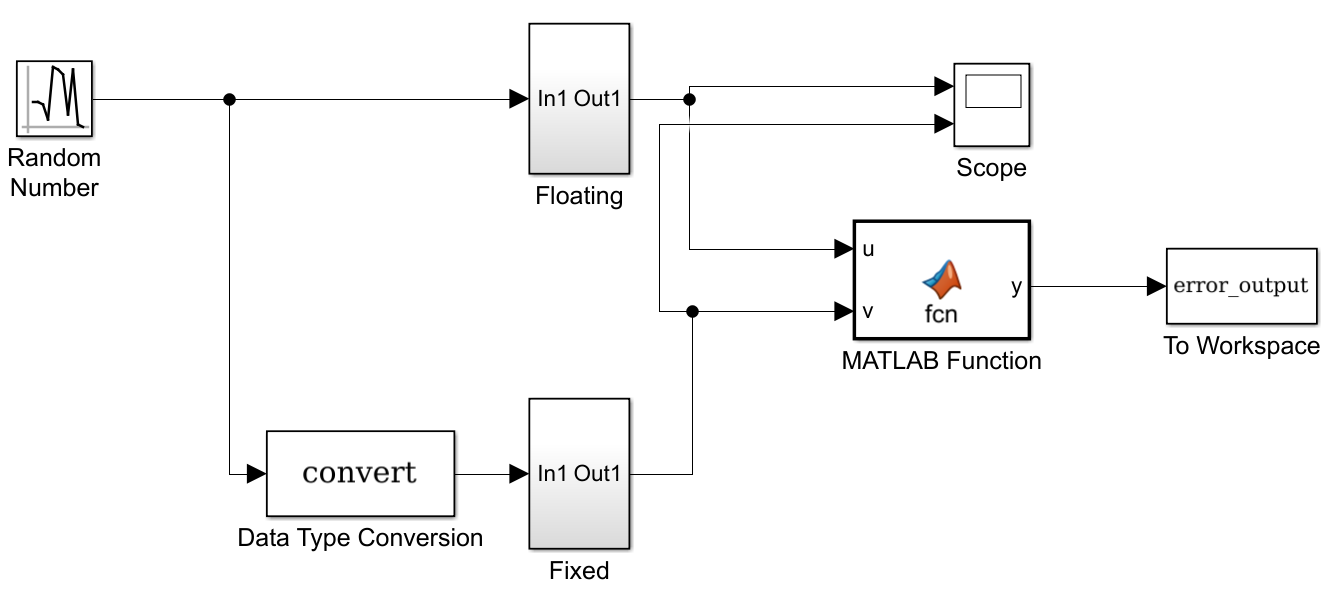
Homework:

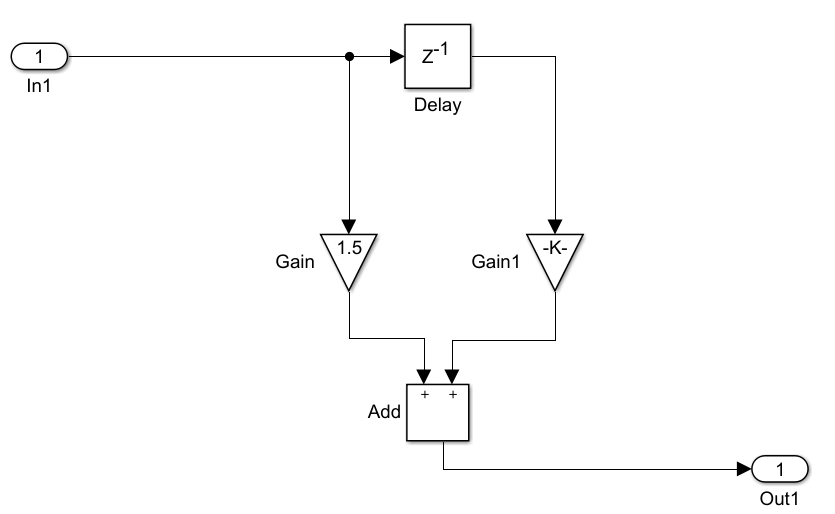
* Simulate the 2-tap filter with Matlab codes.
* Compare the result with that of the fixed-pointed.
* Adjust the bit width and the decimal point position such that the number of bits used in operations is minimized while the result is similar to the floating point.
* Exam the generated VHDL codes and see how the truncation and quantization and conducted.
* Generate the testbench and use Vivado to conduct simulations.

