ABSTRACT

TITLE : **Design and Implementation of a 3-Tap FIR Filter using Basic Digital Logic in eSim**

THEORY / DESCRIPTION:

A Finite Impulse Response (FIR) filter is a digital filter whose output depends on the present and past input samples. A 3-tap FIR filter is defined as:

$$y[n] = h_0 x[n] + h_1 x[n-1] + h_2 x[n-2]$$

In this project, the coefficients are chosen as h0 = h1 = h2 = 3 (binary 11). The filter is implemented using basic digital components such as multipliers, adders, and D flip-flops.

The delay elements store previous input samples, and the outputs of each tap are multiplied by the coefficients and summed to produce the final output.

The referenced paper implements an optimized FIR filter using advanced multiplier and adder architectures. In this work, a simplified version of the FIR filter is implemented using basic digital logic components in eSim to reproduce the fundamental architecture and verify its operation.

APPLICATIONS:

- Noise reduction and signal denoising in audio and sensor data
- Audio processing such as equalizers and echo cancellation
- Digital communication systems for pulse shaping and ISI reduction
- Biomedical signal processing (ECG, EEG filtering)

CIRCUIT DESCRIPTION:

The FIR filter is implemented using the following blocks-

• Delay Unit: Implemented using D flip-flops to generate x[n−1] and x[n−2]
• Multiplier: A 2-bit multiplier is used to multiply the input with coefficient (11_2 = 3) for simplicity.
• Adder: Multi-bit adders are used to accumulate outputs of the three taps. First a 4-bit adder accumulates the 4-bit sum of the first 2 taps along with carry bit to prevent overflow then a 5-bit adder accumulates the previous adder sum and the 3rd tap 4-bit product, where the 4-bit product is extended to 5-bit with its MSB as GND (0 value). Hence the final accumulated output is a 6-bit result.

The overall design forms a sequential digital system.

CIRCUITDIAGRAM(S):

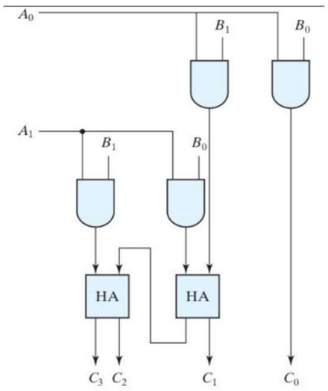


Fig 1: Circuit diagram of 2-bit Multiplier

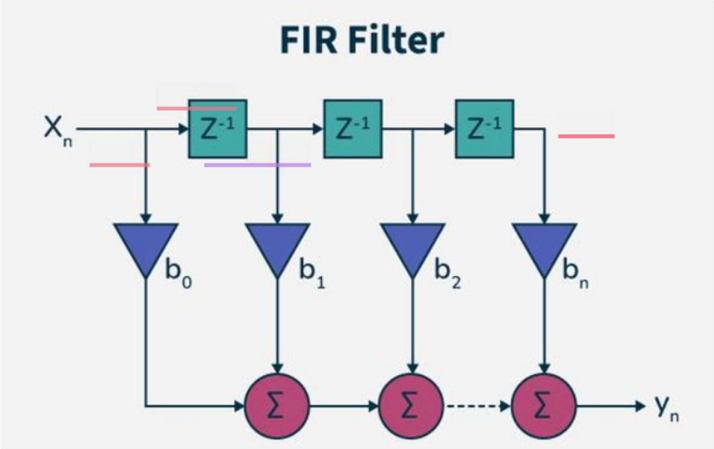


Fig 2: FIR filter flow diagram

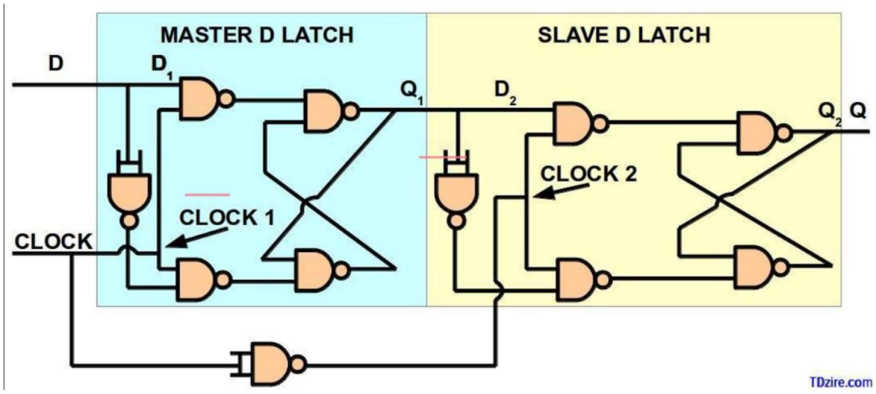


Fig 3: Circuit Diagram of a D flip-flop using NAND gates.

