DSP Function Usage Guide for iCE40 Devices

Technical Note

FPGA-TN-02007-1.2

October 2020
Disclaimers
Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer’s responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice’s product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.
Contents

1. Introduction ........................................................................................................................................... 4
2. DSP Primitive – SB_MAC16 .................................................................................................................... 4
   2.1. SB_MAC16 Primitive ............................................................................................................................ 4
   2.2. SB_MAC16 Functional Diagram .......................................................................................................... 5
   2.3. SB_MAC16 Interface Ports .................................................................................................................. 6
   2.4. SB_MAC16 Parameters ....................................................................................................................... 8
3. Implementing DSP Functions .................................................................................................................. 10
   3.1. Inferencing DSP Functions ................................................................................................................ 10
       3.1.1. 32-bit Accumulator with Async Data In and Sync Data Out ...................................................... 10
       3.1.2. 8 x 8 Multiplier, Unsigned with Sync Data In and Data Out .................................................... 12
   3.2. Instantiation DSP Primitive – SB_MAC16 ......................................................................................... 13
       3.2.1. Dual 16 bit Accumulator with Sync Data Out ................................................................................. 15
       3.2.2. Multiplier 16 x 16 Signed .......................................................................................................... 17
       3.2.3. Multiplier 16 x 16 Unsigned .................................................................................................... 19
3. Technical Support Assistance ................................................................................................................ 21
3. Revision History .................................................................................................................................... 22

Figures

Figure 2.1. SB_MAC16 DSP Primitive Interface Diagram ................................................................. 4
Figure 2.2. SB_MAC16 DSP Functional Diagram ................................................................................. 5

Tables

Table 2.1. SB_MAC16 Ports and their Functional Descriptions .......................................................... 6
Table 2.2. SB_MAC16 Parameter Description ......................................................................................... 8
Table 3.1. Instantiation Guide ................................................................................................................. 13
1. Introduction

This technical note discusses DSP function usage for the iCE40™ device family, specifically iCE40 Ultra™ and iCE40 UltraPlus™. It is intended to be used as a guide on various modes and how to configure them for these devices.

The DSP block, referred to as SB_MAC16 primitive in this guide, is an embedded block available in the iCE40 Ultra and iCE40 UltraPlus devices. This block can be configured into combinations of the following functional units by selecting appropriate parameter values.

- Single 16 x 16 Multiplier (generating a 32-bit product output)
- Two independent 8 x 8 Multiplier (generating two independent 16-bit product outputs)
- Single 32-bit Accumulator
- Two independent 16-bit Accumulators
- Single 32-bit Adder/Subtractor
- Two independent 16-bit Adders/Subtractors

2. DSP Primitive – SB_MAC16

The SB_MAC16 primitive is the dedicated configurable DSP block for the iCE40 Ultra and iCE40 UltraPlus devices. This primitive can be configured into a multiplier, adder, subtractor, accumulator, multiply-add and multiply-sub by setting various parameters.

2.1. SB_MAC16 Primitive

Figure 2.1 provides an overview of the SB_MAC16 primitive with various inputs and outputs.

The inputs and outputs of the functional units can be configured independently into:

- Registered Inputs/Outputs
  - The inputs to the functional units can be either registered or unregistered.
  - The outputs from the functional units can be either registered or unregistered.
  - The intermediate multiplier outputs can be registered (pipelined) for faster clock performance.

- Signed/Unsigned Inputs
  - Inputs to the multiplier block can be either a signed or unsigned number.

These various options and their usage is discussed in more detail in the sections that follow.
2.2. **SB_MAC16 Functional Diagram**

Figure 2.2 shows the functional diagram of the SB_MAC16 primitive.

![SB_MAC16 DSP Functional Diagram](image-url)
2.3. SB_MAC16 Interface Ports

Table 2.1 provides a list of interface ports available in SB_MAC16 and their functional description. It is important to note the Default Values of these ports in this table. These values are useful in determining how to connect the ports that are not used in a particular function during instantiation. This is discussed in detail in the sections that follow.

