ELEN 475 INTRODUCTION TO VLSI

FALL 2000

PROJECT REPORT:

FIR FILTER DESIGN

Rajeshwary Tayade

Student ID 000-07-1573

Introduction:

A finite Impulse response filter (FIR filter) is a system with transfer function given by ,

$$y(n) = \sum_{k=0}^{k=m-1} x(n-k)h(k) \qquad (1)$$

where x is the input, y is the output and h is the impulse response of the system. The above equation is for a discrete case in which both x and y can be considered as samples of a continuos signal. The equation shows that every output of the system is a function of the system coefficients and the last 'm' inputs, where 'm' is the order of the filter. It is similar to averaging the input samples using different weights.

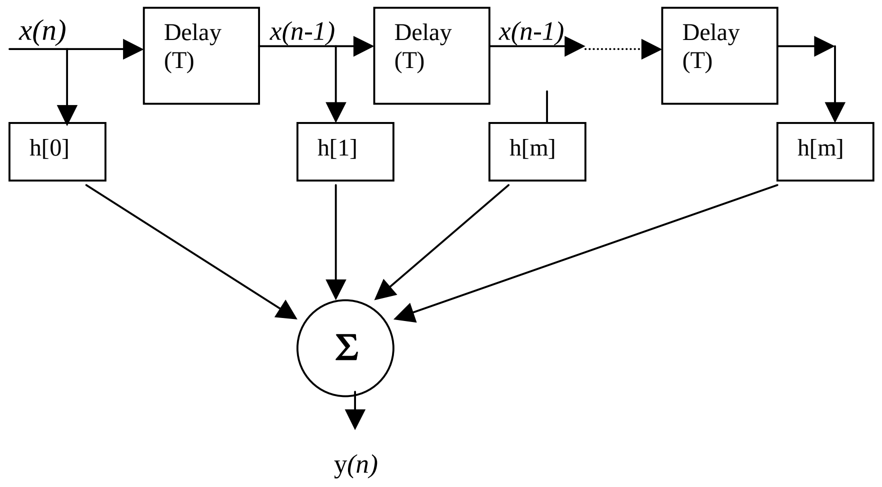


Figure 1 : FIR filter block diagram

The implementation of (1) is shown in Figure 1. In practice since the calculations are done in real time the coefficients and the input samples are stored in memory. Such a filter forms the core of any DSP chip.

Design:

A filter of order '*m*' will have *m* coefficients and thus will require M multipliers and M-1 adders to get an output for every the input sample. If it is linear phase filter then the coefficients are symmetrical i.e

$$h(n) = h(m-1-n) \qquad (2)$$

In this case only m/2 coefficients need to be stored thus reducing the memory. The filter can then be design using dual configuration Figure 2.

The FIR filter that is implemented in the project is of order 7 and it is assumed to have linear phase.The design can be made using dual configuration and only four coefficients (C_k) need to be stored.

