



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





Section I. Cyclone FPGA Family Data Sheet

This section provides designers with the data sheet specifications for Cyclone® devices. The chapters contain feature definitions of the internal architecture, configuration and JTAG boundary-scan testing information, DC operating conditions, AC timing parameters, a reference to power consumption, and ordering information for Cyclone devices.

This section contains the following chapters:

- [Chapter 1. Introduction](#)
- [Chapter 2. Cyclone Architecture](#)
- [Chapter 3. Configuration and Testing](#)
- [Chapter 4. DC and Switching Characteristics](#)
- [Chapter 5. Reference and Ordering Information](#)

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.

Introduction

The Cyclone® field programmable gate array family is based on a 1.5-V, 0.13-µm, all-layer copper SRAM process, with densities up to 20,060 logic elements (LEs) and up to 288 Kbits of RAM. With features like phase-locked loops (PLLs) for clocking and a dedicated double data rate (DDR) interface to meet DDR SDRAM and fast cycle RAM (FCRAM) memory requirements, Cyclone devices are a cost-effective solution for data-path applications. Cyclone devices support various I/O standards, including LVDS at data rates up to 640 megabits per second (Mbps), and 66- and 33-MHz, 64- and 32-bit peripheral component interconnect (PCI), for interfacing with and supporting ASSP and ASIC devices. Altera also offers new low-cost serial configuration devices to configure Cyclone devices.

Features

The Cyclone device family offers the following features:

- 2,910 to 20,060 LEs, see [Table 1-1](#)
- Up to 294,912 RAM bits (36,864 bytes)
- Supports configuration through low-cost serial configuration device
- Support for LVTTTL, LVCMOS, SSTL-2, and SSTL-3 I/O standards
- Support for 66- and 33-MHz, 64- and 32-bit PCI standard
- High-speed (640 Mbps) LVDS I/O support
- Low-speed (311 Mbps) LVDS I/O support
- 311-Mbps RSDS I/O support
- Up to two PLLs per device provide clock multiplication and phase shifting
- Up to eight global clock lines with six clock resources available per logic array block (LAB) row
- Support for external memory, including DDR SDRAM (133 MHz), FCRAM, and single data rate (SDR) SDRAM
- Support for multiple intellectual property (IP) cores, including Altera® MegaCore® functions and Altera Megafunctions Partners Program (AMPPSM) megafunctions.

Table 1-1. Cyclone Device Features (Part 1 of 2)

| Feature | EP1C3 | EP1C4 | EP1C6 | EP1C12 | EP1C20 |
|--------------------------------|-------|-------|-------|--------|--------|
| LEs | 2,910 | 4,000 | 5,980 | 12,060 | 20,060 |
| M4K RAM blocks (128 × 36 bits) | 13 | 17 | 20 | 52 | 64 |

Table 1–1. Cyclone Device Features (Part 2 of 2)

| Feature | EP1C3 | EP1C4 | EP1C6 | EP1C12 | EP1C20 |
|---------------------------|--------|--------|--------|---------|---------|
| Total RAM bits | 59,904 | 78,336 | 92,160 | 239,616 | 294,912 |
| PLLs | 1 | 2 | 2 | 2 | 2 |
| Maximum user I/O pins (1) | 104 | 301 | 185 | 249 | 301 |

Note to [Table 1–1](#):

- (1) This parameter includes global clock pins.

Cyclone devices are available in quad flat pack (QFP) and space-saving FineLine® BGA packages (see [Tables 1–2 through 1–3](#)).

Table 1–2. Cyclone Package Options and I/O Pin Counts

| Device | 100-Pin TQFP (1) | 144-Pin TQFP (1), (2) | 240-Pin PQFP (1) | 256-Pin FineLine BGA | 324-Pin FineLine BGA | 400-Pin FineLine BGA |
|--------|---------------------|--------------------------|---------------------|-------------------------|-------------------------|-------------------------|
| EP1C3 | 65 | 104 | — | — | — | — |
| EP1C4 | — | — | — | — | 249 | 301 |
| EP1C6 | — | 98 | 185 | 185 | — | — |
| EP1C12 | — | — | 173 | 185 | 249 | — |
| EP1C20 | — | — | — | — | 233 | 301 |

Notes to [Table 1–2](#):

- (1) TQFP: thin quad flat pack.
PQFP: plastic quad flat pack.
- (2) Cyclone devices support vertical migration within the same package (i.e., designers can migrate between the EP1C3 device in the 144-pin TQFP package and the EP1C6 device in the same package).

Vertical migration means you can migrate a design from one device to another that has the same dedicated pins, JTAG pins, and power pins, and are subsets or supersets for a given package across device densities. The largest density in any package has the highest number of power pins; you must use the layout for the largest planned density in a package to provide the necessary power pins for migration.

For I/O pin migration across densities, cross-reference the available I/O pins using the device pin-outs for all planned densities of a given package type to identify which I/O pins can be migrated. The Quartus® II software can automatically cross-reference and place all pins for you when given a device migration list. If one device has power or ground pins, but these same pins are user I/O on a different device that is in the migration path, the Quartus II software ensures the pins are not used as user I/O in the Quartus II software. Ensure that these pins are connected

to the appropriate plane on the board. The Quartus II software reserves I/O pins as power pins as necessary for layout with the larger densities in the same package having more power pins.

Table 1–3. Cyclone QFP and FineLine BGA Package Sizes

| Dimension | 100-Pin TQFP | 144-Pin TQFP | 240-Pin PQFP | 256-Pin FineLine BGA | 324-Pin FineLine BGA | 400-Pin FineLine BGA |
|-----------------------------|-----------------|-----------------|-----------------|----------------------------|----------------------------|----------------------------|
| Pitch (mm) | 0.5 | 0.5 | 0.5 | 1.0 | 1.0 | 1.0 |
| Area (mm ²) | 256 | 484 | 1,024 | 289 | 361 | 441 |
| Length × width (mm × mm) | 16×16 | 22×22 | 34.6×34.6 | 17×17 | 19×19 | 21×21 |

Document Revision History

Table 1–4 shows the revision history for this document.

Table 1–4. Document Revision History

| Date and Document Version | Changes Made | Summary of Changes |
|---------------------------|---|--------------------|
| May 2008 v1.5 | Minor textual and style changes. | — |
| January 2007 v1.4 | Added document revision history. | — |
| August 2005 v1.3 | Minor updates. | — |
| October 2003 v1.2 | Added 64-bit PCI support information. | — |
| September 2003 v1.1 | <ul style="list-style-type: none"> ● Updated LVDS data rates to 640 Mbps from 311 Mbps. ● Updated RSFS feature information. | — |
| May 2003 v1.0 | Added document to Cyclone Device Handbook. | — |

Functional Description

Cyclone® devices contain a two-dimensional row- and column-based architecture to implement custom logic. Column and row interconnects of varying speeds provide signal interconnects between LABs and embedded memory blocks.

