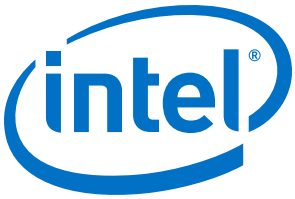


Intel®Arria® 10 Native Floating-Point DSP Intel® FPGA IP User Guide

UG-20028 | 2017.11.06
Latest document on the web: PDF | HTML

Contents

1 Intel® Arria® 10 Native Floating-Point DSP Intel® FPGA IP User Guide

1.1 Parameterizing the Intel Arria 10 Native Floating-Point DSP Intel FPGA IP

Select different parameters to create an IP core suitable for your design.

1. In Intel Quartus® Prime Pro Edition,create a new project that targets a Intel Arria 10 device.
2. In IP Catalog, click **Library ➤ DSP ➤ Primitive DSP ➤ Intel Arria 10 Native Floating Point DSP**.
   The Intel Arria 10 Native Floating-Point DSP IP Core IP parameter editor opens.
3. In the **New IP Variation** dialog box, enter an **Entity Name** and click **OK**.
4. Under **Parameters**, select the **DSP Template** and the **View** you want for your IP core
5. In the **DSP Block View**, switch the clock or reset of each valid register.
6. For **Multiply Add** or **Vector Mode 1**, click the **Chain In** multiplexer in the GUI to select input from `chainin` port or Ax port.
7. Click the **Adder** symbol in the GUI to select addition or subtraction.
8. Click the **Chain Out** multiplexer in the GUI to enable `chainout` port.
9. Click **Generate HDL**.
10. Click **Finish**.

1.2 Intel Arria 10 Native Floating-Point DSP Intel FPGA IP Parameters

Table 1. Parameters

Parameter	Value	Default Value	Description
DSP Template	**Multiply** **Add** **Multiply Add** **Multiply Accumulate** **Vector Mode 1** **Vector Mode 2**	**Multiply**	Select the desired operational mode for the DSP block. The selected operation is reflected in the **DSP Block View**.
View	**Register Enables** **Register Clears**	**Register Enables**	Options to select clocking scheme or reset scheme for registers view. The selected operation is reflected in the **DSP Block View**.

continued...

Parameter	Value	Default Value	Description
			Select **Register Enables** for **DSP Block View** to show registers clocking scheme. You can change the clocks for each of the registers in this view. Select **Register Clears** for **DSP Block View** to show registers reset scheme. Turn on **Use Single Clear** to change the registers reset scheme.
Use Single Clear	On or off	Off	Turn on this parameter if you want a single reset to reset all the registers in the DSP block. Turn off this parameter to use different reset ports to reset the registers. Turn on for clear 0 on output register; turn off for clear 1 on output register. **Clear 0** for input registers uses `aclr[0]` signal. **Clear 1** for output and pipeline registers uses `aclr[1]` signal. All input registers use `aclr[0]` reset signal. All output and pipeline registers use `aclr[1]` reset signal.
DSP View Block.			
Chain In Multiplexer (14)	Enable Disable	Disable	Click on the multiplexer to enable `chainin` port.
Chain Out Multiplexer (12)	Disable Enable	Disable	Click on the multiplexer to enable `chainout` port.
Adder (13)	**+** **-**	**+**	Click on the **Adder** symbol to select addition or subtraction mode.
Register Clock • ax_clock (2) • ay_clock (3) • az_clock (4) • mult_pipeline_clock(5) • ax_chainin_pl_clock (7) • adder_input_clock (9) • adder_input_2_clock (10) • output_clock (11) • accumulate_clock (1) • accum_pipeline_clock (6) • accum_adder_clock (8)	**None** **Clock 0** **Clock 1** **Clock 2**	**Clock 0**	To bypass any register, toggle the register clock to **None**. Toggle the register clock to: • **Clock 0** to use `clk[0]` signal as the clock source • **Clock 1** to use `clk[1]` signal as the clock source • **Clock 2** to use `clk[2]` signal as the clock source You can only change these settings when you select **Register Enables** in **View** parameter.

