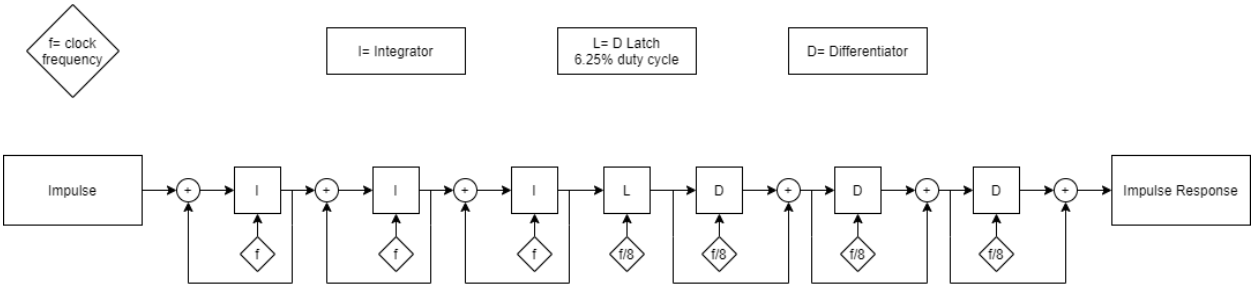
Samuel Darkow
Digital IC Design
5/7/2020

CIC Filter

Introduction

CIC Filter stands for cascaded integrator comb. The integrator is self-explanatory, but the comb is made up of differentiators. The filter design has 3 stages and a 5 bit input. In order to shift the 5 bit input, there must be 14 bits total. These 14 bits are input to an integrator. Three integrators will be cascaded together. These integrators operate at 10MHz. The output of the integrator stage will connect to a 14 bit D latch that decimates the clock to 1.25MHz. The D latch pulses for only one cycle of the 10MHz clock. The output of the D latch then connects to a 14 bit differentiator. This begins the comb stage where three differentiators are cascaded together. These differentiators run at the decimated clock of 1.25MHz. To realize the decimated clock, a 4 bit counter is used. The three main parts for the comb filter can be seen below in the block diagram.



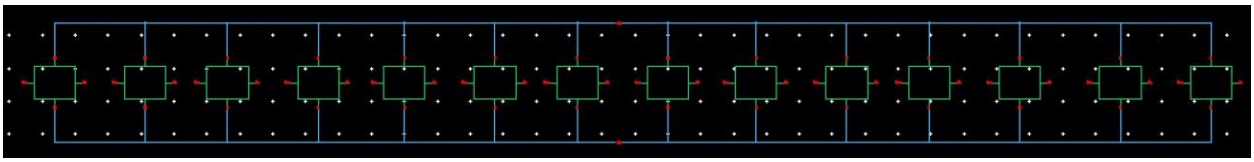
Base Components

The base components needed for the integrators and differentiators are the 14 bit inverter, 14 bit D flip flop, and 14 bit full adder. Another base component is the 14 bit D latch.

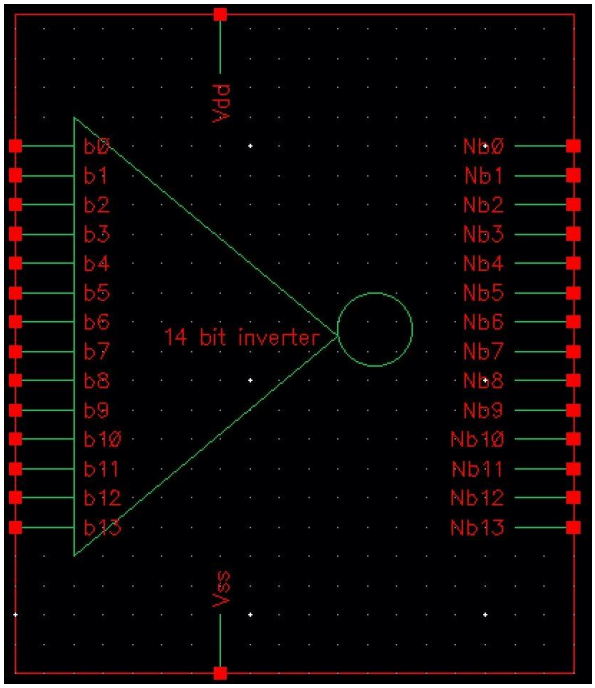
14 bit Inverter

An inverter flips the input signal from 0 to 1 or 1 to 0. Each has one input and one output. To create a 14 bit inverter, the inverters were put in parallel. All power and ground of each one is connected.

Schematic



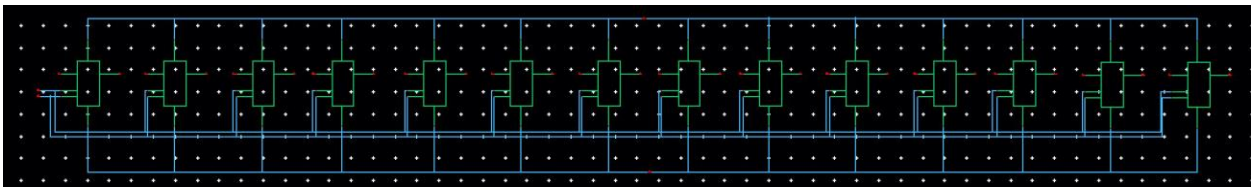
Symbol



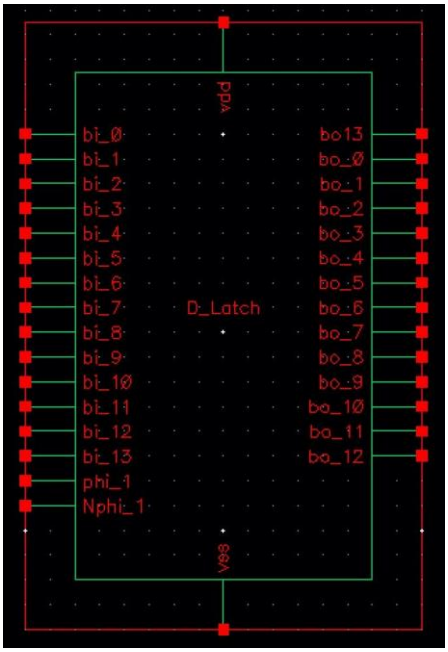
14 bit D Latch

The D latch functions as follows; when the clock is high, the input is passed to the output. When the clock is low the latch holds its output. Each has 3 inputs, the input bit along with the pulse and not pulse. There is only one output. Each latch uses the same pulse. All power and ground of each one is connected.

