**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 to 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 combination of following functional units by selecting appropriate parameter values.

- Single 16x16 Multiplier (generating 32-bit product output).
- Two independent 8x8 Multiplier (generating two independent 16-bit product output).
- Single 32-bit Accumulator.
- Two independent 16-bit Accumulator.
- Single 32-bit Adder/Subtractor.
- Two independent 16-bit Adder/Subtractor.
- Single 32-bit Multiply-Add/Multiply-Sub.
- Two independent 16-bit Multiply-Add/Multiply-Sub.

**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 up various instance parameters.

**SB_MAC16 Primitive**

Figure 1 provides an overview of the SB_MAC16 primitive with various inputs and outputs.

**Figure 1. SB_MAC16 DSP Primitive Interface Diagram**
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 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 details in the sections that follow.

**SB_MAC16 Functional Diagram**

Figure 2 shows the functional model of the SB_MAC16 primitive. The variety of functions can be implemented in this block by interfacing with portions required of the functional model that are needed for these functions.

*Figure 2. SB_MAC16 DSP Functional Model*
SB_MAC16 Interface Ports

The following table, Table 1, provides a list of interface ports available in SB_MAC16 and their functional description. Something of importance to note in this table is the “Default Values” of these ports; as it will be 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 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>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[15:8]</td>
<td>Input</td>
<td>Register B Hold Input. Control data flow input Register B</td>
<td>0: Load 1: Hold</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>CHOLD</td>
<td>Input</td>
<td>Register C Hold Input. Control data flow input Register C</td>
<td>0: Load 1: Hold</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[15:0]</td>
<td>Input</td>
<td>Register D Hold Input. Control data flow input Register D</td>
<td>0: Load 1: Hold</td>
</tr>
<tr>
<td>IRSTTOP</td>
<td>Input</td>
<td>Reset Input to Registers A and C. Also resets upper 8x8 Multiplier Output Register (8x8 MAC Pipeline Register)</td>
<td>0: Not reset 1: Reset</td>
</tr>
<tr>
<td>ORSTTOP</td>
<td>Input</td>
<td>Reset Input to top Accumulator Register (for Adder/Subtractor, Accumulator, and MAC functions)</td>
<td>0: Not reset 1: Reset</td>
</tr>
<tr>
<td>OLOADTOP</td>
<td>Input</td>
<td>Load Control Input to top Accumulator Register (initialize on MAC function)</td>
<td>0: Not load 1: Load data from Register/Input C</td>
</tr>
<tr>
<td>ADDSUBTOP</td>
<td>Input</td>
<td>Add/Subtract Control Input to top Accumulator</td>
<td>0: Add 1: Subtract</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: Load 1: Hold</td>
</tr>
<tr>
<td>Port Name</td>
<td>Direction</td>
<td>Functional Description</td>
<td>Default Values</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>OUTPUT[31:16]</td>
<td>Output</td>
<td>Upper 16 bits of Output</td>
<td></td>
</tr>
<tr>
<td>IRSTBOT</td>
<td>Input</td>
<td>Reset Input to Registers A and C. Also resets upper 8x8 Multiplier Output Register (8x8 MAC Pipeline Register) and the 16x16 Multiplier Output Register (16x16 MAC Pipeline Register)</td>
<td>0</td>
</tr>
<tr>
<td>ORSTBOT</td>
<td>Input</td>
<td>Reset Input to top Accumulator Register (for Adder/Subtractor, Accumulator, and MAC functions)</td>
<td>0</td>
</tr>
<tr>
<td>OLOADBOT</td>
<td>Input</td>
<td>Load Control Input to bottom Accumulator Register (initialize on MAC function)</td>
<td>0</td>
</tr>
<tr>
<td>ADDSUBBOT</td>
<td>Input</td>
<td>Add/Subtract Control Input to bottom Accumulator</td>
<td>0</td>
</tr>
<tr>
<td>OHOLDBOT</td>
<td>Input</td>
<td>Bottom Accumulator Output Register Hold Input. Control data flow into the register.</td>
<td>0</td>
</tr>
<tr>
<td>OUTPUT[15:0]</td>
<td>Output</td>
<td>Lower 16 bits of Output</td>
<td>0</td>
</tr>
<tr>
<td>CI</td>
<td>Input</td>
<td>Cascaded Add/Sub Carry Input from previous DSP block</td>
<td>0</td>
</tr>
<tr>
<td>CO</td>
<td>Output</td>
<td>Cascaded Add/Sub Carry Output to next DSP block</td>
<td>0</td>
</tr>
<tr>
<td>ACCUMCI</td>
<td>Input</td>
<td>Cascaded Accumulator Carry Input from previous DSP block</td>
<td>0</td>
</tr>
<tr>
<td>ACCUMCO</td>
<td>Output</td>
<td>Cascaded Accumulator Carry Output to previous DSP block</td>
<td>0</td>
</tr>
<tr>
<td>SIGNEXTIN</td>
<td>Input</td>
<td>Sign Extension Input from previous DSP block</td>
<td>0</td>
</tr>
<tr>
<td>SIGNEXTOUT</td>
<td>Output</td>
<td>Sign Extension Output to next DSP block</td>
<td>0</td>
</tr>
</tbody>
</table>
SB_MAC16 Parameters

