

The Cyclone[®] III device family (Cyclone III and Cyclone III LS devices) includes a combination of on-chip resources and external interfaces that help to increase performance, reduce system cost, and lower the power consumption of digital signal processing (DSP) systems. The Cyclone III device family, either alone or as DSP device co-processors, are used to improve price-to-performance ratios of DSP systems. Particular focus is placed on optimizing Cyclone III and Cyclone III LS devices for applications that benefit from an abundance of parallel processing resources, which include video and image processing, intermediate frequency (IF) modems used in wireless communications systems, and multi-channel communications and video systems.

This chapter contains the following sections:

- “Embedded Multiplier Block Overview” on page 4–2
- “Architecture” on page 4–3
- “Operational Modes” on page 4–5

Embedded Multiplier Block Overview

Figure 4-1 shows one of the embedded multiplier columns with the surrounding logic array blocks (LABs). The embedded multiplier is configured as either one 18×18 multiplier or two 9×9 multipliers. For multiplications greater than 18×18 , the Quartus® II software cascades multiple embedded multiplier blocks together. There are no restrictions on the data width of the multiplier, but the greater the data width, the slower the multiplication process.

Figure 4-1. Embedded Multipliers Arranged in Columns with Adjacent LABs

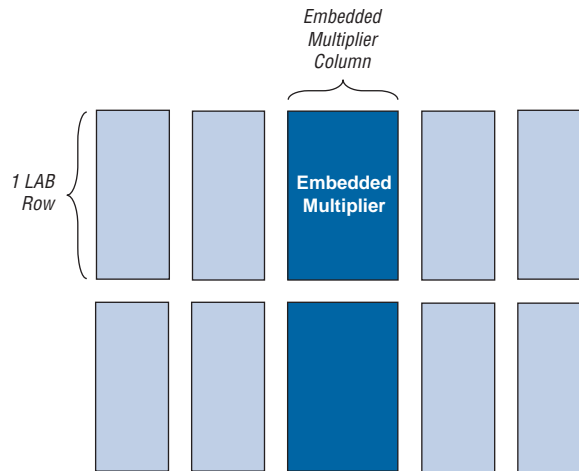


Table 4-1 lists the number of embedded multipliers and the multiplier modes that can be implemented in the Cyclone III device family.

Table 4-1. Number of Embedded Multipliers in the Cyclone III Device Family

Device Family	Device	Embedded Multipliers	9×9 Multipliers ⁽¹⁾	18×18 Multipliers ⁽¹⁾
Cyclone III	EP3C5	23	46	23
	EP3C10	23	46	23
	EP3C16	56	112	56
	EP3C25	66	132	66
	EP3C40	126	252	126
	EP3C55	156	312	156
	EP3C80	244	488	244
	EP3C120	288	576	288
Cyclone III LS	EP3CLS70	200	400	200
	EP3CLS100	276	552	276
	EP3CLS150	320	640	320
	EP3CLS200	396	792	396

Note to Table 4-1:

- (1) These columns show the number of 9×9 or 18×18 multipliers for each device. The total number of multipliers for each device is not the sum of all the multipliers.

In addition to the embedded multipliers in the Cyclone III device family, you can implement soft multipliers by using the M9K memory blocks as look-up tables (LUTs). The LUTs contain partial results from the multiplication of input data with coefficients that implement variable depth and width high-performance soft multipliers for low-cost, high-volume DSP applications. The availability of soft multipliers increases the number of available multipliers in the device.

Table 4–2 lists the total number of multipliers available in the Cyclone III device family using embedded multipliers and soft multipliers.


Table 4–2. Number of Multipliers in the Cyclone III Device Family

Device Family	Device	Embedded Multipliers	Soft Multipliers (16 × 16) ⁽¹⁾	Total Multipliers ⁽²⁾
Cyclone III	EP3C5	23	—	23
	EP3C10	23	46	69
	EP3C16	56	56	112
	EP3C25	66	66	132
	EP3C40	126	126	252
	EP3C55	156	260	416
	EP3C80	244	305	549
Cyclone III LS	EP3CL570	200	333	533
	EP3CLS100	276	483	759
	EP3CLS150	320	666	986
	EP3CLS200	396	891	1287

Notes to Table 4–2:

- (1) Soft multipliers are implemented in sum of multiplication mode. M9K memory blocks are configured with 18-bit data widths to support 16-bit coefficients. The sum of the coefficients requires 18-bits of resolution to account for overflow.
- (2) The total number of multipliers may vary, depending on the multiplier mode you use.

