

# Latest Features in Simulink Fixed Point

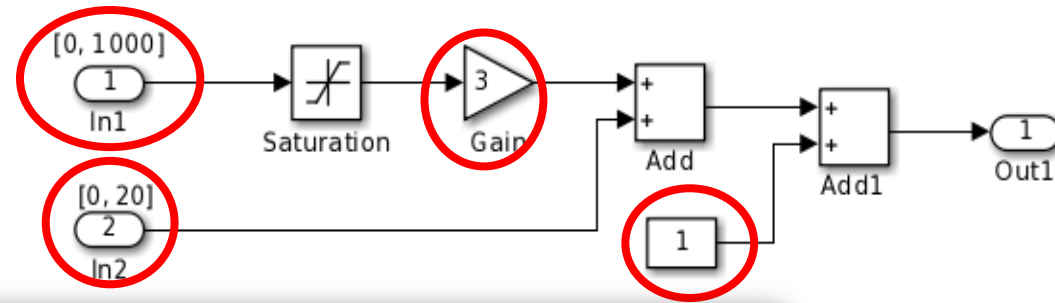
September 2011

**R2011b**

# Derived Minimum and Maximum Ranges

**Autoscale models based on worst-case range analysis**

- No need to simulate
- No need to synthesize test vectors to stress ranges



Analyze model and store derived min/max

The screenshot shows the **Fixed-Point Tool** interface. The **Contents of: derived\_ranges** pane displays the following data:

Name	DerivedMin	DerivedMax	DesignMin	DesignMax
Add : Output 0	0	1520		
Gain	0	1500		
Saturation	0	500		
Add1 : Output 1		1521		
Gain : Gain			0	1000
In1			0	20

The **Options** pane shows the following settings:

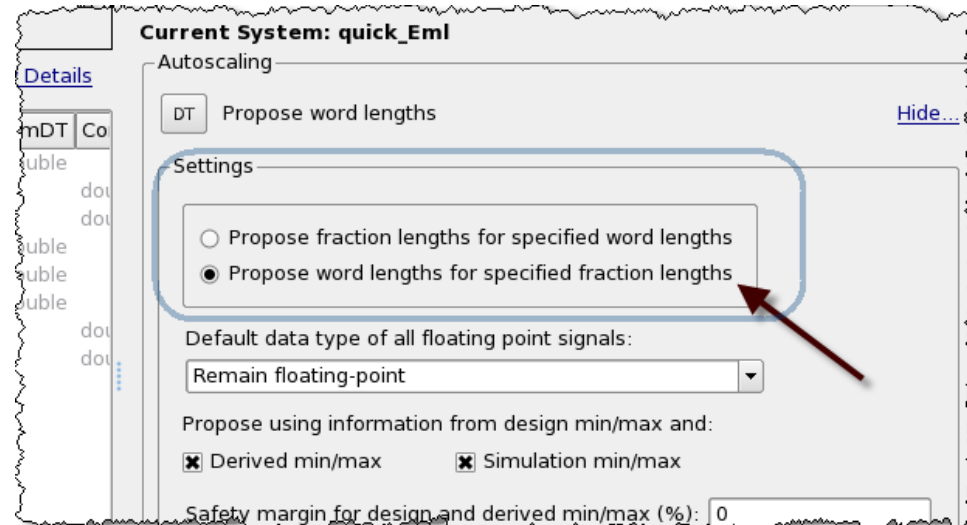
- Data type override:** Use local settings
- Data collection:**
  - Simulate model and store results in Active run
  - Merge instrumentation results from multiple simulations
  - Analyze model and store derived min/max results in Active run
  - Highlight results with potential issues
- Autoscaling:**
  - Propose fraction lengths for specified word lengths
  - Propose word lengths for specified fraction lengths

Name	DerivedMin	DerivedMax
Add : Output 0	0	1520
Gain	0	1500
Saturation	0	500
Add1 : Output 1		1521

# Derived Word Lengths

## Automatically derive word lengths to avoid model and code overflow

- Propose word lengths using fixed fraction lengths and hardware information
  - Such as integer word sizes
- Propose optimized multiword data types when computed ideal word length is greater than the long type supported by hardware
- Fully integrated into the Fixed-Point Advisor and Fixed-Point Tool workflows



# Multiple Runs Support in Fixed-Point Tool

## Create and store more than two runs in Fixed-Point Tool

- No need to move or exchange runs anymore
- Specify run names
- Propose and accept data types
- Difference signals between runs