$$C_0 = C_6,\ C_1 = C_5 \text{ and } C_2 = C_3 \tag{3}$$

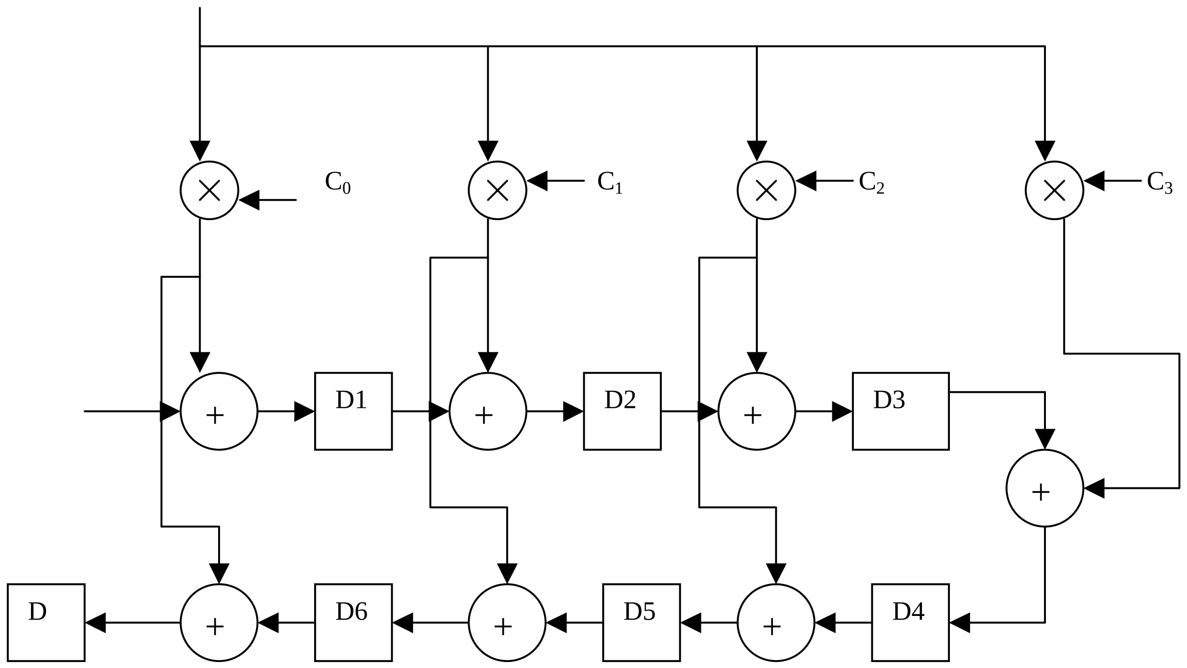


Figure 2 : Signal Flow for a 7th order dual configuration FIR filter

Figure 2 shows the signal flow graph of the FIR filter which uses dual configuration and parallel calculation of all the product terms by using four multipliers. The circuit hardware can be reduced to one multiplier and one adder by designing a sequential circuit and calculating the product terms and partial sums and storing them for a finite amount of time. A pipelined architecture is further used to reduce delays between outputs and for parallel processing

At any instant, the contents of the registers D1 to D6 will be as follows:

$$D1 = x(n) \times C_0 \tag{4}$$

$$D2 = x(n) \times C_1 + x(n-1)C_0 \tag{5}$$

$$D3 = x(n) \times C_2 + x(n-1)C_1 + x(n-2) \times C_0 \tag{6}$$

$$D4 = x(n)\times C_3 + x(n-1)C_2 + x(n-2)\times C_1 + x(n-3)\times C_0 \quad (7)$$

$$D5 = x(n)\times C_2 + x(n-1)C_3 + x(n-2)\times C_2 + x(n-3)\times C_1 + x(n-4)\times C_0 \quad (8)$$

$$D6 = x(n)\times C_1 + x(n-1)C_2 + x(n-2)\times C_3 + x(n-3)\times C_2 + x(n-4)\times C_1 + x(n-5)\times C_0 \quad (9)$$

$$D = x(n)\times C_0 + x(n-1)C_1 + x(n-2)\times C_2 + x(n-3)\times C_3 + x(n-4)\times C_2 + x(n-5)\times C_1 + x(n-6)\times C_0 \quad (10)$$

From equations (4) to (10) it be seen that every register D1 to D6 contains the sum of the product of the current input sample with one of the coefficients and the previous contents of the lower number register.