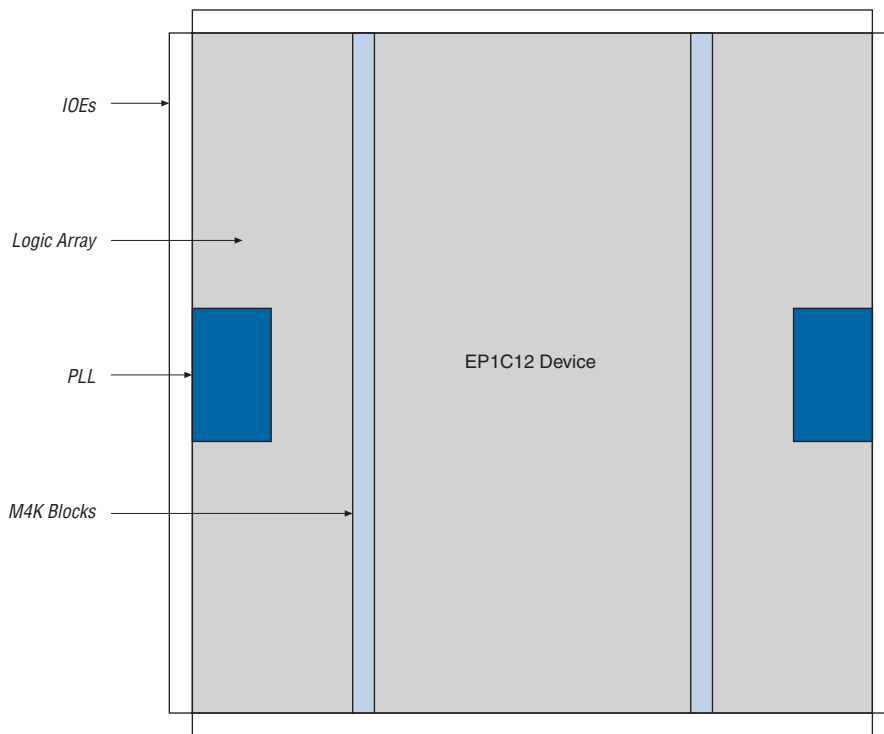
The logic array consists of LABs, with 10 LEs in each LAB. An LE is a small unit of logic providing efficient implementation of user logic functions. LABs are grouped into rows and columns across the device. Cyclone devices range between 2,910 to 20,060 LEs.

M4K RAM blocks are true dual-port memory blocks with 4K bits of memory plus parity (4,608 bits). These blocks provide dedicated true dual-port, simple dual-port, or single-port memory up to 36-bits wide at up to 250 MHz. These blocks are grouped into columns across the device in between certain LABs. Cyclone devices offer between 60 to 288 Kbits of embedded RAM.

Each Cyclone device I/O pin is fed by an I/O element (IOE) located at the ends of LAB rows and columns around the periphery of the device. I/O pins support various single-ended and differential I/O standards, such as the 66- and 33-MHz, 64- and 32-bit PCI standard and the LVDS I/O standard at up to 640 Mbps. Each IOE contains a bidirectional I/O buffer and three registers for registering input, output, and output-enable signals. Dual-purpose DQS, DQ, and DM pins along with delay chains (used to phase-align DDR signals) provide interface support with external memory devices such as DDR SDRAM, and FCRAM devices at up to 133 MHz (266 Mbps).

Cyclone devices provide a global clock network and up to two PLLs. The global clock network consists of eight global clock lines that drive throughout the entire device. The global clock network can provide clocks for all resources within the device, such as IOEs, LEs, and memory blocks. The global clock lines can also be used for control signals. Cyclone PLLs provide general-purpose clocking with clock multiplication and phase shifting as well as external outputs for high-speed differential I/O support.

Figure 2–1 shows a diagram of the Cyclone EP1C12 device.

Figure 2–1. Cyclone EP1C12 Device Block Diagram

The number of M4K RAM blocks, PLLs, rows, and columns vary per device. [Table 2–1](#) lists the resources available in each Cyclone device.

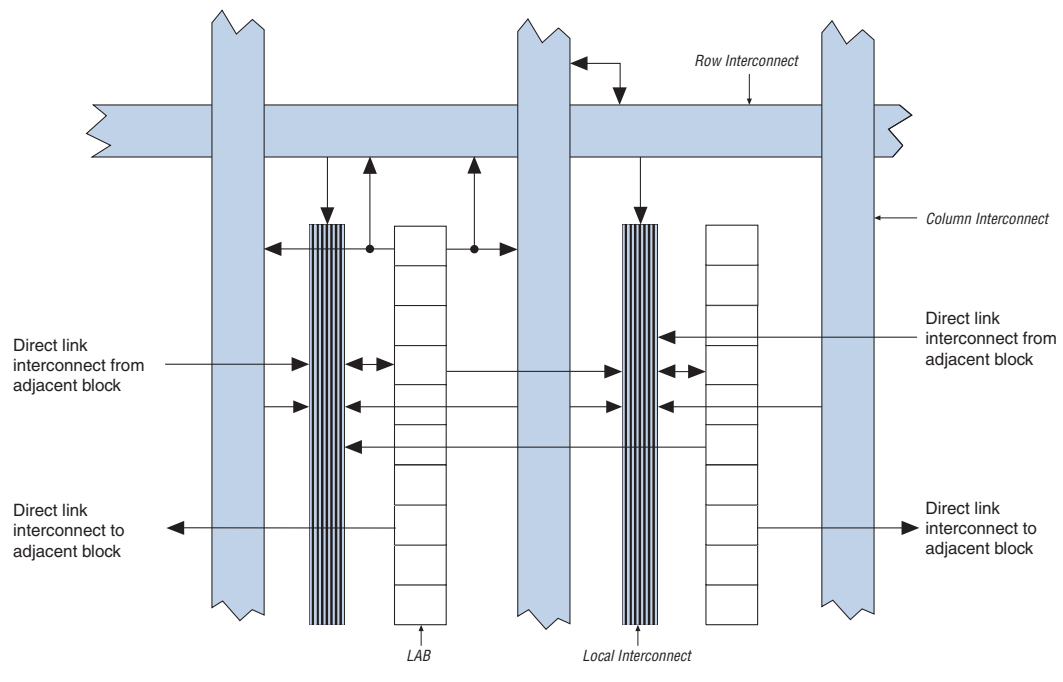
Table 2–1. Cyclone Device Resources

| Device | M4K RAM | | PLLs | LAB Columns | LAB Rows |
|--------|---------|--------|------|-------------|----------|
| | Columns | Blocks | | | |
| EP1C3 | 1 | 13 | 1 | 24 | 13 |
| EP1C4 | 1 | 17 | 2 | 26 | 17 |
| EP1C6 | 1 | 20 | 2 | 32 | 20 |
| EP1C12 | 2 | 52 | 2 | 48 | 26 |
| EP1C20 | 2 | 64 | 2 | 64 | 32 |