Schematic



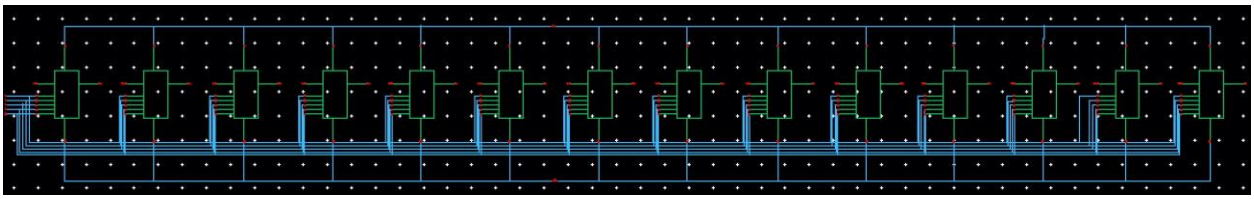
Symbol



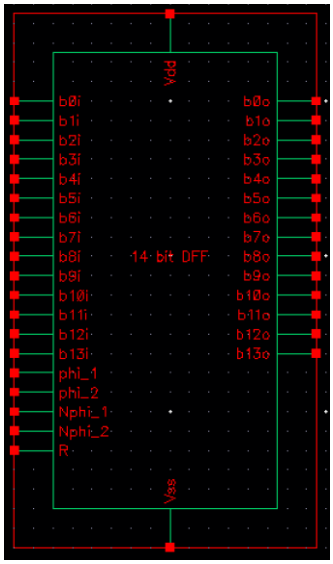
14 Bit D flip flop

A D flip flop is similar to a latch. It differs in that the output is updated on the edge of a clock signal, rather than its current whenever the logic level changes. The D flip flop is triggered by the positive edge of the input clock. A 14 bit D flip flop was created by putting 14 D flip flops in parallel with each having the same clock input. These inputs were phi1, phi2, Not phi1, and Not phi2. Each has 6 inputs and 1 output; the other input is the reset. Before using this component, it must be reset by sending a signal for at least 1 cycle of both two phase clocks. All resets must be tied together for this to work properly. Below are the corresponding schematic and symbol. All power and ground of each one is connected.

Schematic



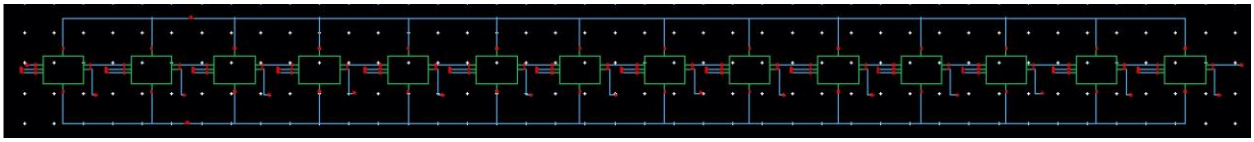
Symbol



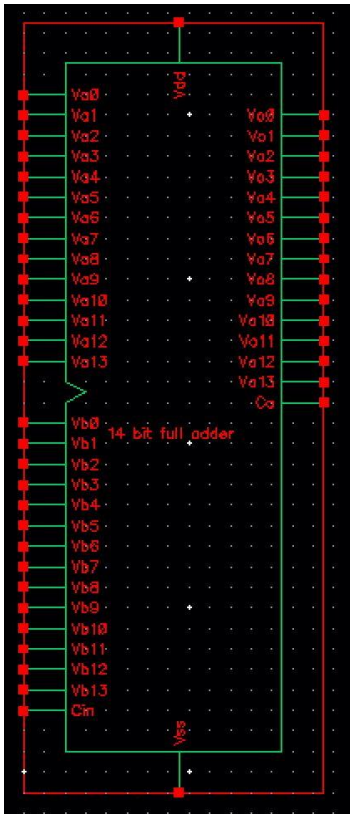
14 Bit Full Adder

A full adder is for binary addition that sums three inputs and produces two outputs. The inputs are A, B, and the carry input(Cin). The outputs are the output carry(Co), which is for overflow, and sum(S). The function is to create a byte-wide adder and cascade the carry bit from one adder to the other. A 1 bit full adder has the logic output for the sum and C out is as follows: Co= A•B+Cin•(A+B) Sum= Cin⊕(A⊕B). The 14 bit full adder is created by putting the full adders in series and the carry out is the input for the next full adders carry in. All power and ground of each one is connected.