 For more information about M9K memory blocks of the Cyclone III device family, refer to the *Memory Blocks in the Cyclone III Device Family* chapter.

 For more information about soft multipliers, refer to the *Implementing Multipliers in FPGA Devices* application note.

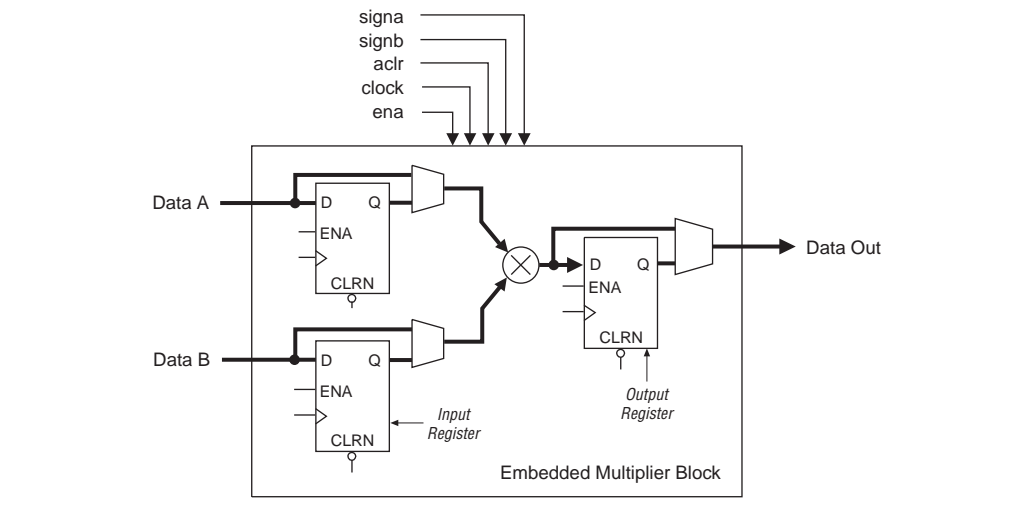
Architecture

Each embedded multiplier consists of the following elements:

- Multiplier stage
- Input and output registers
- Input and output interfaces

Figure 4-2 shows the multiplier block architecture.

Figure 4-2. Multiplier Block Architecture



Input Registers

You can send each multiplier input signal into an input register or directly into the multiplier in 9- or 18-bit sections, depending on the operational mode of the multiplier. Each multiplier input signal can be sent through a register independently of other input signals. For example, you can send the multiplier Data A signal through a register and send the Data B signal directly to the multiplier.

The following control signals are available to each input register in the embedded multiplier:

- clock
- clock enable
- asynchronous clear

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

Multiplier Stage

The multiplier stage of an embedded multiplier block supports 9×9 or 18×18 multipliers as well as other multipliers in between these configurations. Depending on the data width or operational mode of the multiplier, a single embedded multiplier can perform one or two multiplications in parallel. For multiplier information, refer to “Operational Modes” on page 4-5.


Each multiplier operand is a unique signed or unsigned number. Two signals, *signa* and *signb*, control an input of a multiplier and determine if the value is signed or unsigned. If the *signa* signal is high, the Data A operand is a signed number. If the *signa* signal is low, the Data A operand is an unsigned number.

Table 4-3 lists the sign of the multiplication results for the various operand sign representations. The results of the multiplication are signed if any one of the operands is a signed value.