Logic Array Blocks

Each LAB consists of 10 LEs, LE carry chains, LAB control signals, a local interconnect, look-up table (LUT) chain, and register chain connection lines. The local interconnect transfers signals between LEs in the same LAB. LUT chain connections transfer the output of one LE's LUT to the adjacent LE for fast sequential LUT connections within the same LAB. Register chain connections transfer the output of one LE's register to the adjacent LE's register within a LAB. The Quartus® II Compiler places associated logic within a LAB or adjacent LABs, allowing the use of local, LUT chain, and register chain connections for performance and area efficiency. Figure 2-2 details the Cyclone LAB structure.

Figure 2-2. Cyclone LAB Structure

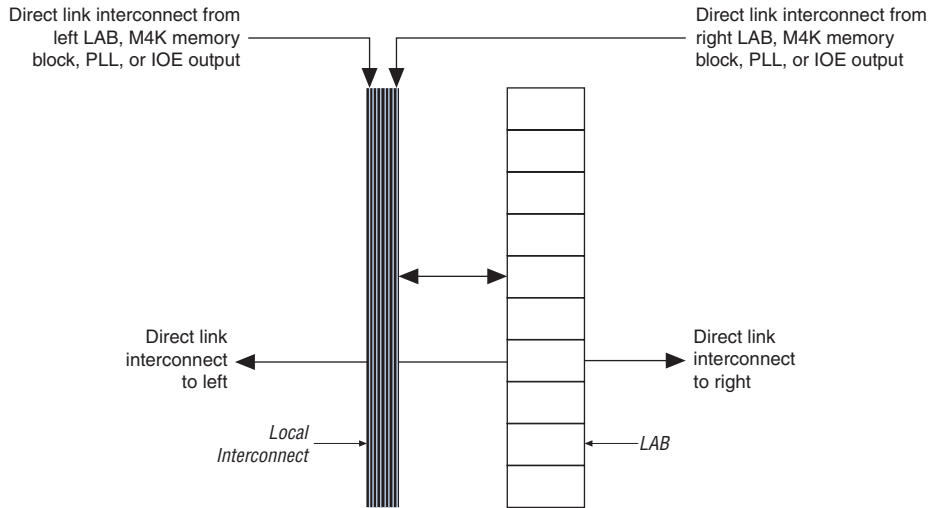


LAB Interconnects

The LAB local interconnect can drive LEs within the same LAB. The LAB local interconnect is driven by column and row interconnects and LE outputs within the same LAB. Neighboring LABs, PLLs, and M4K RAM blocks from the left and right can also drive a LAB's local interconnect through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher

performance and flexibility. Each LE can drive 30 other LEs through fast local and direct link interconnects. Figure 2–3 shows the direct link connection.

Figure 2–3. Direct Link Connection



LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its LEs. The control signals include two clocks, two clock enables, two asynchronous clears, synchronous clear, asynchronous preset/load, synchronous load, and add/subtract control signals. This gives a maximum of 10 control signals at a time. Although synchronous load and clear signals are generally used when implementing counters, they can also be used with other functions.

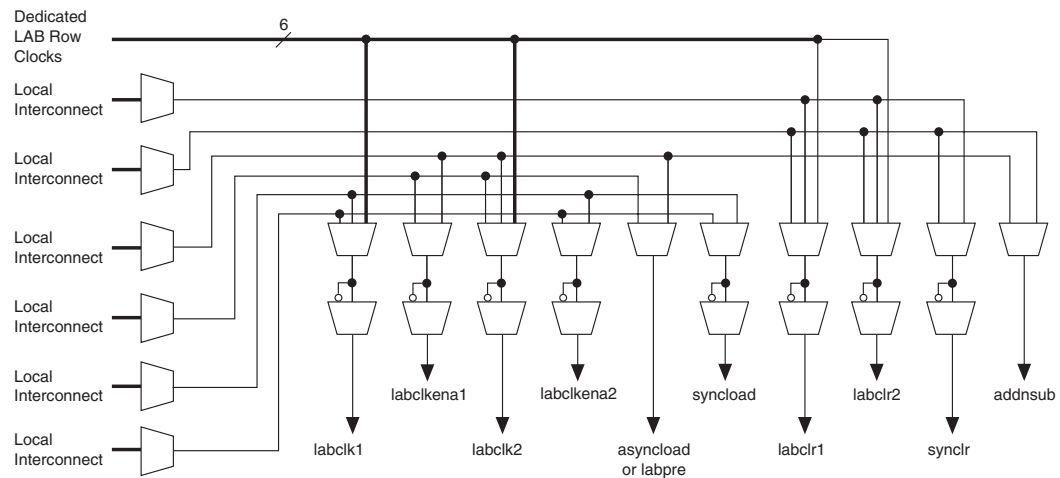
Each LAB can use two clocks and two clock enable signals. Each LAB's clock and clock enable signals are linked. For example, any LE in a particular LAB using the `labclk1` signal will also use `labckena1`. If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock signals. Deasserting the clock enable signal will turn off the LAB-wide clock.

Each LAB can use two asynchronous clear signals and an asynchronous load/preset signal. The asynchronous load acts as a preset when the asynchronous load data input is tied high.

With the LAB-wide `addnsub` control signal, a single LE can implement a one-bit adder and subtractor. This saves LE resources and improves performance for logic functions such as DSP correlators and signed multipliers that alternate between addition and subtraction depending on data.

The LAB row clocks [5 . . 0] and LAB local interconnect generate the LAB-wide control signals. The MultiTrack™ interconnect's inherent low skew allows clock and control signal distribution in addition to data. [Figure 2-4](#) shows the LAB control signal generation circuit.