Model Hierarchy: fxpdemo\_feedback\* > Controller (mimo-d)

Contents of: Controller (mimo-d)

Column View: Autoscaling View

Name	Run	SimDT	CompiledDT	Accept	ProposedDT	SpecifiedDT
Run: DoubleOverride						
Combine Terms : Accumulator	DoubleOverride	double		<input type="checkbox"/>		Inherit: Inherit via intern...
Combine Terms : Output	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,32,12)
Denominator Terms : Accumul...	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,32,12)
Denominator Terms : Output	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,32,12)
Denominator Terms : Product ...	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,32,12)
Down Cast	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,16,5)
In1	DoubleOverride	double		<input type="checkbox"/>		Inherit: auto
Numerator Terms : Accumulator	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,32,12)
Numerator Terms : Output	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,32,12)
Numerator Terms : Product ou...	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,32,12)
Prev Out	DoubleOverride	double		<input type="checkbox"/>		
Up Cast	DoubleOverride	double		<input type="checkbox"/>		fixdt(1,16,14)
Run: FixedPoint						
Combine Terms : Accumulator	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		Inherit: Inherit via intern...
Combine Terms : Output	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		fixdt(1,32,12)
Denominator Terms : Accumul...	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		fixdt(1,32,12)
Denominator Terms : Output	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		fixdt(1,32,12)
Denominator Terms : Product ...	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		fixdt(1,32,12)
Down Cast	FixedPoint	fixdt(1,16,5)		<input type="checkbox"/>		fixdt(1,16,5)
In1	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		Inherit: auto
Numerator Terms : Accumulator	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		fixdt(1,32,12)
Numerator Terms : Output	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		fixdt(1,32,12)
Numerator Terms : Product ou...	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		fixdt(1,32,12)
Prev Out	FixedPoint	fixdt(1,32,12)		<input type="checkbox"/>		
Up Cast	FixedPoint	fixdt(1,16,14)		<input type="checkbox"/>		fixdt(1,16,14)
Run: ControllerDbfRun						
Combine Terms : Accumulator	ControllerDbfRun	double		<input type="checkbox"/>	n/a	Inherit: Inherit via intern...
Combine Terms : Output	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,32,28)	fixdt(1,32,12)
Denominator Terms : Accumul...	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,32,27)	fixdt(1,32,12)
Denominator Terms : Output	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,32,28)	fixdt(1,32,12)
Denominator Terms : Product ...	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,32,27)	fixdt(1,32,12)
Down Cast	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,16,12)	fixdt(1,16,5)
In1	ControllerDbfRun	double		<input type="checkbox"/>	n/a	Inherit: auto
Numerator Terms : Accumulator	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,32,28)	fixdt(1,32,12)
Numerator Terms : Output	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,32,29)	fixdt(1,32,12)
Numerator Terms : Product ou...	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,32,28)	fixdt(1,32,12)
Prev Out	ControllerDbfRun	double		<input type="checkbox"/>	n/a	
Up Cast	ControllerDbfRun	double		<input checked="" type="checkbox"/>	fixdt(1,16,12)	fixdt(1,16,14)

Data collection

Store results in run:

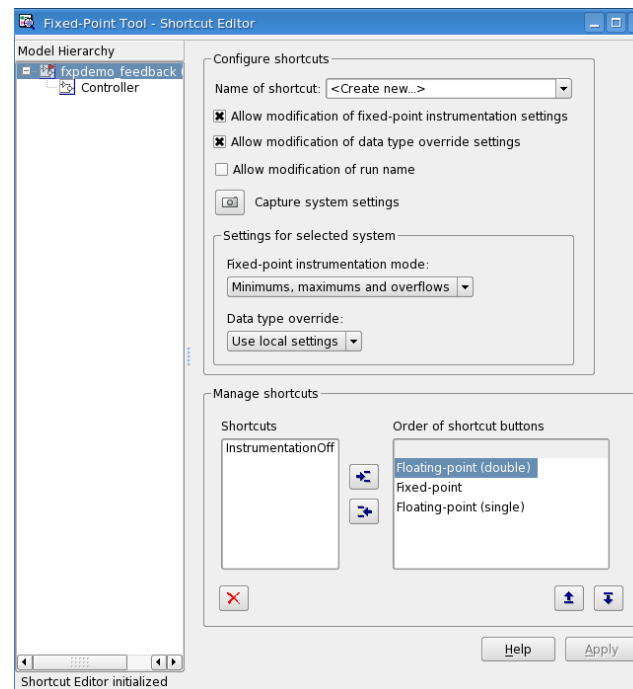
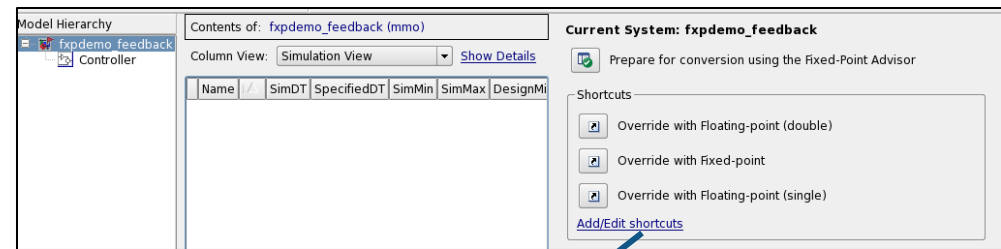
Simulate model and store results

