# Using Embedded Multipliers

## Introduction

Virtex-II devices feature a large number of embedded 18-bit X 18-bit two's-complement embedded multipliers. The embedded multipliers offer fast, efficient means to create 18-bit signed by 18-bit signed multiplication products. The multiplier blocks share routing resources with the Block SelectRAM memory, allowing for increased efficiency for many applications. Cascading of multipliers can be implemented with additional logic resources in local Virtex-II slices.

Applications such as signed-signed, signed-unsigned, and unsigned-unsigned multiplication, logical, arithmetic, and barrel shifters, two's-complement and magnitude return are easily implemented.

Using the CORE Generator, the designer can quickly generate multipliers that make use of the embedded 18-bit x 18-bit two's-complement multipliers (V2.0 or later) of the Multiplier core for Virtex-II devices.

## Two's-Complement Signed Multiplier

### Data Flow

Each embedded multiplier block (MULT18X18 primitive) supports two independent dynamic data input ports: 18-bit signed or 17-bit unsigned. The MULT18X18 primitive is illustrated in Figure 2-66.

In addition, efficient cascading of multipliers up to 35-bit X 35-bit signed can be accomplished by using 4 embedded multipliers, one 36-bit adder, and one 53-bit adder. See Figure 2-67.


*Figure 2-66:* **Embedded Multiplier**

## Library Primitives and Submodules

One library primitive (MULT18X18) is available. Table 2-26 lists the attributes of this primitive.

*Table 2-26:* **Embedded Multiplier Primitive**

| Primitive | A width | B width | P width | Signed/Unsigned |
|---|---|---|---|---|
| MULT18X18 | 18 | 18 | 36 | Signed (2's complement) |

In addition to the primitive, 15 submodules that implement various widths of signed and unsigned multipliers and two's-complement return functions are provided in VHDL and Verilog code. Multipliers using cascaded MULT18X18 primitives are included with registers between stages causing three cycles of latency. Multipliers that make use of the embedded Virtex-II 18-bit by 18-bit two's complement multipliers can be easily generated using V2.0 of the CORE Generator Multiplier module. Table 2-27 lists cascaded multiplier submodules.

*Table 2-27:* **Embedded Multiplier Submodules - Cascaded MULT18X18**

| Submodule | A Width | B Width | P Width | Signed/Unsigned |
|---|---|---|---|---|
| MULT35X35_S | 35 | 35 | 70 | Signed |
| MULT34X34_U | 34 | 34 | 68 | Unsigned |

Figure 2-67 represents the cascaded scheme used to implement a 35-bit by 35-bit signed multiplier utilizing four embedded multipliers and two adders.



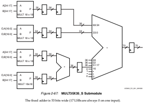

*Figure 2-67:* **MULT35X35_S Submodule**

The fixed adder is 53 bits wide (17 LSBs are always 0 on one input).

The 34-bit by 34-bit unsigned submodule is constructed in a similar manner with the most significant bit on each operand being tied to logic low.

Table 2-27 lists multipliers and two's-complement return functions that utilize one MULT18X18 primitive and are not registered.

*Table 2-28:* **Embedded Multiplier Submodules - Single MULT18X18**

| Submodule | A width | B width | P width | Signed/Unsigned |
|---|---|---|---|---|
| MULT17X17_U | 17 | 17 | 34 | Unsigned |
| MULT8X8_S | 8 | 8 | 16 | Signed |
| MULT8X8_U | 8 | 8 | 16 | Unsigned |
| MULT4X4_S | 4 | 4 | 8 | Signed |
| MULT4X4_U | 4 | 4 | 8 | Unsigned |
| MULT_6X6S_5X5U | 6<br>5 | 6<br>5 | 12<br>10 | Signed<br>Unsigned |
| MULT_5X5S_6X6U | 5<br>6 | 5<br>6 | 10<br>12 | Signed<br>Unsigned |
| MULT_5X5U_5X5U | 5<br>5 | 5<br>5 | 10<br>10 | Unsigned<br>Unsigned |
| MULT_4X4S_7X7U | 4<br>7 | 4<br>7 | 8<br>14 | Signed<br>Unsigned |
| MULT_4X4S_3X3S | 4<br>3 | 4<br>3 | 8<br>6 | Signed<br>Signed |
| TWOS_CMP18 | 18 | - | 18 | - |
| TWOS_CMP9 | 9 | - | 9 | - |
| MAGNTD_18 | 18 | - | 17 | - |