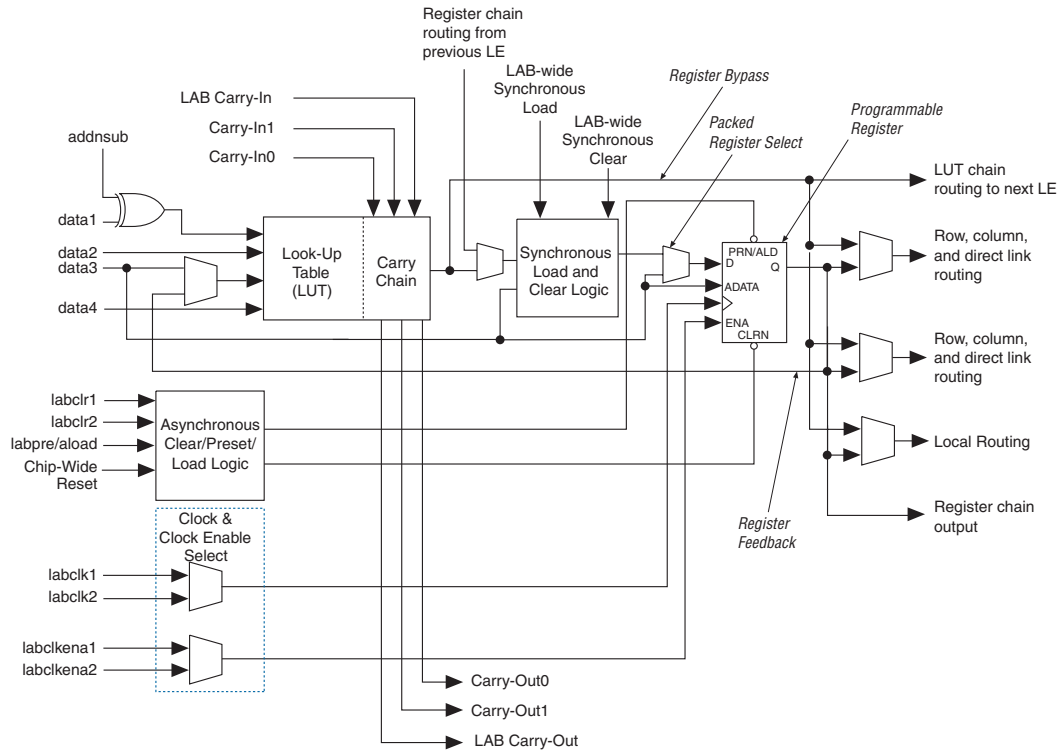
Figure 2-4. LAB-Wide Control Signals



Logic Elements

The smallest unit of logic in the Cyclone architecture, the LE, is compact and provides advanced features with efficient logic utilization. Each LE contains a four-input LUT, which is a function generator that can implement any function of four variables. In addition, each LE contains a programmable register and carry chain with carry select capability. A single LE also supports dynamic single bit addition or subtraction mode selectable by a LAB-wide control signal. Each LE drives all types of interconnects: local, row, column, LUT chain, register chain, and direct link interconnects. See [Figure 2-5](#).

Figure 2-5. Cyclone LE



Each LE's programmable register can be configured for D, T, JK, or SR operation. Each register has data, true asynchronous load data, clock, clock enable, clear, and asynchronous load/preset inputs. Global signals, general-purpose I/O pins, or any internal logic can drive the register's clock and clear control signals. Either general-purpose I/O pins or internal logic can drive the clock enable, preset, asynchronous load, and asynchronous data. The asynchronous load data input comes from the data3 input of the LE. For combinatorial functions, the LUT output bypasses the register and drives directly to the LE outputs.

Each LE has three outputs that drive the local, row, and column routing resources. The LUT or register output can drive these three outputs independently. Two LE outputs drive column or row and direct link routing connections and one drives local interconnect resources. This allows the LUT to drive one output while the register drives another output. This feature, called register packing, improves device utilization because the device can use the register and the LUT for unrelated

functions. Another special packing mode allows the register output to feed back into the LUT of the same LE so that the register is packed with its own fan-out LUT. This provides another mechanism for improved fitting. The LE can also drive out registered and unregistered versions of the LUT output.

LUT Chain and Register Chain

In addition to the three general routing outputs, the LEs within a LAB have LUT chain and register chain outputs. LUT chain connections allow LUTs within the same LAB to cascade together for wide input functions. Register chain outputs allow registers within the same LAB to cascade together. The register chain output allows a LAB to use LUTs for a single combinatorial function and the registers to be used for an unrelated shift register implementation. These resources speed up connections between LABs while saving local interconnect resources. “[MultiTrack Interconnect](#)” on page 2–12 for more information on LUT chain and register chain connections.

addsub Signal

The LE's dynamic adder/subtractor feature saves logic resources by using one set of LEs to implement both an adder and a subtractor. This feature is controlled by the LAB-wide control signal `addsub`. The `addsub` signal sets the LAB to perform either $A + B$ or $A - B$. The LUT computes addition; subtraction is computed by adding the two's complement of the intended subtractor. The LAB-wide signal converts to two's complement by inverting the B bits within the LAB and setting carry-in = 1 to add one to the least significant bit (LSB). The LSB of an adder/subtractor must be placed in the first LE of the LAB, where the LAB-wide `addsub` signal automatically sets the carry-in to 1. The Quartus II Compiler automatically places and uses the adder/subtractor feature when using adder/subtractor parameterized functions.

LE Operating Modes

The Cyclone LE can operate in one of the following modes:

- Normal mode
- Dynamic arithmetic mode

Each mode uses LE resources differently. In each mode, eight available inputs to the LE—the four data inputs from the LAB local interconnect, `carry-in0` and `carry-in1` from the previous LE, the LAB carry-in from the previous carry-chain LAB, and the register chain connection—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, asynchronous

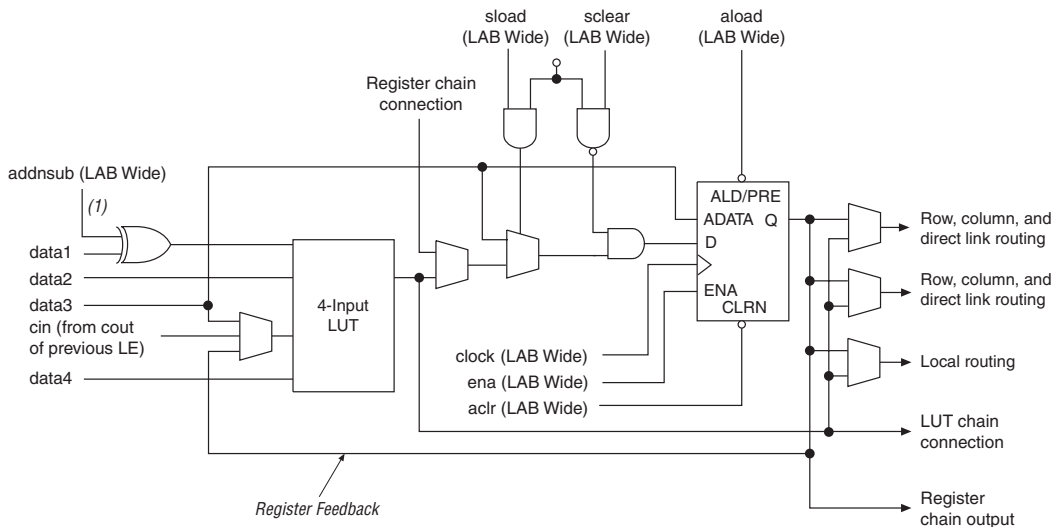
preset/load, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all LE modes. The `addnsub` control signal is allowed in arithmetic mode.