The parameter table below, Table 2, shows a list of parameters to configure the SV_MAC16 block. This table also maps the parameter to the configuration bits shown in the SB_MAC16 Functional Diagram in Figure 2.

Table 2. SB_MAC16 Parameter Description

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Configuration Bit(s)</th>
<th>Parameter Description &amp; Allowed Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEG_TRIGGER</td>
<td>-</td>
<td>Input Clock Polarity:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = rising edge</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = falling edge</td>
<td></td>
</tr>
<tr>
<td>C_REG</td>
<td>C0</td>
<td>Input C Register Control:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>A_REG</td>
<td>C1</td>
<td>Input A Register Control:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>B_REG</td>
<td>C2</td>
<td>Input B Register Control:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>D_REG</td>
<td>C3</td>
<td>Input D Register Control:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>TOP_8x8_MULT_REG</td>
<td>C4</td>
<td>Top 8x8 Multiplier Output Register Control (Pipeline Register for MAC):</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>BOT_8x8_MULT_REG</td>
<td>C5</td>
<td>Bottom 8x8 Multiplier Output Register Control (Pipeline Register for MAC):</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>PIPELINE_16X16_MULT_REG1</td>
<td>C6</td>
<td>16x16 Multiplier Pipeline Register Control:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>PIPELINE_16x16_MULT_REG2</td>
<td>C7</td>
<td>16x16 Multiplier Output Register Control (Pipeline Register for MAC):</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Registered</td>
<td></td>
</tr>
<tr>
<td>TOPOUTPUT_SELECT</td>
<td>C9, C8</td>
<td>Top Output Select:</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00: Adder/Subtractor, not registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>01: Adder/Subtractor, registered</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10: 8x8 Multiplier</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11: 16x16 Multiplier</td>
<td></td>
</tr>
<tr>
<td>TOPADDSUB_LOWERINPUT</td>
<td>C11, C10</td>
<td>Input X of upper Adder/Subtractor:</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00: Input A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>01: 8x8 Multiplier Output at Top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10: 16x16 Multiplier upper 16-bit outputs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11: Sign extension from Z15 (lower Adder/Subtractor input)</td>
<td></td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Configuration Bit(s)</td>
<td>Parameter Description &amp; Allowed Values</td>
<td>Default</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>TOPADDSUB_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>TOPADDSUB_CARRYSELECT</td>
<td>C14, C13</td>
<td>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</td>
<td>00</td>
</tr>
<tr>
<td>BOTOUTPUT_SELECT</td>
<td>C16, C15</td>
<td>Bottom Output Select: 00: Adder/Subtractor, not registered 01: Adder/Subtractor, registered 10: 8x8 Multiplier 11: 16x16 Multiplier</td>
<td>00</td>
</tr>
<tr>
<td>BOTADDSUB_LOWERINPUT</td>
<td>C18, C17</td>
<td>Input Z of upper Adder/Subtractor: 00: Input B 01: 8x8 Multiplier Output at Top 10: 16x16 Multiplier upper 16-bit outputs 11: Sign extension from SIGNEXTIN</td>
<td>00</td>
</tr>
<tr>
<td>BOTADDSUB_UPPERINPUT</td>
<td>C19</td>
<td>Input Y of upper Adder/Subtractor: 00: Output of Adder/Subtractor Output Register (Accumulation Function) 1: Input D</td>
<td>0</td>
</tr>
<tr>
<td>BOTADDSUB_CARRYSELECT</td>
<td>C21, C20</td>
<td>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</td>
<td>00</td>
</tr>
<tr>
<td>MODE_8x8</td>
<td>C22</td>
<td>Select 8x8 Multiplier Mode (Power Saving): 0: Not Selected 1: Selected</td>
<td>0 --&gt; 1</td>
</tr>
<tr>
<td>A_SIGNED</td>
<td>C23</td>
<td>Input A Sign: 0: Input A is un-signed 1: Input A is signed</td>
<td>0</td>
</tr>
<tr>
<td>B_SIGNED</td>
<td>C24</td>
<td>Input B Sign: 0: Input B is un-signed 1: Input B is signed</td>
<td>0</td>
</tr>
</tbody>
</table>
Implementing DSP Function in iCE40 Ultra and iCE40 UltraPlus Devices