Multipliers of form MULT_aXaS_bXbU use one embedded multiplier to implement two multipliers with separate outputs. The submodules listed above use optimized pin assignments to achieve shortest possible through-delay.

Figure 2-68 and Figure 2-69 represent 4-bit by 4-bit signed multiplier and 4-bit by 4-bit unsigned multiplier implementations, respectively.

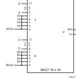


*Figure 2-68:* **MULT4X4_S Submodule**

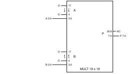


*Figure 2-69:* **MULT4X4_U Submodule**

Submodule MAGNTD_18 performs a magnitude return (i.e., absolute value) of a two's-complement number. An incoming negative number returns with a positive number, while an incoming positive number remains unchanged. Submodules TWOS_CMP18 and TWOS_CMP9 perform a two's-complement return function. The incoming number in two's-complement form (either signed or unsigned) is complemented when the DO_COMP pin is asserted High. Additional slice logic can be used with these submodules to efficiently convert sign-magnitude to two's-complement or vice-versa. Figure 2-70 shows the connections to a MULT18X18 to create the submodule TWOS_CMP9.


*Figure 2-70:* **TWOS_CMP9 Submodule**

## Two Multipliers in a Single Primitive

Two multipliers can be implemented in a single primitive. For simplified illustration purposes, an assumption of two squares being implemented in the same MULT18X18 primitive is used. The following equation shows the form of the multiplication.

**Two Multipliers per Primitive:**

$$(X * 2^n + Y)(X * 2^n + Y) = (X^2 * 2^{2n}) + (Y^2) + (XY * 2^{n+1})$$

$(X * 2^n)$ is the input X appearing on the MSBs while Y appears on the LSBs to form the value $(X * 2^n + Y)$. Two multipliers can coexist in one MULT18X18 primitive, if the conditions in the following inequalities are met when neither X nor Y are 0.

**Inequality Conditions for Two Multipliers per Primitive:**

$$(X^2 * 2^{2n})_{min} > (XY * 2^{n+1})_{max}, (XY * 2^{n+1})_{min} > (Y^2)_{max}$$

For values 0 on X or Y, the equation becomes:

| | |
|---|---|
| $X^2 * 2^{2n}$ | {Y=0} |
| $Y^2$ | {X=0} |
| 0 | {X=0, Y=0} |

Figure 2-71 represents the MULT_6X6S_5X5U submodule.

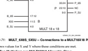


*Figure 2-71:* **MULT_6X6S_5X5U – Connections to a MULT18X18 Primitive**

Table 2-29 shows values for X and Y where these conditions are met.

*Table 2-29:* **Two Multipliers per MULT18X18 Allowable Sizes**

| X * X | | Y * Y | |
|---|---|---|---|
| **Signed Size** | **Unsigned Size** | **Signed Size** | **Unsigned Size** |
| 7 X 7 | 6 X 6 | - | 4 X 4 |
| 6 X 6 | 5 X 5 | - | 5 X 5 |
| 5 X 5 | 4 X 4 | 3 X 3 | 6 X 6 |
| 4 X 4 | 3 X 3 | 3 X 3 | 7 X 7 |
| 3 X 3 | 2 X 2 | 4 X 4 | 8 X 8 |

## VHDL and Verilog Instantiation

VHDL and Verilog instantiation templates are available as examples of primitives and submodules (see "VHDL and Verilog Templates" on page 243).