The Quartus II software, in conjunction with parameterized functions such as library of parameterized modules (LPM) functions, automatically chooses the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions. If required, you can also create special-purpose functions that specify which LE operating mode to use for optimal performance.

Normal Mode

The normal mode is suitable for general logic applications and combinatorial functions. In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT (see Figure 2-6). The Quartus II Compiler automatically selects the carry-in or the `data3` signal as one of the inputs to the LUT. Each LE can use LUT chain connections to drive its combinatorial output directly to the next LE in the LAB. Asynchronous load data for the register comes from the `data3` input of the LE. LEs in normal mode support packed registers.

Figure 2-6. LE in Normal Mode



Note to Figure 2-6:

(1) This signal is only allowed in normal mode if the LE is at the end of an adder/subtractor chain.

Dynamic Arithmetic Mode

The dynamic arithmetic mode is ideal for implementing adders, counters, accumulators, wide parity functions, and comparators. An LE in dynamic arithmetic mode uses four 2-input LUTs configurable as a dynamic adder/subtractor. The first two 2-input LUTs compute two summations based on a possible carry-in of 1 or 0; the other two LUTs generate carry outputs for the two chains of the carry select circuitry. As shown in [Figure 2-7](#), the LAB carry-in signal selects either the `carry-in0` or `carry-in1` chain. The selected chain's logic level in turn determines which parallel sum is generated as a combinatorial or registered output. For example, when implementing an adder, the sum output is the selection of two possible calculated sums:

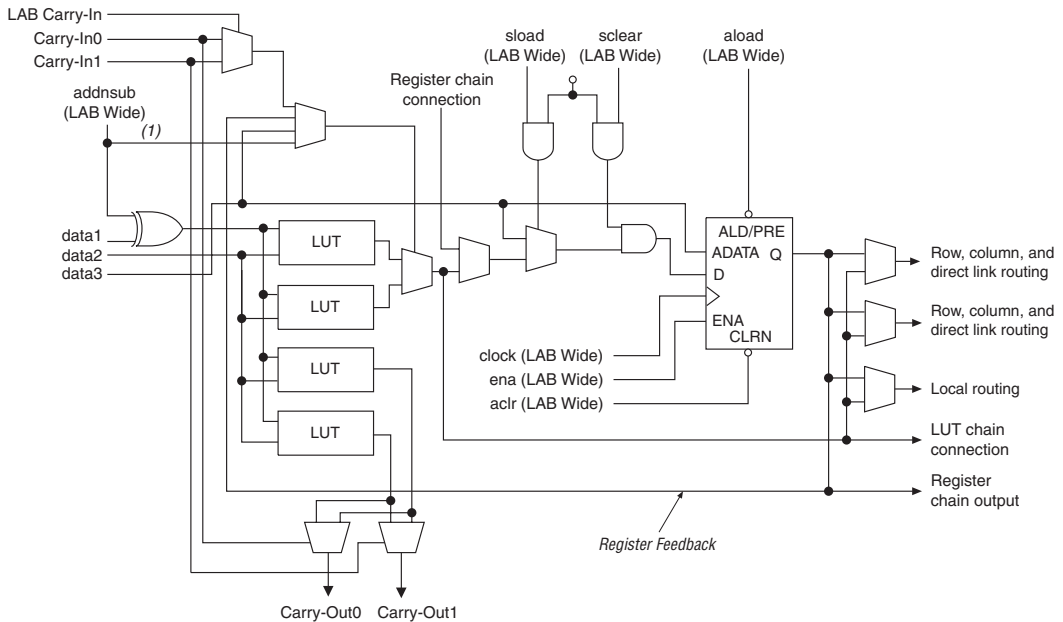
$$\text{data1} + \text{data2} + \text{carry-in0}$$

or

$$\text{data1} + \text{data2} + \text{carry-in1}$$

The other two LUTs use the `data1` and `data2` signals to generate two possible carry-out signals—one for a carry of 1 and the other for a carry of 0. The `carry-in0` signal acts as the carry select for the `carry-out0` output and `carry-in1` acts as the carry select for the `carry-out1` output. LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output.

The dynamic arithmetic mode also offers clock enable, counter enable, synchronous up/down control, synchronous clear, synchronous load, and dynamic adder/subtractor options. The LAB local interconnect data inputs generate the counter enable and synchronous up/down control signals. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. The Quartus II software automatically places any registers that are not used by the counter into other LABs. The `addnsub` LAB-wide signal controls whether the LE acts as an adder or subtractor.

Figure 2-7. LE in Dynamic Arithmetic Mode**Note to Figure 2-7:**

(1) The addnsb signal is tied to the carry input for the first LE of a carry chain only.

