Alimomeni

810100215

CA2/ESL

Part1:

Advantages:

1. **Simpler and faster hardware implementation:** In fixed-point systems, due to simpler computations and fewer required hardware elements, processing speed increases and manufacturing costs decrease.

2. **More precision and control:** Fixed-point systems offer more precision and control over numerical operations in specific, limited ranges, which is useful in real-time and embedded systems.

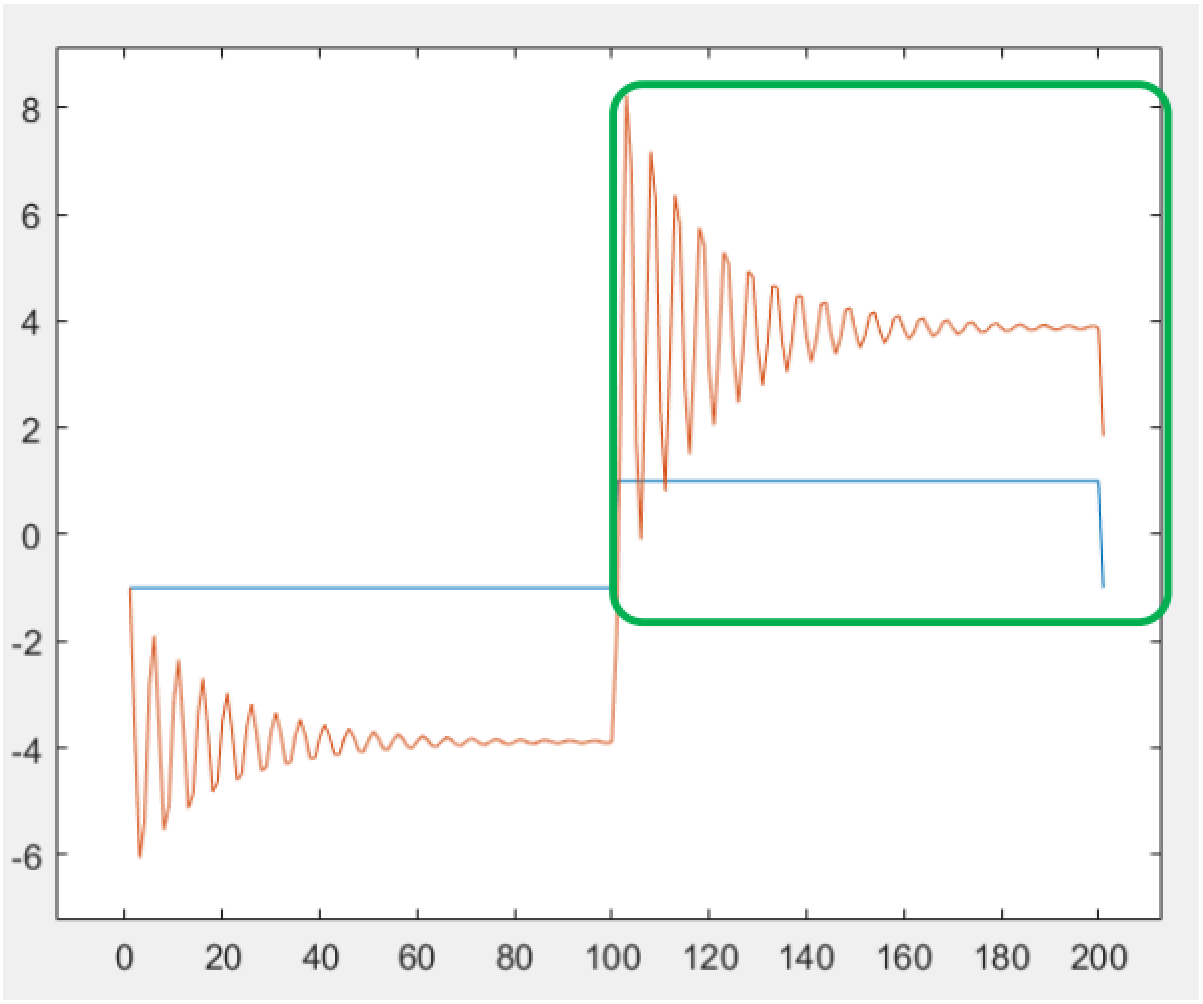
Disadvantages:

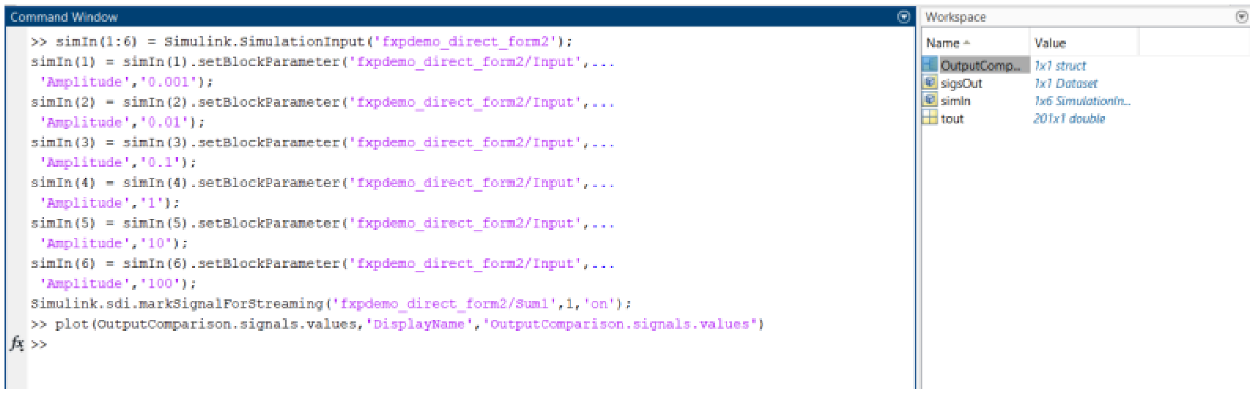
1. **Higher risk of overflow and underflow:** In fixed-point systems, managing overflow and underflow requires careful design, and these issues are more likely compared to floating-point systems.

2. **Limited range:** With the same number of bits, fixed-point systems have a more limited dynamic range compared to floating-point systems, making them less suitable for applications that require handling very large or very small numbers.

Digital Filter Analysis:

Simulation Results and Comparison Output Chart:



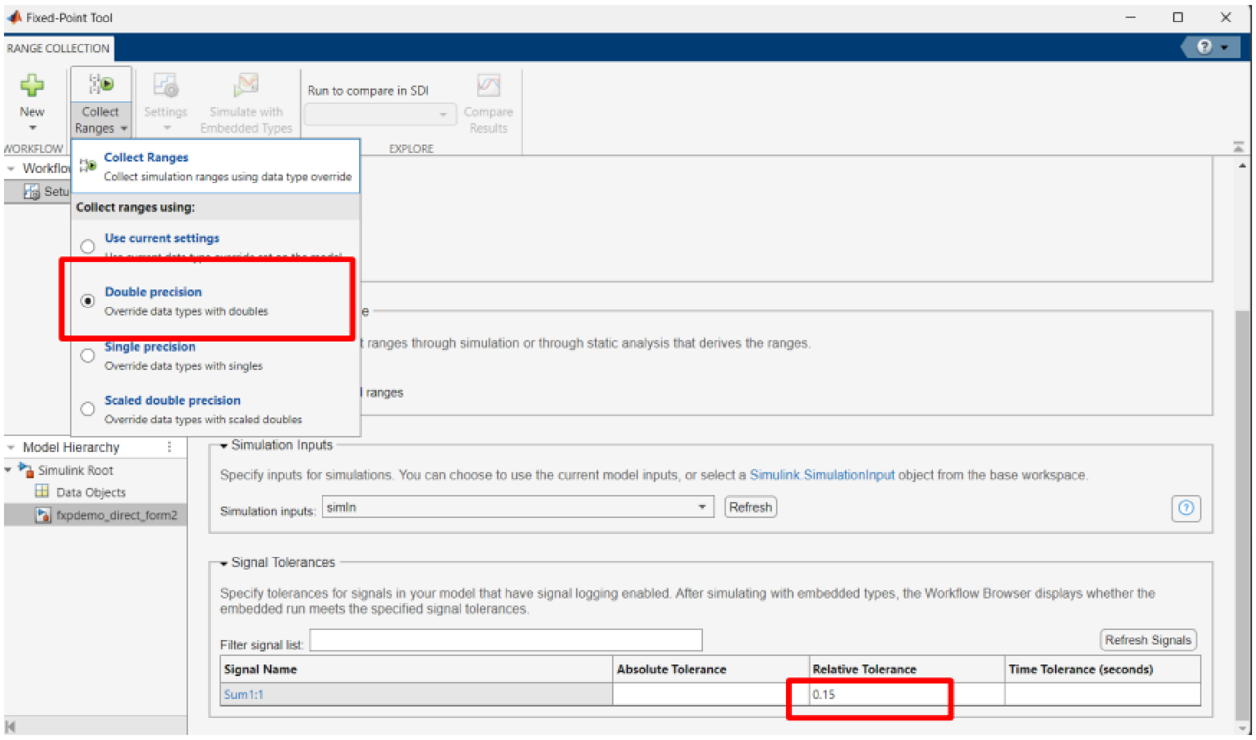


```
>> simIn(1:6) = Simulink.SimulationInput('fxpdemo_direct_form2');
simIn(1) = simIn(1).setBlockParameter('fxpdemo_direct_form2/Input',...
 'Amplitude','0.001');
simIn(2) = simIn(2).setBlockParameter('fxpdemo_direct_form2/Input',...
 'Amplitude','0.01');
simIn(3) = simIn(3).setBlockParameter('fxpdemo_direct_form2/Input',...
 'Amplitude','0.1');
simIn(4) = simIn(4).setBlockParameter('fxpdemo_direct_form2/Input',...
 'Amplitude','1');
simIn(5) = simIn(5).setBlockParameter('fxpdemo_direct_form2/Input',...
 'Amplitude','10');
simIn(6) = simIn(6).setBlockParameter('fxpdemo_direct_form2/Input',...
 'Amplitude','100');
Simulink.sdi.markSignalForStreaming('fxpdemo_direct_form2/Sum1',1,'on');
>> plot(OutputComparison.signals.values,'DisplayName','OutputComparison.signals.values')
fx >>
```

Name	Value
OutputComp...	1x1 struct
sigsOut	1x1 Dataset
simIn	1x6 SimulationIn...
tout	201x1 double

The output chart based on the input, after applying the corresponding MATLAB code, is shown above.
The output approximately corresponds to the green section in the image, which matches and aligns with the main graph.

Converting the Digital Filter to Fixed Point and Comparing it with Floating Point Representation:



Collect Ranges
Collect simulation ranges using data type override

Collect ranges using:

( ) Use current settings

(●) Double precision
Override data types with doubles

( ) Single precision
Override data types with singles

( ) Scaled double precision
Override data types with scaled doubles

Simulation Inputs

Specify inputs for simulations. You can choose to use the current model inputs, or select a Simulink.SimulationInput object from the base workspace.

