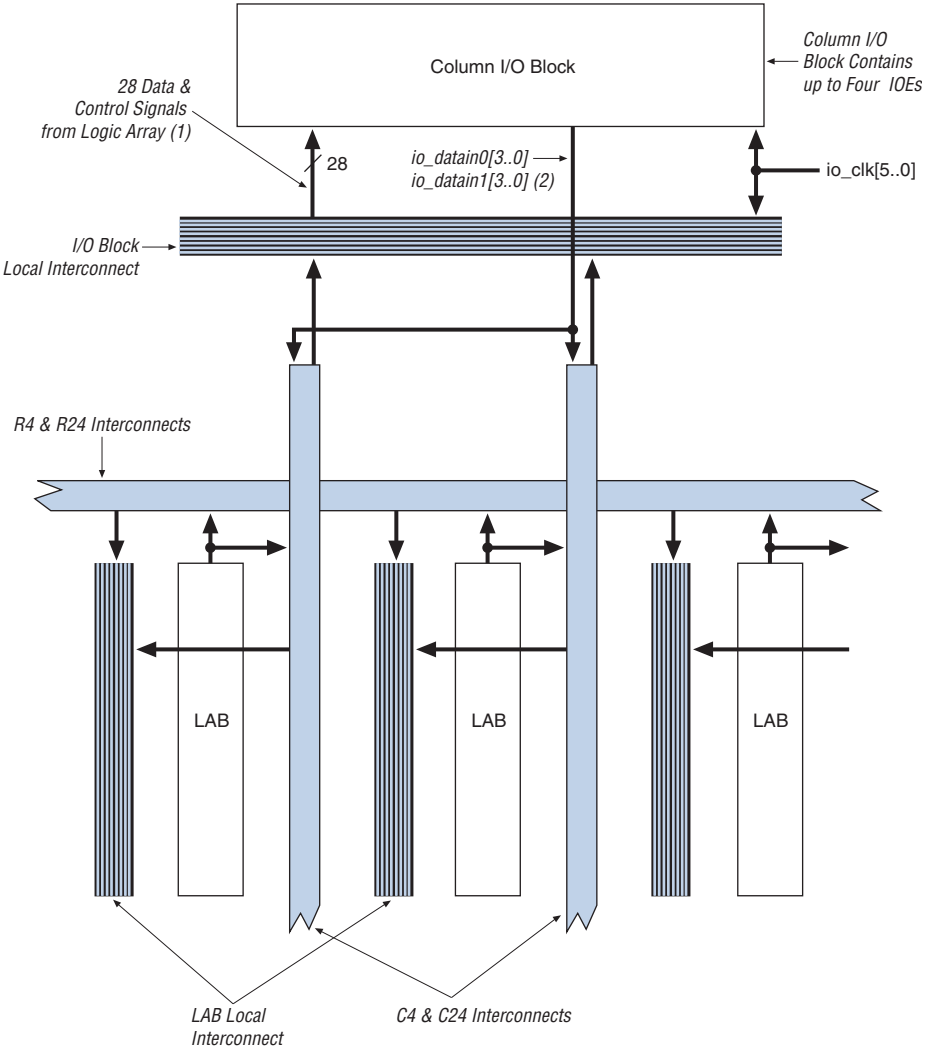
All embedded blocks communicate with the logic array similar to LAB-to-LAB interfaces. Each block (for example, M4K memory, embedded multiplier, or PLL) connects to row and column interconnects and has local interconnect regions driven by row and column interconnects. These blocks also have direct link interconnects for fast connections to and from a neighboring LAB.

Table 2–1 shows the Cyclone II device's routing scheme.

Table 2–1. Cyclone II Device Routing Scheme (Part 1 of 2)

Source	Destination												
	Register Chain	Local Interconnect	Direct Link Interconnect	R4 Interconnect	R24 Interconnect	C4 Interconnect	C16 Interconnect	LE	M4K RAM Block	Embedded Multiplier	PLL	Column IOE	Row IOE
Register Chain								✓					
Local Interconnect								✓	✓	✓	✓	✓	✓
Direct Link Interconnect		✓											
R4 Interconnect		✓		✓	✓	✓	✓						
R24 Interconnect				✓	✓	✓	✓						
C4 Interconnect		✓		✓	✓	✓	✓						
C16 Interconnect				✓	✓	✓	✓						

Figure 2–22. Column I/O Block Connection to the Interconnect



Notes to Figure 2–22:

(1) The 28 data and control signals consist of four data out lines, `io_dataout[3..0]`, four output enables, `io_coe[3..0]`, four input clock enables, `io_cce_in[3..0]`, four output clock enables, `io_cce_out[3..0]`, four clocks, `io_cclk[3..0]`, four asynchronous clear signals, `io_caclr[3..0]`, and four synchronous clear signals, `io_csclr[3..0]`.

(2) Each of the four IOEs in the column I/O block can have two `io_datain` (combinational or registered) inputs.

Table 5–8 shows the recommended operating conditions for user I/O pins with differential I/O standards.