Table 2.1. SB_MAC16 Ports and their Functional Descriptions

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Direction</th>
<th>Functional Description</th>
<th>Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>Input</td>
<td>Clock Input. Applies to all clocked elements.</td>
<td>—</td>
</tr>
<tr>
<td>CE</td>
<td>Input</td>
<td>Clock Enable Input. Applies to all clocked elements.</td>
<td>1</td>
</tr>
<tr>
<td>A[7:0]</td>
<td>Input</td>
<td>Lower 8-Bits data of Input A</td>
<td>8'b0</td>
</tr>
<tr>
<td>A[15:8]</td>
<td>Input</td>
<td>Upper 8-Bits data of Input A</td>
<td>8'b0</td>
</tr>
<tr>
<td>A_HOLD</td>
<td>Input</td>
<td>Register A Hold Input. Control data flow input Register A.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Hold</td>
<td></td>
</tr>
<tr>
<td>B[7:0]</td>
<td>Input</td>
<td>Lower 8-Bits data of Input B</td>
<td>8'b0</td>
</tr>
<tr>
<td>B[15:8]</td>
<td>Input</td>
<td>Upper 8-Bits data of Input B</td>
<td>8'b0</td>
</tr>
<tr>
<td>B_HOLD</td>
<td>Input</td>
<td>Register B Hold Input. Control data flow input Register B.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Hold</td>
<td></td>
</tr>
<tr>
<td>C[15:0]</td>
<td>Input</td>
<td>16-Bits data of Input C</td>
<td>16'b0</td>
</tr>
<tr>
<td>C_HOLD</td>
<td>Input</td>
<td>Register C Hold Input. Control data flow input Register C.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Hold</td>
<td></td>
</tr>
<tr>
<td>D[15:0]</td>
<td>Input</td>
<td>16-Bits data of Input D</td>
<td>16'b0</td>
</tr>
<tr>
<td>D_HOLD</td>
<td>Input</td>
<td>Register D Hold Input. Control data flow input Register D.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Hold</td>
<td></td>
</tr>
<tr>
<td>IRSTTOP</td>
<td>Input</td>
<td>Reset Input to Registers A and C. Also resets upper 8 x 8 Multiplier Output Register (8 x 8 MAC Pipeline Register). It is an active HIGH reset whose deassertion should be synchronized with the clock.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not reset</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Reset</td>
<td></td>
</tr>
<tr>
<td>ORSTTOP</td>
<td>Input</td>
<td>Reset Input to top Accumulator Register (for Adder/Subtractor, Accumulator, and MAC functions). It is an active HIGH reset whose deassertion should be synchronized with the clock.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not reset</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Reset</td>
<td></td>
</tr>
<tr>
<td>OLOADTOP</td>
<td>Input</td>
<td>Load Control Input to top Accumulator Register (initialize on MAC function)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Load data from Register/Input C</td>
<td></td>
</tr>
<tr>
<td>ADDSUBTOP</td>
<td>Input</td>
<td>Add/Subtract Control Input to top Accumulator</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Add</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Subtract</td>
<td></td>
</tr>
<tr>
<td>OHOLDTOP</td>
<td>Input</td>
<td>Top Accumulator Output Register Hold Input. Control data flow into the register.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Hold</td>
<td></td>
</tr>
<tr>
<td>OUTPUT[31:16]</td>
<td>Output</td>
<td>Upper 16 bits of Output</td>
<td>—</td>
</tr>
</tbody>
</table>
## Port Name | Direction | Functional Description | Default Values
--- | --- | --- | ---
IRSTB | Input | Reset Input to Registers A and C. Also resets upper 8 x 8 Multiplier Output Register (8 x 8 MAC Pipeline Register) and the 16 x 16 Multiplier Output Register (16 x 16 MAC Pipeline Register). It is an active HIGH reset whose deassertion should be synchronized with the clock. 0: Not reset 1: Reset | 0
ORSTBOT | Input | Reset Input to top Accumulator Register (for Adder/Subtractor, Accumulator, and MAC functions). It is an active HIGH reset whose deassertion should be synchronized with the clock. 0: Not reset 1: Reset | 0
OLOADBOT | Input | Load Control Input to bottom Accumulator Register (initialize on MAC function) 0: Not load 1: Load data from Register/Input D | 0
ADDSUBB | Input | Add/Subtract Control Input to bottom Accumulator 0: Add 1: Subtract | 0
OHALD | Input | Bottom Accumulator Output Register Hold Input. Control data flow into the register. 0: Load 1: Hold | 0
OUTPUT[15:0] | Output | Lower 16 bits of Output | 0
CI | Input | Cascaded Add/Sub Carry Input from previous DSP block | 0
CO | Output | Cascaded Add/Sub Carry Output to next DSP block | 0
ACCUMCI | Input | Cascaded Accumulator Carry Input from previous DSP block | 0
ACCUMCO | Output | Cascaded Accumulator Carry Output to next DSP block |
SIGNEXTIN | Input | Sign Extension Input from previous DSP block | 0
SIGNEXTOUT | Output | Sign Extension Output to next DSP block |
2.4. **SB_MAC16 Parameters**