Merge instrumentation results from multiple simulations

# Customized Shortcuts in Fixed-Point Tool

Easily change data type override, instrumentation, and run settings using shortcuts

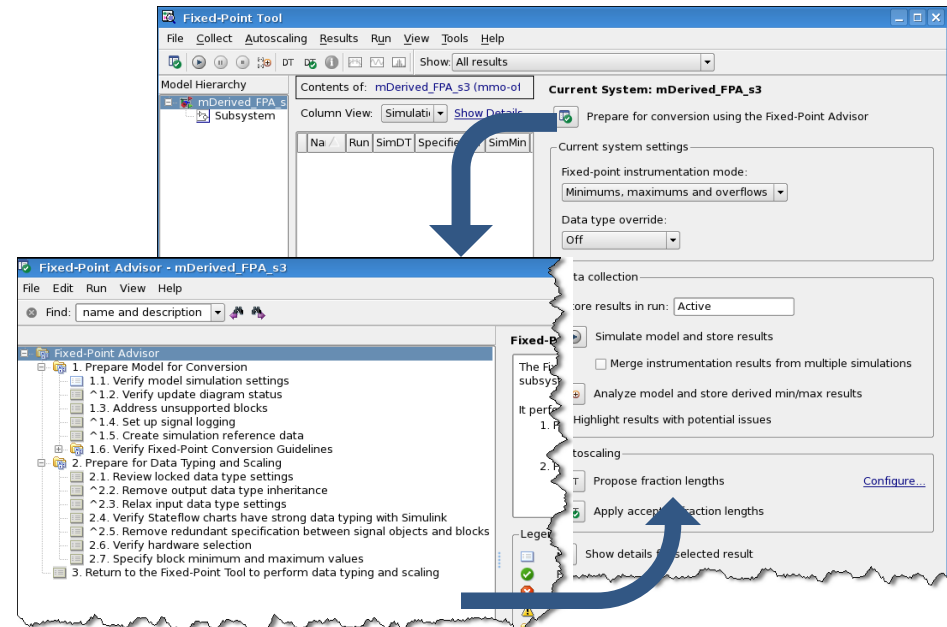
- Helpful when settings change frequently
- Shortcuts saved with model



# Fixed-Point Tool Integration with Fixed-Point Advisor

Convert floating-point models to fixed point using integrated workflow

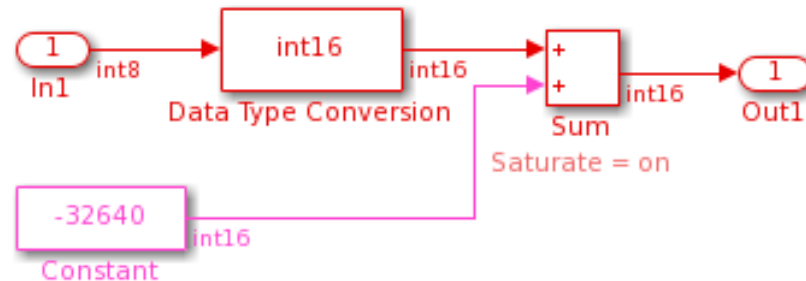
- Launch Fixed-Point Advisor from Fixed-Point Tool and vice versa
- Supports floating-point, fixed-point, and mixed data type designs
- Examines designs to generate optimized production code



# Improved Fixed-Point Saturating Cast

## Reduce single saturation branches

- Improved removal of low-side branch
- Common situation with integer math



R2011a

```
int32_T tmp;
tmp = oss_U.In7 + -32640;
if (tmp <= -32768) {
    tmp = -32768;
}
```