Table 5–8. Recommended Operating Conditions for User I/O Pins Using Differential Signal I/O Standards

I/O Standard	V_{CCIO} (V)			V_{ID} (V) *(1)*			V_{ICM} (V)			V_{IL} (V)		V_{IH} (V)	
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Max	Min	Max
LVDS	2.375	2.5	2.625	0.1	—	0.65	0.1	—	2.0	—	—	—	—
Mini-LVDS *(2)*	2.375	2.5	2.625	—	—	—	—	—	—	—	—	—	—
RSDS *(2)*	2.375	2.5	2.625	—	—	—	—	—	—	—	—	—	—
LVPECL *(3) (6)*	3.135	3.3	3.465	0.1	0.6	0.95	—	—	—	0	2.2	2.1	2.88
Differential 1.5-V HSTL class I and II *(4)*	1.425	1.5	1.575	0.2	—	V_{CCIO} + 0.6	0.68	—	0.9	—	V_{REF} – 0.20	V_{REF} + 0.20	—
Differential 1.8-V HSTL class I and II *(4)*	1.71	1.8	1.89	—	—	—	—	—	—	—	V_{REF} – 0.20	V_{REF} + 0.20	—
Differential SSTL-2 class I and II *(5)*	2.375	2.5	2.625	0.36	—	V_{CCIO} + 0.6	0.5 × V_{CCIO} – 0.2	0.5 × V_{CCIO}	0.5 × V_{CCIO} + 0.2	—	V_{REF} – 0.35	V_{REF} + 0.35	—
Differential SSTL-18 class I and II *(5)*	1.7	1.8	1.9	0.25	—	V_{CCIO} + 0.6	0.5 × V_{CCIO} – 0.2	0.5 × V_{CCIO}	0.5 × V_{CCIO} + 0.2	—	V_{REF} – 0.25	V_{REF} + 0.25	—

Notes to Table 5–8:

(1) Refer to the *High-Speed Differential Interfaces in Cyclone II Devices* chapter of the *Cyclone II Device Handbook* for measurement conditions on V_{ID}.

(2) The RSDS and mini-LVDS I/O standards are only supported on output pins.

(3) The LVPECL I/O standard is only supported on clock input pins. This I/O standard is not supported on output pins.

(4) The differential 1.8-V and 1.5-V HSTL I/O standards are only supported on clock input pins and PLL output clock pins.

(5) The differential SSTL-18 and SSTL-2 I/O standards are only supported on clock input pins and PLL output clock pins.

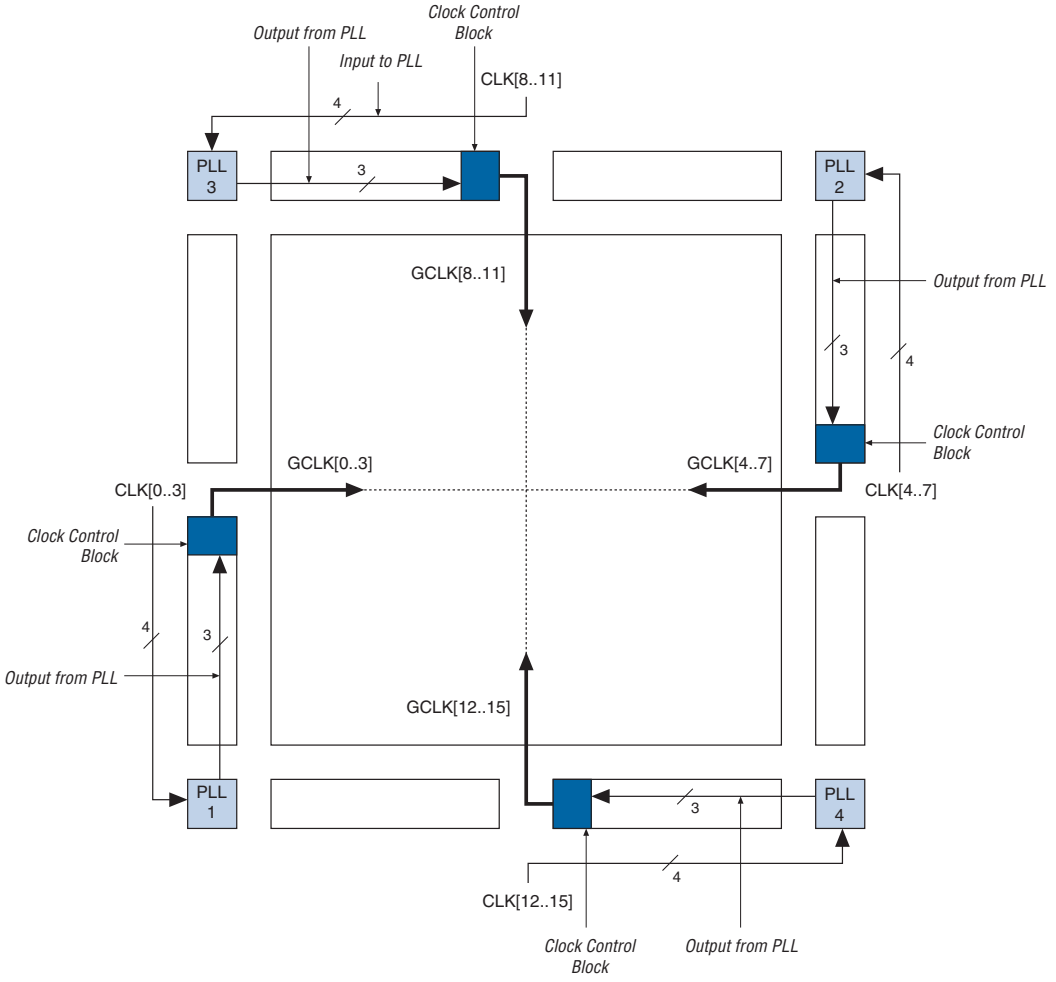
(6) The LVPECL clock inputs are powered by V_{CCINT} and support all V_{CCIO} settings. However, it is recommended to connect V_{CCIO} to typical value of 3.3V.