The parameter table below, Table 2.2, shows a list of parameters to configure the SB_MAC16 block. This table also maps the parameters to the configuration bits shows in the SB_MAC16 Functional Diagram in Figure 2.2. For more information about the parameters, kindly refer to the iCE Technology Library.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Configuration Bit(s)</th>
<th>Parameter Description and Allowed Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEG_TRIGGER</td>
<td>—</td>
<td>Input Clock Polarity: 0: Rising edge 1: Falling edge</td>
<td>0</td>
</tr>
<tr>
<td>C_REG</td>
<td>C0</td>
<td>Input C Register Control: 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>A_REG</td>
<td>C1</td>
<td>Input A Register Control: 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>B_REG</td>
<td>C2</td>
<td>Input B Register Control: 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>D_REG</td>
<td>C3</td>
<td>Input D Register Control: 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>TOP_8x8_MULT_REG</td>
<td>C4</td>
<td>Top 8 x 8 Multiplier Output Register Control (Pipeline Register for MAC): 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>BOT_8x8_MULT_REG</td>
<td>C5</td>
<td>Bottom 8 x 8 Multiplier Output Register Control (Pipeline Register for MAC): 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>PIPELINE_16X16_MULT_REG1</td>
<td>C6</td>
<td>16 x 16 Multiplier Pipeline Register Control: 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>PIPELINE_16X16_MULT_REG2</td>
<td>C7</td>
<td>16 x 16 Multiplier Pipeline Register Control (Pipeline Register for MAC): 0: Not registered 1: Registered</td>
<td>0</td>
</tr>
<tr>
<td>TOPOUTPUT_SELECT</td>
<td>C9, C8</td>
<td>Top Output Select: 00: Adder/Subtractor, not registered 01: Adder/Subtractor, registered 10: 8 x 8 Multiplier 11: 16 x 16 Multiplier</td>
<td>00</td>
</tr>
<tr>
<td>TOPADD_SUB_LOWERINPUT</td>
<td>C11, C10</td>
<td>Input X of upper Adder/Subtractor: 00: Input A 01: 8 x 8 Multiplier Output at Top 10: 16 x 16 Multiplier upper 16-bit Outputs 11: Sign extension from Z15 (lower Adder/Subtractor input)</td>
<td>00</td>
</tr>
<tr>
<td>TOPADD_SUB_UPPERINPUT</td>
<td>C12</td>
<td>Input W of upper Adder/Subtractor: 0: Output of Adder/Subtractor Output Register (Accumulation Function) 1: Input C</td>
<td>0</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Configuration Bit(s)</td>
<td>Parameter Description and Allowed Values</td>
<td>Default</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| TOPADDSUB_CARRYSELECT | C14, C13             | Carry Input Select, Top Adder/Subtractor:  
00: Constant 0  
01: Constant 1  
10: Cascade ACCUMOUT from lower Adder/Subtractor  
11: Cascade CO from lower Adder/Subtractor                                                                                                                                 | 00      |
| BOTOUTPUT_SELECT     | C16, C15             | Bottom Output Select:  
00: Adder/Subtractor, not registered  
01: Adder/Subtractor, registered  
10: 8 x 8 Multiplier  
11: 16 x 16 Multiplier                                                                                                                                                       | 00      |
| BOTADDSUB_LOWERINPUT | C18, C17             | Input Z of upper Adder/Subtractor:  
00: Input B  
01: 8 x 8 Multiplier Output at Bottom  
10: 16 x 16 Multiplier lower 16-bit Outputs  
11: Sign extension from SIGNEXTIN                                                                                                                                            | 00      |
| BOTADDSUB_UPPERINPUT | C19                  | Input Y of upper Adder/Subtractor:  
0: Output of Adder/Subtractor Output Register (Accumulation Function)  
1: Input D                                                                                                                                                                 | 0       |
| BOTADDSUB_CARRYSELECT | C21, C20            | Carry Input Select, Bottom Adder/Subtractor:  
00: Constant 0  
01: Constant 1  
10: Cascade ACCUMOUT from lower DSP block  
11: Cascade CO from lower DSP block                                                                                                                                           | 00      |
| MODE_8x8             | C22                  | Select 8 x 8 Multiplier Mode (Power Saving):  
0: Not Selected  
1: Selected                                                                                                                                                             | 0 --> 1 |
| A_SIGNED             | C23                  | Input A Sign:  
0: Input A is unsigned  
1: Input A is signed                                                                                                                                                      | 0       |
| B_SIGNED             | C24                  | Input B Sign:  
0: Input B is unsigned  
1: Input B is signed                                                                                                                                                      | 0       |
3. Implementing DSP Functions

There are two ways to implement DSP functions in the iCE40 Ultra and iCE40 UltraPlus devices:

- Inferencing DSP functions
  This method requires users to define the functional behavior of the DSP function they wish to implement, and let the software tools map it to the SB_MAC16 DSP block.