1. 2-tap filter system in matlab with floating point variant and fixed point variant:



The matlab function block calculates the error of the fixed point implementation.

Inside the filter blocks:



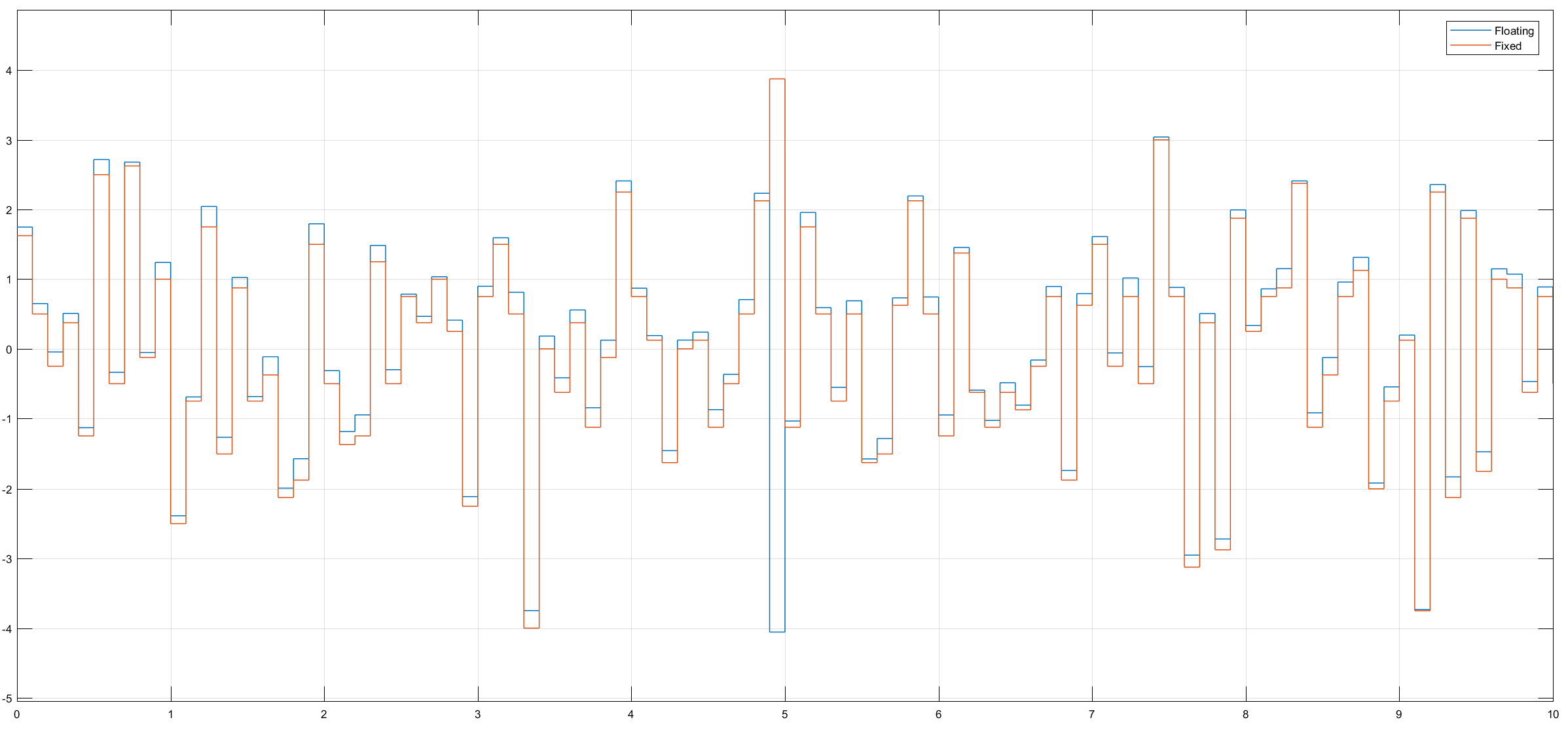
Gain = 1.5 (fixdt(0,2,1))