Table 4-3. Multiplier Sign Representation

Data A		Data B		Result
signa Value	Logic Level	signb Value	Logic Level	
Unsigned	Low	Unsigned	Low	Unsigned
Unsigned	Low	Signed	High	Signed
Signed	High	Unsigned	Low	Signed
Signed	High	Signed	High	Signed

Each embedded multiplier block has only one *signa* and one *signb* signal to control the sign representation of the input data to the block. If the embedded multiplier block has two 9×9 multipliers, the *Data A* input of both multipliers share the same *signa* signal, and the *Data B* input of both multipliers share the same *signb* signal. You can dynamically change the *signa* and *signb* signals to modify the sign representation of the input operands at run time. You can send the *signa* and *signb* signals through a dedicated input register. The multiplier offers full precision, regardless of the sign representation.

 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 register the embedded multiplier output using output registers in either 18- or 36-bit sections, depending on the operational mode of the multiplier. The following control signals are available for each output register in the embedded multiplier:


- clock
- clock enable
- asynchronous clear

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

Operational Modes

You can use an embedded multiplier block in one of two operational modes, depending on the application needs:

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

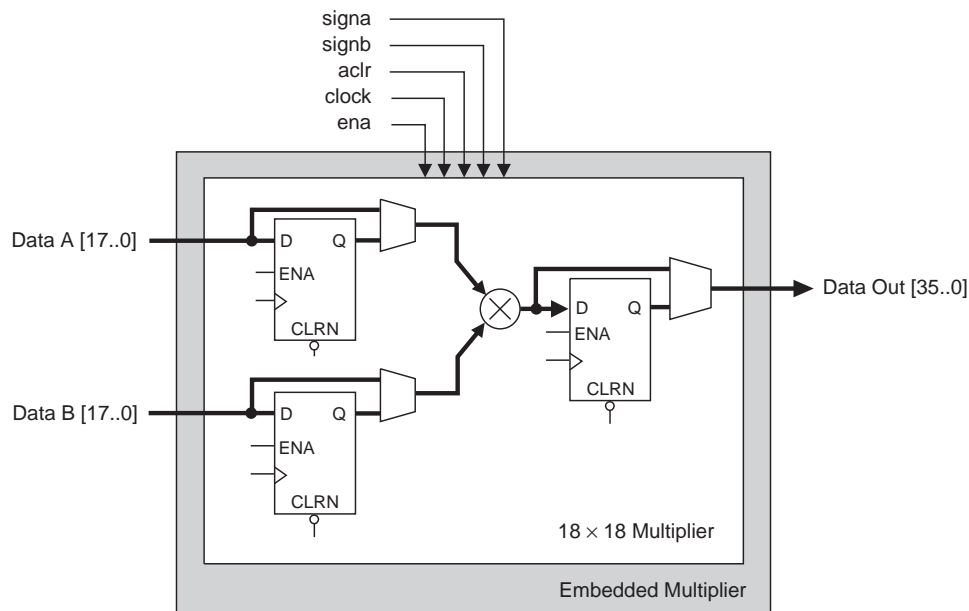
 You can also use embedded multipliers of the Cyclone III device family to implement multiplier adder and multiplier accumulator functions, in which the multiplier portion of the function is implemented using embedded multipliers, and the adder or accumulator function is implemented in logic elements (LEs).

18-Bit Multipliers

You can configure each embedded multiplier to support a single 18×18 multiplier for input widths of 10 to 18 bits.

Figure 4-3 shows the embedded multiplier configured to support an 18-bit multiplier.

