12. Embedded Multipliers in Cyclone II Devices

CII51012-1.2

Introduction

Use Cyclone® II FPGAs alone or as digital signal processing (DSP) co-processors to improve price-to-performance ratios for DSP applications. You can implement high-performance yet low-cost DSP systems with the following Cyclone II device features and design support:

- Up to 150 18 x 18 multipliers
- Up to 1.1 Mbit of on-chip embedded memory
- High-speed interface to external memory
- DSP Intellectual Property (IP) cores
- DSP Builder interface to the Mathworks Simulink and Matlab design environment
- DSP Development Kit, Cyclone II Edition

This chapter focuses on the Cyclone II embedded multiplier blocks.

Cyclone II devices have embedded multiplier blocks optimized for multiplier-intensive low-cost DSP applications. These embedded multipliers combined with the flexibility of programmable logic devices (PLDs), provide you with the ability to efficiently implement various cost sensitive DSP functions easily. Consumer-based application systems such as digital television (DTV) and home entertainment systems typically require a cost effective solution for implementing multipliers to perform signal processing functions like finite impulse response (FIR) filters, fast Fourier transform (FFT) functions, and discrete cosine transform (DCT) functions.

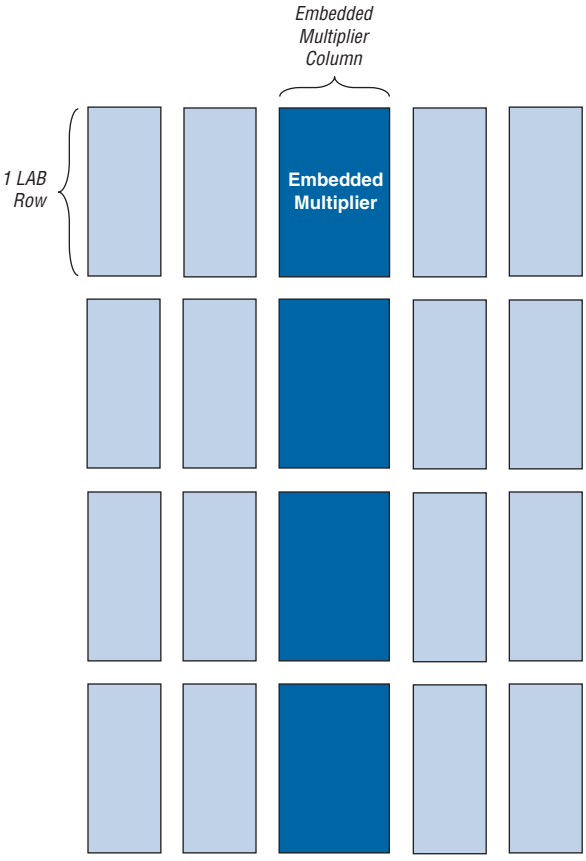
Along with the embedded multipliers, the M4K memory blocks in Cyclone II devices also support various soft multiplier implementations. These, in combination with the embedded multipliers increase the available number of multipliers in Cyclone II devices and provide the user with a wide variety of implementation options and flexibility when designing their systems.

See the Cyclone II Device Family Data Sheet section in Volume 1 of the ***Cyclone II Device Handbook*** for more information on Cyclone II devices.

Embedded Multiplier Block Overview

Each Cyclone II device has one to three columns of embedded multipliers that implement multiplication functions. Figure 12–1 shows one of the embedded multiplier columns with the surrounding LABs. Each embedded multiplier can be configured to support one 18 × 18 multiplier or two 9 × 9 multipliers.

Figure 12–1. Embedded Multipliers Arranged in Columns with Adjacent LABs



The number of embedded multipliers per column and the number of columns available increases with device density. Table 12–1 shows the number of embedded multipliers in each Cyclone II device and the multipliers that you can implement.

Table 12–1. Number of Embedded Multipliers in Cyclone II Devices

Device	Embedded Multipliers	9 × 9 Multipliers *(1)*	18 × 18 Multipliers *(1)*
EP2C5	13	26	13
EP2C8	18	36	18
EP2C20	26	52	26
EP2C35	35	70	35
EP2C50	86	172	86
EP2C70	150	300	150

Note to Table 12–1:
(1) Each device has either the number of 9 × 9 or 18 × 18 multipliers shown. The total number of multipliers for each device is not the sum of all the multipliers.

In addition to the embedded multipliers, you can also implement soft multipliers using Cyclone II M4K memory blocks. The availability of soft multipliers increases the number of multipliers available within the device. Table 12–2 shows the total number of multipliers available in Cyclone II devices using embedded multipliers and soft multipliers.