Gain1 = -0.25 (fixdt(1,3,2))

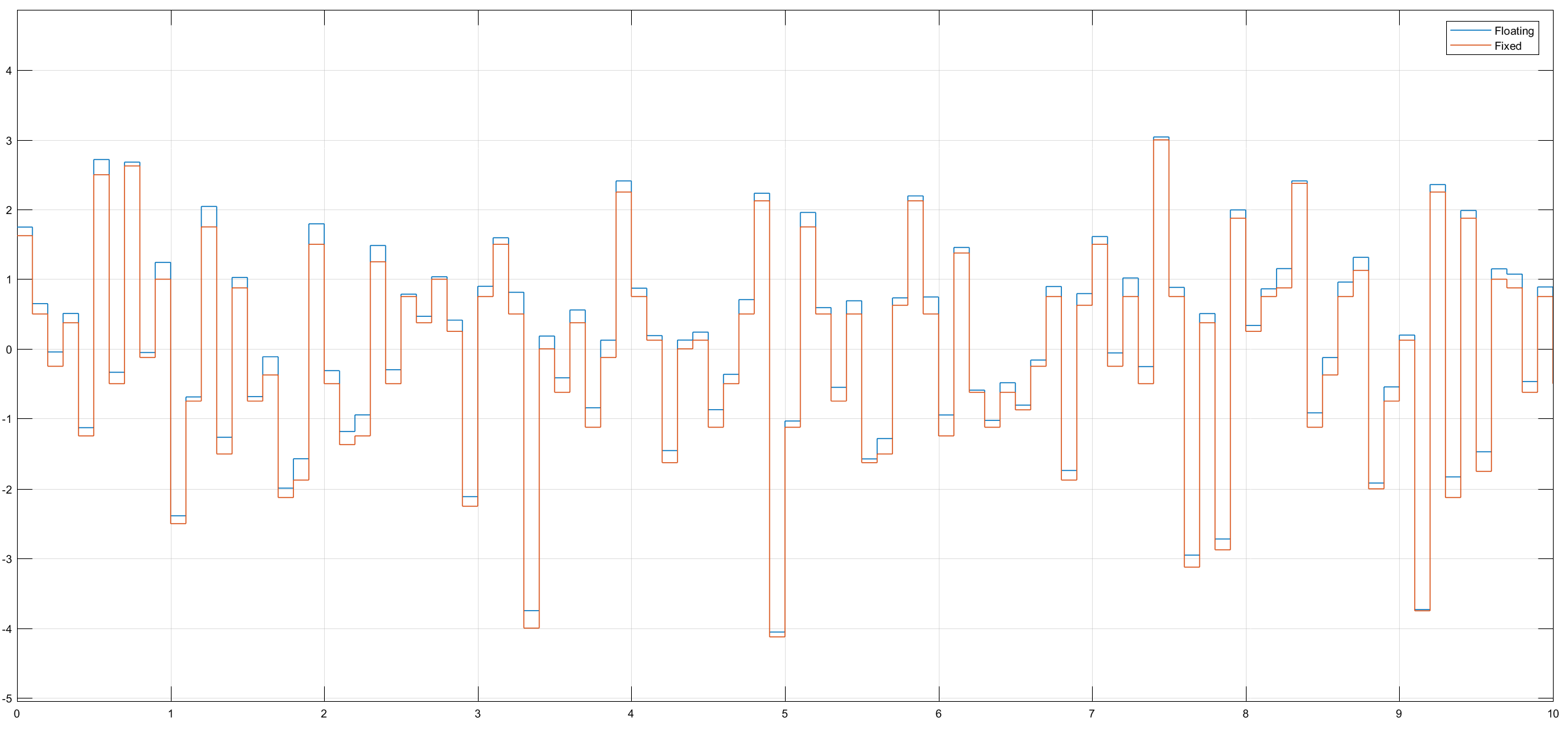
The format chosen for the parameters is enough to represent the numbers without precision loss.