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

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

- **Instantiating DSP Primitive SB_MAC16**
  This method involves the 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. The detailed discussion of the methodology is discussed below.

Both these methods are discussed in details in following sections.

**Inferencing DSP Functions**

This method involves defining desired DSP function as a behavioral model in the standard HDL. The code does not require you to know the details of the DSP primitive, and 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.

**32-bit Accumulator with Async Data In & 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**

```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 8x8 multiplier, with both inputs and outputs registered.

8x8 Multiplier, Unsigned with Sync Data In & Data Out

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

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);
BEGIN

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;

Instantiation DSP Primitive – SB_MAC16

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

Table 3 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 references 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 1.

In certain cases, the DSP block can have two independent functions, for example two 8x8 multipliers, generating two 16-bit outputs. Such cases are referenced as Top and Bottom Signals in the table below. In such cases, one of the 8x8 multipliers can be implemented using Top Signals, and other using Bottom Signals.
<table>
<thead>
<tr>
<th>Port Name</th>
<th>Input/Output</th>
<th>8x8 Multiplier</th>
<th>16x16 Multiplier</th>
<th>16x16 Accumulate</th>
<th>32x32 Accumulate</th>
<th>16x16 Adder/Subtractor</th>
<th>32x32 Adder/Subtractor</th>
<th>8x8 MAC</th>
<th>16x16 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>
<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>
<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>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>AHOLED</td>
<td>Input</td>
<td>Signal</td>
<td>Signal</td>
<td>Top -&gt; Signal</td>
<td>Signal</td>
<td>Top -&gt; Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</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>Signal</td>
<td>Signal</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>Top</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>BHOLED</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>
<td>Signal</td>
</tr>
<tr>
<td>C[15:0]</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default</td>
<td>Top</td>
<td>Default -&gt; Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>CHOLD</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default</td>
<td>Top</td>
<td>Default -&gt; Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>D[15:0]</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default</td>
<td>Bottom</td>
<td>Default -&gt; Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>DHOLED</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default</td>
<td>Bottom</td>
<td>Default -&gt; Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>IRSTTOP</td>
<td>Input</td>
<td>Top -&gt; Signal</td>
<td>Signal</td>
<td>Top -&gt; Signal</td>
<td>Signal</td>
<td>Top -&gt; Signal</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</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>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>OLOADTOP</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Signal</td>
<td>Signal</td>
<td>0</td>
<td>Signal</td>
<td>Signal</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>
<td>0</td>
</tr>
<tr>
<td>OLOADBOT</td>
<td>Input</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
</tr>
<tr>
<td>ADDSUBBOT</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Signal - default</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>OHOLDBOT</td>
<td>Input</td>
<td>Bottom - default</td>
<td>Signal</td>
<td>Bottom - default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Signal</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>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>OUTPUT[15:0]</td>
<td>Output</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>ADDSUBBOT</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Signal</td>
<td>0</td>
<td>0 = Add</td>
<td>0 = Add</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OLOADBOT</td>
<td>Input</td>
<td>Bottom - default</td>
<td>Signal</td>
<td>Bottom - default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>OUTPUT[15:0]</td>
<td>Output</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>CI</td>
<td>Input</td>
<td>Default</td>
<td>Default</td>
<td>Default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Default</td>
<td>Default</td>
<td>Default</td>
</tr>
<tr>
<td>CO</td>
<td>Output</td>
<td>Default</td>
<td>Default</td>
<td>Default</td>
<td>Top</td>
<td>Default -&gt; Signal</td>
<td>Top</td>
<td>Signal</td>
<td>Signal</td>
</tr>
<tr>
<td>ACCUMCI</td>
<td>Input</td>
<td>Default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Default</td>
<td>Bottom</td>
<td>Default -&gt; Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
</tr>
<tr>
<td>ACCUMCO</td>
<td>Output</td>
<td>Default</td>
<td>Top</td>
<td>Signal</td>
<td>Default</td>
<td>Bottom</td>
<td>Default -&gt; Top</td>
<td>Bottom</td>
<td>Bottom</td>
</tr>
<tr>
<td>SIGNEXTIN</td>
<td>Input</td>
<td>Default</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Signal</td>
<td>Bottom</td>
<td>Bottom</td>
<td>Bottom</td>
</tr>
<tr>
<td>SIGNEXTOUT</td>
<td>Output</td>
<td>Default</td>
<td>Top</td>
<td>Signal</td>
<td>Top</td>
<td>Default -&gt; Top</td>
<td>Default -&gt; Signal</td>
<td>Signal</td>
<td>Signal</td>
</tr>
</tbody>
</table>
As an example, let us look at instantiation sections for 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 instantiations will be based on the tables discussed in the sections above.

Accumulator 16x2 Sync Data Out
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.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.TOPADDSUB_LOWERINPUT = 2'b00;
  defparam i_sbmac16.TOPADDSUB_UPPERINPUT = 1'b0;
  defparam i_sbmac16.TOPADDSUB_CARRYSELECT = 2'b00;
```