e.g $D3 = x(n)\times C_2 + D2_{old}$

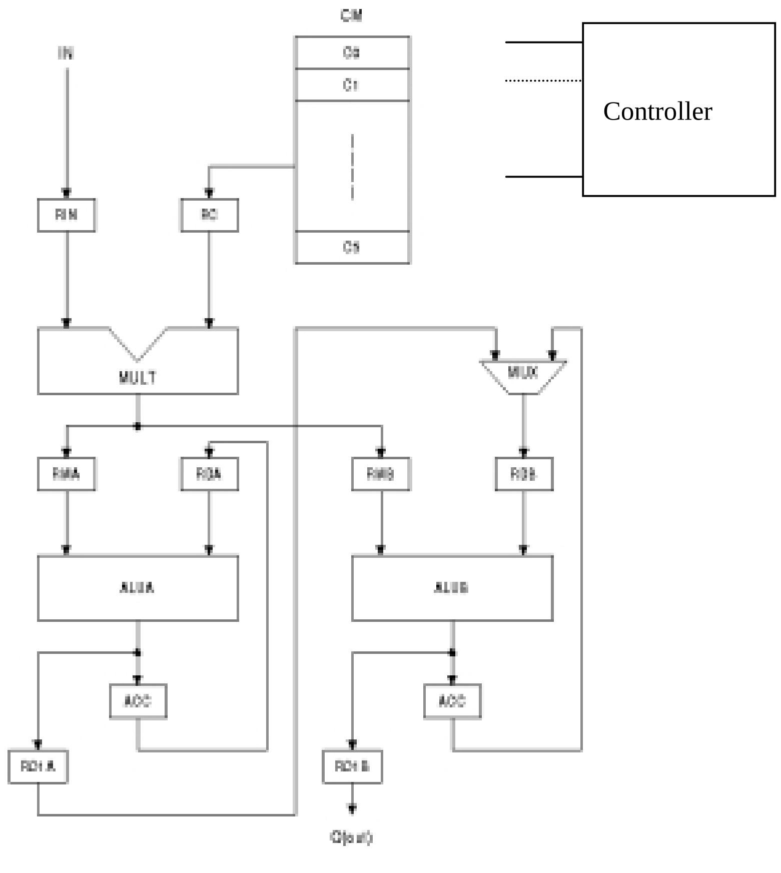


Figure 3 : Block Diagram of sequential ,pipelined FIR design

Components:

The block diagram sequential circuit implementing the signal flow mentioned above is shown in Figure 3. It consists of the following functional modules:

1. **Memory:** The memory is used to store the filter coefficients .In the implementation, each coefficient is of six bits and only four words need to be stored. The memory is build using simple negative edge triggered D-type latches.

2. **Multiplier:** The multiplier is a 6-bit Array multiplier, with a output of 12-bit product. Schematic is attached in Appendix A.

3. **ALUA/B:** The ALU is a simple 12-bit adder . The carry output of the adder is not taken for further calculation and it is an overflow indicator.

4. **Accumulator A/B:** The accumulators are FIFO, 12-bit shift registers and are used to calculate and store the intermediate terms for the future outputs. Accumulator A consists of 4 registers and Accumulator B is made of 3 registers.

5. **Multiplexer:** A 2-input 12-bit multiplexer is used to choose between R01A and the output of Accumulator B.

6. **Registers:** The intermediate registers Rin, Rc, RDA, RMA, RMB, RDA, R01A and R01B are used to account for the finite time delay in the combinational logic. All the registers are connected to the common internal clock. Also the registers can be selectively enabled or disabled using a write-enable signal which is active high.

7. **Controller:** The controller is basically a 2-bit counter that is driven by the internal circuit clock. The output lines of the counter are used to address the memory for reading the coefficients. The controller also generates the control signals for the circuit operation.