- Instantiating SB_MAC16 DSP Primitives
  This method involves instantiating the SB_MAC16 primitive in the user code. The ports discussed in the above sections need to be port-mapped for each function, or tied off to their default value.

Both of these methods are discussed in detail in the following sections.

3.1. Inferencing DSP Functions

This method involves defining the desired DSP function in behavioral HDL code. This does not require you to know the details of the DSP primitive, and the function is inferred automatically based on the code.

Here is an example of inferencing a 32-bit Accumulator with asynchronous data input and synchronous (registered) data out.

3.1.1. 32-bit Accumulator with Async Data In and Sync Data Out

Verilog

```verilog
module accum32_syncdataout (clk, accumdata_syncout, dataAB);
  input clk;
  input [31:0] dataAB;
  output [31:0] accumdata_syncout;
  reg [31:0] accumdata_syncout;
  always@(posedge clk)
    begin
      accumdata_syncout <= accumdata_syncout + dataAB;
    end
endmodule
```

```
VHDL

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY accum32_syncdataout IS PORT (clk : IN STD_LOGIC;
accumdata_syncout : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
dataAB : IN STD_LOGIC_VECTOR(31 DOWNTO 0)
);
END accum32_syncdataout;

ARCHITECTURE arch OF accum32_syncdataout IS
-- Declare intermediate signals for referenced outputs

SIGNAL accumdata_syncout_xhdl0 : STD_LOGIC_VECTOR(31 DOWNTO 0); BEGIN
-- Drive referenced outputs

accumdata_syncout <= accumdata_syncout_xhdl0;

PROCESS (clk) BEGIN
IF (clk'EVENT AND clk = '1') THEN
accumdata_syncout_xhdl0 <= accumdata_syncout_xhdl0 + dataAB; END IF;
END PROCESS;

END arch;
Another example is of an 8 x 8 multiplier, with both inputs and outputs registered.

### 3.1.2. 8 x 8 Multiplier, Unsigned with Sync Data In and Data Out

#### Verilog

```verilog
module mult8x8_inoutreg_unsigned (clk, prod, a_in, b_in);
  input [7:0] a_in;
  input [7:0] b_in;
  input clk;
  output [15:0] prod;
  reg [15:0] prod;

  reg [7:0] a_reg, b_reg;
  wire [15:0] mult_out;

  assign mult_out = a_reg * b_reg;

  always@(posedge clk)
  begin
    a_reg <= a_in;
    b_reg <= b_in;
    prod <= mult_out;
  end
endmodule
```

#### VHDL

```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY mult8x8_inoutreg_unsigned IS PORT ( 
  clk: IN STD_LOGIC;
  prod: OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
  a_in: IN STD_LOGIC_VECTOR(7 DOWNTO 0);
  b_in: IN STD_LOGIC_VECTOR(7 DOWNTO 0)
);
END mult8x8_inoutreg_unsigned;

ARCHITECTURE arch OF mult8x8_inoutreg_unsigned IS

  SIGNAL a_reg : STD_LOGIC_VECTOR(7 DOWNTO 0);
  SIGNAL b_reg : STD_LOGIC_VECTOR(7 DOWNTO 0);
  SIGNAL mult_out : STD_LOGIC_VECTOR(15 DOWNTO 0);

  mult_out <= ("00000000" & a_reg * b_reg);

  PROCESS (clk)
  BEGIN
    IF (clk'EVENT AND clk = '1') THEN
      a_reg <= a_in;
      b_reg <= b_in;
      prod <= mult_out;
    END IF;
  END PROCESS;