Simulation inputs: simIn

Signal Tolerances

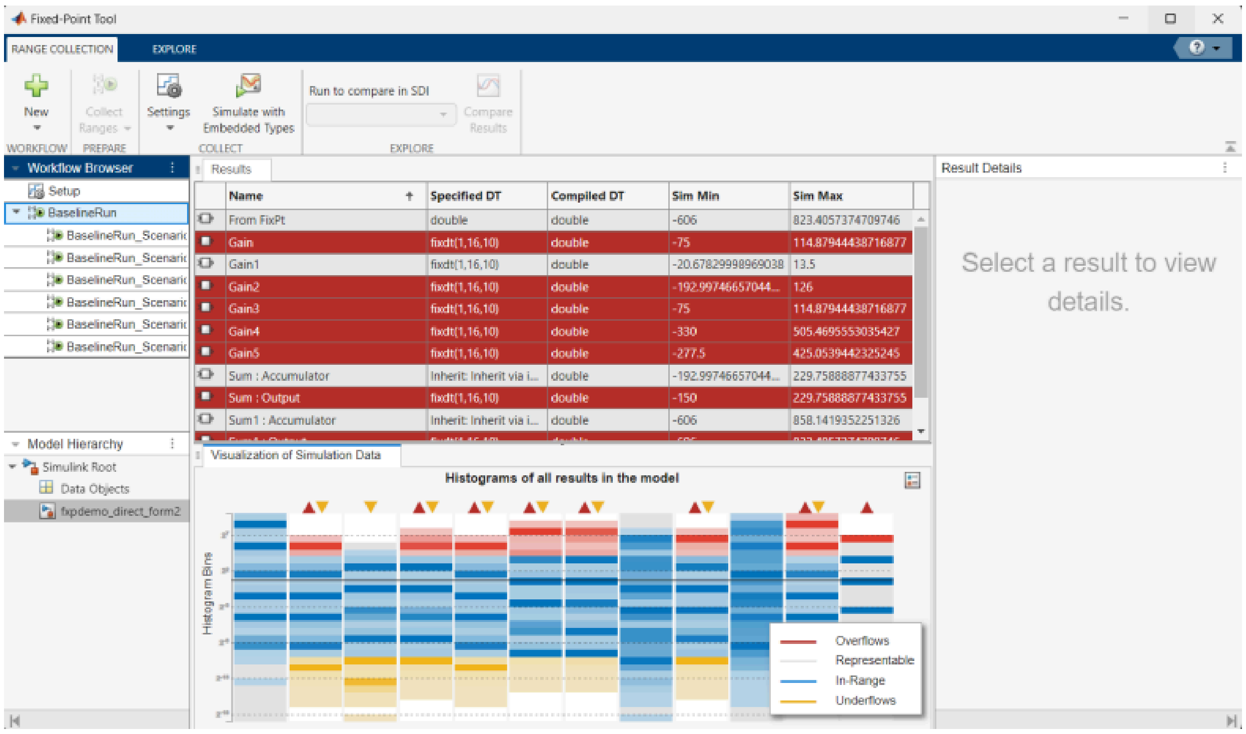
Specify tolerances for signals in your model that have signal logging enabled. After simulating with embedded types, the Workflow Browser displays whether the embedded run meets the specified signal tolerances.

Filter signal list:

Signal Name	Absolute Tolerance	Relative Tolerance	Time Tolerance (seconds)
Sum1:1		0.15	

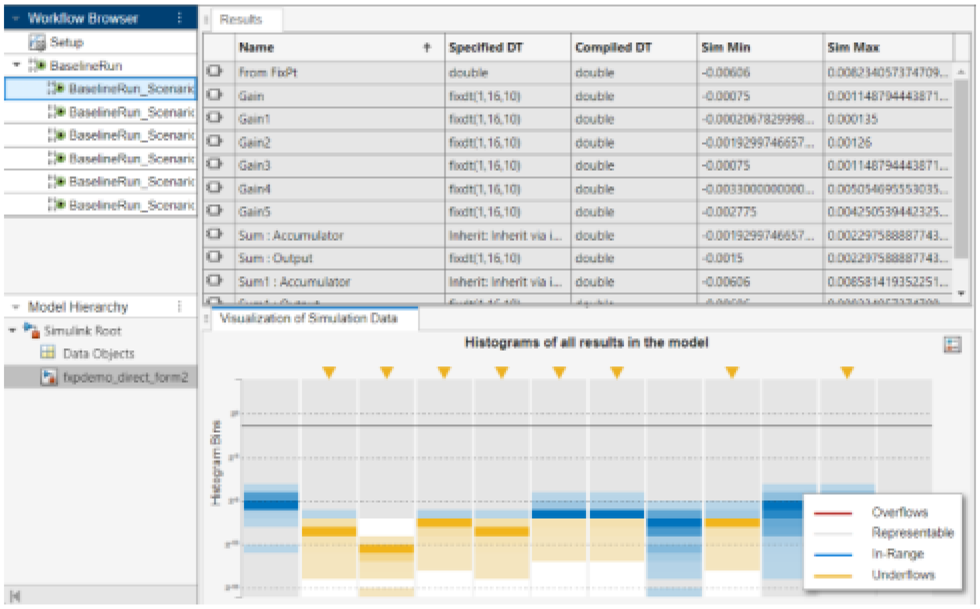
3.Underflow and Overflow Status for Each Set of 6 Inputs and Their Relation to the Input Signal:

It can be observed that a significant amount of underflow occurs initially, which is expected since the input values are small at the beginning, causing the system to experience underflow.
As the input amplitude increases, the amount of underflow is expected to decrease, which is evident in the chart as well. At the stage where the input values reach a moderate level, underflow no longer occurs, and its value becomes zero.
Additionally, if the input increases excessively, the system is expected to experience overflow, which is also observed in the graphs.

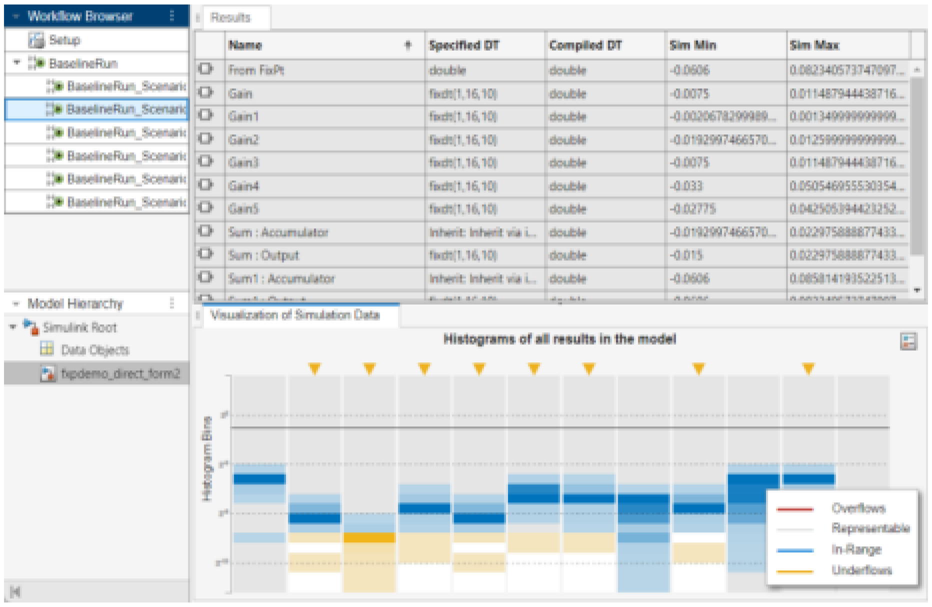


Name	Specified DT	Compiled DT	Sim Min	Sim Max
From FixPt	double	double	-606	823.4057374709746
Gain	fixdt(1,16,10)	double	-75	114.87944438716877
Gain1	fixdt(1,16,10)	double	-20.67829998969038	13.5
Gain2	fixdt(1,16,10)	double	-192.99746657044...	126
Gain3	fixdt(1,16,10)	double	-75	114.87944438716877
Gain4	fixdt(1,16,10)	double	-330	505.4695553035427
Gain5	fixdt(1,16,10)	double	-277.5	425.0539442325245
Sum : Accumulator	Inherit: Inherit via i...	double	-192.99746657044...	229.75888877433755
Sum : Output	fixdt(1,16,10)	double	-150	229.75888877433755
Sum1 : Accumulator	Inherit: Inherit via i...	double	-606	858.1419352251326

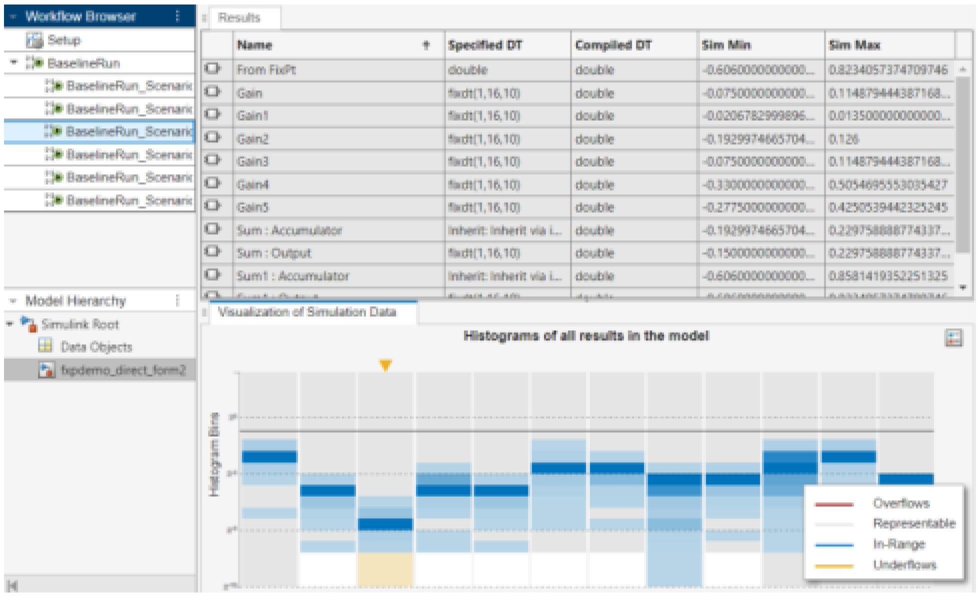
The charts for scenarios 1 to 6 individually are as follows:



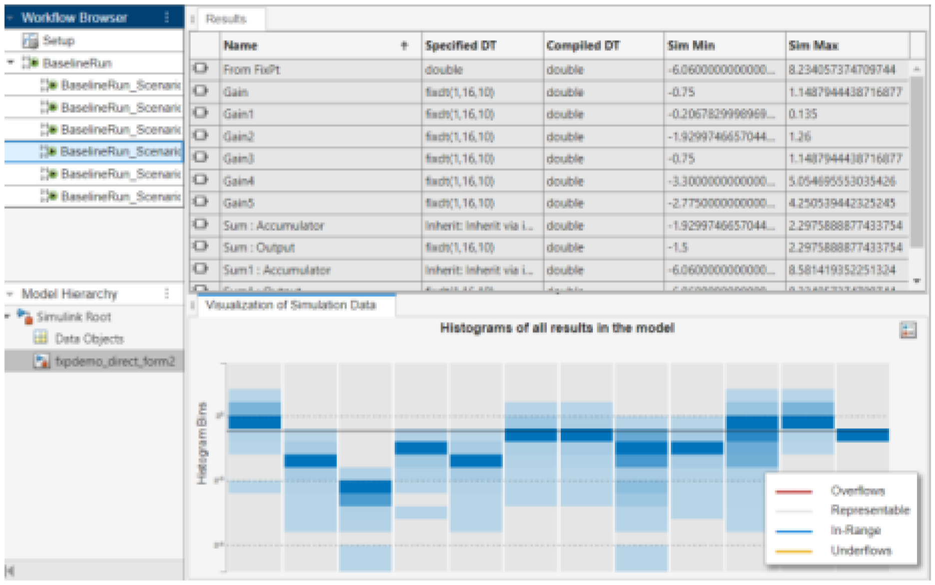
Name	Specified DT	Compiled DT	Sim Min	Sim Max
From FixPt	double	double	-0.00606	0.008234057374709...
Gain	fixdt(1,16,10)	double	-0.00075	0.001148794443871...
Gain1	fixdt(1,16,10)	double	-0.0002067829998...	0.000135
Gain2	fixdt(1,16,10)	double	-0.0019299746657...	0.00126
Gain3	fixdt(1,16,10)	double	-0.00075	0.001148794443871...
Gain4	fixdt(1,16,10)	double	-0.0033000000000...	0.005054695553035...
Gain5	fixdt(1,16,10)	double	-0.002775	0.004250539442325...
Sum : Accumulator	Inherit: Inherit via i...	double	-0.0019299746657...	0.002297588887743...
Sum : Output	fixdt(1,16,10)	double	-0.0015	0.002297588887743...
Sum1 : Accumulator	Inherit: Inherit via i...	double	-0.00606	0.008581419352251...



	Name	Specified DT	Compiled DT	Sim Min	Sim Max
	From FixPt	double	double	-0.0606	0.082340573747097...
	Gain	fixdt(1,16,10)	double	-0.0075	0.011487944438716...
	Gain1	fixdt(1,16,10)	double	-0.0020678299989...	0.001349999999999...
	Gain2	fixdt(1,16,10)	double	-0.0192997466570...	0.012599999999999...
	Gain3	fixdt(1,16,10)	double	-0.0075	0.011487944438716...
	Gain4	fixdt(1,16,10)	double	-0.033	0.050546955530354...
	Gain5	fixdt(1,16,10)	double	-0.02775	0.042505394423252...
	Sum : Accumulator	Inherit: Inherit via i...	double	-0.0192997466570...	0.022975888877433...
	Sum : Output	fixdt(1,16,10)	double	-0.015	0.022975888877433...
	Sum1 : Accumulator	Inherit: Inherit via i...	double	-0.0606	0.085814193522513...



	Name	Specified DT	Compiled DT	Sim Min	Sim Max
	From FixPt	double	double	-0.6060000000000...	0.8234057374709746
	Gain	fixdt(1,16,10)	double	-0.0750000000000...	0.114879444387168...
	Gain1	fixdt(1,16,10)	double	-0.0206782999896...	0.013500000000000...
	Gain2	fixdt(1,16,10)	double	-0.1929974665704...	0.126
	Gain3	fixdt(1,16,10)	double	-0.0750000000000...	0.114879444387168...
	Gain4	fixdt(1,16,10)	double	-0.3300000000000...	0.5054695553035427
	Gain5	fixdt(1,16,10)	double	-0.2775000000000...	0.4250539442325245
	Sum : Accumulator	Inherit: Inherit via i...	double	-0.1929974665704...	0.229758888774337...
	Sum : Output	fixdt(1,16,10)	double	-0.1500000000000...	0.229758888774337...
	Sum1 : Accumulator	Inherit: Inherit via i...	double	-0.6060000000000...	0.8581419352251325



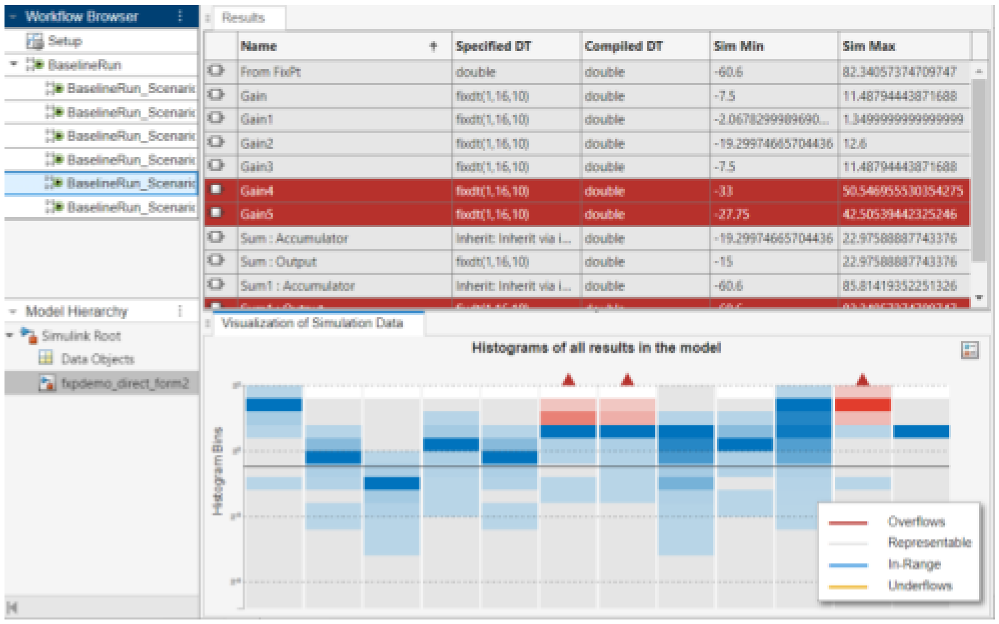
Workflow Browser
Setup
BaselineRun
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario

Model Hierarchy
Simulink Root
Data Objects
fxpdemo_direct_form2

Results

	Name	Specified DT	Compiled DT	Sim Min	Sim Max
	From FixPt	double	double	-6.0600000000000...	8.234057374709744
	Gain	fixdt(1,16,10)	double	-0.75	1.1487944438716877
	Gain1	fixdt(1,16,10)	double	-0.2067829998969...	0.135
	Gain2	fixdt(1,16,10)	double	-1.9299746657044...	1.26
	Gain3	fixdt(1,16,10)	double	-0.75	1.1487944438716877
	Gain4	fixdt(1,16,10)	double	-3.3000000000000...	5.054695553035426
	Gain5	fixdt(1,16,10)	double	-2.7750000000000...	4.250539442325245
	Sum : Accumulator	Inherit: Inherit via i...	double	-1.9299746657044...	2.2975888877433754
	Sum : Output	fixdt(1,16,10)	double	-1.5	2.2975888877433754
	Sum1 : Accumulator	Inherit: Inherit via i...	double	-6.0600000000000...	8.581419352251324

Visualization of Simulation Data



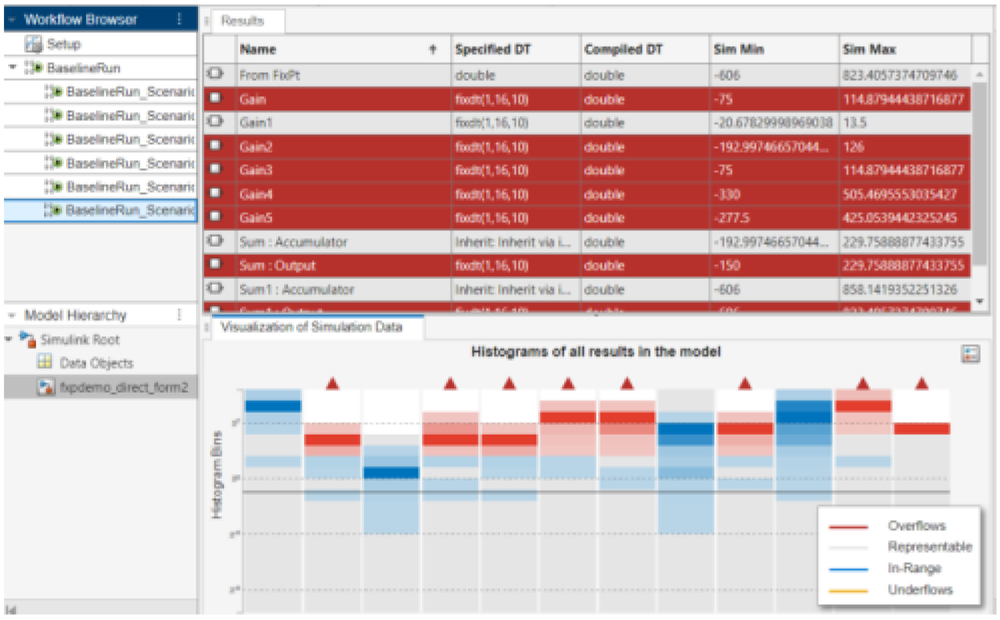
Workflow Browser
Setup
BaselineRun
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario
BaselineRun_Scenario

Model Hierarchy
Simulink Root
Data Objects
fxpdemo_direct_form2

Results

	Name	Specified DT	Compiled DT	Sim Min	Sim Max
	From FixPt	double	double	-60.6	82.34057374709747
	Gain	fixdt(1,16,10)	double	-7.5	11.48794443871688
	Gain1	fixdt(1,16,10)	double	-2.0678299989690...	1.3499999999999999
	Gain2	fixdt(1,16,10)	double	-19.29974665704436	12.6
	Gain3	fixdt(1,16,10)	double	-7.5	11.48794443871688
	Gain4	fixdt(1,16,10)	double	-33	50.546955530354275
	Gain5	fixdt(1,16,10)	double	-27.75	42.50539442325246
	Sum : Accumulator	Inherit: Inherit via i...	double	-19.29974665704436	22.975888877433376
	Sum : Output	fixdt(1,16,10)	double	-15	22.975888877433376
	Sum1 : Accumulator	Inherit: Inherit via i...	double	-60.6	85.81419352251326