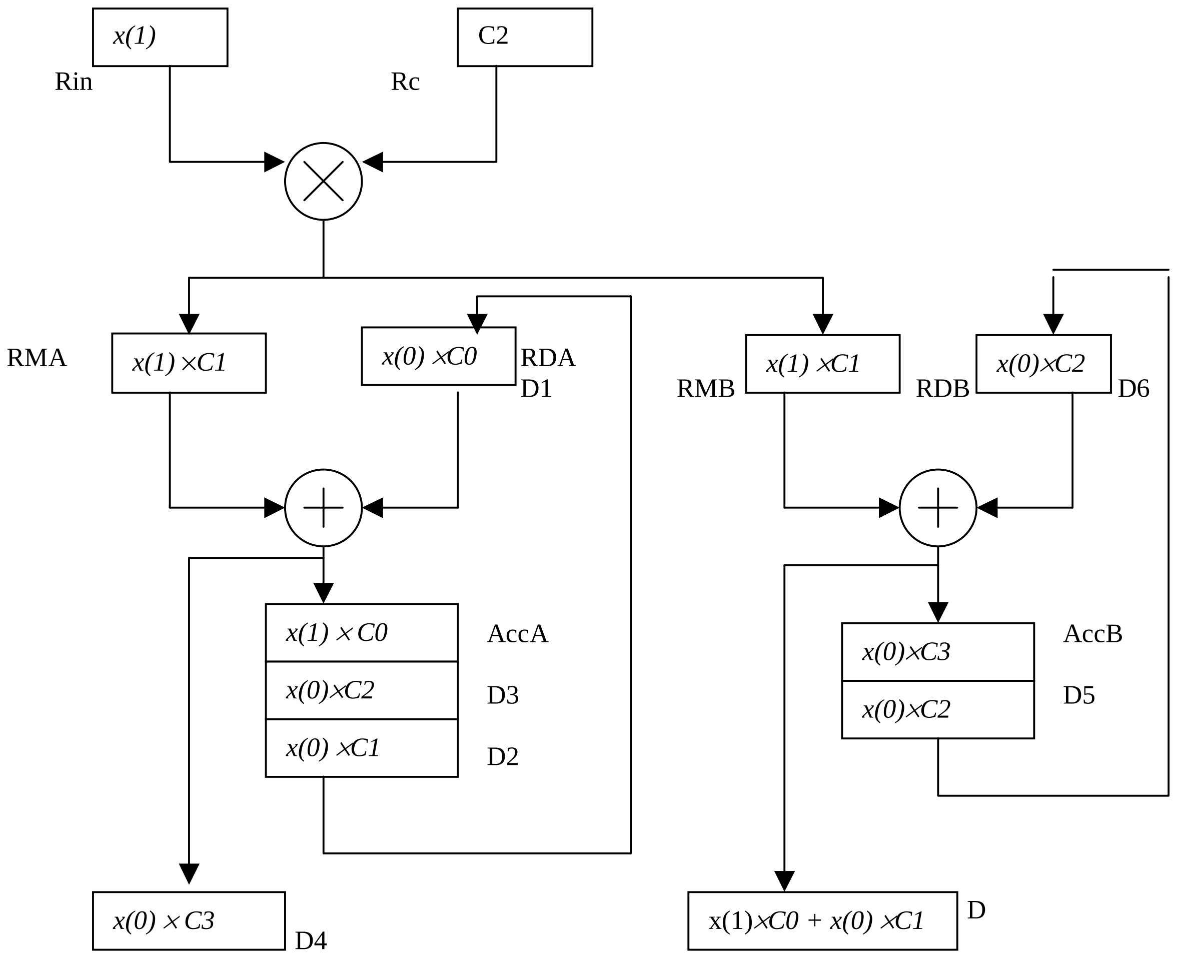


Figure 4 : snapshot of the register contents at the instant of latching *y(1)*

The mapping of the various memory latches of the signal flow diagram of figure 2 to the accumulators of figure 3 is shown in figure 4. The block diagram shown in Figure 4 maps the signal flow diagram of Figure 2 to the block diagram of Figure 3. At every fourth internal clock cycle the registers labeled D1 to D6 in figure 4 contains exactly the same terms that would be present in every cycle in D1 to D6 of the signal flow graph. Thus to achieve any state of the signal flow diagram, the circuit needs four internal clock cycles.

The order of the filter is 7 and therefore a minimum of seven memory words will be required (D1 to D). Hence Accumulator A needs to have only three registers, one for storing the new sum and the other two map to D1 and D2, RDA maps to D3 and R01A corresponds to D4 of figure 3. Similarly Accumulator B needs to have only two registers one for storing the new sum and the other for storing the D5 value and RDB corresponds to D6. The output register D is similar to register R01B.

Initialization:

The memory is initialized using a separate clock that is at a much higher frequency than the internal clock. This is because the coefficients for any filter have to be written only once. Once all the coefficients are written the write-enable of the memory should be made low.

For every negative edge of the counter clock, a address is generated and the contents of that address is placed on the memory bus. However this data is latched into RC only in the following negative edge. In order to synchronize the latching of the first coefficient with that of the new input the counter has to be cleared once just before the first input sample is latched.

The sample clock is obtained from the same internal clock using a divide-by-two circuit. Thus the lower bit of the counter output can be used as the sample clock.

Operation:

For every input sample *x(n),* there are four internal clock cycles and the following operations are performed for every internal clock :

1. The input sample *x(n)* is latched in register Rc and the first coefficient C_0 is latched into register Rin at the same time at the negative edge of the first internal clock
2. The product of the previous register contents Rin and Rc that was available at the multiplier output is latched into registers RMA and RMB.
3. The sum of the previous contents of RMA and RDA is latched into the first register of AccA also the sum of the previous contents of RDB and RMB is latched into the first register of AccB.
4. Accumulators A and B are shifted by one position so that the output of AccA is latched into RDA and that of AccB is latched into RDB.

A valid output value has to be latched into register R01B every sixth clock cycle from the instant a new input is latched into Rin. The delay of two extra clocks is introduced by registers Rc and RDA, however the output frequency is the same as the sample clock. The operation performed in the four clock cycles starting from the first negative edge when a value is latched into the accumulators can be understood from Table1 which maps the shifting and calculation to the signal flow graph.

The various control signals that need to be generated for the signal flow are

1. RDA clear: The register RDA has to be cleared every time D1 is calculated .The clear Signal is to be given only for one half of the internal clock cycle.

2. R01B enable: A valid output is available a every fourth negative edge of the clock e, not including the initial two clocks required for latching in RC and RMA. Thus the write enable signal of register R01B is to be made high every time the counter output is ’10’.