END arch;
```
3.2. Instantiation DSP Primitive – SB_MAC16

In order to implement various functions in the DSP block, users are required to instantiate the SB_MAC16 block in their top level HDL code. Different combinations of ports are connected to the user logic for various functions.

Table 3.1 provides a summary of port connections in instantiation based on functions that are required to be implemented. The column on the left provides various signals that are needed to be port mapped during HDL instantiation. The top row provides various functions that can be implemented. The cross referenced cells indicate whether the port connection is Signal or Default.

The term Signal means that this is one of the signals that user will have to port map to, while implementing the function. The Default implies that this port has to be connected to its default value during port mapping.

The default value of a port can be referenced from Table 2.1.

In certain cases, the DSP block can have two independent functions, for example two 8 x 8 multipliers, generating two 16-bit outputs. Such cases are referenced as Top Signals and Bottom Signals in Table 3.1. In such cases, one of the 8 x 8 multipliers can be implemented using Top Signals, and the other using Bottom Signals.

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Input/ Output</th>
<th>8 x 8 Multiplier</th>
<th>16 x 16 Multiplier</th>
<th>16 bit Accumulator</th>
<th>32 bit Accumulator</th>
<th>16 bit Adder/ Subtractor</th>
<th>32 bit Adder/ Subtractor</th>
<th>8 x 8 MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>Input</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>CE</td>
<td>Input</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>A[7:0]</td>
<td>Input</td>
<td>Bottom</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Bottom</td>
</tr>
<tr>
<td>A[15:8]</td>
<td>Input</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
</tr>
<tr>
<td>B[7:0]</td>
<td>Input</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
</tr>
<tr>
<td>B[15:8]</td>
<td>Input</td>
<td>Top</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Top</td>
</tr>
<tr>
<td>BHOHD</td>
<td>Input</td>
<td>Signal</td>
<td>Signal</td>
<td>Bottom - Signal</td>
<td>Signal</td>
<td>Bottom - Signal</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>C[15:0]</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default -&gt; Top</td>
<td>Default -&gt; Top</td>
<td>Top</td>
<td>Default -&gt; Top</td>
<td>Top</td>
</tr>
<tr>
<td>CHOHD</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default -&gt; Top</td>
<td>Default -&gt; Top</td>
<td>Top</td>
<td>Default -&gt; Top</td>
<td>Top</td>
</tr>
<tr>
<td>D[15:0]</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default -&gt; Bottom</td>
<td>Default -&gt; Signal</td>
<td>Bottom</td>
<td>Default -&gt; Signal</td>
<td>Bottom</td>
</tr>
<tr>
<td>DHOHD</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default -&gt; Bottom</td>
<td>Default -&gt; Signal</td>
<td>Bottom</td>
<td>Default -&gt; Signal</td>
<td>Bottom</td>
</tr>
<tr>
<td>ORSTTOP</td>
<td>Input</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
</tr>
<tr>
<td>OLOADTOP</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Signal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Signal</td>
</tr>
<tr>
<td>ADDSUBTOP</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>0</td>
<td>0</td>
<td>0 = Add</td>
<td>0 = Add</td>
<td>0</td>
</tr>
<tr>
<td>OHOLDTOP</td>
<td>Input</td>
<td>Top -&gt; default</td>
<td>Signal - default</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
</tr>
<tr>
<td>OUTPUT[31:16]</td>
<td>Output</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
</tr>
<tr>
<td>IRSTBOT</td>
<td>Input</td>
<td>Bottom -&gt; Signal</td>
<td>Signal</td>
<td>Bottom -&gt; Signal</td>
<td>Signal</td>
<td>Bottom -&gt; Signal</td>
<td>Signal</td>
<td>Bottom</td>
</tr>
<tr>
<td>ORSTBOT</td>
<td>Input</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
</tr>
<tr>
<td>OLOADBOT</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Signal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Signal</td>
</tr>
</tbody>
</table>
As an example, let us look at instantiating a 16-bit Accumulator with synchronous data out (registered outputs). The example below shows the port mapping and parameters that need to be set. Setting the ports and parameters is based on the tables discussed in the sections above above.

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Input/Output</th>
<th>8 x 8 Multiplier</th>
<th>16 x 16 Multiplier</th>
<th>16 bit Accumulator</th>
<th>32 bit Accumulator</th>
<th>16 bit Adder/Subtractor</th>
<th>32 bit Adder/Subtractor</th>