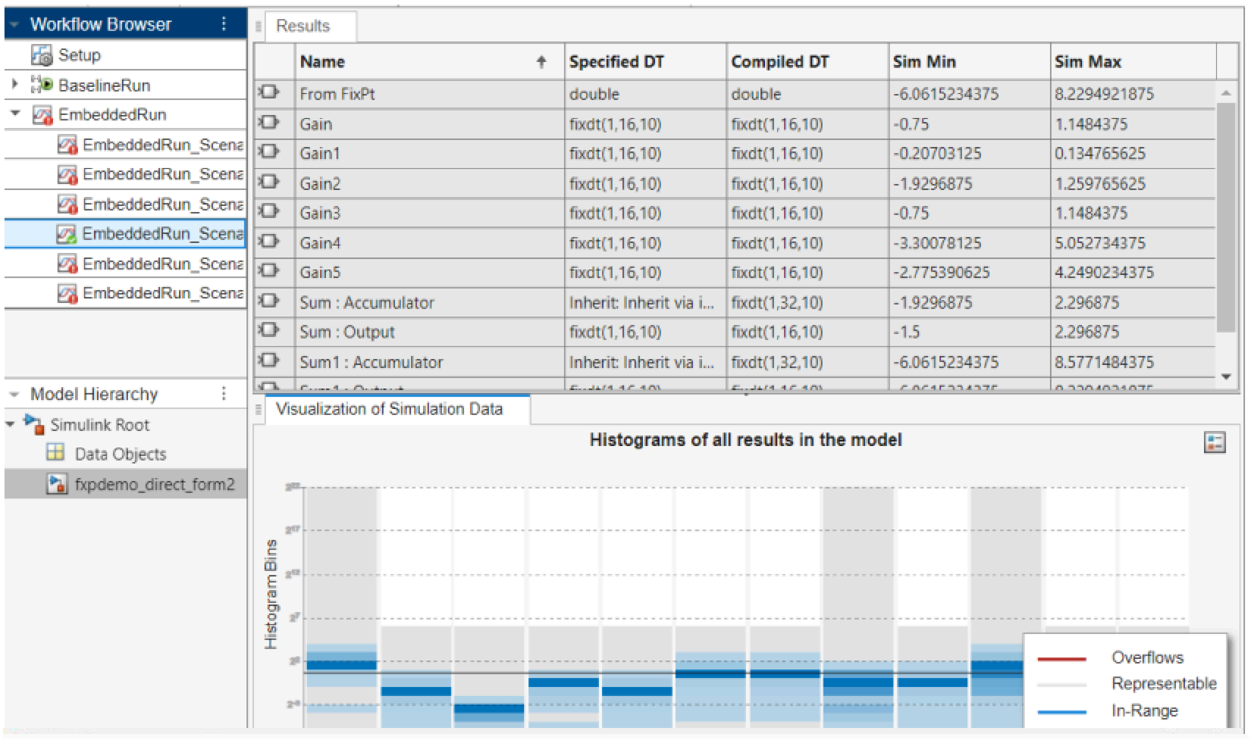
Visualization of Simulation Data



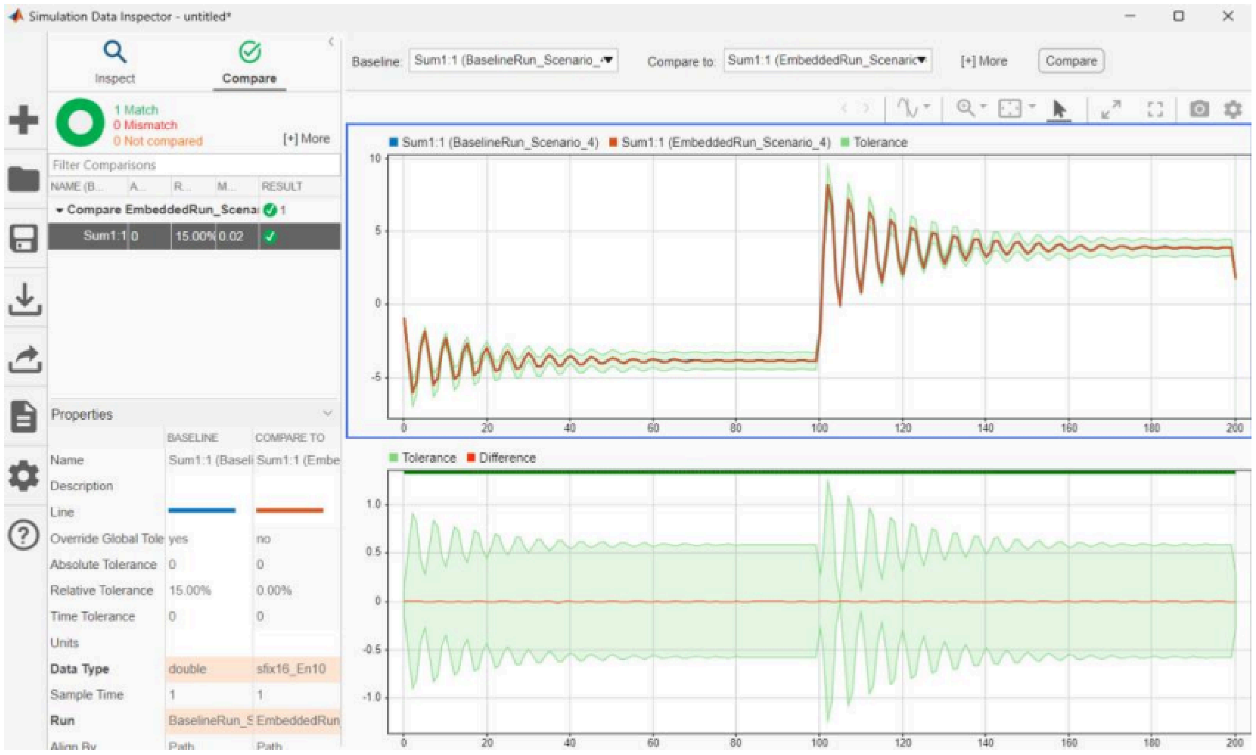
Name	Specified DT	Compiled DT	Sim Min	Sim Max
From FixPt	double	double	-606	823.4057374709746
Gain	fixdt(1,16,10)	double	-75	114.87944438716877
Gain1	fixdt(1,16,10)	double	-20.67829998969038	13.5
Gain2	fixdt(1,16,10)	double	-192.99746657044...	126
Gain3	fixdt(1,16,10)	double	-75	114.87944438716877
Gain4	fixdt(1,16,10)	double	-330	505.4695553035427
Gain5	fixdt(1,16,10)	double	-277.5	425.0539442325245
Sum : Accumulator	Inherit: Inherit via i...	double	-192.99746657044...	229.75888877433755
Sum : Output	fixdt(1,16,10)	double	-150	229.75888877433755
Sum1 : Accumulator	Inherit: Inherit via i...	double	-606	858.1419352251326

4. Displayed Screen Image:

As can be seen, only scenario number 4 has produced the desired response, which is as follows:



Name	Specified DT	Compiled DT	Sim Min	Sim Max
From FixPt	double	double	-6.0615234375	8.2294921875
Gain	fixdt(1,16,10)	fixdt(1,16,10)	-0.75	1.1484375
Gain1	fixdt(1,16,10)	fixdt(1,16,10)	-0.20703125	0.134765625
Gain2	fixdt(1,16,10)	fixdt(1,16,10)	-1.9296875	1.259765625
Gain3	fixdt(1,16,10)	fixdt(1,16,10)	-0.75	1.1484375
Gain4	fixdt(1,16,10)	fixdt(1,16,10)	-3.30078125	5.052734375
Gain5	fixdt(1,16,10)	fixdt(1,16,10)	-2.775390625	4.2490234375
Sum : Accumulator	Inherit: Inherit via i...	fixdt(1,32,10)	-1.9296875	2.296875
Sum : Output	fixdt(1,16,10)	fixdt(1,16,10)	-1.5	2.296875
Sum1 : Accumulator	Inherit: Inherit via i...	fixdt(1,32,10)	-6.0615234375	8.5771484375


Simulation Data Inspector - untitled*

Baseline: Sum1:1 (BaselineRun_Scenario_4)
Compare to: Sum1:1 (EmbeddedRun_Scenario_4)

1 Match
0 Mismatch
0 Not compared

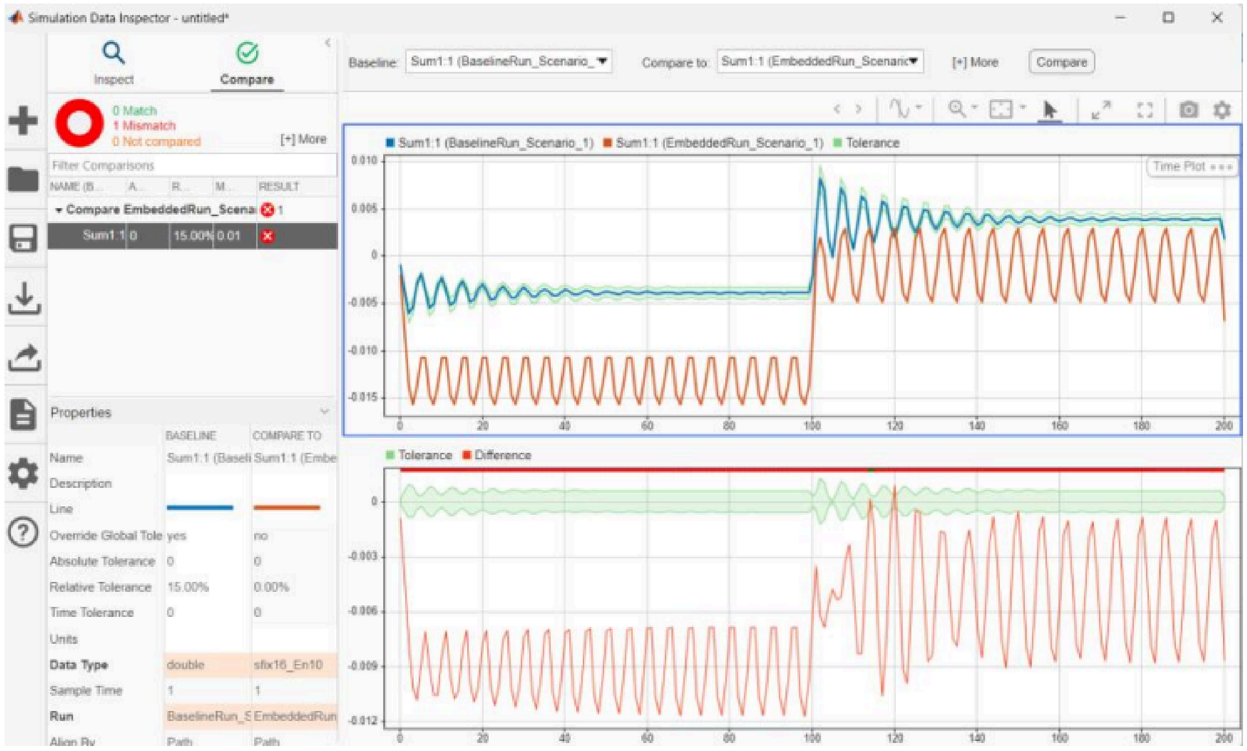
NAME (B...	A...	R...	M...	RESULT
Compare EmbeddedRun_Scena				✓ 1
Sum1:1	0	15.00%	0.02	✓

Properties

	BASELINE	COMPARE TO
Name	Sum1:1 (Baseli	Sum1:1 (Embe
Description		
Line		
Override Global Tole	yes	no
Absolute Tolerance	0	0
Relative Tolerance	15.00%	0.00%
Time Tolerance	0	0
Units		
Data Type	double	sfix16_En10
Sample Time	1	1
Run	BaselineRun_S	EmbeddedRun
Align By	Path	Path

Inspect
Compare
[+] More
Filter Comparisons
Sum1:1 (BaselineRun_Scenario_4)
Sum1:1 (EmbeddedRun_Scenario_4)
Tolerance
Difference

5. Step 4 for Scenario 1 and Scenario 6:


Simulation Data Inspector - untitled*

Baseline: Sum1:1 (BaselineRun_Scenario_1)
Compare to: Sum1:1 (EmbeddedRun_Scenario_1)