Table 12–2. Number of Multipliers in Cyclone II Devices

Device	Embedded Multipliers (18 × 18)	Soft Multipliers (16 × 16) *(1)*	Total Multipliers *(2)*
EP2C5	13	26	39
EP2C8	18	36	54
EP2C20	26	52	78
EP2C35	35	105	140
EP2C50	86	129	215
EP2C70	150	250	400

Notes to Table 12–2:
(1) Soft multipliers are implemented in sum of multiplication mode. The M4K 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 according to the multiplier mode used.

See the *Cyclone II Memory Blocks* chapter in Volume 1 of the ***Cyclone II Device Handbook*** for more information on Cyclone II M4K memory blocks.

Refer to *AN 306: Techniques for Implementing Multipliers in FPGA Devices* for more information on soft multipliers.

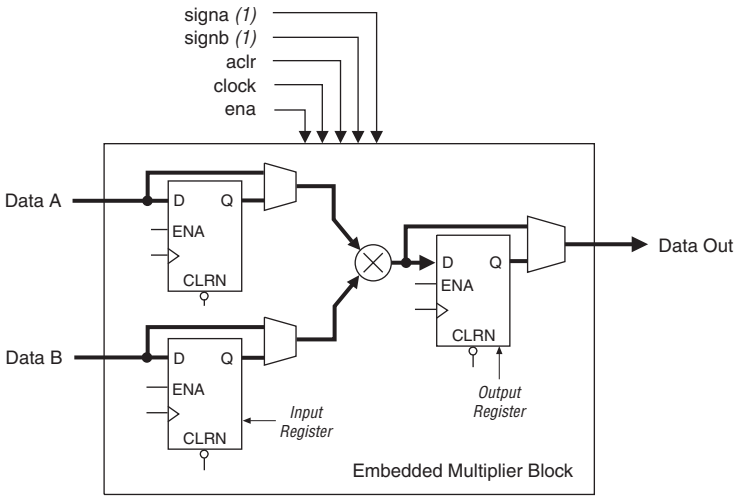
Architecture

Each embedded multiplier consists of the following elements:

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

Figure 12–2 shows the multiplier block architecture.

Figure 12–2. Multiplier Block Architecture



Note to Figure 12–2:
(1) If necessary, you can send these signals through one register to match the data signal path.

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. You can send each multiplier input signal through a register independently of each other (e.g., you can send the multiplier's

data A signal through a register and send the data B signal directly to the multiplier). The following control signals are available to each 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.

Multiplier Stage

The multiplier stage supports 9 × 9 or 18 × 18 multipliers as well as other smaller multipliers in between these configurations. See "Operational Modes" on page 12–6 for details. Depending on the data width or operational mode of the multiplier, a single embedded multiplier can perform one or two multiplications in parallel.

Each multiplier operand can be a unique signed or unsigned number. Two signals, `signa` and `signb`, control whether a multiplier's input is a signed or unsigned value. If the `signa` signal is high, the data A operand is a signed number, and if the `signa` signal is low, the data A operand is an unsigned number. Table 12–3 shows the sign of the multiplication result for the various operand sign representations. The result of the multiplication is signed if any one of the operands is a signed value.

Table 12–3. 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

There is only one `signa` and one `signb` signal for each embedded multiplier. The `signa` and `signb` signals can be changed dynamically 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 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.

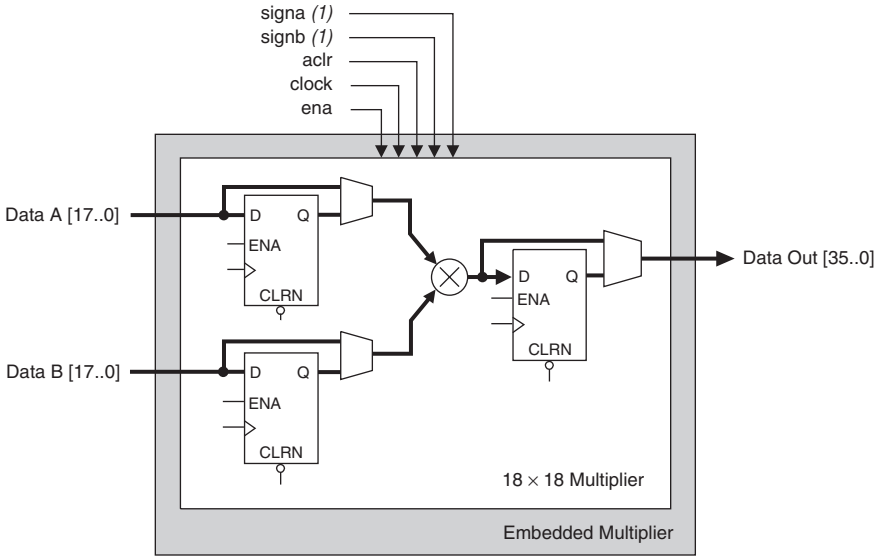
The Cyclone II embedded multipliers can also be used to implement multiplier adder and multiplier accumulator functions where the multiplier portion of the function is implemented using embedded multipliers and the adder or accumulator function is implemented in logic elements (LEs).