---

LATTICE
SEMICONDUCTOR

DSP Function Usage Guide for iCE40 Devices

11
defparam i_sbmac16.BOTOUTPUT_SELECT = 2'b01;  // accum regisiter output at O[15:0]
defparam i_sbmac16.BOTADDSUB_LOWERINPUT = 2'b00;
defparam i_sbmac16.BOTADDSUB_UPPERINPUT = 1'b0;
defparam i_sbmac16.BOTADDSUB_CARRYSELECT = 2'b00;
defparam i_sbmac16.MODE_8x8 = 1'b0;

defparam i_sbmac16.A_SIGNED = 1'b0;
defparam i_sbmac16.B_SIGNED = 1'b0;

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

endmodule

VHDL

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 (  
  A => A, 
  B => B, 
  C => C, 
  D => D, 
  O => O, 
  CLK => CLK, 
  CE => CE, 
  IRSTTOP => IRSTTOP, 
)
Another common module used for DSP applications is a multiplier. The two examples below show the instantiation of 16-bit multipliers both signed and unsigned.

**Multiplier 16x16 Signed**

*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,
    AHOlland => AHOlland,
    BHOlland => BHOlland,
    CHOlland => CHOlland,
    DHOlland => DHOlland,
    OHOllandTOP => OHOllandTOP,
    OHOllandBOT => OHOllandBOT,
    OLOADTOP => OLOADTOP,
    OLOADBOT => OLOADBOT,
    ADDSOUTOP => ADDSOUTOP,
    ADDSOUTOBOT => ADDSOUTOBOT,
    CO => CO,
    CI => CI,
    ACCUMCI => Open,
    ACCUMCO => Open,
    SIGNEXTIN => Open,
    SIGNEXTOUT => Open
);
Multiplier 16x16 Unsigned
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

VHDL

i_sbmac16: SB_MAC16
GENERIC MAP (  
  TOPOUTPUT_SELECT => 2'b11,
  BOTOUTPUT_SELECT => 2'b11,
  PIPELINE_16x16_MULT_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,  
B HOLD => B HOLD,  
C HOLD => C HOLD,  
D HOLD => D HOLD,  
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  

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2016</td>
<td>1.1</td>
<td>Updated Introduction section. Added iCE40 UltraPlus and removed &quot;MX series&quot; to introductory paragraph.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated DSP Primitive – SB_MAC16 section. Added iCE40 UltraPlus and removed &quot;MS series&quot; to introductory paragraph.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Implementing DSP Function in iCE40 Ultra and iCE40 UltraPlus Devices section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Revised section heading to include iCE40 UltraPlus.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Added iCE40 UltraPlus to introductory sentence.</td>
</tr>
<tr>
<td>June 2014</td>
<td>01.0</td>
<td>Updated Technical Support Assistance section.</td>
</tr>
</tbody>
</table>