Fixed Point Simulation

SystemVue provides an extensive set of Fixed Point parts/models in its Hardware Design Library. SystemVue Fixed Point models use SystemC™ fixed point type based on IEEE Std. 1666™ Language Reference Manual (LRM).

For detailed description of SystemC™ fixed-point type, consult section 7.10 of SystemC™ IEEE Std. 1666™ Language Reference Manual (LRM).

Hardware Design Parts in Hardware Design Library can be used to build, simulate and analyze fixed point systems. A range of functions from low level logic elements to more advanced signal processing parts such as filters and fast Fourier transforms (FFTs) are available.

The fixed-to-float and float-to-fixed conversion parts provide a means of interfacing fixed point components with other SystemVue blocks. Fxp parts can also be configured to automatically collect information on dynamic range, overflows and underflows and be shown in Fixed Point Analysis Table in order to help with system optimization.

When used in conjunction with the HDL Code Generation, Verilog/VHDL RTL code describing the fixed point system can be automatically generated. See HDL Code Generation, in the Simulation > Data Flow section for more information.

Using Fixed-Point Parts

Conversion between Fixed-Point and Non-Fixed-Point Data

SystemVue does not support automatic conversion from non-fixed-point to fixed-point data. To convert floating-point to fixed-point data use FloatToFxp part. You could also convert from fixed-point to floating-point data using FxpToFloat part.

Fixed-Point Precision at the Input Port

SystemVue resolves the fixed-point precision at the input port automatically based on the output precision of the previous component in the data flow schematic. A user does not have to specify the precision for the input port.

Fixed-Point Precision at the Output Port

For most of the parts you need to specify the fixed-point precision at the output port explicitly. There are cases where output port precision is tightly coupled with fixed-point precision at the input ports, for example, MuxFxp, and DelayFxp, in such cases it may not be required to specify output precision. The fixed-point models, which require a user to specify output port precision, share the following common parameters.

Wordlength

This parameter is used to specify the total number of bits in the fixed-point, including fractional and integer bits.

IntegerWordlength

This parameter is used to specify the number of integer bits in the fixed-point including the sign bit. Based on the values of Wordlength and IntegerWordlength the fixed point precision could have one of the following three scenarios

- **Wordlength < IntegerWordlength**: There will be (IntegerWordlength - Wordlength) zeros added between the Least Significant Bit (LSB) and the binary point.
- **0 <= IntegerWordlength <= Wordlength**: The fixed point will have IntegerWordlength integer bits and (Wordlength - IntegerWordlength) fractional bits. The binary point will be fully contained in the precision specified.
- **IntegerWordlength < 0**: There are (-IntegerWordlength) signed-extended bits added between binary point and Most Significant Bit (MSB).

Please consult section 7.10.1 of SystemC™ IEEE Std. 1666™ Language Reference Manual (LRM) for more details.

IsSigned

This parameter is used to specify that either the fixed-point is signed or un-signed type.

Quantization

This method is used to specify the quantization mode for the fixed-point. There are seven possible modes

- **RND**: Rounding to plus infinity. The corresponding SystemC™ mode is SC_RND.
**RND_ZERO**: Rounding to zero. The corresponding SystemC™ mode is SC_RND_ZERO.

**RND_MIN_INF**: Rounding to minus infinity. The corresponding SystemC™ mode is SC_RND_MIN_INF.

**RND_INF**: Rounding to infinity. The corresponding SystemC™ mode is SC_RND_INF.

**RND_CONV**: Convergent rounding. The corresponding SystemC™ mode is SC_RND_CONV.

**TRN**: Truncation. The corresponding SystemC™ mode is SC_TRN.

**TRN_ZERO**: Truncation to zero. The corresponding SystemC™ mode is SC_TRN_ZERO.

Please consult section 7.10.9 of SystemC™ IEEE Std. 1666™ Language Reference Manual (LRM) for more details.

**Overflow**

This parameter is used to specify overflow mode. The overflow occurs when specified precision is not enough to hold the value. There are five possible modes:

- **SAT**: Saturation. The corresponding SystemC™ mode is SC_SAT.
- **SAT_ZERO**: Saturation to zero. The corresponding SystemC™ mode is SC_SAT_ZERO.
- **SAT_SYM**: Symmetrical saturation. The corresponding SystemC™ mode is SC_SAT_SYM.
- **WRAP**: Wrap-around. The corresponding SystemC™ mode is SC_WRAP.
- **WRAP_SM**: Sign magnitude wrap-around. The corresponding SystemC™ mode is SC_WRAP_SM.

Please consult section 7.10.9 of SystemC™ IEEE Std. 1666™ Language Reference Manual (LRM) for more details about overflow mode.

**SaturationBits**

This parameter is used to provide number of saturation bits for WRAP and WRAP_SM Overflow modes.

Please consult Table 39 and Table 40 in section 7.10.9 of SystemC™ IEEE Std. 1666™ Language Reference Manual (LRM) for more details about saturation bits.

**TypeOverride**

All fixed point models share a common parameter **TypeOverride**. This parameter is used to override the numeric format of fixed point block. There are 3 possible modes:

- **OFF**: The Fixed point model will be simulated with fixed-point precision. This is the default mode of operation.
- **Double**: The Fixed point model will be simulated with double floating-point precision.
- **Integer**: The Fixed point model will be simulated with integer precision.

**Note**

SystemVue does not support automatic conversion to fixed-point, you may need to consider this when switching the TypeOverride parameter.

**HDL Code Generation and Automatic HDL Cosimulation.**

SystemVue allows its users to generate VHDL or Verilog using the synthesizable parts in its Hardware Design Library. Please consult HDL Code Generation document for further details. SystemVue also provides its users the ability to automatically Cosimulate SystemVue generated HDL with ModelSim™ SE HDL simulator.

**Cosimulating with User HDL**

While simulating fixed-point design, it is possible in SystemVue to cosimulate with user written HDL using ModelSim™ SE HDL simulator by using HdICosim part. However, HDL Code Generation cannot synthesize a fixed-point model that includes user written HDL code, because HDLCosim part is not synthesizable. Also automatic HDL cosimulation is not possible with user written code. Please consult HDL Cosimulation, and HdICosim part document for more details.

**Fixed Point Analysis Table**

The fixed point analysis table is created automatically after simulating a design that contains fixed point parts. It can also be invoked from the right-click menu on the Data Set with fixed point analysis results on the Workspace Tree.

The fixed point analysis table contains 8 columns:

1. Name - the name of the design block.
2. Part - the type of the block.
3. Signal - the name of the fixed point signal.
4. Precision - the fixed point number representation (± indicates signed, the first number in the angle brackets is the register word length, the second number in the angle brackets is the length of the integer part).
5. Overflows - shows the total number and percentage of overflows on the signal.
6. Underflows - shows the total number and percentage of underflows on the signal.
7. Max - the maximum value of the signal.
8. Min - the minimum value of the signal.

As overflows and underflows are generally undesirable, they are shown in red.

Fixed Point Examples

SystemVue is shipped with examples using fixed point analysis with HDL code generation. These examples are located in the directory <SystemVue installation directory>\Examples\Hardware Design.