Figure 4-3. 18-Bit Multiplier Mode



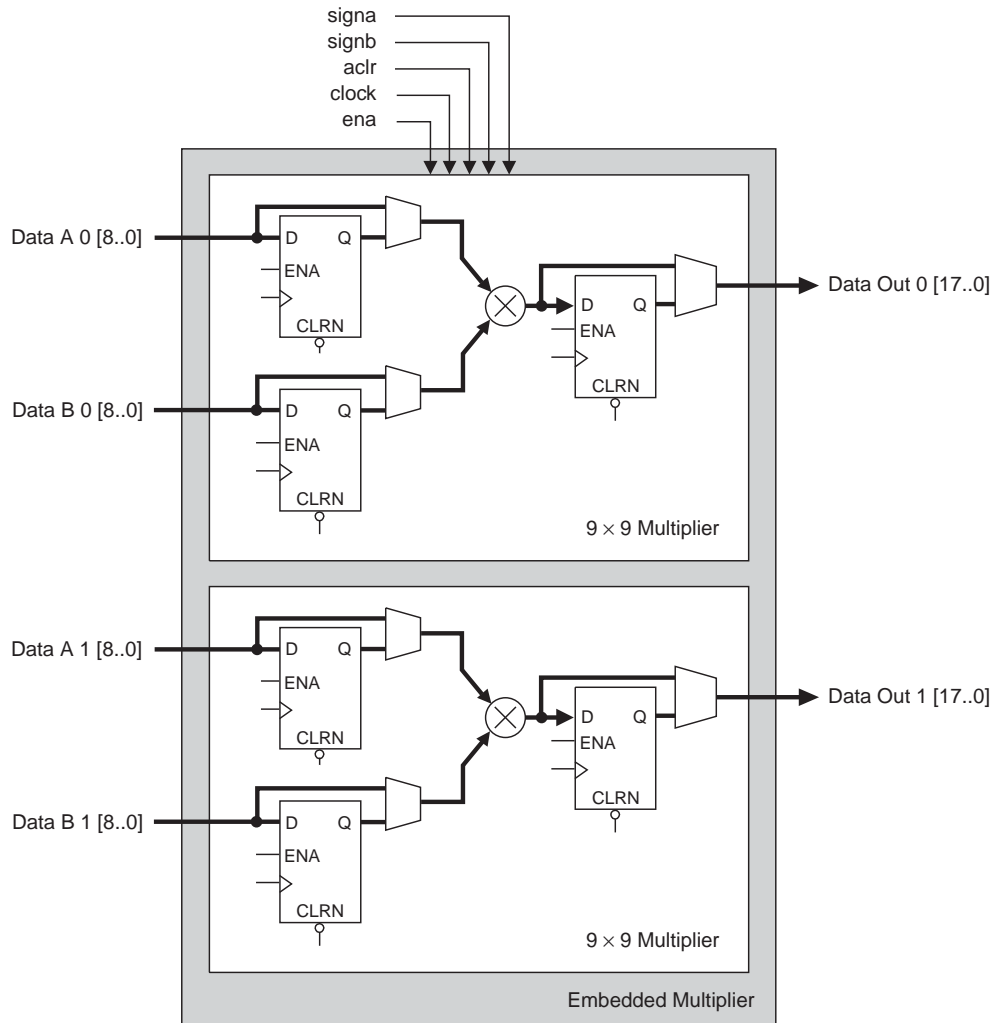
All 18-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Also, you can dynamically change the `signa` and `signb` signals and send these signals through dedicated input registers.

9-Bit Multipliers

You can configure each embedded multiplier to support two 9×9 independent multipliers for input widths of up to 9 bits.

Figure 4-4 shows the embedded multiplier configured to support two 9-bit multipliers.

Figure 4-4. 9-Bit Multiplier Mode



All 9-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Two 9×9 multipliers in the same embedded multiplier block share the same *signa* and *signb* signal. Therefore, all the *Data A* inputs feeding the same embedded multiplier must have the same sign representation. Similarly, all the *Data B* inputs feeding the same embedded multiplier must have the same sign representation.

Document Revision History

Table 4-4 lists the revision history for this document.

Table 4-4. Document Revision History

Date	Version	Changes
December 2011	2.3	Minor text edits.
December 2009	2.2	Minor changes to the text.
July 2009	2.1	Made minor correction to the part number.
June 2009	2.0	Updated to include Cyclone III LS information <ul style="list-style-type: none"> ■ Updated chapter part number. ■ Updated “Introduction” on page 4-1. ■ Updated “Embedded Multiplier Block Overview” on page 4-1. ■ Updated Table 4-1 on page 4-2 and Table 4-2 on page 4-2. ■ Updated “Input Registers” on page 4-4.
October 2008	1.2	Updated chapter to new template.
July 2007	1.1	Added EP3C120 information. <ul style="list-style-type: none"> ■ Updated “Introduction” section. ■ Updated Table 4-1 and Table 4-2. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.