<th>8 x 8 MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDSUBBOT</td>
<td>0 = Add</td>
<td>0</td>
<td>0</td>
<td>0 = Add</td>
<td>0 = Add</td>
<td>0 = Add</td>
<td>0 = Add</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 = Sub</td>
<td>1 = Sub</td>
<td>1 = Sub</td>
<td>1 = Sub</td>
<td>1 = Sub</td>
<td>1 = Sub</td>
<td>1 = Sub</td>
<td></td>
</tr>
<tr>
<td>OHOLDBOT</td>
<td>Input</td>
<td>Bottom -&gt; default</td>
<td>Signal -&gt; default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
</tr>
<tr>
<td>OUTPUT[15:0]</td>
<td>Output</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Bottom</td>
</tr>
<tr>
<td>CI</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default -&gt; Top</td>
<td>Default -&gt; Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Default -&gt; Top</td>
<td></td>
</tr>
<tr>
<td>CO</td>
<td>Output</td>
<td>Default</td>
<td>Default -&gt; Top</td>
<td>Default -&gt; Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Default -&gt; Top</td>
<td></td>
</tr>
<tr>
<td>ACCUMCI</td>
<td>Input</td>
<td>Default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Default -&gt; Bottom</td>
<td>Signal</td>
<td>Default -&gt; Bottom</td>
<td></td>
</tr>
<tr>
<td>ACCUMCO</td>
<td>Output</td>
<td>Default</td>
<td>Top</td>
<td>Signal</td>
<td>Default -&gt; Top</td>
<td>Signal</td>
<td>Default -&gt; Top</td>
<td></td>
</tr>
<tr>
<td>SIGNEXTIN</td>
<td>Input</td>
<td>Default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Default -&gt; Top</td>
<td>Signal</td>
<td>Default -&gt; Bottom</td>
<td></td>
</tr>
<tr>
<td>SIGNEXTOUT</td>
<td>Output</td>
<td>Default</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Default -&gt; Top</td>
<td></td>
</tr>
</tbody>
</table>
3.2.1. Dual 16 bit Accumulator with Sync Data Out

Verilog

SB_MAC16 i_sbmac16
  (  // port interfaces
    .A(A),
    .B(B),
    .C(C),
    .D(D),
    .O(O),
    .CLK(CLK),
    .CE(CE),
    .IRSTTOP(IRSTTOP),
    .IRSTBOT(IRSTBOT),
    .ORSTTOP(ORSTTOP),
    .ORSTBOT(ORSTBOT),
    .AHOLD(AHOLD),
    .BHOLD(BHOLD),
    .CHOLD(CHOLD),
    .DHOLD(DHOLD),
    .OHOLDTOP(OHOLDTOP),
    .OHOLDBOT(OHOLDBOT),
    .OLOADTOP(OLOADTOP),
    .OLOADBOT(OLOADBOT),
    .ADDSUBTOP(ADDSUBTOP),
    .ADDSUBBOT(ADDSUBBOT),
    .CO(CO),
    .CI(CI),
    .ACCUMCI(),
    .ACCUMCO(),
    .SIGNEXTIN(),
    .SIGNEXTOUT()
  );

  defparam i_sbmac16.NEG_TRIGGER = 1'b0;
  defparam i_sbmac16.C_REG = 1'b0;
  defparam i_sbmac16.A_REG = 1'b0;
  defparam i_sbmac16.B_REG = 1'b0;
  defparam i_sbmac16.D_REG = 1'b0;
  defparam i_sbmac16.TOP_8x8_MULT_REG = 1'b0;
  defparam i_sbmac16.BOT_8x8_MULT_REG = 1'b0;
  defparam i_sbmac16.PIPELINE_16x16_MULT_REG1 = 1'b0;
  defparam i_sbmac16.PIPELINE_16x16_MULT_REG2 = 1'b0;
  defparam i_sbmac16.TOPOUTPUT_SELECT = 2'b01; // accum register output at O[31:16]
  defparam i_sbmac16.TOPADDSSUB_LOWERINPUT = 2'b00;
  defparam i_sbmac16.TOPADDSSUB_UPPERINPUT = 1'b0;
  defparam i_sbmac16.TOPADDSSUB_CARRYSELECT = 2'b00;
  defparam i_sbmac16.TOPOUTPUT_SELECT = 2'b01; // accum register output at O[31:16]

  //defparam i_sbmac16.BOTOUTPUT_SELECT = 2'b01 ;// accum register output at O[15:0].
  //defparam i_sbmac16.TOPOUTPUT_SELECT = 2'b01 ;// accum register output at O[31:16]

Endmodule
i_sbmac16: SBMAC16
GENERIC MAP ( 
    NEG_TRIGGER => 1'b0,
    C_REG => 1'b0,
    A_REG => 1'b0,
    B_REG => 1'b0,
    D_REG => 1'b0,
    TOP_8x8_MULT_REG => 1'b0,
    BOT_8x8_MULT_REG => 1'b0,
    PIPELINE_16x16_MULT_REG1 => 1'b0,
    PIPELINE_16x16_MULT_REG2 => 1'b0,
    TOPOUTPUT_SELECT => 2'b01, -- accum register output at O[31:16]
    TOPADDSUB_LOWERINPUT => 2'b00,
    TOPADDSUB_UPPERINPUT => 1'b0,
    TOPADDSUB_CARRYSELECT => 2'b00,
    BOTOUTPUT_SELECT => 2'b01, -- accum register output at O[15:0]
    BOTADDSUB_LOWERINPUT => 2'b00,
    BOTADDSUB_UPPERINPUT => 1'b0,
    BOTADDSUB_CARRYSELECT => 2'b00,
    MODE_8x8 => 1'b0,
    A_SIGNED => 1'b0,
    B_SIGNED => 1'b0
)
PORT MAP ( -- port interfaces
    A => A,
    B => B,
    C => C,
    D => D,
    O => O,
    CLK => CLK,
    CE => CE,
    IRSTTOP => IRSTTOP,
    IRSTBOT => IRSTBOT,
    ORSTTOP => ORSTTOP,
    ORSTBOT => ORSTBOT,
    AHOLOD => AHOLOD,
    BHOLD => BHOLD,
    CHOLD => CHOLD,
    DHOLD => DHOLD,
    OHOLODTOP => OHOLODTOP,
    OHOLODBOT => OHOLODBOT,
    OLOADTOP => OLOADTOP,
    OLOADBOT => OLOADBOT,
    ADDSUBTOP => ADDSUBTOP,
    ADDSUBBOT => ADDSUBBOT,
    CO => CO,
    CI => CI,
    ACCUMCI => Open,
    ACCUMCO => Open,
    SIGNEXTIN => Open,
    SIGNEXTOUT => Open
);
Another common function used for DSP applications is a multiplier. The two examples below show the instantiation of 16-bit multipliers both signed and unsigned.

### 3.2.2. Multiplier 16 x 16 Signed

Verilog

