Giaan Nguyen

Lab 1: FIR and IIR Filters

ECEN 689-600: FPGA Information Processing Systems
Wednesday, February 3, 2021

FIR Simulation

Note: to verify that I did the lab/simulation, I have included a signature within each screenshot at the lower right corner.

Table 1 shows the results that we would expect to get. We know that since there is not a physical representation of a decimal point, our signed values would be $2^4 = 16$ times the actual values. For instance, the input is $x[n] = \{-5, -4, \ldots, 0, 1, \ldots, 5\}$. In the simulation, we expect to see $x = 16x[n] = \{-80, -64, \ldots, 0, 16, \ldots, 80\}$. For 8 bits, we are limited to values [-128, 127]. The second column of Table 1 lists the output calculated by hand. When applying a left shift of 4 (or multiplying by 2^4) to both input and output, we see from the third and fourth columns that all values fall within [-128, 127].

Table 1. Calculations of output values for the FIR filter.

$x[n]$	$y[n] = 0.5x[n] - 1.5x[n-1] + 2.0x[n-2]$	$x[n] \times 2^4$	$y[n] \times 2^4$
-5	0.5*(-5) = -2.5	-80	-40
-4	0.5*(-4)-1.5*(-5) = 5.5	-64	88
-3	0.5*(-3)-1.5*(-4)+2.0*(-5) = -5.5	-48	-88
-2	-4.5	-32	-72
-1	-3.5	-16	-56
0	-2.5	0	-40
1	-1.5	16	-24
2	-0.5	32	-8
3	0.5	48	8
4	1.5	64	24
5	0.5*5-1.5*4+2.0*3 = 2.5	80	40
–	0.5*5-1.5*5+2.0*4 = 3.0	–	48
–	0.5*5-1.5*5+2.0*5 = 5.0	–	80

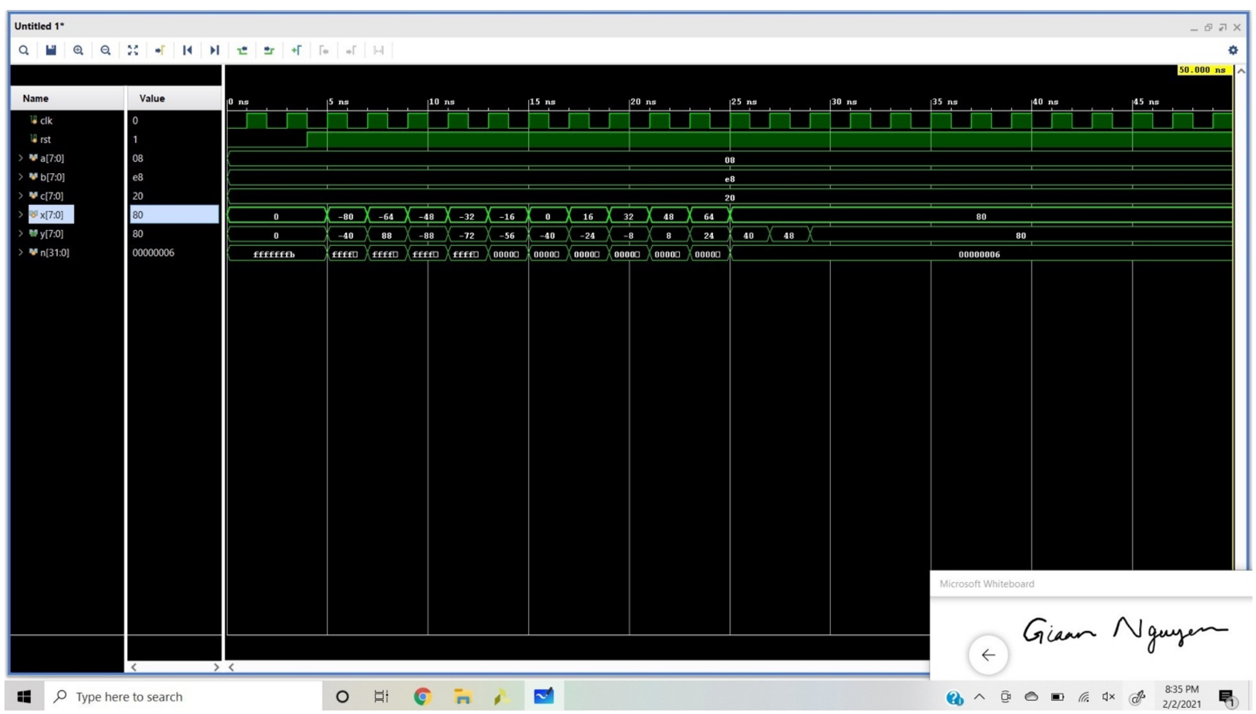

Figure 1. Simulation of the FIR filter.