Figure 1. DSP Block View

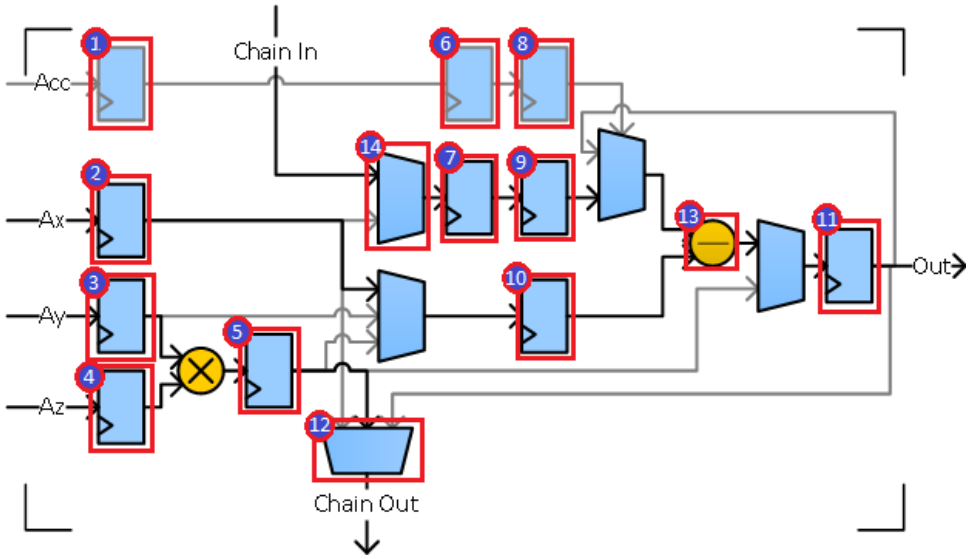


Table 2. DSP Templates

DSP Templates	Description
Multiply	Performs single precision multiplication operation and applies the following equation: • Out = Ay * Az
Add	Performs single precision addition or subtraction operation and applies the following equations:. • Out = Ay + Ax • Out = Ay - Ax
Multiply Add	This mode performs single precision multiplication, followed by addition or subtraction operations and applies the following equations. • Out = (Ay * Az) - chainin • Out = (Ay * Az) + chainin • Out = (Ay * Az) - Ax • Out = (Ay * Az) + Ax
Multiply Accumulate	Performs floating-point multiplication followed by floating-point addition or subtraction with the previous multiplication result and applies the following equations: • Out(t) = [Ay(t) * Az(t)] - Out (t-1) when `accumulate` signal is driven high. • Out(t) = [Ay(t) * Az(t)] + Out (t-1) when `accumulate` port is driven high. • Out(t) = Ay(t) * Az(t) when `accumulate` port is driven low.
Vector Mode 1	Performs floating-point multiplication followed by floating-point addition or subtraction with the chainin input from the previous variable DSP block and applies the following equations:.

continued...

DSP Templates	Description
	• Out = (Ay * Az) - chainin • Out = (Ay * Az) + chainin • Out = (Ay * Az) , chainout = Ax
Vector Mode 2	Performs floating-point multiplication where the IP core feeds the multiplication result is directly to `chainout`. The IP core then adds or subtracts the `chainin` input from the previous variable DSP block from input Ax as the output result. This mode applies the following equations: • Out = Ax - chainin , chainout = Ay * Az • Out = Ax + chainin , chainout = Ay * Az • Out = Ax , chainout = Ay * Az

1.3 Intel Arria 10 Native Floating-Point DSP Intel FPGA IP Signals

Figure 2. Intel Arria 10 Native Floating-Point DSP Intel FPGA IP Signals

The figure shows the input and output signals of the IP core.