Table 7–8 shows the clock sources connectivity to the global clock networks.

Table 7–8. Global Clock Network Connections (Part 1 of 3)

Global Clock Network Clock Sources	Global Clock Networks															
	All Cyclone II Devices								EP2C15 through EP2C70 Devices Only							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLK0/LVDSCLK0p	✓		✓													
CLK1/LVDSCLK0n		✓	✓													
CLK2/LVDSCLK1p	✓			✓												
CLK3/LVDSCLK1n		✓		✓												
CLK4/LVDSCLK2p					✓		✓									
CLK5/LVDSCLK2n						✓	✓									
CLK6/LVDSCLK3p					✓			✓								
CLK7/LVDSCLK3n						✓		✓								
CLK8/LVDSCLK4n									✓		✓					
CLK9/LVDSCLK4p										✓	✓					
CLK10/LVDSCLK5n									✓			✓				
CLK11/LVDSCLK5p										✓		✓				
CLK12/LVDSCLK6n													✓		✓	
CLK13/LVDSCLK6p														✓	✓	
CLK14/LVDSCLK7n													✓			✓
CLK15/LVDSCLK7p														✓		✓
PLL1_c0	✓	✓		✓												
PLL1_c1	✓		✓	✓												
PLL1_c2		✓	✓													
PLL2_c0					✓	✓		✓								
PLL2_c1					✓		✓	✓								
PLL2_c2						✓	✓									
PLL3_c0									✓	✓		✓				
PLL3_c1									✓		✓	✓				
PLL3_c2										✓	✓					

Figure 7–12. Cyclone II Clock Control Blocks Placement



The inputs to the four clock control blocks on each side are chosen from among the following clock sources:

- Four clock input pins
- Three PLL counter outputs
- Two DPCLK pins and two CDPCLK pins from both the left and right sides and four DPCLK pins and two CDPCLK pins from both the top and bottom
- Four signals from internal logic

Section III. Memory

This section provides information on embedded memory blocks in Cyclone® II devices and the supported external memory interfaces.

This section includes the following chapters:

- Chapter 8, Cyclone II Memory Blocks
- Chapter 9, External Memory Interfaces

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the complete handbook.

case writing is controlled only by the write enable signals. There is no clear port to the byte enable registers. M4K blocks support byte enables when the write port has a data width of 1, 2, 4, 8, 9, 16, 18, 32, or 36 bits. When using data widths of 1, 2, 4, 8, and 9 bits, the byte enable behaves as a redundant write enable because the data width is less than or equal to a single byte. Table 8–3 summarizes the byte selection.

Table 8–3. Byte Enable for Cyclone II M4K Blocks *Note (1)*

byteena[3..0]	Affected Bytes								
	datain ×1	datain ×2	datain ×4	datain ×8	datain ×9	datain ×16	datain ×18	datain ×32	datain ×36
[0] = 1	[0]	[1..0]	[3..0]	[7..0]	[8..0]	[7..0]	[8..0]	[7..0]	[8..0]
[1] = 1	-	-	-	-	-	[15..8]	[17..9]	[15..8]	[17..9]
[2] = 1	-	-	-	-	-	-	-	[23..16]	[26..18]
[3] = 1	-	-	-	-	-	-	-	[31..24]	[35..27]

Note to Table 8–3:
(1) Any combination of byte enables is possible.

Table 8–4 shows the byte enable port control for true dual-port mode.

Table 8–4. Byte Enable Port Control for True Dual-Port Mode

byteena [3:0]	Affected Port
[1:0]	Port A *(1)*
[3:2]	Port B *(1)*

Note to Table 8–4:
(1) For any data width up to ×18 for each port.

Figure 8–2 shows how the `wren` and `byteena` signals control the operations of the RAM.

When a byte enable bit is de-asserted during a write cycle, the corresponding data byte output appears as a "don't care" or unknown value. When a byte enable bit is asserted during a write cycle, the corresponding data byte output is the newly written data.