In VHDL, each template has a component declaration section and an architecture section. Each part of the template should be inserted within the VHDL design file. The port map of the architecture section should include the design signals names.

## Port Signals

### Data In - A

The data input provides new data (up to 18 bits) to be used as one of the multiplication operands.

### Data In - B

The data input provides new data (up to 18 bits) to be used as one of the multiplication operands.

### Data Out - P

The data output bus P provides the data value (up to 36 bits) of two's-complement multiplication for operands A and B.

## Location Constraints

Each embedded multiplier has location coordinates of the form XrowYcolumn. To constrain placement, multiplier instances can have LOC properties attached to

MULT18X18 embedded multiplier instances can have LOC properties attached to them to constrain placement. MULT18X18 placement locations differ from the convention used for naming CLB locations, allowing LOC properties to transfer easily from array to array.

The LOC properties use the following form:

LOC = MULT18X18_X#Y#

For example, MULT18X18_X0Y0 is the bottom-left MULT18X18 location on the device.

## VHDL and Verilog Templates



VHDL and Verilog templates are available for the primitive and submodules.

The following is a template for the primitive:

- SIGNED_MULT_18X18 (primitive: MULT18X18)

The following are templates for submodules:

- SIGNED_MULT_35X35 (submodule: MULT35X35_S)
- UNSIGNED_MULT_34X34 (submodule: MULT34X34_U)
- UNSIGNED_MULT_17X17 (submodule: MULT17X17_U)
- SIGNED_MULT_8X8 (submodule: MULT8X8_S)
- UNSIGNED_MULT_8X8 (submodule: MULT8X8_U)
- SIGNED_MULT_4X4 (submodule: MULT4X4_S)
- UNSIGNED_MULT_4X4 (submodule: MULT4X4_U)
- DUAL_MULT_6X6S_5X5U (submodule: MULT_6X6S_5X5U)
- DUAL_MULT_5X5S_6X6U (submodule: MULT_5X5S_6X6U)
- DUAL_MULT_5X5U_5X5U (submodule: MULT_5X5U_5X5U)
- DUAL_MULT_4X4S_7X7U (submodule: MULT_4X4S_7X7U)
- DUAL_MULT_4X4S_3X3S (submodule: MULT_4X4S_3X3S)
- TWOS_COMPLEMENTER_18BIT (submodule: TWOS_CMP18)
- TWOS_COMPLEMENTER_9BIT (submodule: TWOS_CMP9)
- MAGNITUDE_18BIT (submodule: MAGNTD_18)

The corresponding submodules have to be synthesized with the design.

Templates for the SIGNED_MULT_18X18 module are provided in VHDL and Verilog code as an example.

### VHDL Template:

```
-- Module: SIGNED_MULT_18X18
-- Description: VHDL instantiation template
-- 18-bit X 18-bit embedded signed multiplier (asynchronous)
--
-- Device: Virtex-II Family
---------------------------------------------------------------------
-- Components Declarations
component MULT18X18
  port(
       A :  in std_logic_vector (17 downto 0);
       B :  in std_logic_vector (17 downto 0);
       P : out std_logic_vector (35 downto 0)
  );
end component;
--
-- Architecture Section
--
U_MULT18X18 : MULT18X18
  port map (
    A => , -- insert input signal #1
    B => , -- insert input signal #2
    P =>   -- insert output signal
  );
```

### Verilog Template:

```
// Module: SIGNED_MULT_18X18
// Description: Verilog instantiation template
// 18-bit X 18-bit embedded signed multiplier (asynchronous)
//
// Device: Virtex-II Family
//-------------------------------------------------------------------
// Instantiation Section
//
MULT18X18 U_MULT18X18
  (
   .A () , // insert input signal #1
   .B () , // insert input signal #2
   .P ()   // insert output signal
  );
```