0 Match
1 Mismatch
0 Not compared

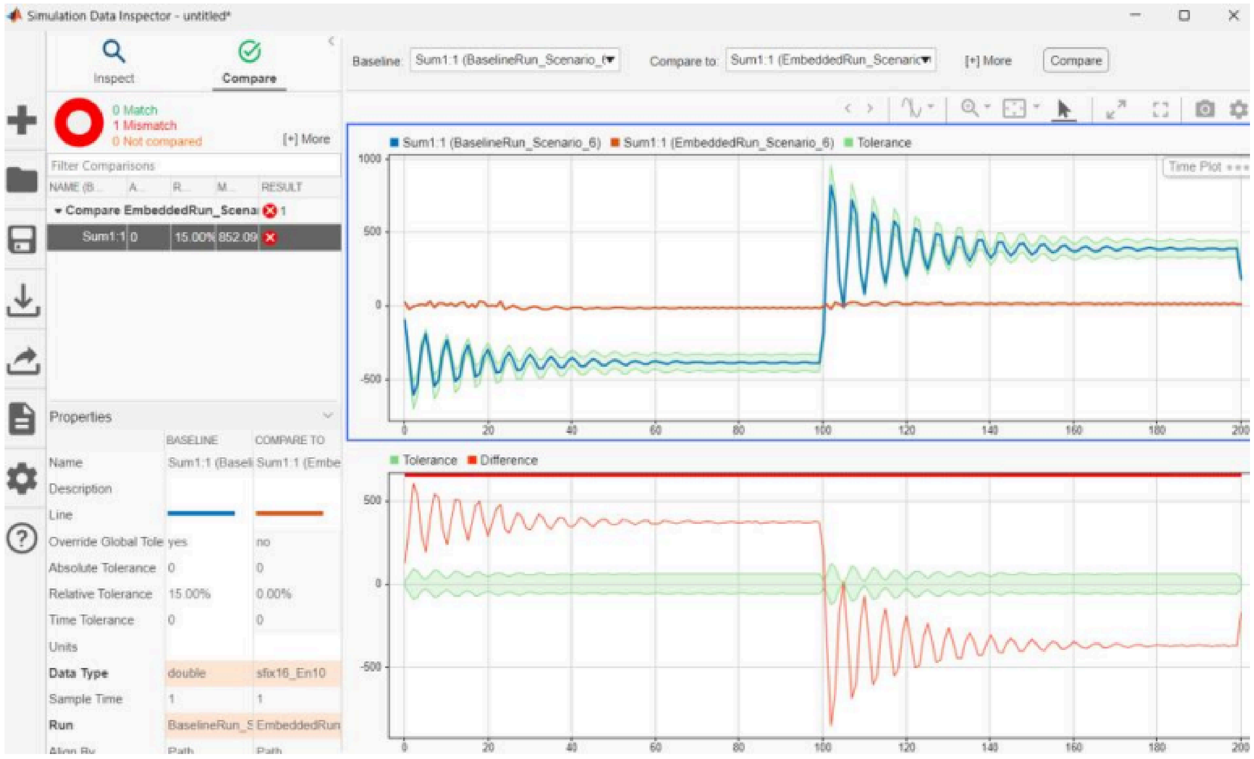
NAME (B...	A...	R...	M...	RESULT
Compare EmbeddedRun_Scena				✗ 1
Sum1:1	0	15.00%	0.01	✗

Properties

	BASELINE	COMPARE TO
Name	Sum1:1 (Baseli	Sum1:1 (Embe
Description		
Line		
Override Global Tole	yes	no
Absolute Tolerance	0	0
Relative Tolerance	15.00%	0.00%
Time Tolerance	0	0
Units		
Data Type	double	sfix16_En10
Sample Time	1	1
Run	BaselineRun_S	EmbeddedRun
Align By	Path	Path

Scenario 1:

In this case, the output has experienced underflow, which is why, as shown in the image, the output is not placed in the desired and displayable section


Simulation Data Inspector - untitled*
Inspect
Compare
Baseline: Sum1:1 (BaselineRun_Scenario_6)
Compare to: Sum1:1 (EmbeddedRun_Scenario_6)
[+] More
Compare
Sum1:1 (BaselineRun_Scenario_6)
Sum1:1 (EmbeddedRun_Scenario_6)
Tolerance
Difference
Time Plot

0 Match
1 Mismatch
0 Not compared
[+] More

Filter Comparisons

NAME (B...	A...	R...	M...	RESULT
▾ Compare EmbeddedRun_Scena				✖ 1
Sum1:1	0	15.00%	852.09	✖

Properties

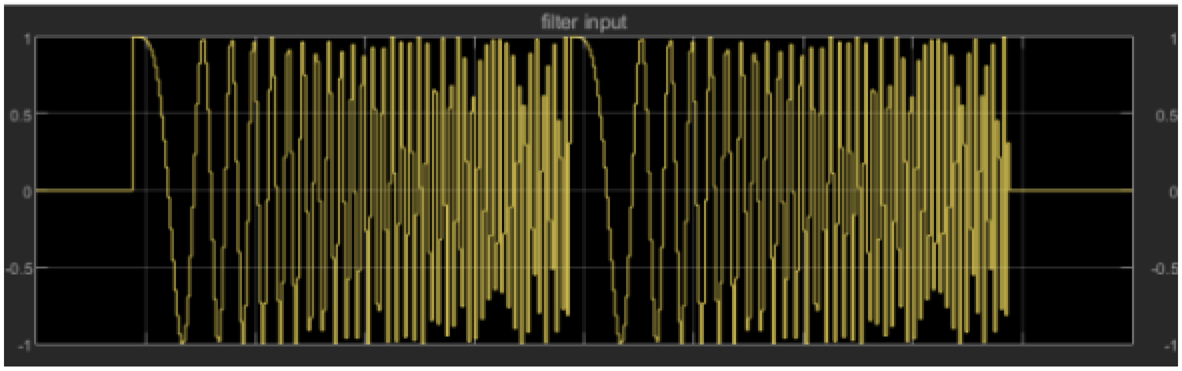
	BASELINE	COMPARE TO
Name	Sum1:1 (Baseli	Sum1:1 (Embe
Description		
Line	🟦	🟧
Override Global Tole	yes	no
Absolute Tolerance	0	0
Relative Tolerance	15.00%	0.00%
Time Tolerance	0	0
Units		
Data Type	double	sfix16_En10
Sample Time	1	1
Run	BaselineRun_S	EmbeddedRun
Align By	Path	Path

:Scenario 6

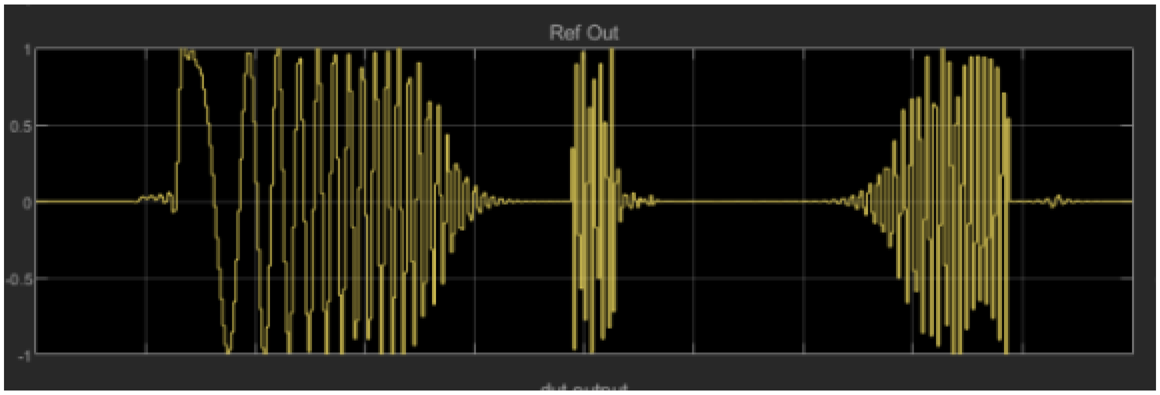
In this case, the system has experienced overflow, which is why, as shown in the image, the output is not placed in the desired and displayable section

Part2:

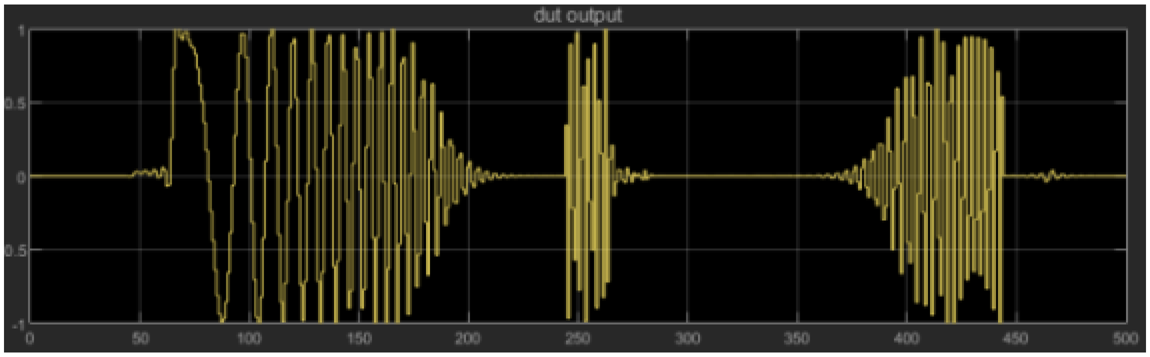
1.