10. Selectable I/O Standards in Cyclone II Devices

CII51010-2.4

Introduction

The proliferation of I/O standards and the need for improved I/O performance have made it critical that low-cost devices have flexible I/O capabilities. Selectable I/O capabilities such as SSTL-18, SSTL-2, and LVDS compatibility allow Cyclone® II devices to connect to other devices on the same printed circuit board (PCB) that may require different operating and I/O voltages. With these aspects of implementation easily manipulated using the Altera® Quartus® II software, the Cyclone II device family allows you to use low cost FPGAs while keeping pace with increasing design complexity.

This chapter is a guide to understanding the input and output capabilities of the Cyclone II devices, including:

- Supported I/O standards
- Cyclone II I/O banks
- Programmable current drive strength
- I/O termination
- Pad placement and DC guidelines

For information on hot socketing, refer to the *Hot Socketing & Power-On Reset* chapter in volume 1 of the *Cyclone II Device Handbook*.

For information on ESD specifications, refer to the *Altera Reliability Report*.

Supported I/O Standards

Cyclone II devices support the I/O standards shown in Table 10–1.

For more details on the I/O standards discussed in this section, including target data rates and voltage values for each I/O standard, refer to the *DC Characteristics and Timing Specifications* chapter in volume 1 of the *Cyclone II Device Handbook*.

- Cyclone II FPGA (EP2C15 or larger)
- Altera PCI Express Compiler ×1 MegaCore® function
- External PCI Express transceiver/PHY

2.5-V LVTTL (EIA/JEDEC Standard EIA/JESD8-5)

The 2.5-V I/O standard is used for 2.5-V LVTTL applications. This standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 2.5-V devices.

The 2.5-V standard does not require input reference voltages or board terminations. Cyclone II devices support input and output levels for 2.5-V LVTTL.

2.5-V LVCMOS (EIA/JEDEC Standard EIA/JESD8-5)

The 2.5-V I/O standard is used for 2.5-V LVCMOS applications. This standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 2.5-V parts.

The 2.5-V standard does not require input reference voltages or board terminations. Cyclone II devices support input and output levels for 2.5-V LVCMOS.

SSTL-2 Class I and II (EIA/JEDEC Standard JESD8-9A)

The SSTL-2 I/O standard is a 2.5-V memory bus standard used for applications such as high-speed double data rate (DDR) SDRAM interfaces. This standard defines the input and output specifications for devices that operate in the SSTL-2 logic switching range of 0.0 to 2.5 V. This standard improves operations in conditions where a bus must be isolated from large stubs. The SSTL-2 standard specifies an input voltage range of $-0.3\ V \leq V_I \leq V_{CCIO} + 0.3\ V$. SSTL-2 requires a V_{REF} value of 1.25 V and a V_{TT} value of 1.25 V connected to the termination resistors (refer to Figures 10–1 and 10–2).

You can use I/O pins and internal logic to implement a high-speed I/O receiver and transmitter in Cyclone II devices. Cyclone II devices do not contain dedicated serialization or deserialization circuitry. Therefore, shift registers, internal global phase-locked loops (PLLs), and I/O cells are used to perform serial-to-parallel conversions on incoming data and parallel-to-serial conversion on outgoing data.

I/O Standards Support

This section provides information on the I/O standards that Cyclone II devices support.

LVDS Standard Support in Cyclone II Devices

The LVDS I/O standard is a high-speed, low-voltage swing, low power, and general purpose I/O interface standard. The Cyclone II device meets the ANSI/TIA/EIA-644 standard.

I/O banks on all four sides of the Cyclone II device support LVDS channels. See the pin tables on the Altera web site for the number of LVDS channels supported throughout different family members. Cyclone II LVDS receivers (input) support a data rate of up to 805 Mbps while LVDS transmitters (output) support up to 640 Mbps. The maximum internal clock frequency for a receiver and for a transmitter is 402.5 MHz. The maximum input data rate of 805 Mbps and the maximum output data rate of 640 Mbps is only achieved when DDIO registers are used. The LVDS standard does not require an input reference voltage; however, it does require a 100-Ω termination resistor between the two signals at the input buffer.

For LVDS data rates in Cyclone II devices with different speed grades, see the *DC Characteristics & Timing Specifications* chapter of the *Cyclone II Device Handbook*.

Table 11–1 shows LVDS I/O specifications.

Table 11–1. LVDS I/O Specifications (Part 1 of 2) *Note (1)*

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{CCINT}	Supply voltage		1.15	1.2	1.25	V
V_{CCIO}	I/O supply voltage		2.375	2.5	2.625	V
V_{OD}	Differential output voltage	R_L = 100 Ω	250		600	mV
ΔV_{OD}	Change in V_{OD} between H and L	R_L = 100 Ω			50	mV
V_{OS}	Output offset voltage	R_L = 100 Ω	1.125	1.25	1.375	V

When the `signa` and `signb` signals are unused, the Quartus® II software sets the multiplier to perform unsigned multiplication by default.