3. R01A enable: The D4 term is stored in register R01A which is not a part of the accumulator and hence a separate write-enable signal has to be given every time the counter output is ‘11’.

4. Multiplexer Select: As shown in Table 1, the select input of the multiplexer has to be changed from 0 to 1 every time the counter output is ‘00’.

5. AccA disable: The write-enable of Accumulator A has to be made low every time the register R01A is written into.

6. AccB disable : The write-enable of Accumulator B has to be made low every time the Register R01B is written into.

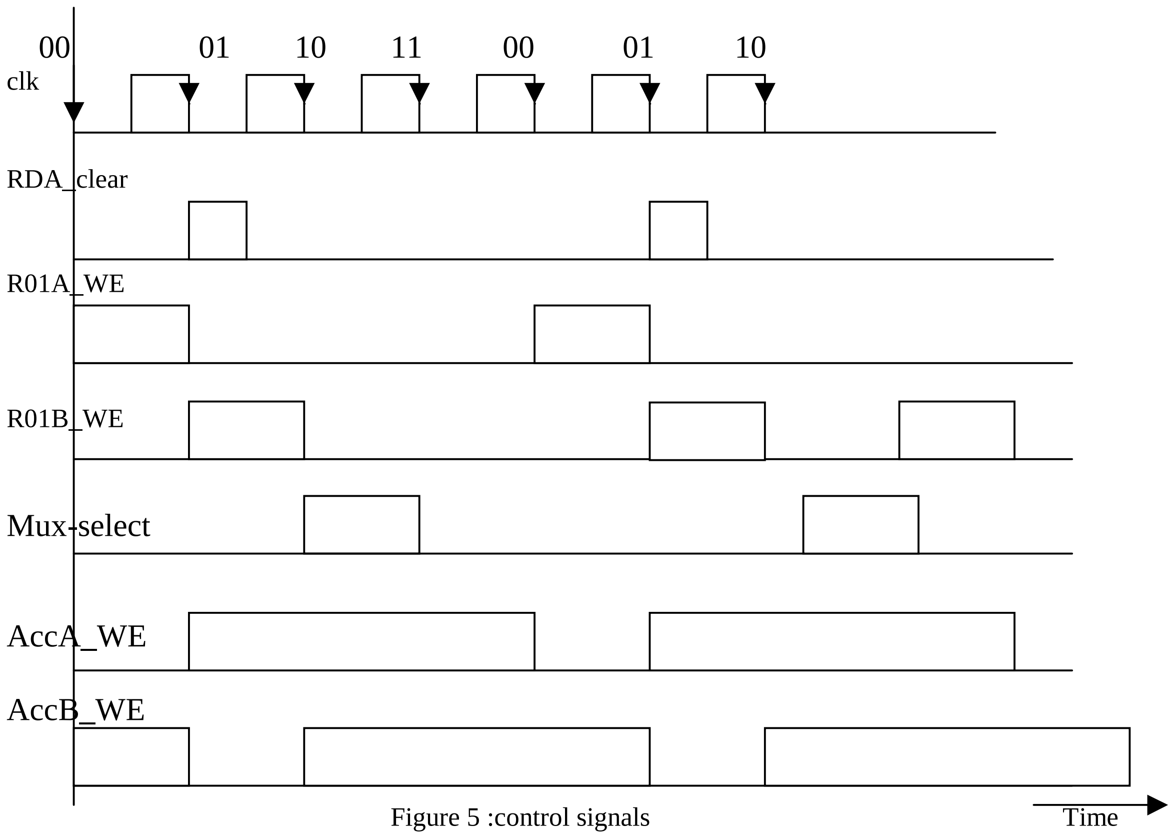


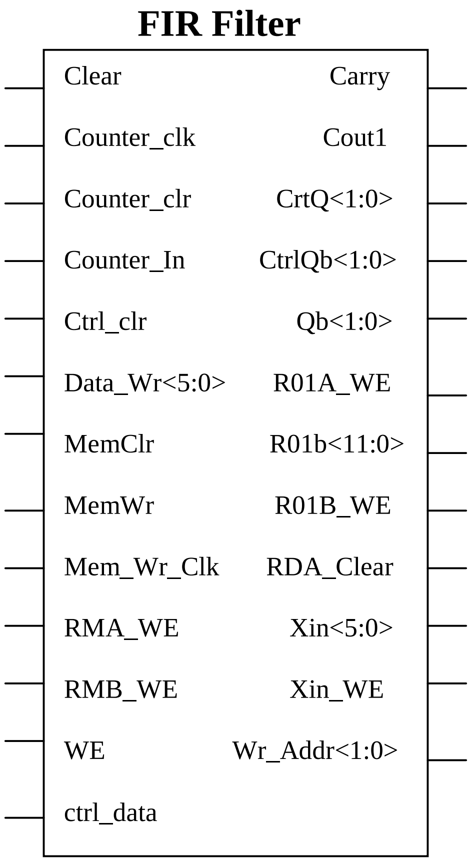
Figure 5 :control signals

These control signals are generated from the counter output using a simple decoding logic .
The complete schematic of the circuit and the verilog simulation file is attached in Appendix B. The internal clock period used is 10ns and the system clock period is 40ns.

Internal Clock cycle	Operations
1.	Latch D1 into AccA and D into R01B Note: The accumulator B should not be shifted in this cycle. Calculate D2 in ALUA and D6 in ALUB
2	Latch D2 into AccA and D6 into AccB Calculate D3 in ALUA and D5 in ALUB Note: for calculating D5 the previous contents of D4 i.e R01A is required and therefore the multiplexer select signal needs to be changed at this clock.
3.	Latch D3 into AccA and D5 into AccB Calculate D4 in ALUA .In this cycle ALUB produces an invalid sum .
4.	Latch D4 into R01A. Note: The contents of AccA should not be shifted in this cycle as all the terms will already be there in place. Calculate D1 in ALUA and D in ALUB Note: To calculate D1 term register RDA needs to be cleared at this point.

Table 1: Signal Flow for the Accumulators A/B

Pinout:



Sr. no.	Pin name	Type :Input/output	Description
1.	Clear	Input	Common Clear Signal for the all the registers .Active Low
2	Counter_Clk	Input	The input to the clock pin of the counter. It is also the internal clock of the circuit
3.	Counter_Clr	Input	Active low clear for the counter
4.	Counter_In	Input	The data input to the T-flipflops inside the counter. This input should be high for the counter to function.
5.	Ctrl_clr	Input	Active low Clear for the controller.