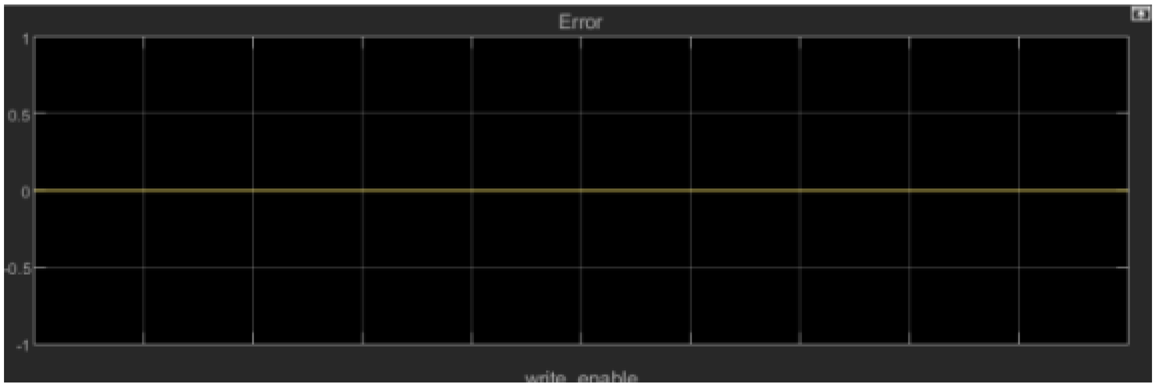
This shows the input signal



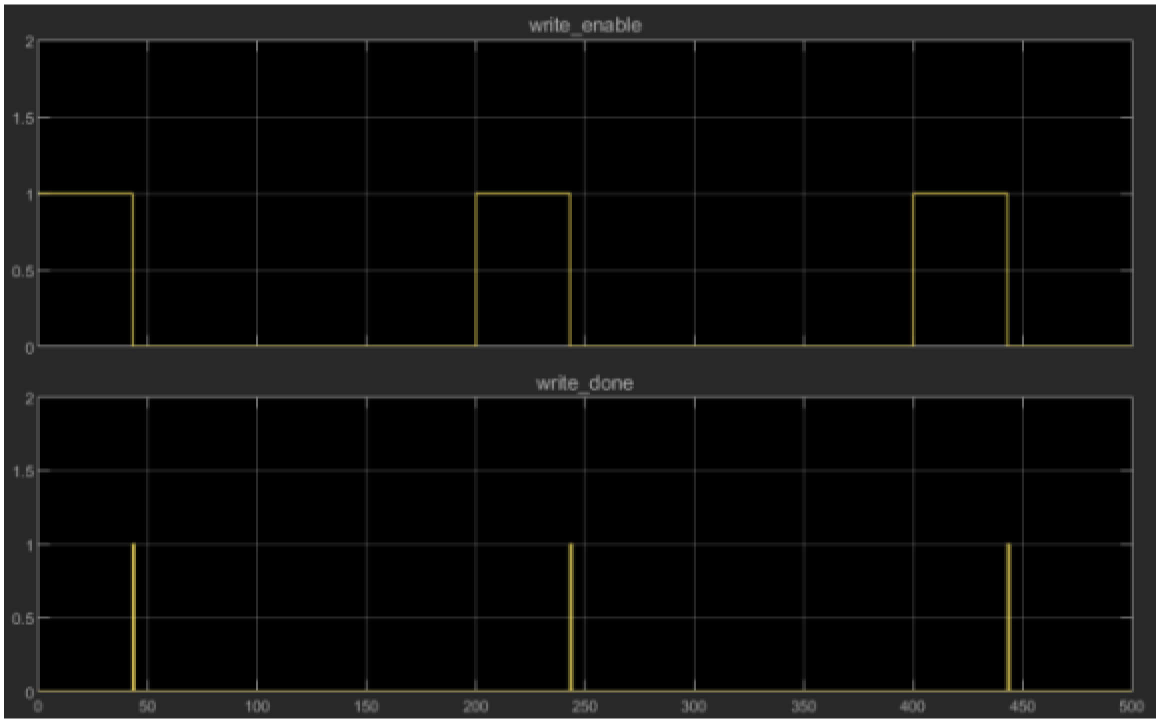
This shows the ideal filter output



This is the output signal of the circuit

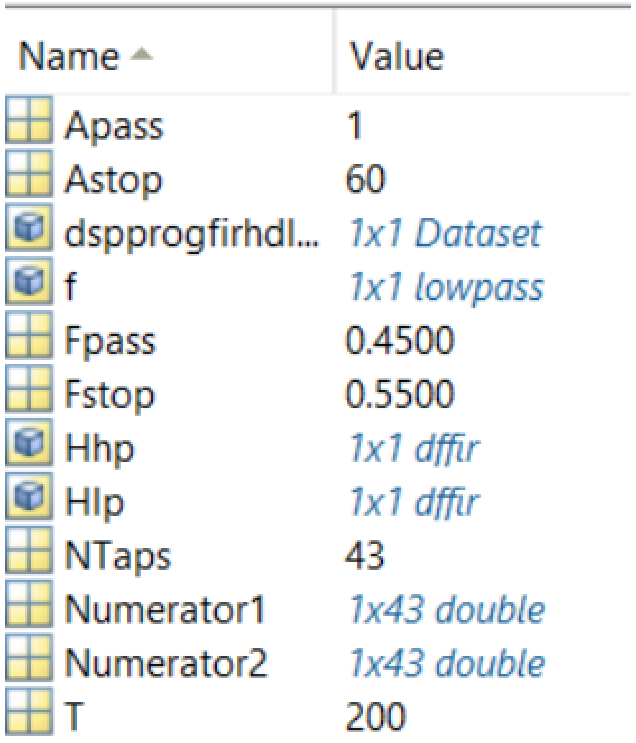


Error is the difference between Ref out and dut out



These two are control signals of the circuit

2.

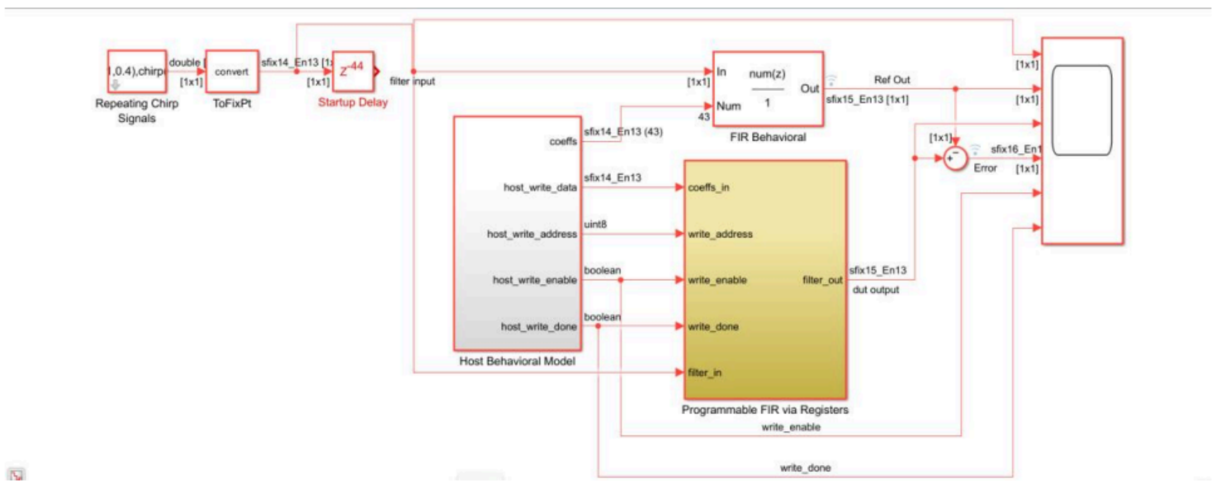


Name	Value
Apass	1
Astop	60
dspprogfirhdl...	*1x1 Dataset*
f	*1x1 lowpass*
Fpass	0.4500
Fstop	0.5500
Hhp	*1x1 dffir*
Hlp	*1x1 dffir*
NTaps	43
Numerator1	*1x43 double*
Numerator2	*1x43 double*
T	200

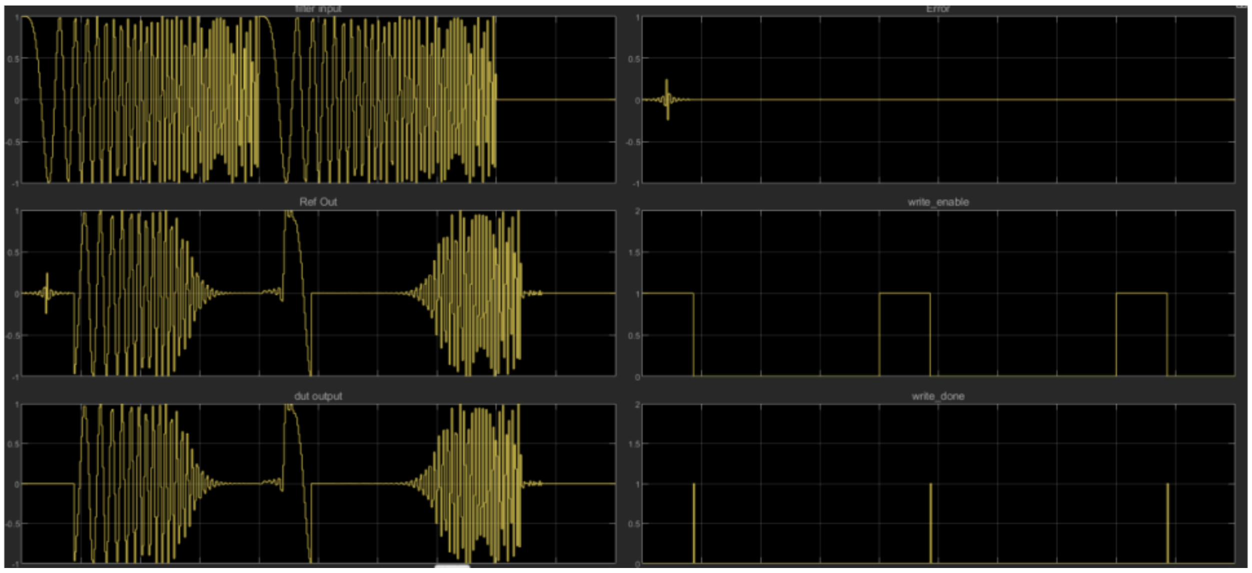
We can assume that the filter has 43 coeffs based on Numerator 1 and Numerator 2.

3:

The circuit if we bypass delay:



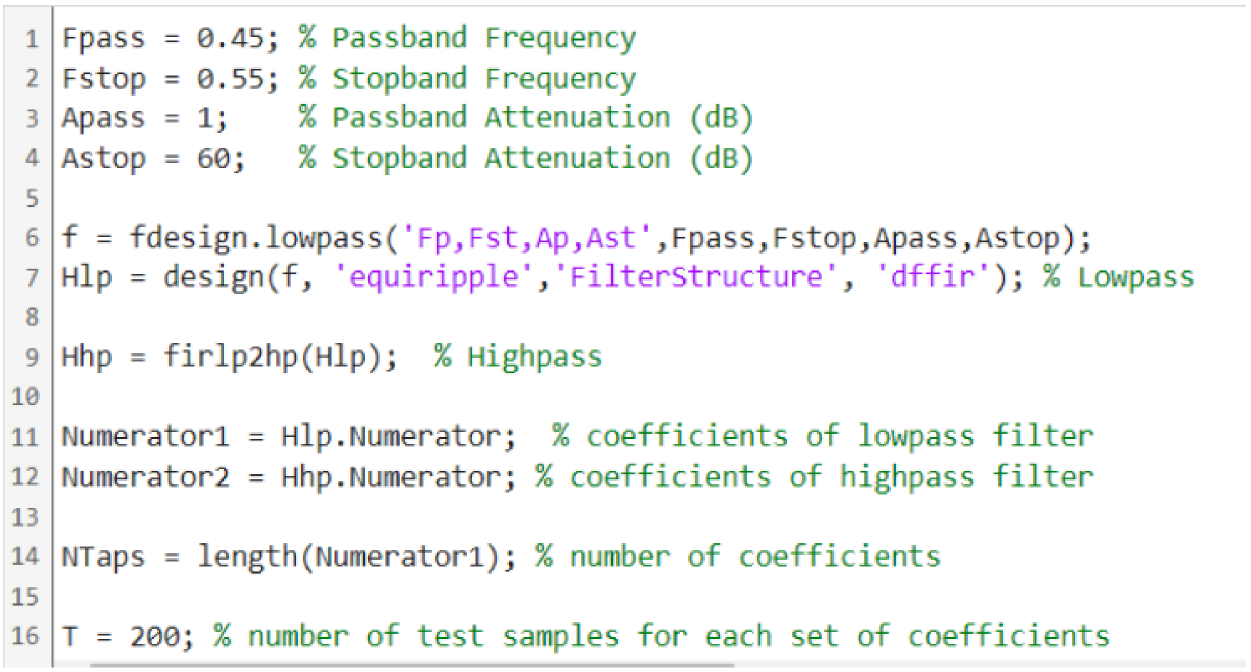
the output waveform:



The change in this case, compared to the scenario where there is a delay unit, is the non-zero error at the beginning of the simulation. This is due to the unstable and incorrect output of the block. The reason for this is that both blocks require full loading of all coefficients. The delay unit used introduces a 44-cycle delay in the start of the operation to ensure the coefficients have enough time to load. In the implemented block in the system, using registers and considering that we have two control signals, the output is not generated until the signal becomes equal to 1.