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

18-Bit Multipliers

Each embedded multiplier can be configured to support a single 18 × 18 multiplier for input widths from 10- to 18-bits. Figure 12–3 shows the embedded multiplier configured to support an 18-bit multiplier.

Figure 12–3. 18-Bit Multiplier Mode



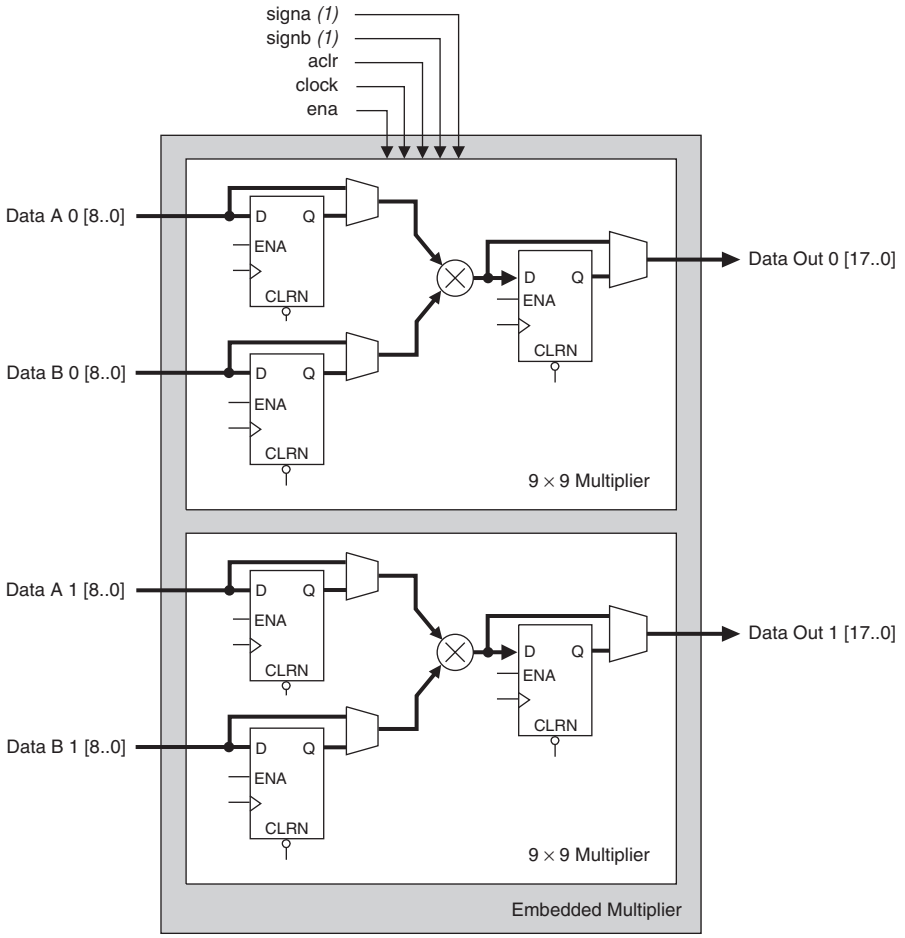
Note to Figure 12–3:
(1) If necessary, you can send these signals through one register to match the data signal path.

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

9-Bit Multipliers

Each embedded multiplier can also be configured to support two 9 × 9 independent multipliers for input widths up to 9-bits. Figure 12–4 shows the embedded multiplier configured to support two 9-bit multipliers.

Figure 12–4. 9-Bit Multiplier Mode



Note to Figure 12–4:
(1) If necessary, you can send these signals through one register to match the data signal path.

All 9-bit multiplier inputs and results can be independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Each embedded multiplier only has one `signa` signal to control the sign representation of both data A inputs (one for each 9 × 9 multiplier) and one `signb` signal to control the sign representation of both data B inputs. Therefore, all of the data A inputs feeding the same embedded multiplier must have the same sign representation. Similarly, all of the data B inputs feeding the same embedded multiplier must have the same sign representation.

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.

Document Revision History

Table 12–4 shows the revision history for this document.

Table 12–4. Document Revision History

Date & Document Version	Changes Made	Summary of Changes
February 2007 v1.2	• Added document revision history. • Updated "Software Support" section.	• Removed reference to third-party synthesis tool: LeonardoSpectrum and Synplify.
November 2005 v2.1	Updated Introduction.	
June 2004 v1.0	Added document to the Cyclone II Device Handbook.	