Output Registers

You can choose to register the embedded multiplier output using the output registers in 18- or 36-bit sections depending on the operational mode of the multiplier. The following control signals are available to each output register within the embedded multiplier:

- clock
- clock enable
- asynchronous clear

All input and output registers within a single embedded multiplier are fed by the same clock, clock enable, or asynchronous clear signal.

See the *Cyclone II Architecture* chapter in Volume 1 of the ***Cyclone II Device Handbook*** for more information on the embedded multiplier routing and interface.

Operational Modes

The embedded multiplier can be used in one of two operational modes, depending on the application needs:

- One 18-bit multiplier
- Up to two 9-bit independent multipliers

The Quartus II software includes megafunctions used to control the mode of operation of the multipliers. After you have made the appropriate parameter settings using the megafunction's MegaWizard® Plug-In Manager, the Quartus II software automatically configures the embedded multiplier.

The Cyclone II embedded multipliers can also be used to implement multiplier adder and multiplier accumulator functions where the multiplier portion of the function is implemented using embedded multipliers and the adder or accumulator function is implemented in logic elements (LEs).

For more information on megafunction and Quartus II support for Cyclone II embedded multipliers, see the "Software Support" section.

Software Support

Altera provides two methods for implementing multipliers in your design using embedded multiplier resources: instantiation and inference. Both methods use the following three Quartus II megafunctions:

- `lpm_mult`
- `altmult_add`
- `altmult_accum`

You can instantiate the megafunctions in the Quartus II software to use the embedded multipliers. You can use the `lpm_mult` and `altmult_add` megafunctions to implement multipliers. Additionally, you can use the `altmult_add` megafunctions to implement multiplier-adders where the embedded multiplier is used to implement the multiply function and the adder function is implemented in LEs. The `altmult_accum` megafunction implements multiply accumulate functions where the embedded multiplier implements the multiplier and the accumulator function is implemented in LEs.

See Quartus II On-Line Help for instructions on using the megafunctions and the MegaWizard Plug-In Manager.

For information on our complete DSP Design and Intellectual Property offerings, see **www.Altera.com**.

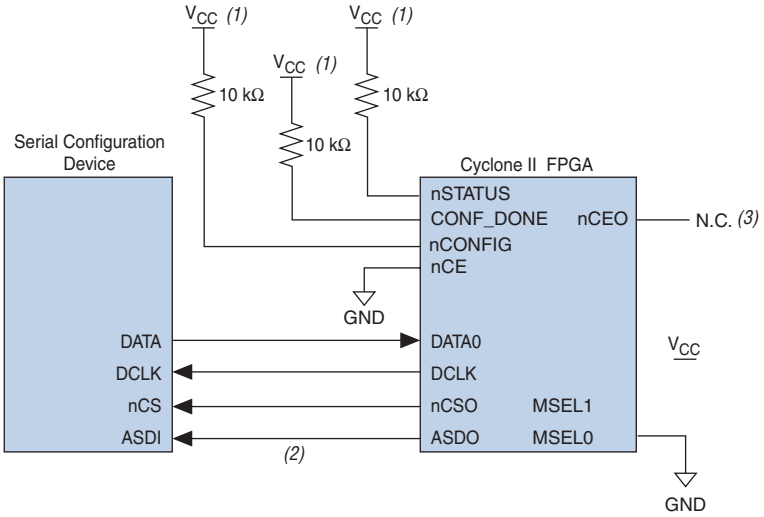
You can also infer the megafunctions by creating an HDL design and synthesize it using Quartus II integrated synthesis or a third-party synthesis tool that recognizes and infers the appropriate multiplier megafunction. Using either method, the Quartus II software maps the multiplier functionality to the embedded multipliers during compilation.

See the Synthesis section in Volume 1 of the ***Quartus II Handbook*** for more information.

Conclusion

The Cyclone II device embedded multipliers are optimized to support multiplier-intensive DSP applications such as FIR filters, FFT functions and encoders. These embedded multipliers can be configured to implement multipliers of various bit widths up to 18-bits to suit a particular application resulting in efficient resource utilization and improved performance and data throughput. The Quartus II software, together with the LeonardoSpectrum and Synplify software provide a complete and easy-to-use flow for implementing multiplier functions using embedded multipliers.

Figure 13–3. Single Device AS Configuration



Notes to Figure 13–3:

(1) Connect the pull-up resistors to a 3.3-V supply.

(2) Cyclone II devices use the ASDO to ASDI path to control the configuration device.

(3) The nCEO pin can be left unconnected or used as a user I/O pin when it does not feed another device's nCE pin.