```verilog
SB_MAC16 i_sbmac16
    (    // port interfaces
        .A(A),
        .B(B),
        .C(C),
        .D(D),
        .O(O),
        .CLK(CLK),
        .CE(CE),
        .IRSTTOP(IRSTTOP),
        .IRSTBOT(IRSTBOT),
        .ORSTTOP(ORSTTOP),
        .ORSTBOT(ORSTBOT),
        .AHOLD(AHOLD),
        .BHOLD(BHOLD),
        .CHOLD(CHOLD),
        .DHOLD(DHOLD),
        .OHOLDTOP(OHOLDTOP),
        .OHOLDBOT(OHOLDBOT),
        .OLOADTOP(OLOADTOP),
        .OLOADBOT(OLOADBOT),
        .ADDSUBTOP(ADDSUBTOP),
        .ADDSUBBOT(ADDSUBBOT),
        .CO(CO),
        .CI(CI),
        .ACCUMCI(),
        .ACCUMCO(),
        .SIGNEXTIN(),
        .SIGNEXTOUT()
    );

    defparam i_sbmac16.TOPOUTPUT_SELECT = 2'b11; //Mult16x16 data output
    defparam i_sbmac16.BOTOUTPUT_SELECT = 2'b11;
    defparam i_sbmac16.PIPELINE_16x16_MULT_REG2 = 1'b1; //Mult16x16 output registered
    defparam i_sbmac16.A_SIGNED = 1'b1; //Signed Inputs
    defparam i_sbmac16.B_SIGNED = 1'b1;
endmodule
```
VHDL

i_sbmac16: SB_MAC16
GENERIC MAP (  
  TOPOUTPUT_SELECT => 2'b11,  
  BOTOUTPUT_SELECT => 2'b11,  
  PIPELINE_16x16_MULT_REG2 => 1'b1,  
  A_SIGNED => 1'b1,  
  B_SIGNED => 1'b1  
)

PORT MAP (  
  A => A,  
  B => B,  
  C => C,  
  D => D,  
  O => O,  
  CLK => CLK,  
  CE => CE,  
  IRSTTOP => IRSTTOP,  
  IRSTBOT => IRSTBOT,  
  ORSTTOP => ORSTTOP,  
  ORSTBOT => ORSTBOT,  
  AHOLD => AHOLD,  
  BHOLD => BHOLD,  
  CHOLD => CHOLD,  
  DHold => DHold,  
  OHOLDTOP => OHOLDTOP,  
  OHOLDBOT => OHOLDBOT,  
  OLOADTOP => OLOADTOP,  
  OLOADBOT => OLOADBOT,  
  ADDSUBTOP => ADDSUBTOP,  
  ADDSUBBOT => ADDSUBBOT,  
  CO => CO,  
  CI => CI,  
  ACCUMCI => Open,  
  ACCUMCO => Open,  
  SIGNEXTIN => Open,  
  SIGNEXTOUT => Open  
);
### 3.2.3. Multiplier 16 x 16 Unsigned

**Verilog**

```verilog
SB_MAC16 i_sbmac16
    (    // port interfaces
        .A(A),
        .B(B),
        .C(C),
        .D(D),
        .O(O),
        .CLK(CLK),
        .CE(CE),
        .IRSTTOP(IRSTTOP),
        .IRSTBOT(IRSTBOT),
        .ORSTTOP(ORSTTOP),
        .ORSTBOT(ORSTBOT),
        .AHOLD(AHOLD),
        .BHOLD(BHOLD),
        .CHOLD(CHOLD),
        .DHOLD(DHOLD),
        .OHOLDTOP(OHOLDTOP),
        .OHOLDBOT(OHOLDBOT),
        .OLOADTOP(OLOADTOP),
        .OLOADBOT(OLOADBOT),
        .ADDSUBTOP(ADDSUBTOP),
        .ADDSUBBOT(ADDSUBBOT),
        .CO(CO),
        .CI(CI),
        .ACCUMCI(),
        .ACCUMCO(),
        .SIGNEXTIN(),
        .SIGNEXTOUT()
    );