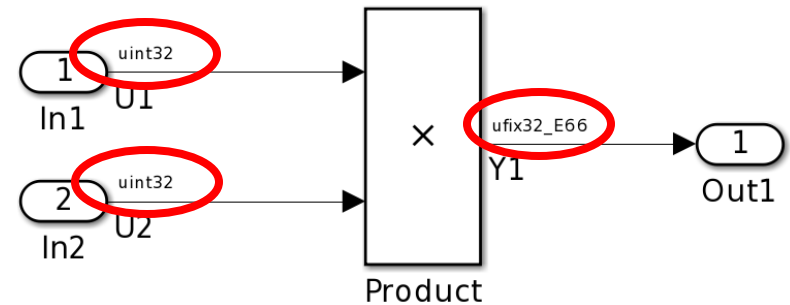
R2011b

```
oss_Y.Out1 = (int16_T) (oss_U.In1 + -32640);
```

# Excessive Shift Optimization

## Avoid multiplication helpers for large shift cases

- For shift left or right in all rounding modes
- Can occur when instantiating library subsystems with fixed-point types



R2011a

```
Y1 =
    mul_u32_u32_u32_sr64(U1, U2);
```

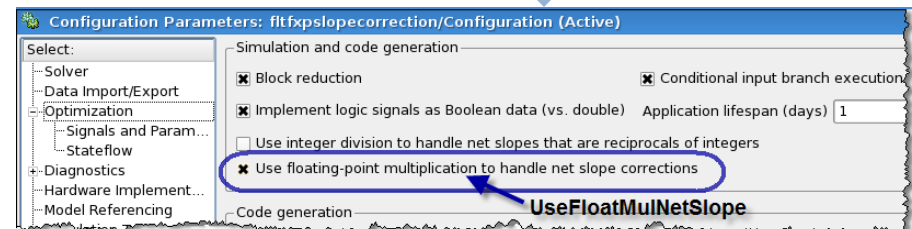
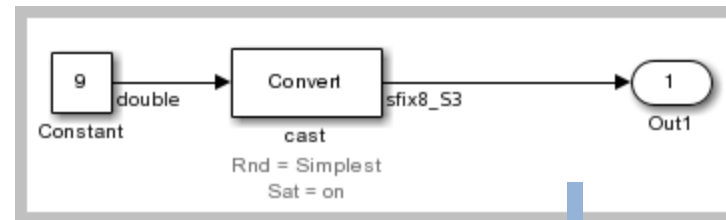
R2011b

```
Y1 = 0U;
```

# Multiplication for Net Slope Correction in Mixed Cast Operations

## Option to avoid floating-point division for slope correction

- When converting from floating point to fixed point, correcting net slope using division improves precision.
- Using floating-point multiplication of the reciprocal improves code efficiency for some processors.
- Optimization is selectable to evaluate tradeoffs.



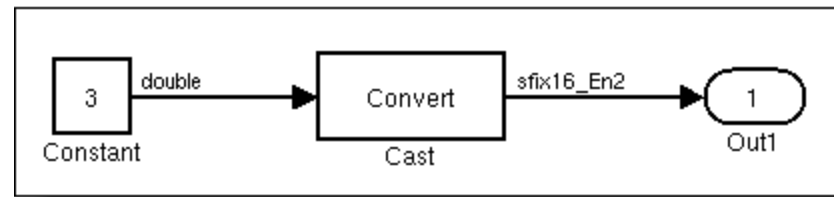
```
/* UseFloatMulNetSlope = 'off' */
tmp = mdl_P.Constant_Value / 3.0;
```

```
/* UseFloatMulNetSlope = 'on' */
tmp = mdl_P.Constant_Value * 0.333333333333333331;
```

# Remove Use of LDEXP for Mixed Casts

## More efficient mixed cast operations

- Use multiplication by power of two instead of `ldexp` function
- May improve range analysis of code and facilitate verification
- More readable generated code
  - No efficiency disadvantage



**R2011b**

```

/* DataTypeConversion: '<Root>/Cast' incorporates:
 * Constant: '<Root>/Constant'
 */
tmp = cast_P.Constant_Value * 4.0;
    
```

**R2011a**

```

/* DataTypeConversion: '<Root>/Cast' incorporates:
 * Constant: '<Root>/Constant'
 */
tmp = ldexp(cast_P.Constant_Value, 2);
    
```