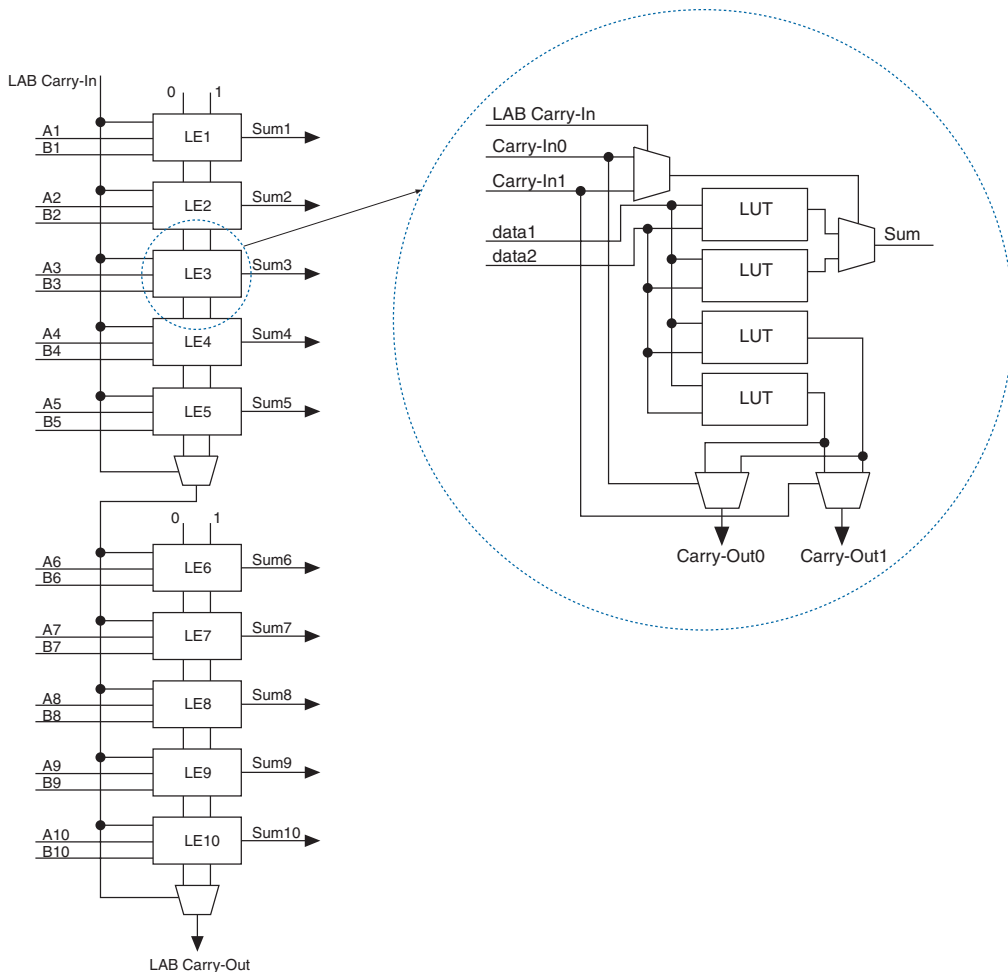
Carry-Select Chain

The carry-select chain provides a very fast carry-select function between LEs in dynamic arithmetic mode. The carry-select chain uses the redundant carry calculation to increase the speed of carry functions. The LE is configured to calculate outputs for a possible carry-in of 0 and carry-in of 1 in parallel. The carry-in0 and carry-in1 signals from a lower-order bit feed forward into the higher-order bit via the parallel carry chain and feed into both the LUT and the next portion of the carry chain. Carry-select chains can begin in any LE within a LAB.

The speed advantage of the carry-select chain is in the parallel pre-computation of carry chains. Since the LAB carry-in selects the pre-computed carry chain, not every LE is in the critical path. Only the propagation delays between LAB carry-in generation (LE 5 and LE 10) are now part of the critical path. This feature allows the Cyclone architecture to implement high-speed counters, adders, multipliers, parity functions, and comparators of arbitrary width.

Figure 2–8 shows the carry-select circuitry in a LAB for a 10-bit full adder. One portion of the LUT generates the sum of two bits using the input signals and the appropriate carry-in bit; the sum is routed to the output of the LE. The register can be bypassed for simple adders or used for accumulator functions. Another portion of the LUT generates carry-out bits. A LAB-wide carry-in bit selects which chain is used for the addition of given inputs. The carry-in signal for each chain, *carry-in0* or *carry-in1*, selects the carry-out to carry forward to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it is fed to local, row, or column interconnects.

Figure 2–8. Carry Select Chain



The Quartus II Compiler automatically creates carry chain logic during design processing, or you can create it manually during design entry. Parameterized functions such as LPM functions automatically take advantage of carry chains for the appropriate functions.

The Quartus II Compiler creates carry chains longer than 10 LEs by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically allowing fast horizontal connections to M4K memory blocks. A carry chain can continue as far as a full column.

Clear and Preset Logic Control

LAB-wide signals control the logic for the register's clear and preset signals. The LE directly supports an asynchronous clear and preset function. The register preset is achieved through the asynchronous load of a logic high. The direct asynchronous preset does not require a NOT-gate push-back technique. Cyclone devices support simultaneous preset/ asynchronous load and clear signals. An asynchronous clear signal takes precedence if both signals are asserted simultaneously. Each LAB supports up to two clears and one preset signal.

In addition to the clear and preset ports, Cyclone devices provide a chip-wide reset pin (DEV_CLRn) that resets all registers in the device. An option set before compilation in the Quartus II software controls this pin. This chip-wide reset overrides all other control signals.

MultiTrack Interconnect

In the Cyclone architecture, connections between LEs, M4K memory blocks, and device I/O pins are provided by the MultiTrack interconnect structure with DirectDrive™ technology. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different speeds used for inter- and intra-design block connectivity. The Quartus II Compiler automatically places critical design paths on faster interconnects to improve design performance.

DirectDrive technology is a deterministic routing technology that ensures identical routing resource usage for any function regardless of placement within the device. The MultiTrack interconnect and DirectDrive technology simplify the integration stage of block-based designing by eliminating the re-optimization cycles that typically follow design changes and additions.

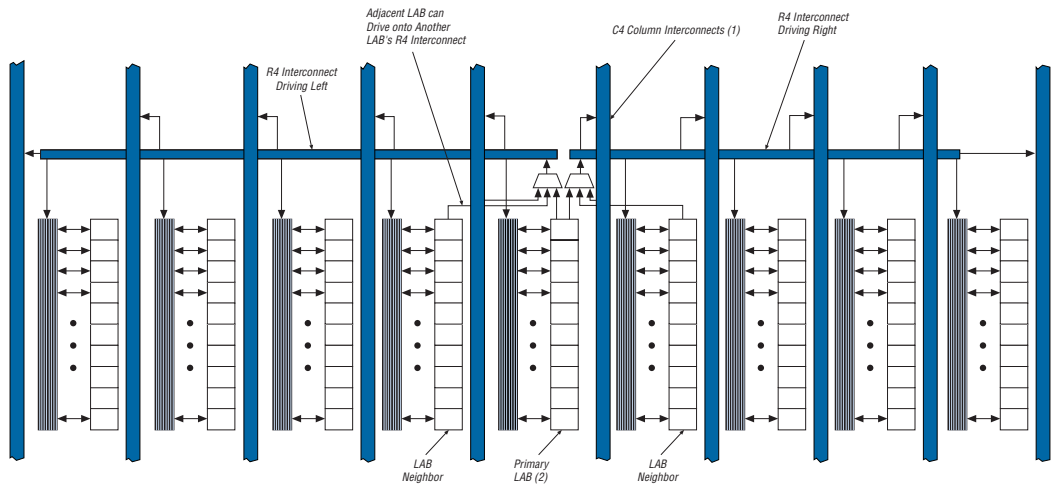
The MultiTrack interconnect consists of row and column interconnects that span fixed distances. A routing structure with fixed length resources for all devices allows predictable and repeatable performance when

migrating through different device densities. Dedicated row interconnects route signals to and from LABs, PLLs, and M4K memory blocks within the same row. These row resources include:

- Direct link interconnects between LABs and adjacent blocks
- R4 interconnects traversing four blocks to the right or left

The direct link interconnect allows a LAB or M4K memory block to drive into the local interconnect of its left and right neighbors. Only one side of a PLL block interfaces with direct link and row interconnects. The direct link interconnect provides fast communication between adjacent LABs and/or blocks without using row interconnect resources.