Schematic



Symbol

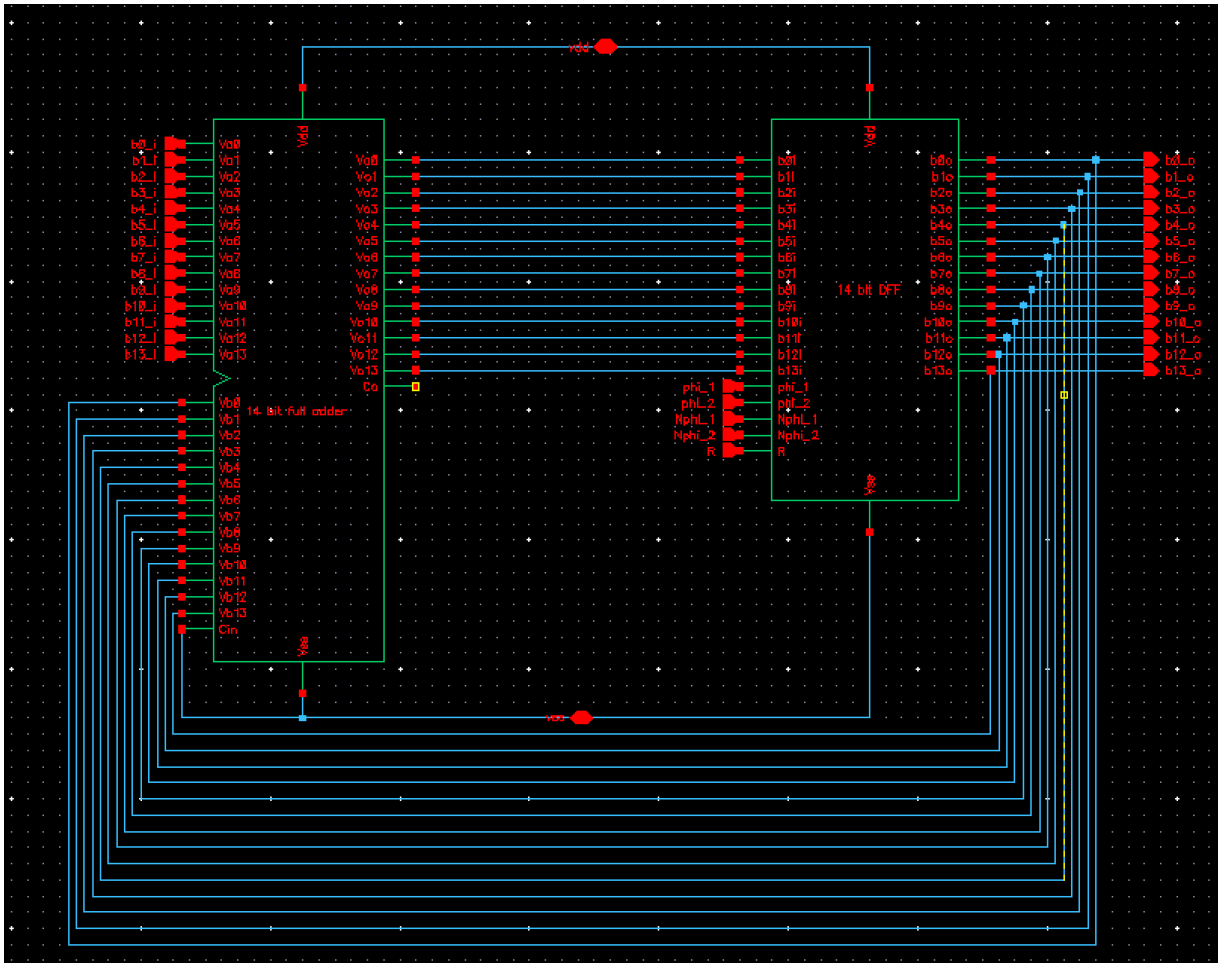


Main Components

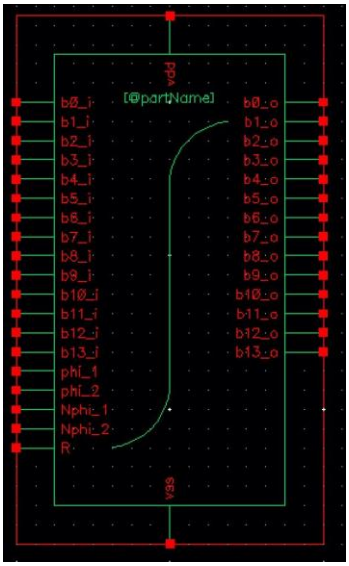
Integrator

The components used are a 14 bit adder and a 14 bit D flip flop. The output of an integrator is equal to the sum of the current input and the previous output. The transfer function to realize this is Vo(n)=Vi(n)+Vo(n−1). The D flip flop can be used to delay the signal and the full adder to sum. To implement an integrator one input for the adder will come from a 14 bit input and the other will be feedback from the D flip flop output. The input to the D flip flop will come from the output of the adder, the carry out is not used and left floating. The output of the D flip flop is the output of the integrator. To simulate and verify if the integrator works, the least significant bit would be set high and the rest of the inputs grounded. This would make the output of the integrator increase by one each clock cycle. This can be seen in the simulation below.