Therefore, we don't have an incorrect output, and the error that occurs is because the output is incorrect.

4:



```
Fpass = 0.45; % Passband Frequency
Fstop = 0.55; % Stopband Frequency
Apass = 1;    % Passband Attenuation (dB)
Astop = 60;   % Stopband Attenuation (dB)

f = fdesign.lowpass('Fp,Fst,Ap,Ast',Fpass,Fstop,Apass,Astop);
Hlp = design(f, 'equiripple','FilterStructure', 'dffir'); % Lowpass

Hhp = firlp2hp(Hlp);  % Highpass

Numerator1 = Hlp.Numerator;  % coefficients of lowpass filter
Numerator2 = Hhp.Numerator; % coefficients of highpass filter

NTaps = length(Numerator1); % number of coefficients

T = 200; % number of test samples for each set of coefficients
```

In digital signal processing, when designing filters, it is common practice to normalize the frequencies in relation to the Nyquist frequency, which is half the sampling frequency (**Fs/2**). This ensures that the filter design remains consistent regardless of the specific sampling rate used in the system. In this study, we are working with the human hearing range, with a passband frequency (**Fpass**) of 80 Hz and a stopband frequency (**Fstop**) of 8 kHz.

Step-by-Step Frequency Normalization

1. **Define the Sampling Frequency (Fs):**
   The sampling frequency for this system is assumed to be **44,100 Hz**, which is standard in many audio applications (e.g., CD-quality audio). This corresponds to a Nyquist frequency of:

Passband and Stopband Frequencies:

- The **passband frequency** (**Fpass**) is set at **80 Hz**, which is the minimum frequency to pass through the filter.

- The **stopband frequency** (**Fstop**) is set at **8,000 Hz (8 kHz)**, which is the maximum frequency to be attenuated by the filter.

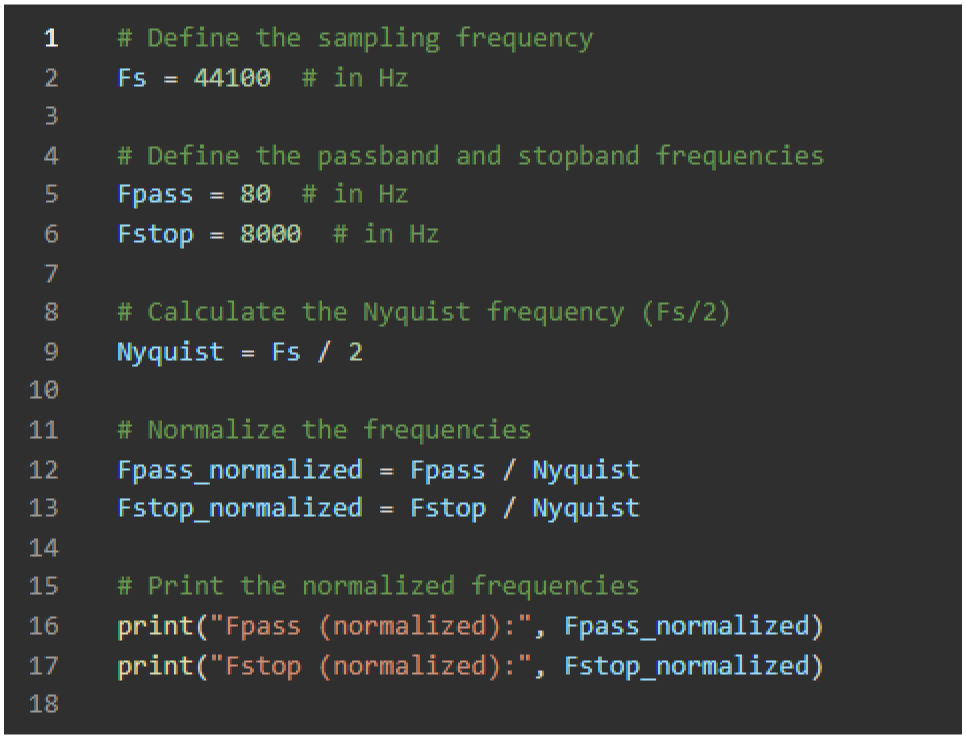
Normalization of Frequencies:
To normalize the passband and stopband frequencies, we divide each frequency by the Nyquist frequency

These normalized values represent the passband and stopband frequencies in terms of the Nyquist frequency. In a digital filter design, these normalized frequencies are used as inputs to algorithms that calculate filter coefficients.

Importance of Normalization:

Normalization is crucial in digital filter design as it allows the filter's behavior to remain consistent, regardless of the sampling rate. It ensures that the filter's performance is based on relative frequencies, making it easier to apply the same filter design principles across different systems with varying sampling rates. This process also avoids issues that can arise from frequency aliasing and ensures accurate representation of the desired frequency ranges.

Python code to automate the process:



```python
# Define the sampling frequency
Fs = 44100  # in Hz

# Define the passband and stopband frequencies
Fpass = 80  # in Hz
Fstop = 8000  # in Hz

# Calculate the Nyquist frequency (Fs/2)
Nyquist = Fs / 2

# Normalize the frequencies
Fpass_normalized = Fpass / Nyquist
Fstop_normalized = Fstop / Nyquist

# Print the normalized frequencies
print("Fpass (normalized):", Fpass_normalized)
print("Fstop (normalized):", Fstop_normalized)

```

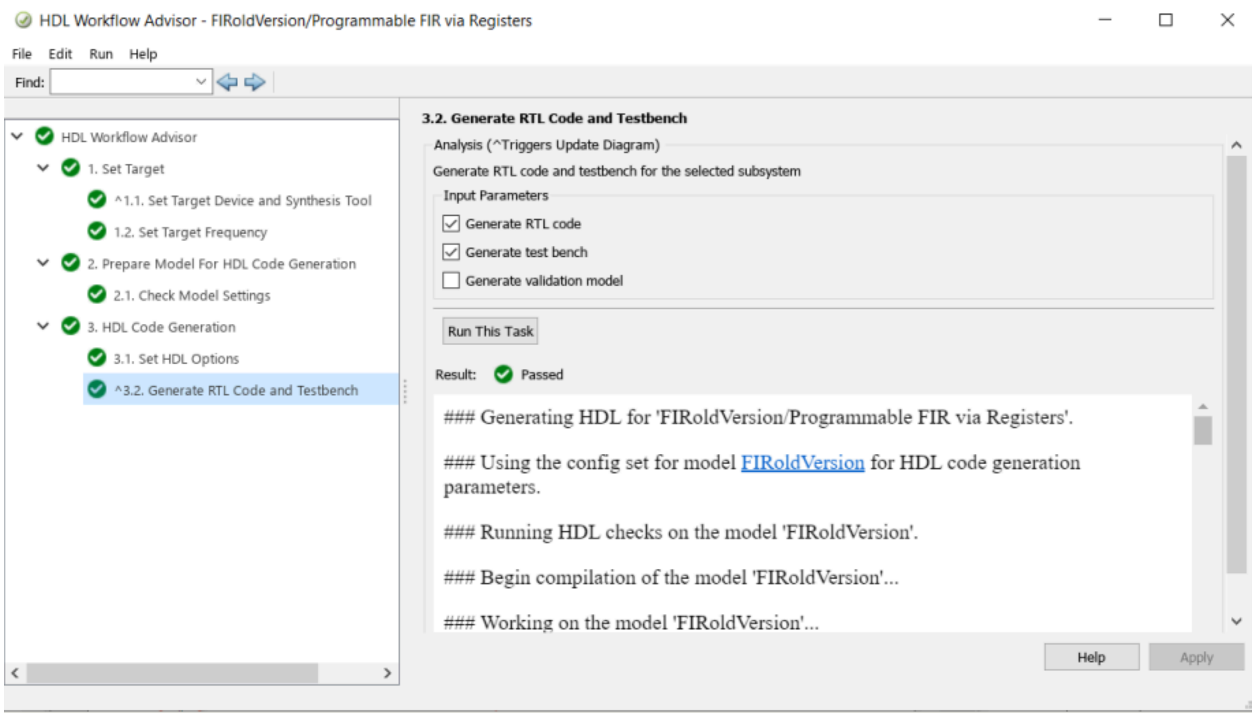
```
Fpass (normalized): 0.0036271221744613O8 Fstop (normalized):
0.3634251097954708
```

Updated MATLAB code:

```matlab
Fpass = 0.0036; % Passband Frequency  %80/(44100/2)
Fstop = 0.36; % Stopband Frequency  %8000/(44100/2)
Apass = 1;    % Passband Attenuation (dB)
Astop = 60;   % Stopband Attenuation (dB)

f = fdesign.lowpass('Fp,Fst,Ap,Ast',Fpass,Fstop,Apass,Astop);
Hlp = design(f, 'equiripple','FilterStructure', 'dffir'); % Lowpass

Hhp = firlp2hp(Hlp);  % Highpass

Numerator1 = Hlp.Numerator;  % coefficients of lowpass filter
Numerator2 = Hhp.Numerator; % coefficients of highpass filter

NTaps = length(Numerator1); % number of coefficients

T = 200; % number of test samples for each set of coefficients
```

5:

Compiled successfully:


HDL Workflow Advisor - FIRoldVersion/Programmable FIR via Registers
File
Edit
Run
Help
Find:
HDL Workflow Advisor
1. Set Target
^1.1. Set Target Device and Synthesis Tool
1.2. Set Target Frequency
2. Prepare Model For HDL Code Generation
2.1. Check Model Settings
3. HDL Code Generation
3.1. Set HDL Options
^3.2. Generate RTL Code and Testbench
Run This Task
Help
Apply

3.2. Generate RTL Code and Testbench

Analysis (^Triggers Update Diagram)

Generate RTL code and testbench for the selected subsystem

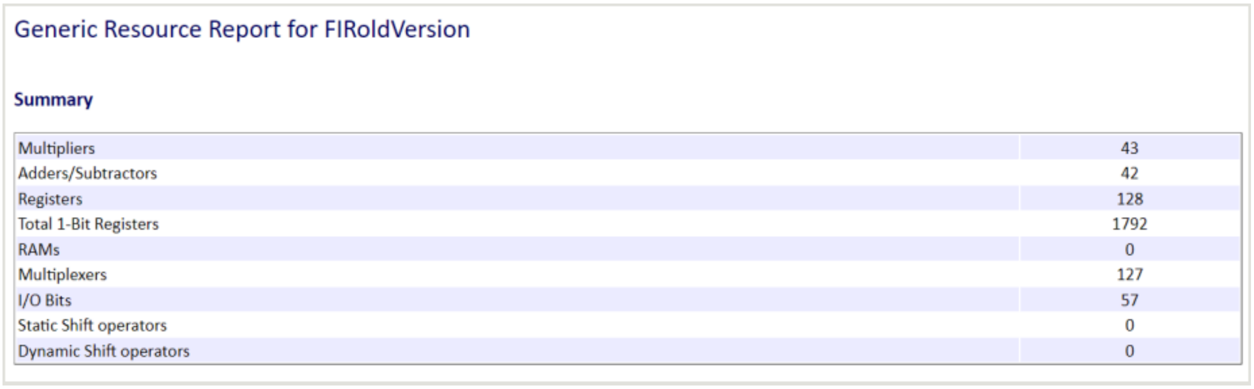
Input Parameters

- [✓] Generate RTL code
- [✓] Generate test bench
- [ ] Generate validation model