1. Comparison:

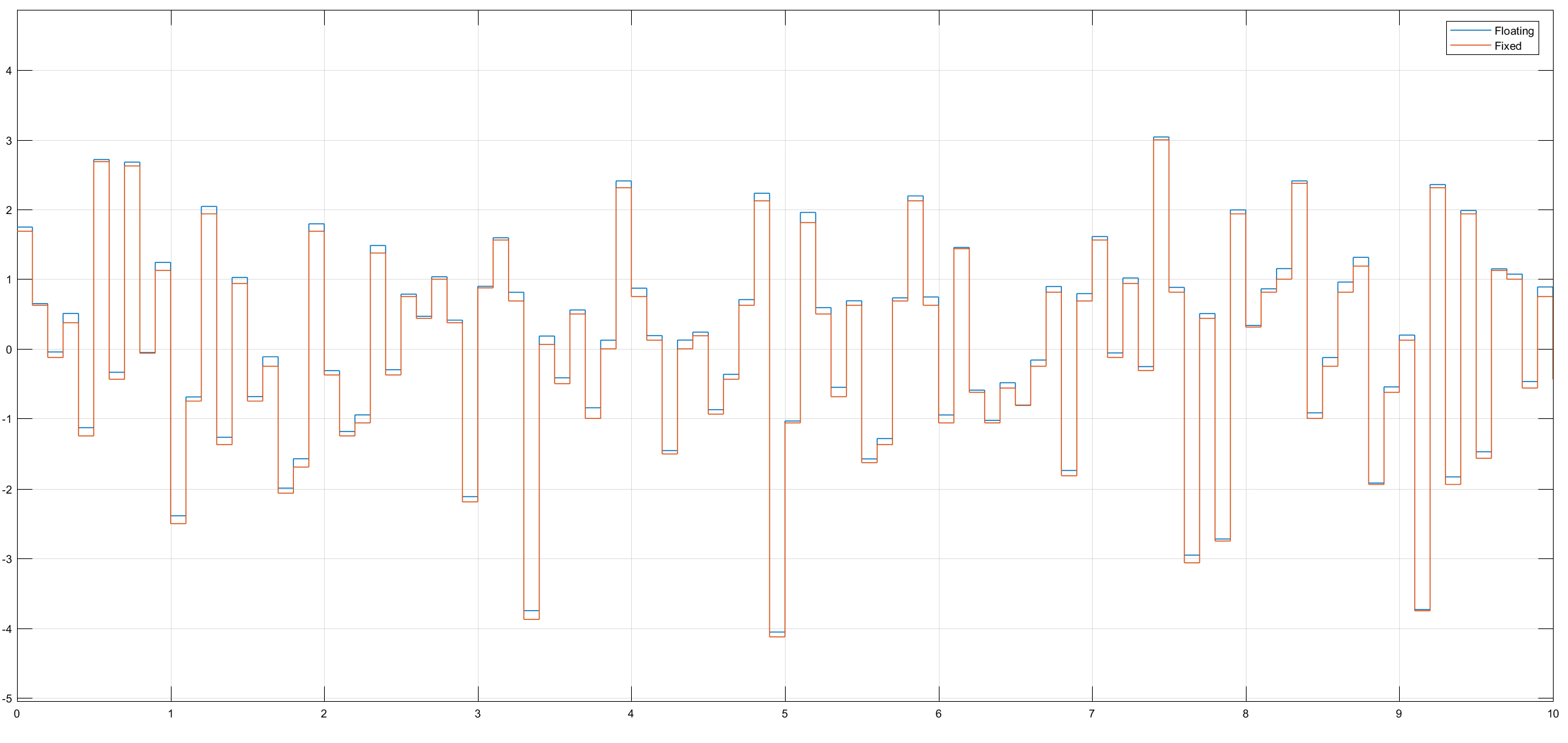
For fixed point format of fixdt(1,6,3):



For fixed point format of fixdt(1,7,3):