Schematic



Symbol

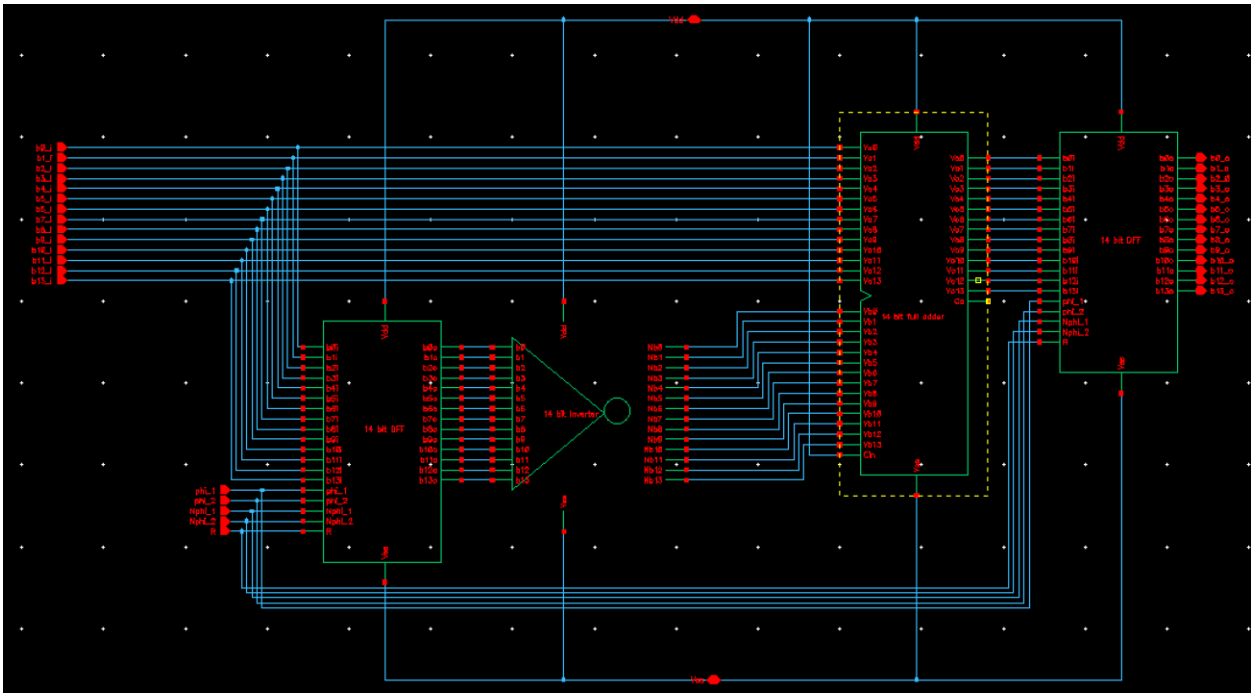


Simulation

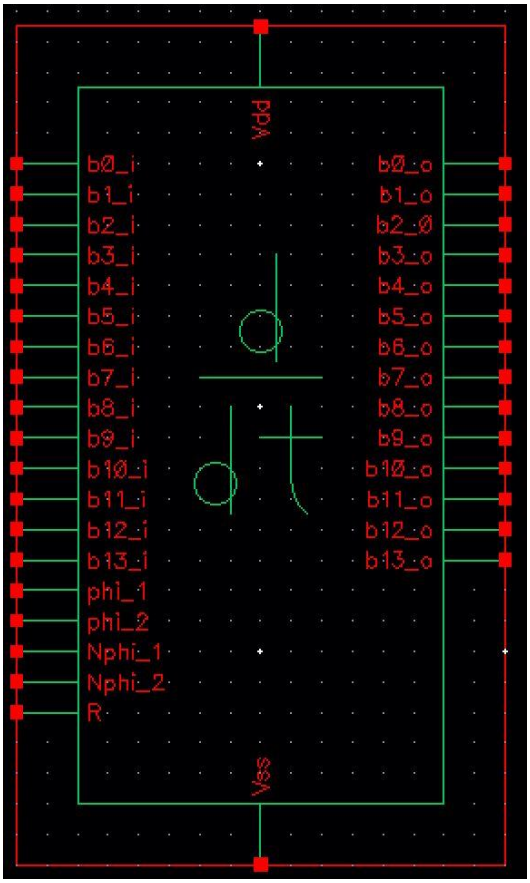
Differentiator

The components used are two 14 bit D flip flops, a 14 bit inverter, and a 14 bit adder. The output of a differentiator is equal to the current input minus the previous input. The transfer function to realize this is Vo(n)=Vi(n)−Vi(n−1). The D flip flop can be used again to delay but unlike the integrator, the differentiator delays the input. Subtraction using 2's complement is completed by inverting the subtrahend and adding one. The minuend and subtrahend are then added together. In order to realize this in the schematic, Cin is always high to add one and the 14 bit inverter is used. To create the schematic for the differentiator, a 14 bit input will be connected to both an input of the adder and a D flip flop. As mentioned earlier, the output of the D flip flop will connect to the input of the inverter. The output of the inverter will connect to the other input of the adder. The output of the adder will connect to the other D flip flop, and its output will be the differentiator output. To simulate and verify if the differentiator works, the least significant bit would be set high and the rest of the inputs grounded. This would make the output of the differentiator increase by one for one clock cycle. The next clock the output will return to zero and remain there.