Figure 1 shows the results from the simulation for the FIR filter. As expected, we see that $x = 16x[n]$. Cross-examining the waveforms from Figure 1 (for $x[7:0]$ and $y[7:0]$, which has displayed radix of signed decimal) to our expectations from Table 1 (columns 3 and 4), we see that the values confirm each other; that is, both our calculations and our design are valid. Additionally, the output y is changing on the same clock cycle as the input x. Therefore, our design works.

IIR Simulation

As with the FIR case, we now attempt to perform hand calculations for the IIR filter. However, from the first four rows in Table 2, we can already see that the outputs will exceed the range [-128, 127]. That is, there will be overflow.

Table 2. Condensed table of the first attempted calculations for the IIR filter.

$x[n]$	$y[n] = 0.5x[n] - 1.5x[n-1] + 2.0y[n-1] - 1.0y[n-2]$	$x[n] \times 2^4$	$y[n] \times 2^4$
-5	0.5*(-5) = -2.5	-80	-40
-4	0.5*(-4)-1.5*(-5)+2.0*(-2.5) = 0.5	-64	8
-3	0.5*(-3)-1.5*(-4)+2.0*(0.5)-1.0(-2.5) = 8.0	-48	128
-2	19.0	-32	304

Therefore, we will refer to Table 3 for verification, where any values exceeding the range [-128, 127] will be converted to binary, truncated to bits 7:0, and reconverted back to decimal. The Excel function DEC2BIN() is used to help with quick calculations. (Alternatively, applying mod 256 to values outside the range would also work.)

Table 3. Calculations of output values for the IIR filter with considerations for 8-bit length.

$x[n]$	$0.5x[n] \times 2^4$	$-1.5x[n-1] \times 2^4$	$2.0y_S[n-1]$ (truncated)	$-1.0y_S[n-2]$ (truncated)	$y_S[n]$ (no carry)	$y[n] = \frac{y_S[n]}{2^4}$
-5	-40	–	–	–	-40	-2.5
-4	-32	120	2*(-40) = -80	–	-32+120+(-80) = 8	0.5
-3	-24	96	2*8 = 16	-1*(-40) = 40	-24+96+16+40 = 128 (1000 0000) => -128	-8.0
-2	-16	72	2*(-128) = -256 (0000 0000) => 0	-1*8 = -8	-16+72+0+(-8) = 48	3.0
-1	-8	48	2*48 = 96	-1*-128 = 128 (1000 0000) => -128	-8+48+96+(-128) = 8	0.5
0	0	24	2*8 = 16	-1*48 = -48	-8	-0.5
1	8	0	2*(-8) = -16	-1*8 = -8	-16	-1.0
2	16	-24	2*(-16) = -32	-1*(-8) = 8	-32	-2.0
3	24	-48	2*(-32) = -64	-1*(-16) = 16	-72	-6.0
4	32	-72	2*(-72) = -144 (0111 0000) => 112	-1*(-32) = 32	32+(-72)+112+32 = 104	6.5
5	40	-96	2*104 = 208 (1101 0000) => -48	-1*(-72) = 72	40+(-96)+(-48)+72 = -32	-2.0

Figure 2 shows the simulation results for the IIR filter. As with the FIR filter, we expect x and y (with radix set to signed decimal) to be 16 times the actual values. Comparing x and y from Figure 2 to the first and penultimate columns in Table 3, we see that the values match up. This confirms that our hand calculations are correct and confirms the validity of our design, also demonstrated by the syncing of the input and output for each clock cycle.