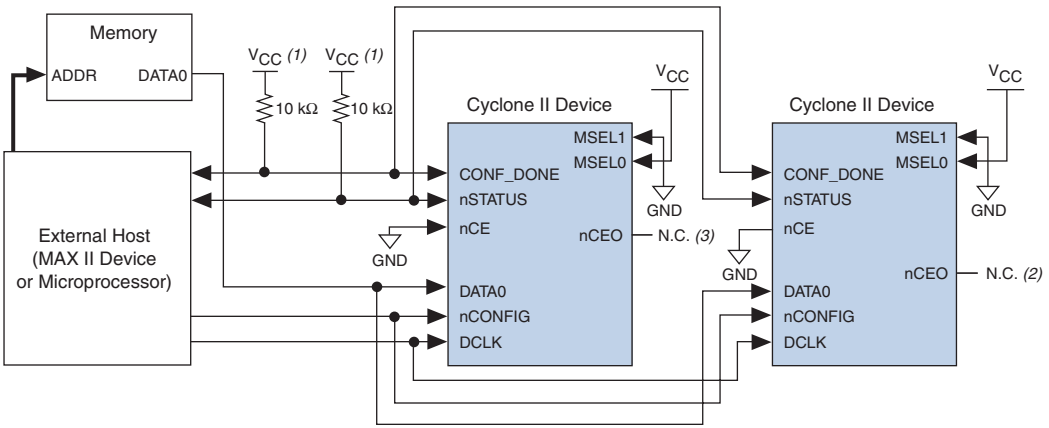
Upon power-up, the Cyclone II device goes through a POR. During POR, the device resets, holds nSTATUS and CONF_DONE low, and tri-states all user I/O pins. After POR, which typically lasts 100 ms, the Cyclone II device releases nSTATUS and enters configuration mode when the external 10-kΩ resistor pulls the nSTATUS pin high. Once the FPGA successfully exits POR, all user I/O pins continue to be tri-stated. Cyclone II devices have weak pull-up resistors on the user I/O pins which are on before and during configuration.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration are available in the *DC Characteristics & Timing Specifications* chapter of the *Cyclone II Device Handbook*.

The configuration cycle consists of the reset, configuration, and initialization stages.

If your system has multiple Cyclone II devices (in the same density and package) with the same configuration data, you can configure them in one configuration cycle by connecting all device's nCE pins to ground and connecting all the Cyclone II device's configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) together. You can also use the nCEO pin as a user I/O pin after configuration. The configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Make sure the DCLK and DATA lines are buffered for every fourth device. All devices start and complete configuration at the same time. Figure 13–11 shows multiple device PS configuration when both Cyclone II devices are receiving the same configuration data.

Figure 13–11. Multiple Device PS Configuration When Both FPGAs Receive the Same Data



Notes to Figure 13–11:

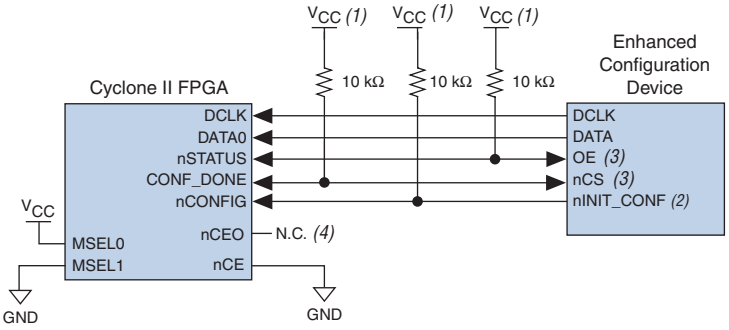
(1) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the devices and the external host.

(2) The nCEO pins of both devices can be left unconnected or used as user I/O pins when configuring the same configuration data into multiple devices.

You can use a single configuration chain to configure Cyclone II devices with other Altera devices. Connect all the Cyclone II device's and all other Altera device's CONF_DONE and nSTATUS pins together so all devices in the chain complete configuration at the same time or that an error reported by one device initiates reconfiguration in all devices.

For more information on configuring multiple Altera devices in the same configuration chain, see *Configuring Mixed Altera FPGA Chains* in the *Configuration Handbook*.

Figure 13–13. Single Device PS Configuration Using an Enhanced Configuration Device



Notes to Figure 13–13:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device. This pull-up resistor is 10 kΩ

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active, meaning an external pull-up resistor should not be used on the nINIT_CONF to nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor (if reconfiguration is required, a resistor is necessary).

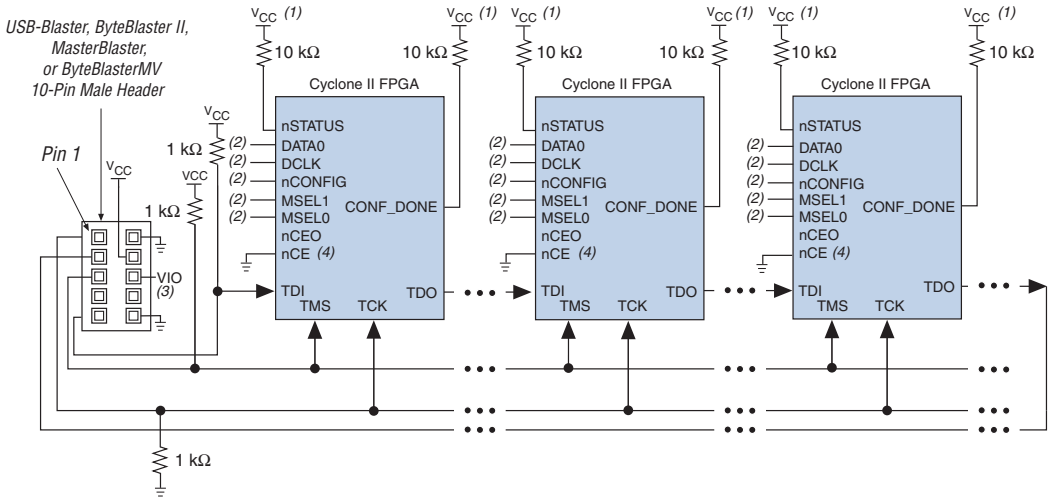
(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