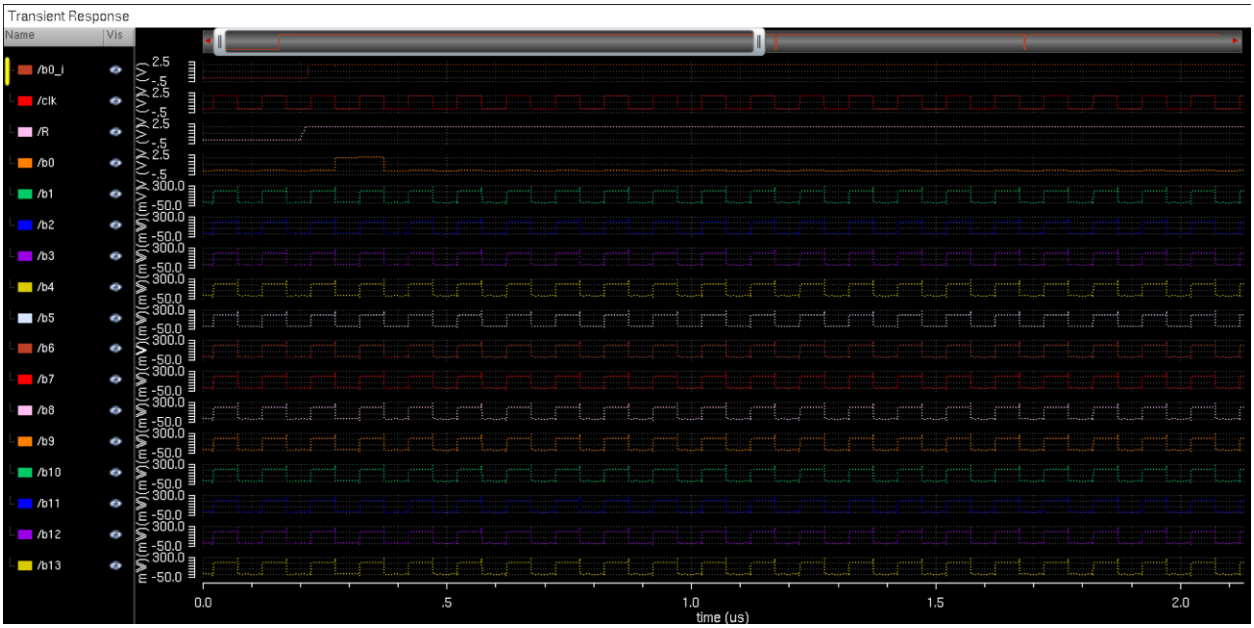
Schematic



Symbol



Simulation

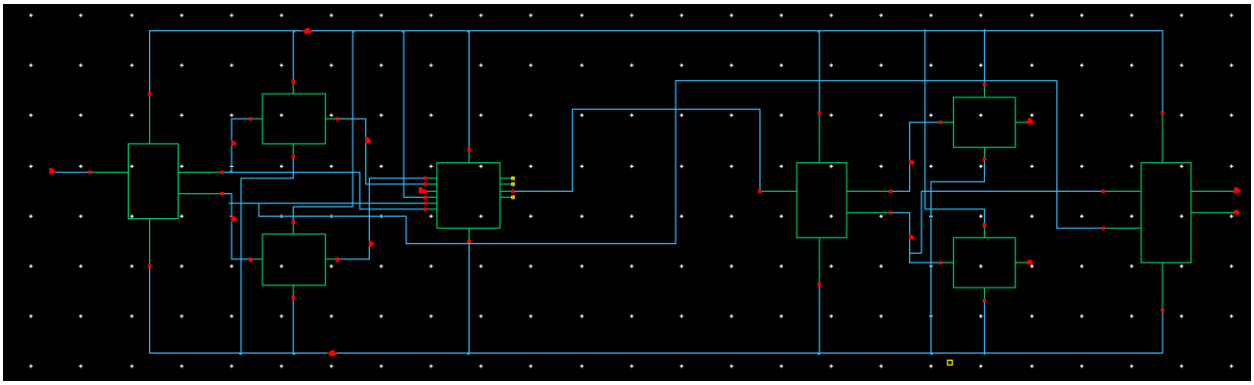


Note: all outputs b1 through b13 are exhibiting noise in the mV range.

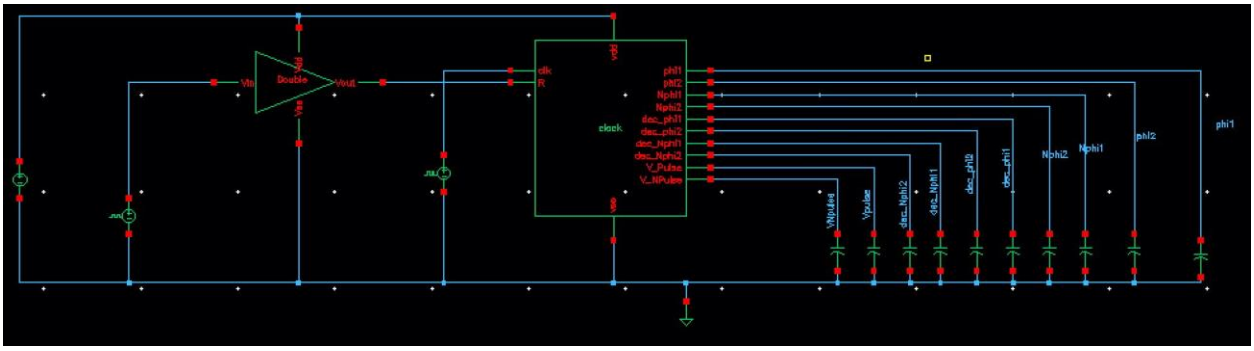
Clocks

There are three clocks that need to be implemented, the first being the sampling frequency for the integrators. The input for the 2 phase clock generator must be a frequency of 10MHz. This is realized by inputting a pulse with a 100ns period and 50% duty cycle. The output of the 2 phase clock generator is phi1 and phi2. These outputs are then inverted to create four outputs, phi1, phi2, not phi1, and not phi 2. To create a second clock with a frequency decimated by eight, the four outputs are input into a 4 bit counter. The third bit of the 4 bit counter is the output since it will output high after four clock cycles, and then low after another four, for a total of 8 clock cycles. This clock is for the differentiators. The third bit is the input to another 2 phase clock generator. Another pair of inverters are used to create four outputs, decimated phi1, decimated phi2, not decimated phi1, and not decimated phi2. The third clock is for the D latch. The D latch needs the decimate clock frequency period with the sampling clock pulse width. To accomplish this, a pulse sequencer is implemented, and the two inputs are phi1 and decimated phi1. The pulse sequencer has two outputs, pulse and not pulse. To simulate and verify if the clocks are working, the sampling frequency should be 10MHz or have a period of 100ns, the second clock should have a period of 800ns or have a positive edge every 8 clock cycles, and the third should have a period of 800ns and a pulse width of 50ns.