The R4 interconnects span four LABs, or two LABs and one M4K RAM block. These resources are used for fast row connections in a four-LAB region. Every LAB has its own set of R4 interconnects to drive either left or right. [Figure 2-9](#) shows R4 interconnect connections from a LAB. R4 interconnects can drive and be driven by M4K memory blocks, PLLs, and row IOEs. For LAB interfacing, a primary LAB or LAB neighbor can drive a given R4 interconnect. For R4 interconnects that drive to the right, the primary LAB and right neighbor can drive on to the interconnect. For R4 interconnects that drive to the left, the primary LAB and its left neighbor can drive on to the interconnect. R4 interconnects can drive other R4 interconnects to extend the range of LABs they can drive. R4 interconnects can also drive C4 interconnects for connections from one row to another.

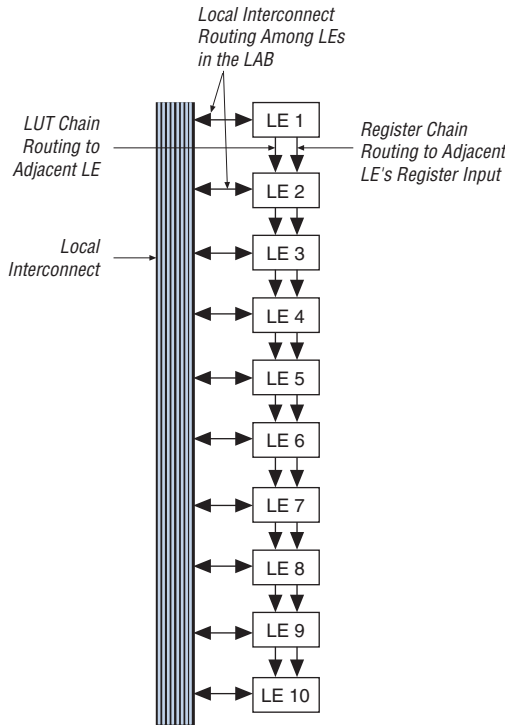
Figure 2–9. R4 Interconnect Connections**Notes to Figure 2–9:**

- (1) C4 interconnects can drive R4 interconnects.
- (2) This pattern is repeated for every LAB in the LAB row.

The column interconnect operates similarly to the row interconnect. Each column of LABs is served by a dedicated column interconnect, which vertically routes signals to and from LABs, M4K memory blocks, and row and column IOEs. These column resources include:

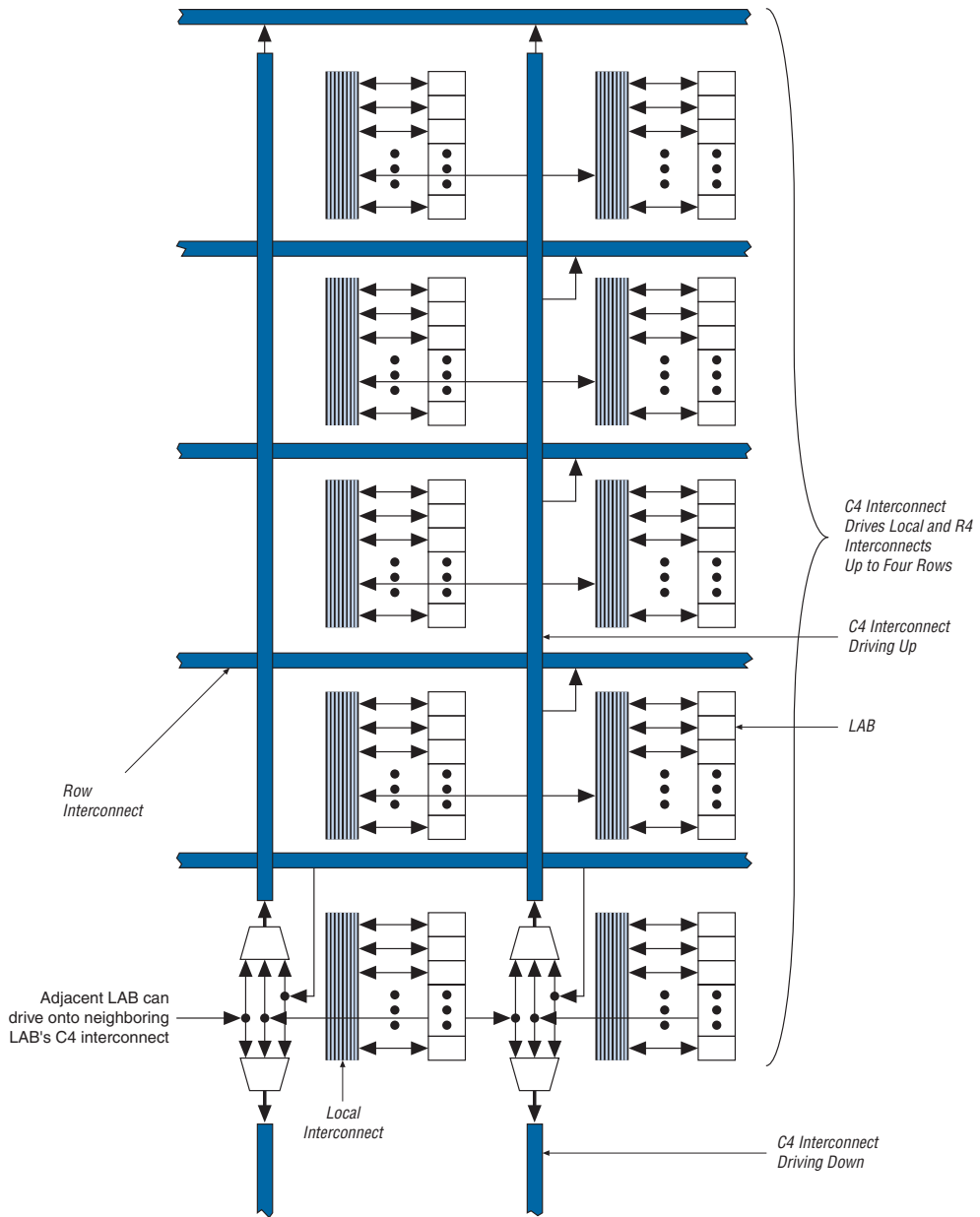
- LUT chain interconnects within a LAB
- Register chain interconnects within a LAB
- C4 interconnects traversing a distance of four blocks in an up and down direction

Cyclone devices include an enhanced interconnect structure within LABs for routing LE output to LE input connections faster using LUT chain connections and register chain connections. The LUT chain connection allows the combinatorial output of an LE to directly drive the fast input of the LE right below it, bypassing the local interconnect. These resources can be used as a high-speed connection for wide fan-in functions from LE 1 to LE 10 in the same LAB. The register chain connection allows the register output of one LE to connect directly to the register input of the next LE in the LAB for fast shift registers. The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance. Figure 2–10 shows the LUT chain and register chain interconnects.