Result: Passed

```
### Generating HDL for 'FIRoldVersion/Programmable FIR via Registers'.
### Using the config set for model FIRoldVersion for HDL code generation
parameters.
### Running HDL checks on the model 'FIRoldVersion'.
### Begin compilation of the model 'FIRoldVersion'...
### Working on the model 'FIRoldVersion'...
```

resources:

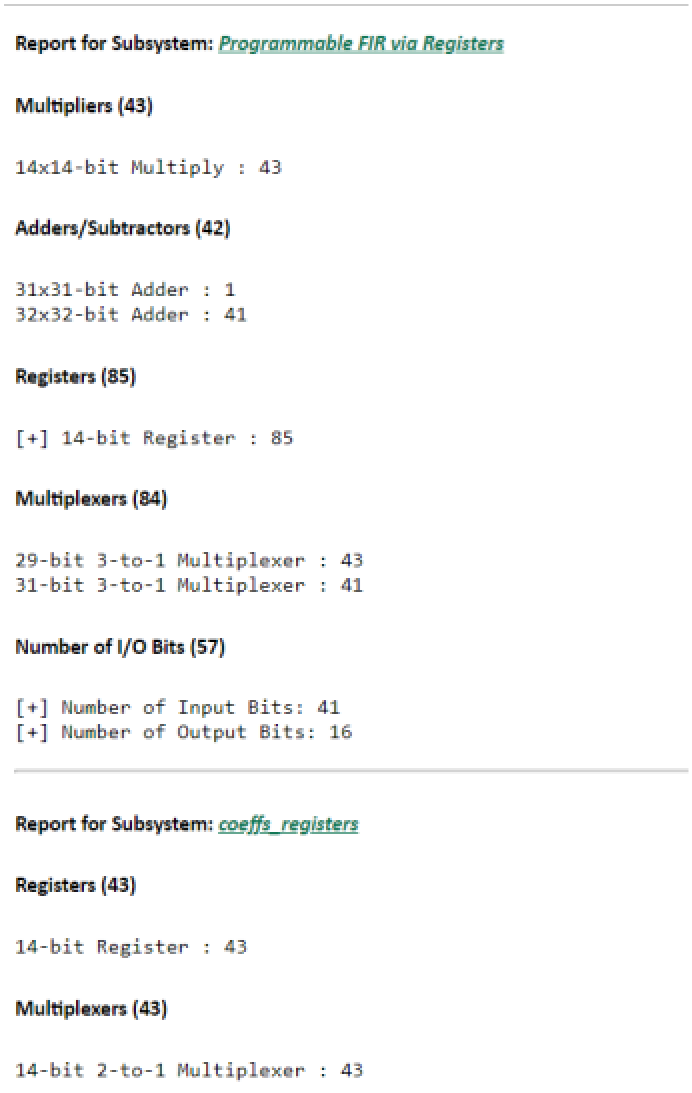


Generic Resource Report for FIRoldVersion

Summary

Multipliers	43
Adders/Subtractors	42
Registers	128
Total 1-Bit Registers	1792
RAMs	0
Multiplexers	127
I/O Bits	57
Static Shift operators	0
Dynamic Shift operators	0

Resources in detail:



Report for Subsystem: *Programmable FIR via Registers*

Multipliers (43)

```
14x14-bit Multiply : 43
```

Adders/Subtractors (42)

```
31x31-bit Adder : 1
32x32-bit Adder : 41
```

Registers (85)

```
[+] 14-bit Register : 85
```

Multiplexers (84)

```
29-bit 3-to-1 Multiplexer : 43
31-bit 3-to-1 Multiplexer : 41
```

Number of I/O Bits (57)

```
[+] Number of Input Bits: 41
[+] Number of Output Bits: 16
```

Report for Subsystem: *coeffs_registers*

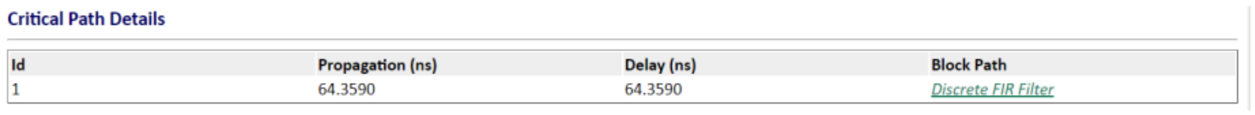
Registers (43)

```
14-bit Register : 43
```

Multiplexers (43)

```
14-bit 2-to-1 Multiplexer : 43
```

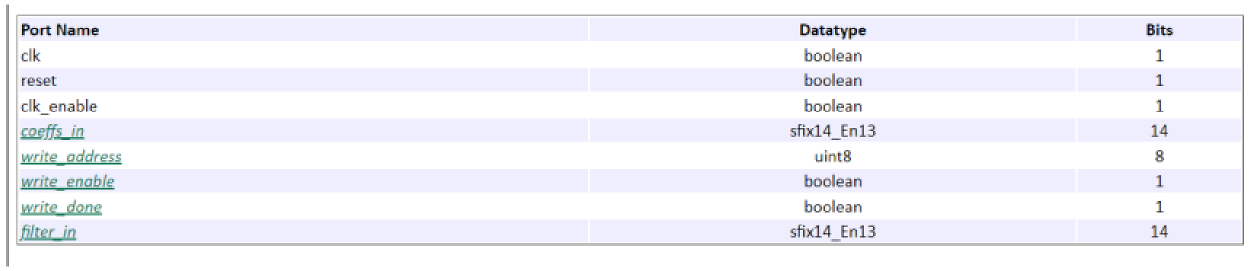
Critical path:



Critical Path Details

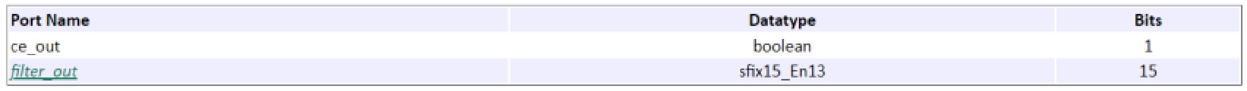
Id	Propagation (ns)	Delay (ns)	Block Path
1	64.3590	64.3590	*Discrete FIR Filter*

Input ports:



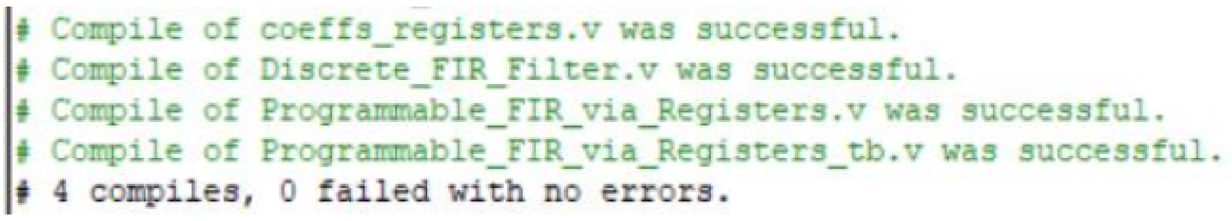
Port Name	Datatype	Bits
clk	boolean	1
reset	boolean	1
clk_enable	boolean	1
coeffs_in	sfix14_En13	14
write_address	uint8	8
write_enable	boolean	1
write_done	boolean	1
filter_in	sfix14_En13	14

Output ports:



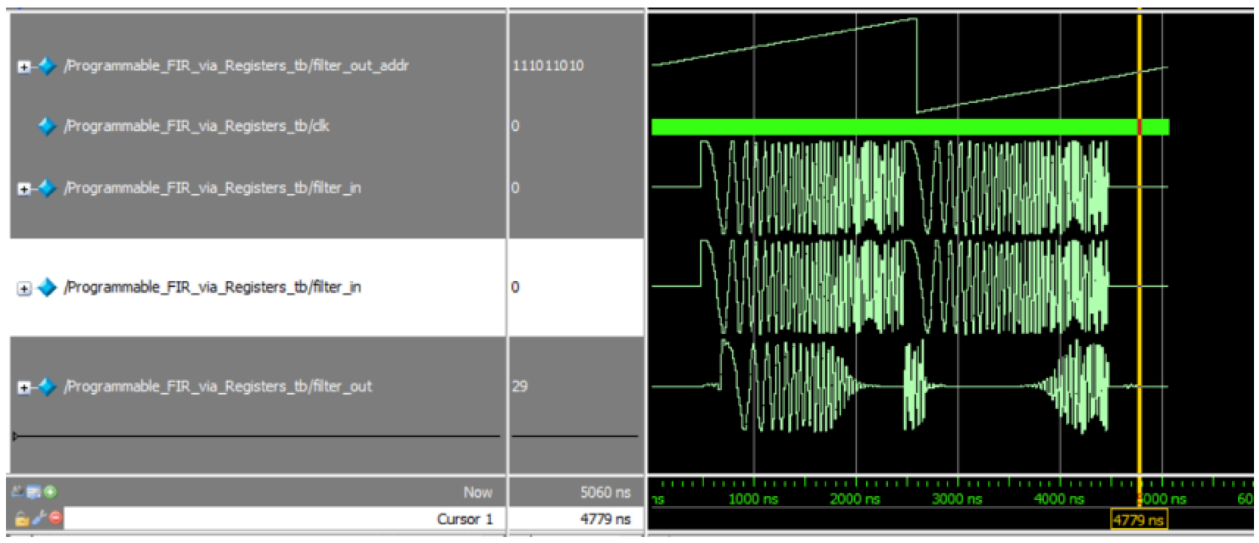
Port Name	Datatype	Bits
ce_out	boolean	1
filter_out	sfix15_En13	15

Modelsim verification:



```
# Compile of coeffs_registers.v was successful.
# Compile of Discrete_FIR_Filter.v was successful.
# Compile of Programmable_FIR_via_Registers.v was successful.
# Compile of Programmable_FIR_via_Registers_tb.v was successful.
# 4 compiles, 0 failed with no errors.
```

Output waveform:


/Programmable_FIR_via_Registers_tb/filter_out_addr
111011010
/Programmable_FIR_via_Registers_tb/clk
0
/Programmable_FIR_via_Registers_tb/filter_in
0
/Programmable_FIR_via_Registers_tb/filter_in
0
/Programmable_FIR_via_Registers_tb/filter_out
29
Now
5060 ns
Cursor 1
4779 ns
1000 ns
2000 ns
3000 ns
4000 ns
5000 ns
4779 ns

As we expected the output is filtered and it matches the simulink output.