(4) The nCEO pin can be left unconnected or used as a user I/O pin when it does not feed other device's nCE pin.

The value of the internal pull-up resistors on the enhanced configuration devices and EPC2 devices can be found in the *Enhanced Configuration Devices (EPC4, EPC8, & EPC16) Data Sheet* or the *Configuration Devices for SRAM-based LUT Devices Data Sheet*.

When using enhanced configuration devices or EPC2 devices, you can connect the Cyclone II nCONFIG pin to the configuration device nINIT_CONF pin, which allows the INIT_CONF JTAG instruction to initiate FPGA configuration. You do not need to connect the nINIT_CONF pin if you are not using it. If nINIT_CONF is not used or not available (e.g., on EPC1 devices), pull the nCONFIG signal to V_{CC} either directly or through a resistor (if reconfiguration is required, a resistor is necessary). An internal pull-up resistor on the nINIT_CONF pin is always active in enhanced configuration devices and EPC2 devices. Therefore, you do not need an external pull-up if nCONFIG is connected to nINIT_CONF.

Figure 13–24. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 13–24:

(1) The pull-up resistor should be connected to the same supply voltage as the USB-Blaster, MasterBlaster (VIO pin), ByteBlaster II or ByteBlasterMV cable.

(2) Connect the nCONFIG and MSEL[1..0] pins to support a non-JTAG configuration scheme. If only JTAG configuration is used, connect the nCONFIG pin to V_{CC}, and the MSEL[1..0] pins to ground. In addition, pull DCLK and DATA0 to either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable Data Sheet* for this value. In the ByteBlasterMV cable, this pin is a no connect. In the USB-Blaster and ByteBlaster II cable, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

(4) nCE must be connected to ground or driven low for successful JTAG configuration.

Connect the nCE pin to GND or pull it low during JTAG configuration. In multiple device AS and PS configuration chains, connect the first device's nCE pin to GND and connect its nCEO pin to the nCE pin of the next device in the chain or you can use it as a user I/O pin after configuration.

After the first device completes configuration in a multiple device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, you should make sure the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multiple device configuration chain, the nCEO pin of the previous device drives the nCE pin of the next device low when it has successfully been JTAG configured.

Document Revision History

Table 13–14 shows the revision history for this document.

Table 13–14. Document Revision History

Date & Document Version	Changes Made	Summary of Changes
February 2007 v3.1	• Added document revision history. • Added *Note (1)* to Table 13–1. • Added *Note (1)* to Table 13–4. • Updated Figure 13–3. • Updated Figures 13–6 and 13–7. • Updated *Note (2)* to Figure 13–13. • Updated "Single Device PS Configuration Using a Configuration Device" section. • Updated *Note (2)* to Figure 13–14. • Updated *Note (2)* to Figure 13–15. • Updated *Note (2)* to Figure 13–16. • Updated *Note (2)* to Figure 13–17. • Updated *Note (4)* to Figure 13–21. • Updated *Note (2)* to Figure 13–25.	• Changed unit 'kw' to 'kΩ' in Figures 13–6 and 13–7. • Added note about serial configuration devices supporting 20 MHz and 40 MHz DCLK. • Added infomation about the need for a resistor on nCONFIG if reconfiguration is required. • Added information about MSEL[1..0] internal pull-down resistor value.
July 2005 v2.0	• Updated "Configuration Stage" section. • Updated "PS Configuration Using a Download Cable" section. • Updated Figures 13–8, 13–12, and 13–18.	—
November 2004 v1.1	• Updated "Configuration Stage" section in "Single Device AS Configuration" section. • Updated "Initialization Stage" section in "Single Device AS Configuration" section. • Updated Figure 13–8. • Updated "Initialization Stage" section in "Single Device PS Configuration Using a MAX II Device as an External Host" section. • Updated Table 13–7. • Updated "Single Device PS Configuration Using a Configuration Device" section. • Updated "Initialization Stage" section in "Single Device PS Configuration Using a Configuration Device" section. • Updated Figure 13–18. • Updated "Single Device JTAG Configuration" section.	—
June 2004 v1.0	Added document to the Cyclone II Device Handbook.	—