Figure 2–10. LUT Chain and Register Chain Interconnects

The C4 interconnects span four LABs or M4K blocks up or down from a source LAB. Every LAB has its own set of C4 interconnects to drive either up or down. [Figure 2–11](#) shows the C4 interconnect connections from a LAB in a column. The C4 interconnects can drive and be driven by all types of architecture blocks, including PLLs, M4K memory blocks, and column and row IOEs. For LAB interconnection, a primary LAB or its LAB neighbor can drive a given C4 interconnect. C4 interconnects can drive each other to extend their range as well as drive row interconnects for column-to-column connections.

Figure 2-11. C4 Interconnect Connections *Note (1)*



Note to Figure 2-11:

(1) Each C4 interconnect can drive either up or down four rows.

All embedded blocks communicate with the logic array similar to LAB-to-LAB interfaces. Each block (i.e., M4K memory 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 2–2 shows the Cyclone device's routing scheme.

| Source | Destination | | | | | | | | | | |
|--------------------------|-------------|----------------|--------------------|--------------------------|-----------------|-----------------|----|---------------|-----|------------|---------|
| | LUT Chain | Register Chain | Local Interconnect | Direct Link Interconnect | R4 Interconnect | C4 Interconnect | LE | M4K RAM Block | PLL | Column IOE | Row IOE |
| LUT Chain | — | — | — | — | — | — | ✓ | — | — | — | — |
| Register Chain | — | — | — | — | — | — | ✓ | — | — | — | — |
| Local Interconnect | — | — | — | — | — | — | ✓ | ✓ | ✓ | ✓ | ✓ |
| Direct Link Interconnect | — | — | ✓ | — | — | — | — | — | — | — | — |
| R4 Interconnect | — | — | ✓ | — | ✓ | ✓ | — | — | — | — | — |
| C4 Interconnect | — | — | ✓ | — | ✓ | ✓ | — | — | — | — | — |
| LE | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | — | — | — | — | — |
| M4K RAM Block | — | — | ✓ | ✓ | ✓ | ✓ | — | — | — | — | — |
| PLL | — | — | — | ✓ | ✓ | ✓ | — | — | — | — | — |
| Column IOE | — | — | — | — | — | ✓ | — | — | — | — | — |
| Row IOE | — | — | — | ✓ | ✓ | ✓ | — | — | — | — | — |

Embedded Memory

The Cyclone embedded memory consists of columns of M4K memory blocks. EP1C3 and EP1C6 devices have one column of M4K blocks, while EP1C12 and EP1C20 devices have two columns (refer to [Table 1–1 on page 1–1](#) for total RAM bits per density). Each M4K block can implement various types of memory with or without parity, including true dual-port, simple dual-port, and single-port RAM, ROM, and FIFO buffers. The M4K blocks support the following features:

- 4,608 RAM bits
- 250 MHz performance
- True dual-port memory
- Simple dual-port memory
- Single-port memory
- Byte enable
- Parity bits
- Shift register
- FIFO buffer
- ROM
- Mixed clock mode

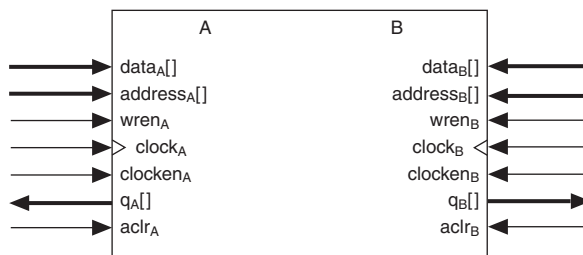


Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Memory Modes

The M4K memory blocks include input registers that synchronize writes and output registers to pipeline designs and improve system performance. M4K blocks offer a true dual-port mode to support any combination of two-port operations: two reads, two writes, or one read and one write at two different clock frequencies. [Figure 2–12](#) shows true dual-port memory.

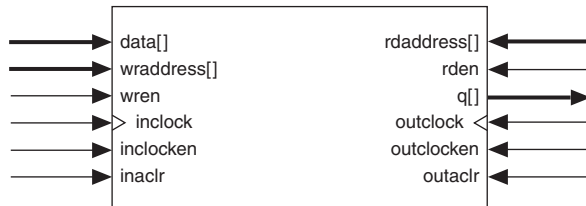
Figure 2–12. True Dual-Port Memory Configuration



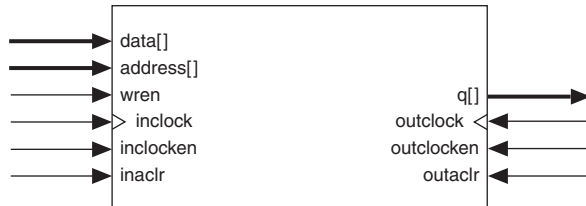
In addition to true dual-port memory, the M4K memory blocks support simple dual-port and single-port RAM. Simple dual-port memory supports a simultaneous read and write. Single-port memory supports non-simultaneous reads and writes. Figure 2–13 shows these different M4K RAM memory port configurations.

Figure 2–13. Simple Dual-Port and Single-Port Memory Configurations

Simple Dual-Port Memory



Single-Port Memory (1)



Note to Figure 2–13:

- (1) Two single-port memory blocks can be implemented in a single M4K block as long as each of the two independent block sizes is equal to or less than half of the M4K block size.

The memory blocks also enable mixed-width data ports for reading and writing to the RAM ports in dual-port RAM configuration. For example, the memory block can be written in $\times 1$ mode at port A and read out in $\times 16$ mode from port B.

The Cyclone memory architecture can implement fully synchronous RAM by registering both the input and output signals to the M4K RAM block. All M4K memory block inputs are registered, providing synchronous write cycles. In synchronous operation, the memory block generates its own self-timed strobe write enable (*wren*) signal derived from a global clock. In contrast, a circuit using asynchronous RAM must generate the RAM *wren* signal while ensuring its data and address signals meet setup and hold time specifications relative to the *wren*