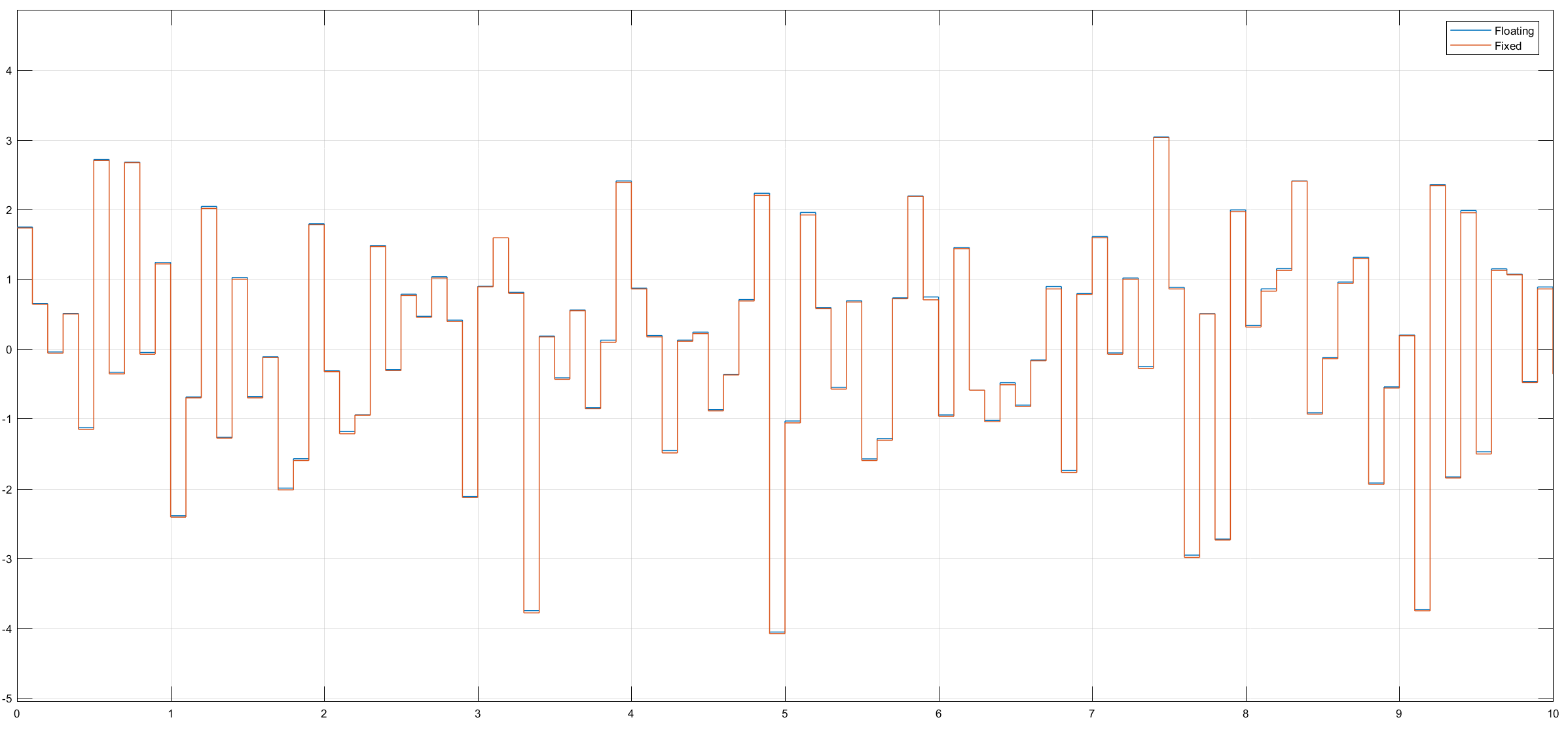


For fixed point format of fixdt(1,8,4):



As the input is between -4 and 4, 4 integer bits are needed to represent the whole interval accurately. The precision can then be adjusted by adding fractional bits.

1. Fixdt of (1,10,6) is very similar to the floating point implementation:



1. Generated VHDL code:
2. -- -------------------------------------------------------------
3. --
4. -- File Name: hdlsrc\TapFilter\Fixed.vhd
5. -- Created: 2018-10-21 10:40:11
6. --
7. -- Generated by MATLAB 9.2 and HDL Coder 3.10
8. --
9. --
10. -- -------------------------------------------------------------
11. -- Rate and Clocking Details
12. -- -------------------------------------------------------------
13. -- Model base rate: 0.1
14. -- Target subsystem base rate: 0.1
15. --
16. --
17. -- Clock Enable Sample Time
18. -- -------------------------------------------------------------
19. -- ce\_out 0.1
20. -- -------------------------------------------------------------
21. --
22. --
23. -- Output Signal Clock Enable Sample Time
24. -- -------------------------------------------------------------
25. -- Out1 ce\_out 0.1
26. -- -------------------------------------------------------------
27. --
28. -- -------------------------------------------------------------