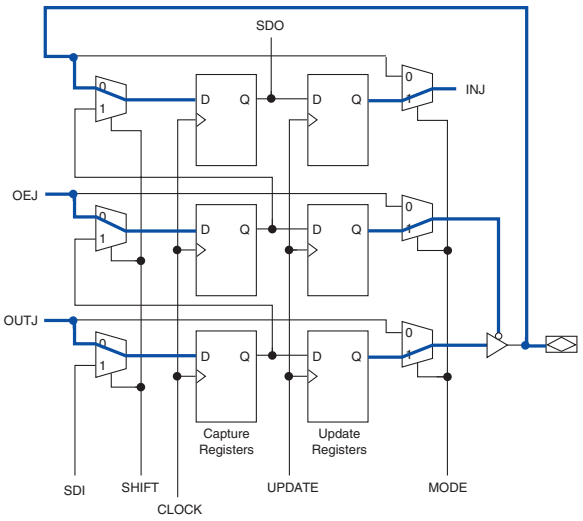
Figure 14–10 shows the capture, shift, and update phases of the EXTEST mode.

Figure 14–10. IEEE Std. 1149.1 BST EXTEST Mode

Capture Phase

In the capture phase, the signals at the pin, OEJ and OUTJ, are loaded into the capture registers. The CLOCK signals are supplied by the TAP controller's CLOCKDR output. Previously retained data in the update registers drive the PIN_IN, INJ, and allows the I/O pin to tri-state or drive a signal out.

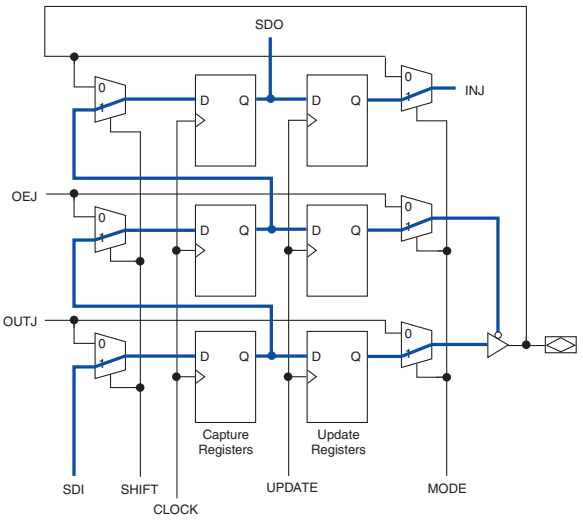
A "1" in the OEJ update register tri-states the output buffer.



Shift & Update Phases

In the shift phase, the previously captured signals at the pin, OEJ and OUTJ, are shifted out of the boundary-scan register via the TDO pin using CLOCK. As data is shifted out, the patterns for the next test can be shifted in via the TDI pin.

In the update phase, data is transferred from the capture registers to the update registers using the UPDATE clock. The update registers then drive the PIN_IN, INJ, and allow the I/O pin to tri-state or drive a signal out.



- Perform a SAMPLE/PRELOAD test cycle prior to the first EXTEST test cycle to ensure that known data is present at the device pins when the EXTEST mode is entered. If the OEJ update register contains a 0, the data in the OUTJ update register is driven out. The state must be known and correct to avoid contention with other devices in the system.
- Do not perform EXTEST testing during ICR. This instruction is supported before or after ICR, but not during ICR. Use the CONFIG_IO instruction to interrupt configuration, then perform testing, or wait for configuration to complete.
- If performing testing before configuration, hold the nCONFIG pin low.
- After configuration, any pins in a differential pin pair cannot be tested. Therefore, performing BST after configuration requires editing BSC group definitions that correspond to these differential pin pairs. The BSC group should be redefined as an internal cell. See the BSDL file for more information on editing.

For more information on boundary scan testing, contact Altera Applications.

Boundary-Scan Description Language (BSDL) Support

The Boundary-Scan Description Language (BSDL), a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.1 BST-capable device that can be tested. Test software development systems then use the BSDL files for test generation, analysis, and failure diagnostics. For more information, or to receive BSDL files for IEEE Std. 1149.1-compliant Cyclone II devices, visit the Altera web site at **www.altera.com**.

Conclusion

The IEEE Std. 1149.1 BST circuitry available in Cyclone II devices provides a cost-effective and efficient way to test systems that contain devices with tight lead spacing. Circuit boards with Altera and other IEEE Std. 1149.1-compliant devices can use the EXTEST, SAMPLE/PRELOAD, BYPASS, IDCODE, USERCODE, CLAMP, and HIGHZ modes to create serial patterns that internally test the pin connections between devices and check device operation.

References

Bleeker, H., P. van den Eijnden, and F. de Jong. *Boundary-Scan Test: A Practical Approach*. Eindhoven, The Netherlands: Kluwer Academic Publishers, 1993.

Institute of Electrical and Electronics Engineers, Inc. *IEEE Standard Test Access Port and Boundary-Scan Architecture* (IEEE Std 1149.1-2001). New York: Institute of Electrical and Electronics Engineers, Inc., 2001.