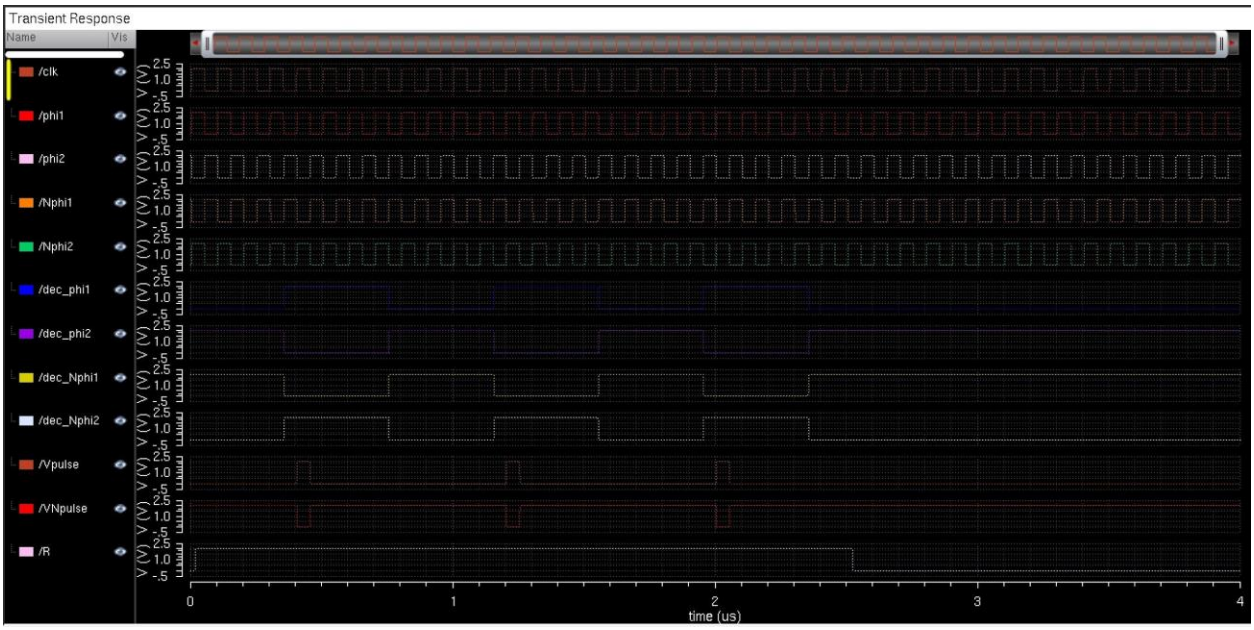
Schematic



Symbol



Simulation



Comb Filter

To realize the comb filter, the components that were created in the previous steps were used. To know if the filter is working properly, an impulse can be sent into the system that is clocked by a single clock edge and compared to an impulse response obtained via MatLab. There is some delay before the first value is realized because of the D latch. The comb stage is decimated by 8, so every 8th output will be realized.

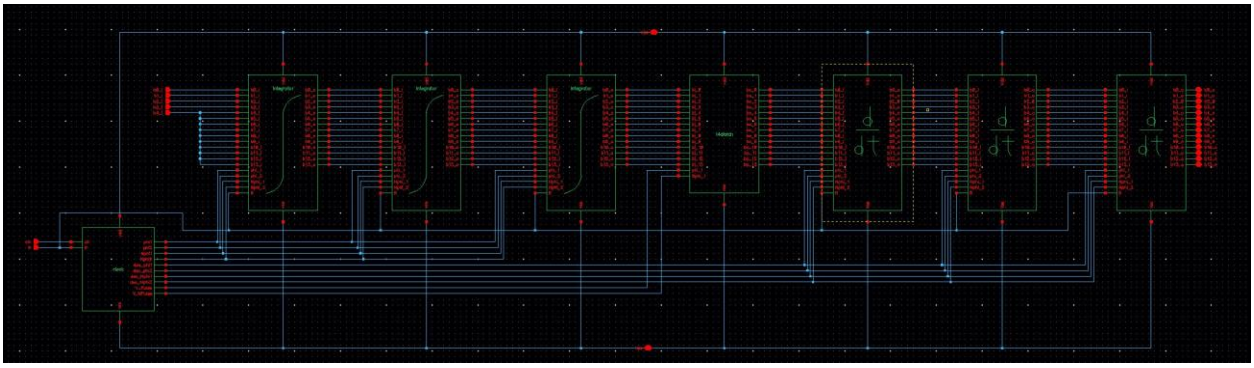
Impulse Response from Matlab

000001, 000011, 000110, 001010, 001111, 010101, 011100, 100100, 101010, 101110, 110000, 110000, 101110, 101010, 100100, 011100, 010101, 001111, 001010, 000110, 000011, 000001

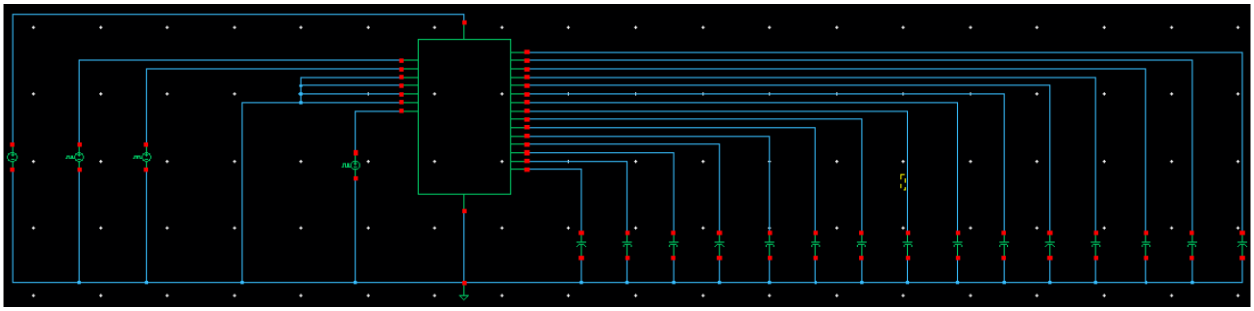
The output of each stage is as depicted below, my latch locked on the second output of the integrator stage, so this table represents my output based on that.

integrator 1 output	integrator 2 output	integrator 3 output		D-Latch Output		Differentiator 1 output	Differentiator 1 output	Differentiator 1 output		output in binary
1	0	0								
1	1	0								
1	2	1								
1	3	3		3		3	0	0		000000
1	4	6								
1	5	10								
1	6	15								
1	7	21								
1	8	28								
1	9	36								
1	10	45								
1	11	55		55		52	3	0		000000
1	12	66								
1	13	78								
1	14	91								
1	15	105								
1	16	120								
1	17	136								
1	18	153								
1	19	171		171		116	49	3		000011
1	20	190								
1	21	210								
1	22	231								
1	23	253								
1	24	276								
1	25	300								
1	26	325								
1	27	351		351		180	64	46		101110
1	28	378								
1	29	406								
1	30	435								
1	31	465								
1	32	496								
1	33	528								
1	34	561								
1	35	595		595		244	64	15		001111
1	36	630								
1	37	666								
1	38	703								
1	39	741								
1	40	780								
1	41	820								
1	42	861								
1	43	903		903		308	64	0		000000