31. -- -------------------------------------------------------------
32. --
33. -- Module: Fixed
34. -- Source Path: TapFilter/Fixed
35. -- Hierarchy Level: 0
36. --
37. -- -------------------------------------------------------------
38. LIBRARY IEEE;
39. USE IEEE.std\_logic\_1164.ALL;
40. USE IEEE.numeric\_std.ALL;
42. ENTITY Fixed IS
43. PORT( clk : IN std\_logic;
44. reset : IN std\_logic;
45. clk\_enable : IN std\_logic;
46. In1 : IN std\_logic\_vector(9 DOWNTO 0); -- sfix10\_En6
47. ce\_out : OUT std\_logic;
48. Out1 : OUT std\_logic\_vector(9 DOWNTO 0) -- sfix10\_En6
49. );
50. END Fixed;

53. ARCHITECTURE rtl OF Fixed IS
55. -- Signals
56. SIGNAL enb : std\_logic;
57. SIGNAL In1\_signed : signed(9 DOWNTO 0); -- sfix10\_En6
58. SIGNAL Gain\_mul\_temp : signed(12 DOWNTO 0); -- sfix13\_En7
59. SIGNAL Gain\_cast : signed(11 DOWNTO 0); -- sfix12\_En7
60. SIGNAL Gain\_out1 : signed(9 DOWNTO 0); -- sfix10\_En6
61. SIGNAL Delay\_out1 : signed(9 DOWNTO 0); -- sfix10\_En6
62. SIGNAL Gain1\_mul\_temp : signed(12 DOWNTO 0); -- sfix13\_En8
63. SIGNAL Gain1\_out1 : signed(9 DOWNTO 0); -- sfix10\_En6
64. SIGNAL Add\_add\_cast : signed(31 DOWNTO 0); -- sfix32\_En6
65. SIGNAL Add\_add\_cast\_1 : signed(31 DOWNTO 0); -- sfix32\_En6
66. SIGNAL Add\_add\_temp : signed(31 DOWNTO 0); -- sfix32\_En6
67. SIGNAL Add\_out1 : signed(9 DOWNTO 0); -- sfix10\_En6
69. BEGIN
70. In1\_signed <= signed(In1);
72. enb <= clk\_enable;
74. -- <S1>/Gain
75. Gain\_mul\_temp <= to\_signed(16#3#, 3) \* In1\_signed;
76. Gain\_cast <= Gain\_mul\_temp(11 DOWNTO 0);
77. Gain\_out1 <= Gain\_cast(10 DOWNTO 1);
79. -- <S1>/Delay
80. Delay\_process : PROCESS (clk, reset)
81. BEGIN
82. IF reset = '1' THEN
83. Delay\_out1 <= to\_signed(16#000#, 10);
84. ELSIF clk'EVENT AND clk = '1' THEN
85. IF enb = '1' THEN
86. Delay\_out1 <= In1\_signed;
87. END IF;
88. END IF;
89. END PROCESS Delay\_process;

92. -- <S1>/Gain1
93. Gain1\_mul\_temp <= to\_signed(-16#1#, 3) \* Delay\_out1;
94. Gain1\_out1 <= Gain1\_mul\_temp(11 DOWNTO 2);
96. -- <S1>/Add
97. Add\_add\_cast <= resize(Gain\_out1, 32);
98. Add\_add\_cast\_1 <= resize(Gain1\_out1, 32);
99. Add\_add\_temp <= Add\_add\_cast + Add\_add\_cast\_1;
100. Add\_out1 <= Add\_add\_temp(9 DOWNTO 0);
102. Out1 <= std\_logic\_vector(Add\_out1);
104. ce\_out <= clk\_enable;
106. END rtl;

Truncation and quantization is done in each Gain block (77-80 and 95-97) by casting to the right format and then truncating the result to fit the correct output size.

5) Vivado simulation