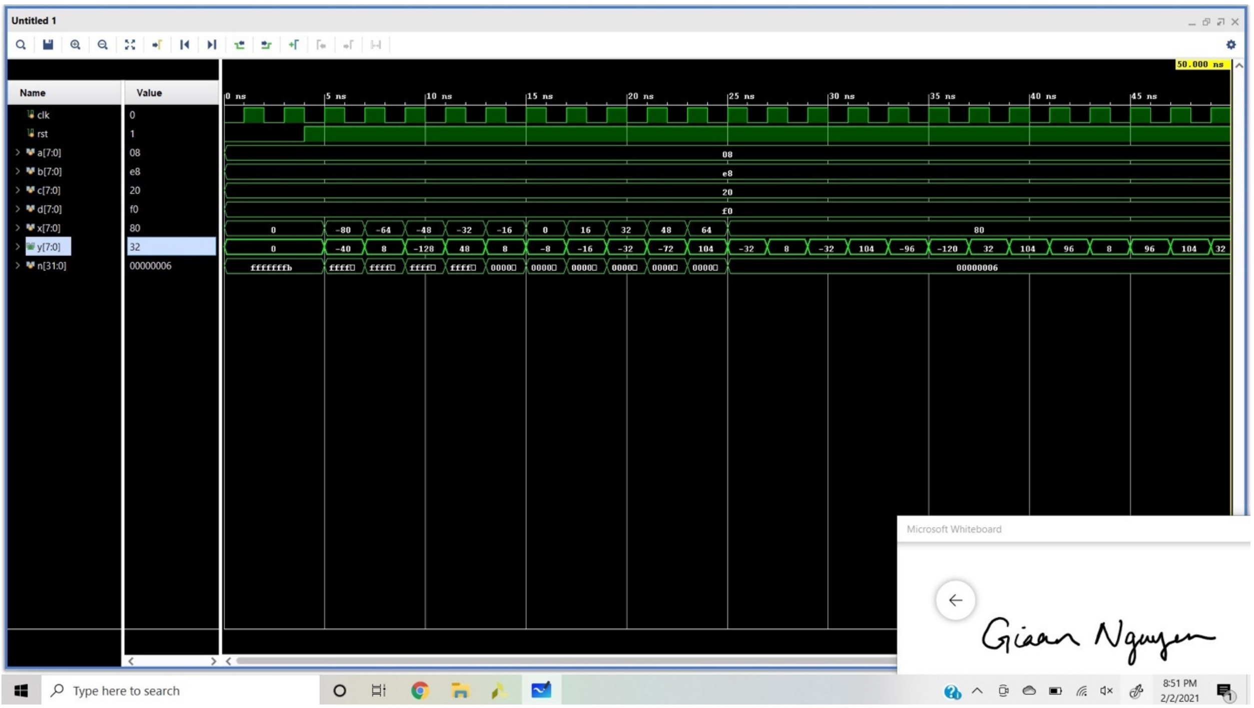

Figure 2. Simulation of the IIR filter.

Questions

1. **Are FIR filters inherently stable? Please specify your reasoning.**

Yes, FIR filters are inherently stable. From the simulation, the difference equation for the FIR filter is given by:

$$y[n] = 0.5x[n] - 1.5x[n-1] + 2.0x[n-2],$$

and thus its transfer function is given by:

$$H(z) = \frac{Y(z)}{X(z)} = 0.5 - 1.5z^{-1} + 2.0z^{-2} = 0.5z^{-2}(z^2 - 3z + 4) = \frac{0.5(z^2 - 3z + 4)}{z^2}.$$

From the transfer function, we see there are two poles at $z = 0$. Assuming that the filter is causal, the condition for BIBO stability is that all poles must lie within the unit circle. That is, for all poles p, the condition $|p| < 1$ must be true. Since the FIR filter gives us two poles which lie within the unit circle, the FIR filter is stable, as further evidenced by the settling nature when the input no longer increments.

In fact, for the general FIR filter:

$$y[n] = \sum_{k=0}^{m} b_k x[n-k]$$

$$H_{FIR}(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{m} b_k z^{-k} = z^{-m} \sum_{k=0}^{m} z^{m-k} = \frac{\sum_{k=0}^{m} z^{m-k}}{z^m}$$

The FIR filter always has m poles at the origin. Therefore, the FIR filter is always stable.

2. Are IIR filters inherently stable? Please specify your reasoning.

No, IIR filters are inherently unstable. From the simulation, the difference equation and transfer function for the IIR filter is given by:

$$y[n] = 0.5x[n] - 1.5x[n-1] + 2.0y[n-1] - 1.0y[n-2]$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.5 - 1.5z^{-1}}{1 - 2.0z^{-1} + 1.0z^{-2}} = \frac{0.5z^{-1}(z-3)}{z^{-2}(z^2 - 2z + 1)} = \frac{0.5z(z-3)}{(z-1)^2}$$

From the transfer function, we see two poles at $z = 1$. Assuming that the filter is causal, the IIR filter gives us two poles at $z = 1$, which lie on but not within the unit circle. Therefore, the IIR filter is unstable, as further evidenced by the oscillatory nature when the input no longer increments.

Additionally, for the general IIR filter:

$$y[n] = \sum_{k=0}^{p} b_k x[n-k] - \sum_{k=1}^{q} a_k y[n-k]$$

$$H_{IIR}(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{p} b_k z^{-k}}{1 + \sum_{k=1}^{q} a_k z^{-k}}$$

Since the IIR filter has poles that may or may not necessarily be within the unit circle, the IIR filter is inherently unstable and must be dealt with precaution.