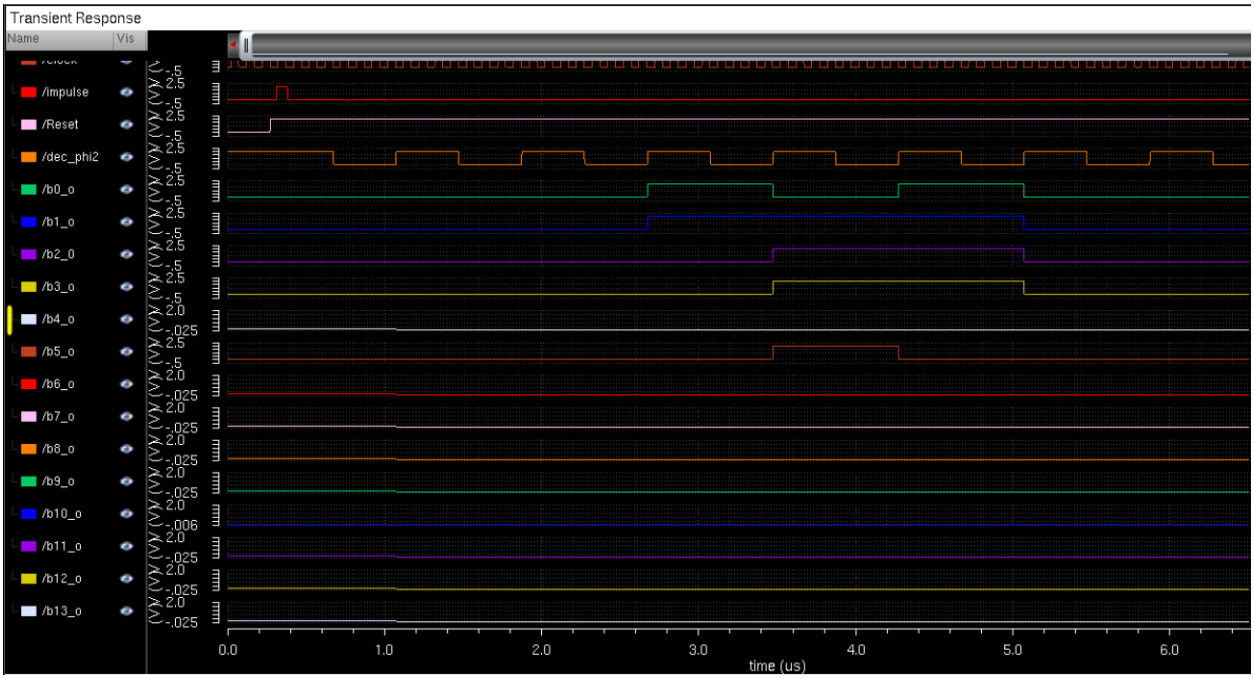
Schematic



Symbol



Simulation



Note: on the third positive edge of decimated phi 2 the output is 00000000000011 00000000101110 00000000001111 00000000000000 which matches the impulse response.

Testing

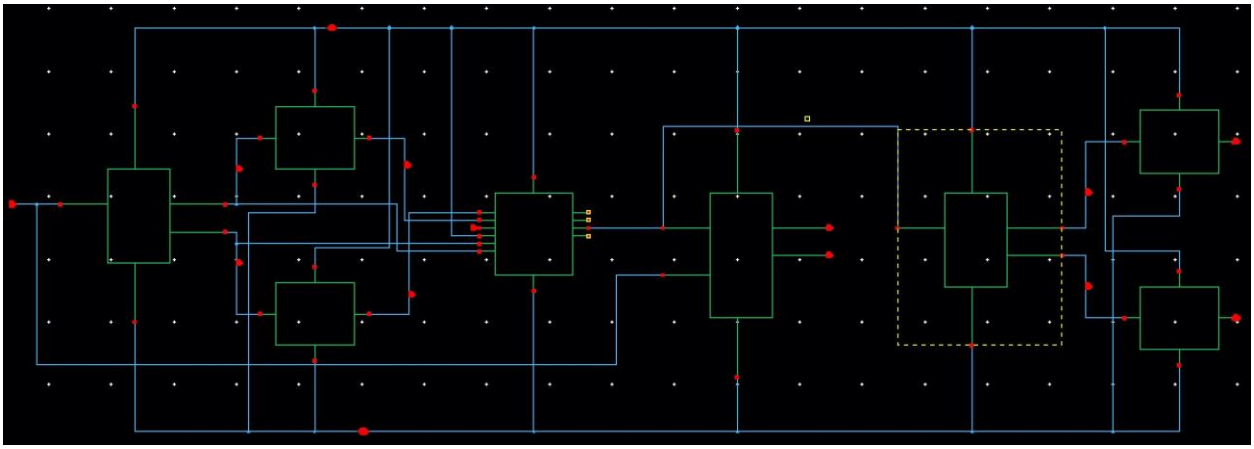
Compared to creating components, I spent about 60-70% verifying, testing, and debugging. I initially started my debugging by testing the differentiator which at the time was the only component in which I had not tested. I tested bits 0-5 and everything seemed fine. I went back and noticed that there was a slight time shift in my simulation since I used the input clock for the pulse. This improved the alignment but did not fix the issue I was having.

I then decided to test each phase sequentially in my filter. I started with the timing of the input signal and reset, and then the integrator stage. Next, I tested the D latch, and finally verified the comb stage. I made sure that I was starting after 2 clock cycles to reset the D flip flops so that both phi1 and phi2 could go high. I made sure that the input impulse encompassed one pulse width. After this, I tested the integrator stage and the output matched what I had in the integrator 3 column for verification. I then knew that my integrator stage was not the problem. I then tested the next section of my filter, the D latch. This was latching every 8th output from the integrator stage and holding it for 8 clock cycles, so I knew that this was not the issue either.