defparam i_sbmac16.TOPOUTPUT_SELECT = 2'b11;
defparam i_sbmac16.BOTOUTPUT_SELECT = 2'b11;
defparam i_sbmac16.PIPELINE_16x16_MULT_REG2 = 1'b1;
defparam i_sbmac16.A_SIGNED = 1'b0;
defparam i_sbmac16.B_SIGNED = 1'b0;

endmodule
```
i_sbmac16: SB_MAC16

```vhdl
GENERIC MAP (
  TOPOUTPUT_SELECT => 2'b11,
  BOTOUTPUT_SELECT => 2'b11,
  PIPELINE_16x16_MULTI_REG2 => 1'b1,
  A_SIGNED => 1'b0,
  B_SIGNED => 1'b0)

PORT MAP (
  A => A,
  B => B,
  C => C,
  D => D,
  O => O,
  CLK => CLK,
  CE => CE,
  IRSTTOP => IRSTTOP,
  IRSTBOT => IRSTBOT,
  ORSTTOP => ORSTTOP,
  ORSTBOT => ORSTBOT,
  AHOlD => AHOlD,
  BHOlD => BHOlD,
  CHOlD => CHOlD,
  DHOlD => DHOlD,
  OHOlDTOP => OHOlDTOP,
  OHOlDBOT => OHOlDBOT,
  OLOADTOP => OLOADTOP,
  OLOADBOT => OLOADBOT,
  ADDSUBTOP => ADDSUBTOP,
  ADDSUBBOT => ADDSUBBOT,
  CO => CO,
  CI => CI,
  ACCUMCI => Open,
  ACCUMCO => Open,
  SIGNEXTIN => Open,
  SIGNEXTOUT => Open)
```

Technical Support Assistance

Submit a technical support case via www.latticesemi.com/techsupport.
## Revision History

### Revision 1.2, October 2020

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>• Changed document number from TN1295 to FPGA-TN-02007.</td>
</tr>
<tr>
<td></td>
<td>• Updated document template.</td>
</tr>
<tr>
<td>Disclaimers</td>
<td>Added this section.</td>
</tr>
<tr>
<td>Introduction</td>
<td>Updated list of functional units.</td>
</tr>
<tr>
<td>DSP Primitive – SB_MAC16</td>
<td>• Added reference to iCE Technology Library in the SB_MAC16 Parameters section.</td>
</tr>
<tr>
<td></td>
<td>• Updated IRSTSTOP, ORSTSTOP, IRSTBOT, ORSTBOT, and ACCUMCO descriptions in Table 2.1.</td>
</tr>
<tr>
<td></td>
<td>• Updated BOTADDSUB_LOWERINPUT description in Table 2.2.</td>
</tr>
<tr>
<td></td>
<td>• Updated Figure 2.2.</td>
</tr>
<tr>
<td>Implementing DSP Functions</td>
<td>• Updated section heading to Implementing DSP Functions.</td>
</tr>
<tr>
<td></td>
<td>• Updated sub-section heading to Dual 16 bit Accumulator with Sync Data Out.</td>
</tr>
<tr>
<td></td>
<td>• General update to section introduction.</td>
</tr>
<tr>
<td></td>
<td>• Updated Inferencing DSP Functions section.</td>
</tr>
<tr>
<td></td>
<td>• Updated Table 3.1.</td>
</tr>
<tr>
<td></td>
<td>• Fixed code errors.</td>
</tr>
<tr>
<td></td>
<td>• Editorial changes to correct grammar and improve readability.</td>
</tr>
<tr>
<td></td>
<td>• Minor adjustments in style/formatting.</td>
</tr>
</tbody>
</table>

### Revision 1.1, June 2016

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Updated Introduction section. Added iCE40 UltraPlus and removed MX series to introductory paragraph.</td>
</tr>
<tr>
<td>DSP Primitive – SB_MAC16</td>
<td>Updated DSP Primitive – SB_MAC16 section. Added iCE40 UltraPlus and removed MS series to introductory paragraph.</td>
</tr>
<tr>
<td>Implementing DSP Function in iCE40 Ultra and iCE40 UltraPlus Devices</td>
<td>Updated Implementing DSP Function in iCE40 Ultra and iCE40 UltraPlus Devices section.</td>
</tr>
<tr>
<td></td>
<td>• Revised section heading to include iCE40 UltraPlus.</td>
</tr>
<tr>
<td></td>
<td>• Added iCE40 UltraPlus to introductory sentence.</td>
</tr>
<tr>
<td>Technical Support Assistance</td>
<td>Updated Technical Support Assistance section.</td>
</tr>
</tbody>
</table>

### Revision 1.0, June 2014

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>