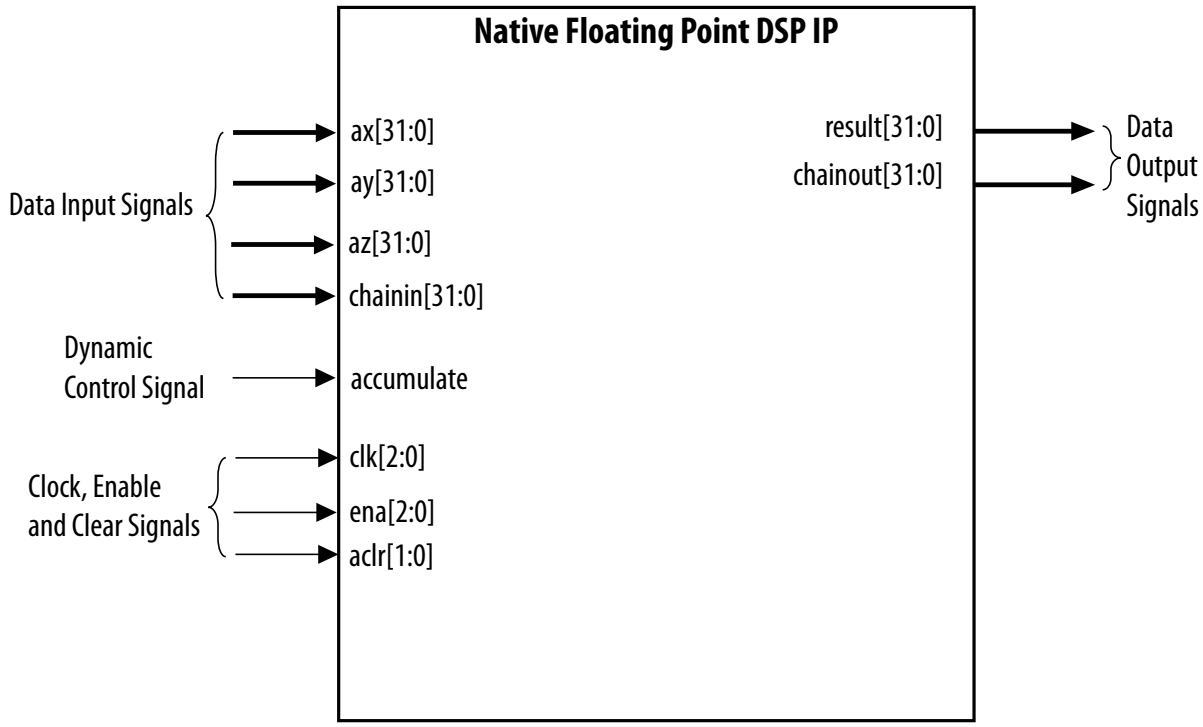
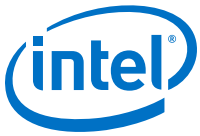


Table 3. Intel Arria 10 Native Floating-Point DSP Intel FPGA IP Input Signals

Signal Name	Type	Width	Default	Description
ax[31:0]	Input	32	Low	Input data bus to the multiplier. Available in: • Add mode • Multiply-Add mode without chainin and chainout feature • Vector Mode 1 • Vector Mode 2
ay[31:0]	Input	32	Low	Input data bus to the multiplier. Available in all floating-point operational modes.
az[31:0]	Input	32	Low	Input data bus to the multiplier. Available in: • Multiply • Multiply Add • Multiply Accumulate • Vector Mode 1 • Vector Mode 2
chainin[31:0]	Input	32	Low	Connect these signals to the chainout signals from the preceding floating-point DSP IP core.
clk[2:0]	Input	3	Low	Input clock signals for all registers. These clock signals are only available if any of the input registers, pipeline registers, or output register is set to **Clock0** or **Clock1** or **Clock2**.
ena[2:0]	Input	3	High	Clock enable for clk[2:0]. These signals are active-High. • ena[0] is for **Clock0** • ena[1] is for **Clock1** • ena[2] is for **Clock2**
aclr[1:0]	Input	2	Low	Asynchronous clear input signals for all registers. These signals are active-high. Use **aclr[0]** for all input registers and use **aclr[1]** for all pipeline and output registers.
accumulate	Input	1	Low	Input signal to enable or disable the accumulator feature. • Assert this signal to enable feedback the adder's output. • De-assert this signal to disable the feedback mechanism. You can assert or de-assert this signal during run-time. Available in Multiply Accumulate mode.
chainout[31:0]	Output	32	—	Connect these signals to the chainin signals of the next floating-point DSP IP core.
result[31:0]	Output	32	—	Output data bus from IP core.

1.4 Document Revision History

Document revision history for the Intel Arria 10 Native Floating-Point DSP Intel FPGA IP User Guide.



Date	Version	Changes
November 2017	2017.11.06	Removed **Clear type** parameter.
May 2016	2016.05.30	Initial release.