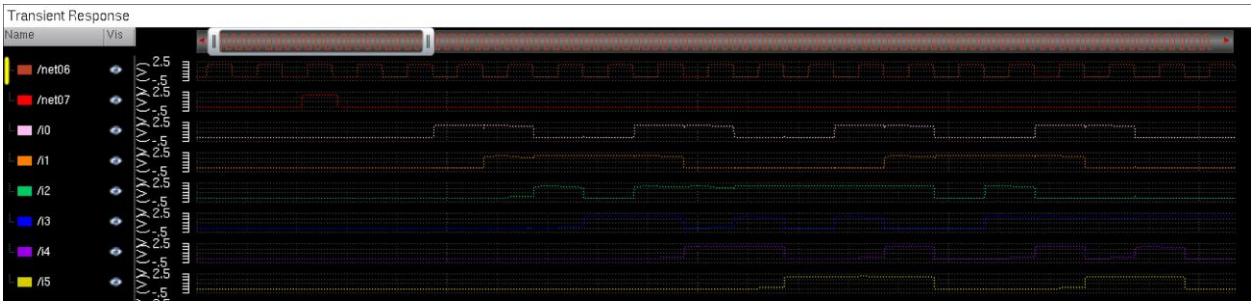
I then tested my first differentiator after the latch. The output was fine for the first six bits, but the rest were all going high. The output was supposed to be 3 in binary, so I went back through the original schematic to make sure I did not cross any wires. I then tested it again on its own and I was seeing the same result as in the filter. I checked to see what the input to the adders was to make sure the D flip flop was working, and the signal was being delayed and inverted correctly. This is when I made sure I was performing 2’s complement correctly and that Cin was in fact supposed to be high.

I was extremely confused as to what was wrong with my full adder since it worked in the integration stage but then realized that 2’s complement utilizes the Co in counting. I thought that setting it high would fix it. However, this did not correct the issue and made it much worse. I then noticed that the schematic on the lab manual had Co grounded. Once I was informed that this needed to be left floating, I modified my differentiator. I then verified that it was working correctly. I then tested the CIC filter again and it worked. For extra verification, I also tested the second differentiator in the comb stage, and this also had the correct output.

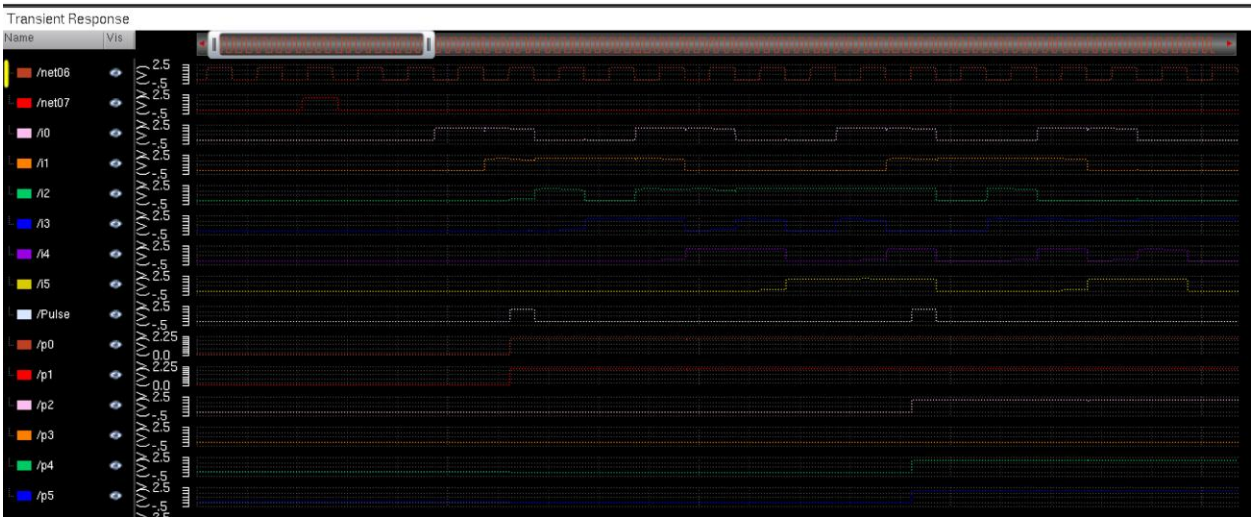
Original Clock with slightly mismatched phases



Testing of the integrator stage



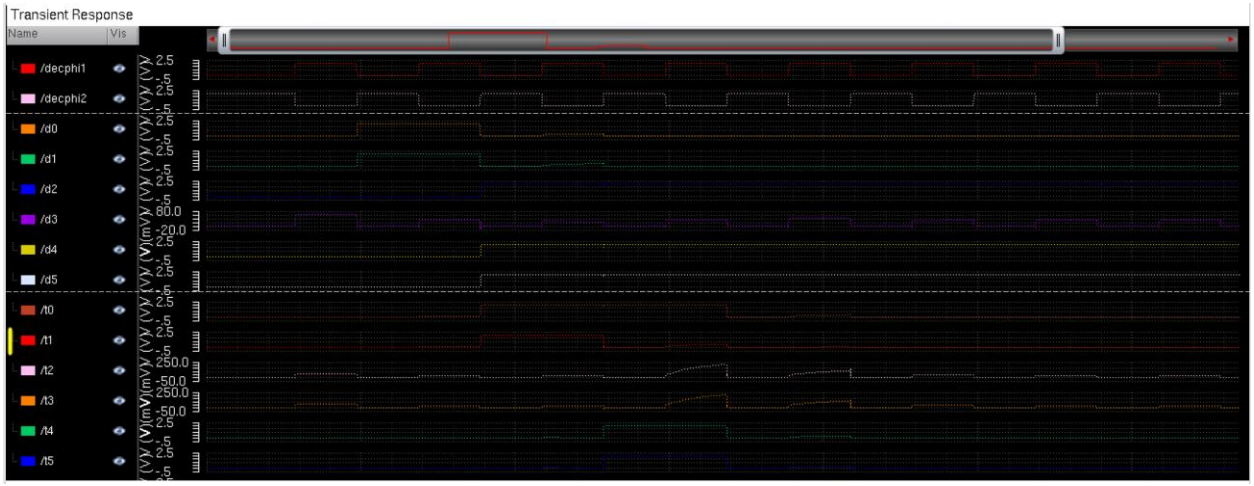
Testing pulse and D Latch



Test of differentiator

Note: testing the differentiator revealed that the count was not correct because my C out was grounded messing up the count sequence when reaching the maximum value.

Verification of first and second differentiator (6 bits only)



Note: the dashed lines separate the decimated clock signal and the first six outputs. d0 through d6 are the outputs of the first differentiator which the outputs are 000011 110100 110100. The decimal values are 3, 52, 116, which is correct. The second outputs 000011 110001 000000, the decimal values that it is supposed to output are 3 49 and 64. These values are correct.