6.	Data_wr<5:0>	Input	The data input bus to the coefficient memory
7.	MemClr	Input	Active Low clear signal for the memory
8.	MemWr	Input	Active high write-enable signal for the memory
9.	RMA_WE	Input	Active high write-enable signal for the register RMA
10.	RMB_WE	Input	Active high write-enable signal for the register RMB
11.	WE	Input	Common Active high write-enable signal for registers Rin and Rc
12.	Wr_Addr<1:0>	Output	The address generated by the counter for writing into the coefficient memory.
13.	Xin<5:0>	Input	The sample data

14	Xin_WE	Input	Active high write enable for the Rin register
15.	ctrl_data	Input	The data input to the counter in the controller. Should be kept high for normal operation.
16.	Carry	Output	Carry out of the ALUA, is an overflow indicator
17.	Cout1	Output	Carry out of ALUB ,is and over flow indicator
18.	CrtlQ	Output	The output lines of the second counter used in the controller.
19.	CrtrlQb	Output	The output lines of the second counter used in the controller.
20.	Qb	Output	The output lines of the counter used for generating memory address.
21.	R01A_WE	Output	Control signal for enabling the register R01A can be monitored at this pin. The signal is generated internally by the controller.
22.	R01B<11:0>	Output	The contents of register R01B can be read at this pin. This is the filter output *y(n)*
23.	R01B_WE	Output	Control signal for enabling the register R01B can be monitored at this pin. The signal is generated internally by the controller.
24.	RDA_Clear	Output	Control signal for clearing the register RDA can be monitored at this pin. The signal is generated internally by the controller.
25.	S	Output	Control signal for the multiplexer for selecting R01A can be monitored at this pin. The signal is generated internally by the controller.

Results:

The Coefficients values used for simulation are
C0 = 3, C1 = 4, C2 = 5, C3 = 1

The circuit was tested for the following input combinations:

Sr. No	Input sample *x(n)*	Filter Output *y(n)*
1.	4	12
2.	8	40
3.	9	79
4.	10	110
5.	11	146
6.	15	204

Table 2: Results

The schematics compacted view and the LVS for the different components of the circuit are attached in Appendix A. The output of the simulations are attached in Appendix B.

Appendix A

Index:

Appendix B