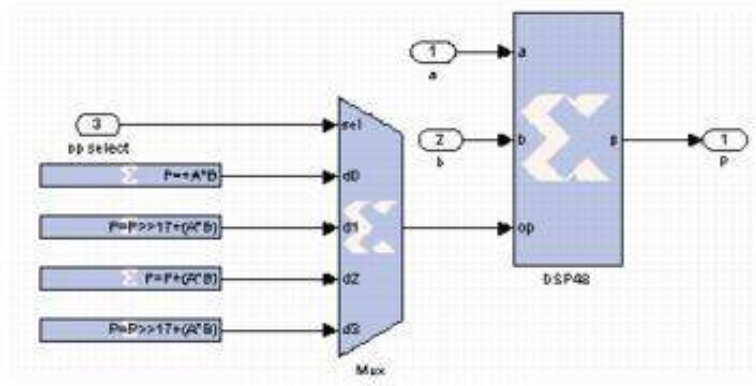
Constant

This block is listed in the following Xilinx Blockset libraries: Basic Elements, Control Logic, Math, Floating-Point and Index.

The Xilinx Constant block generates a constant that can be a fixed-point value, a Boolean value, or a DSP48 instruction. This block is similar to the Simulink constant block, but can be used to directly drive the inputs on Xilinx blocks.

DSP48 Instruction Mode

The constant block, when set to create a DSP48 instruction, is useful for generating DSP48 control sequences. The the figure below shows an example. The example implements a 35x35-bit multiplier using a sequence of four instructions in a DSP48 block. The constant blocks supply the desired instructions to a multiplexer that selects each instruction in the desired sequence.



Block Parameters

The block parameters dialog box can be invoked by double-clicking the icon in your Simulink model.

Basic tab

Parameters specific to the Basic tab are as follows:

Constant Value

> Specifies the value of the constant. When changed, the new value appears on the block icon. If the constant data type is specified as fixed-point and cannot be expressed exactly in the specified fixed-point type, its value is rounded and saturated as needed. A positive value is implemented as an unsigned number, a negative value as signed.

Output Precision

- Specifies the data type of the output. Can be **Boolean**, **Fixed-point**, or **Floating-point.**

  Arithmetic Type: If the Output Type is specified as Fixed-point, you can select **Signed (2's comp)**, **Unsigned** or **DSP48 instruction** as the Arithmetic Type.

  Fixed-point Precision

  - **Number of bits**: specifies the bit location of the binary point of the output number, where bit zero is the least significant bit.
  - **Binary point**: position of the binary point. in the fixed-point output

Floating-point Precision

  - **Single**: Specifies single precision (32 bits)
  - **Double**: Specifies double precision (64 bits)
  - **Custom**: Activates the field below so you can specify the Exponent width and the Fraction width.

  Exponent width: Specify the exponent width

  Fraction width: Specify the fraction width

Sample Period

- **Sampled Constant**: allows a sample period to be associated with the constant output and inherited by blocks that the constant block drives. (This is useful mainly because the blocks eventually target hardware and the Simulink sample periods are used to establish hardware clock periods.)

DSP48 tab

DSP48 Instruction

When DSP48 Instruction is selected for type, the DSP48 tab is activated. A detailed description of the DSP48 can be found in the DSP48 block description.

- **DSP48 operation**: displays the selected DSP48 instruction.
- **Operation select**: allows the selection of a DSP48 instruction. Selecting custom reveals mask parameters that allow the formation of an instruction in the form z_mux +/-(yx_mux + carry).

Custom Instruction

- **Z Mux**: specifies the 'Z' source to the DSP48's adder to be one of {'0', 'C', 'PCIN', 'P','C', 'PCIN>>17',' P>>17'}.
- **Operand**: specifies whether the DSP48's adder is to perform addition or subtraction.
- **YX Muxes**: specifies the 'YX' source to the DSP48's adder to be one of {'0','P', 'A:B', 'A*B', 'C', 'P+C', 'A:B+C'}. 'A:B' implies that A[17:0] is concatenated with B[17:0] to produce a 36-bit value to be used as an input to the DSP48 adder.
- **Carry input**: specifies the 'carry' source to the DSP48's adder to be one of {'0', '1', 'CIN', '~SIGN(P or PCIN)', '~SIGN(A:B or A*B)' ,. '~SIGND(A:B or A*B)'}. '~SIGN (P or PCIN)' implies that the carry source is either P or PCIN depending on the Z Mux setting. '~SIGN(A*B or A:B)' implies that the carry source is either A*B or A:B depending on the YX Mux setting. The option '~SIGND (A*B or A:B)' selects a delayed version of '~SIGN(A*B or A:B)'.

Other parameters used by this block are explained in the topic Common Options in Block Parameter Dialog Boxes.

Appendix: DSP48 Control Instruction Format

Instruction Field Name	Location	Mnemonic	Description
YX Mux	op[3:0]	0	0
		P	DSP48 output register
		A:B	Concat inputs A and B (A is MSB)
		A*B	Multiplication of inputs A and B
		C	DSP48 input C
		P+C	DSP48 input C plus P
		A:B+C	Concat inputs A and B plus C register
Z Mux	op[6:4]	0	0
		PCIN	DSP48 cascaded input from PCOUT
		P	DSP48 output register
		C	DSP48 C input
		PCIN>>17	Cascaded input downshifted by 17
		P>>17	DSP48 output register downshifted by 17
Operand	op[7]	+	Add
		-	Subtract
Carry In	op[8]	0 or 1	Set carry in to 0 or 1
		CIN	Select cin as source
		'~SIGN(P or PCIN)	Symmetric round P or PCIN
		'~SIGN(A:B or A*B)	Symmetric round A:B or A*B
		'~SIGND(A:B or A*B)	Delayed symmetric round of A:B or A*B