ESIM SCHEMATIC DIAGRAM(S):

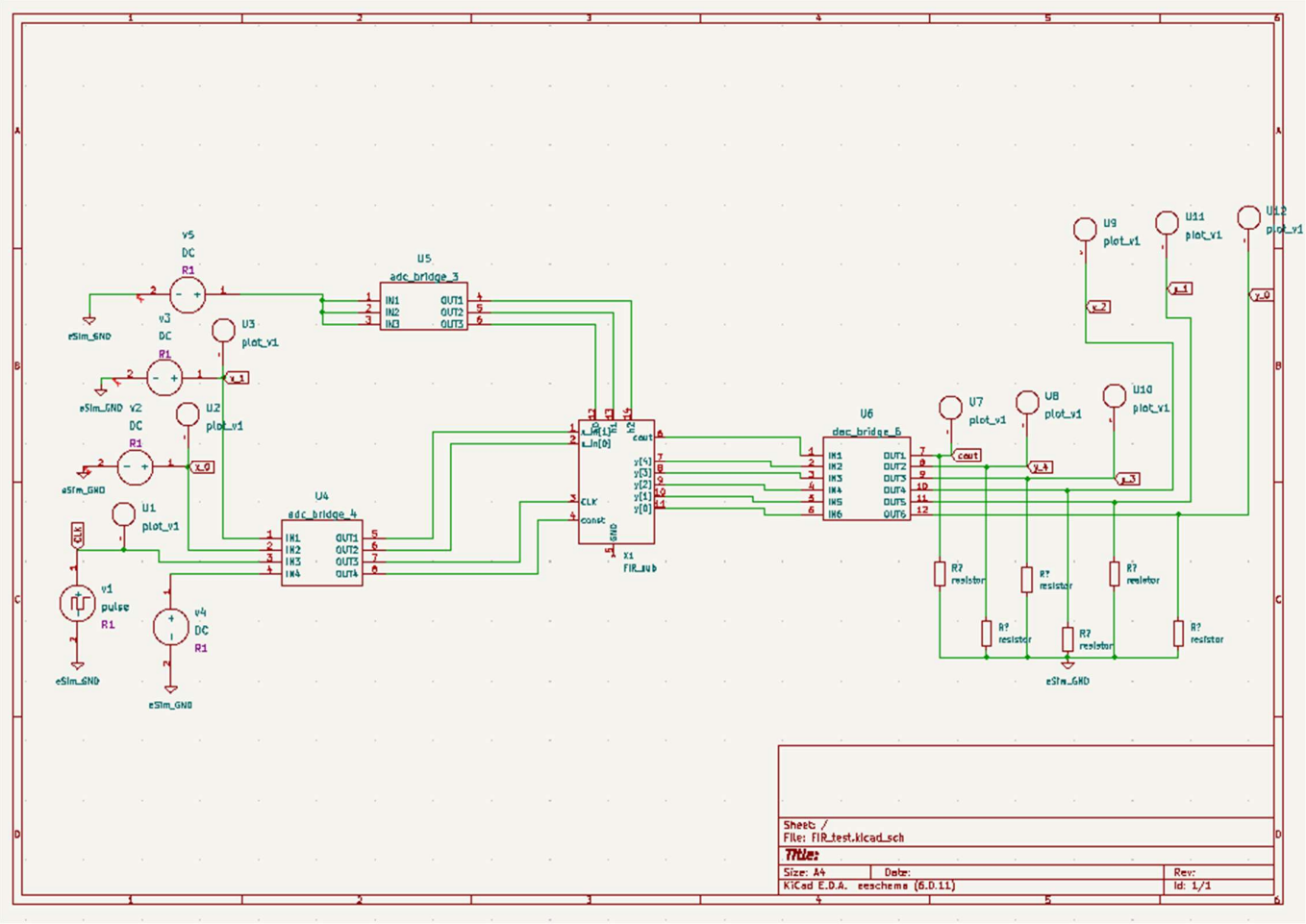


Fig 1 : 3-tap FIR Filter test circuit (root schematic) implemented in eSim KiCad.

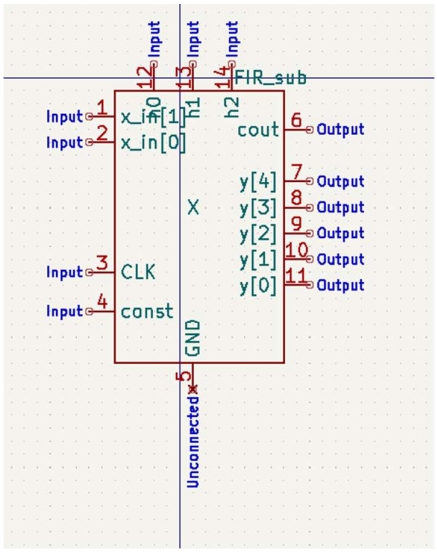


Fig 2 : Custom-made symbol of 3-tap FIR Filter in eSim KiCad. 14 pin structure for clock, 2 bit input x[1:0], constant pin for DFF, GND, coefficient pins h, 6 bit output pins y[4:0] and cout.

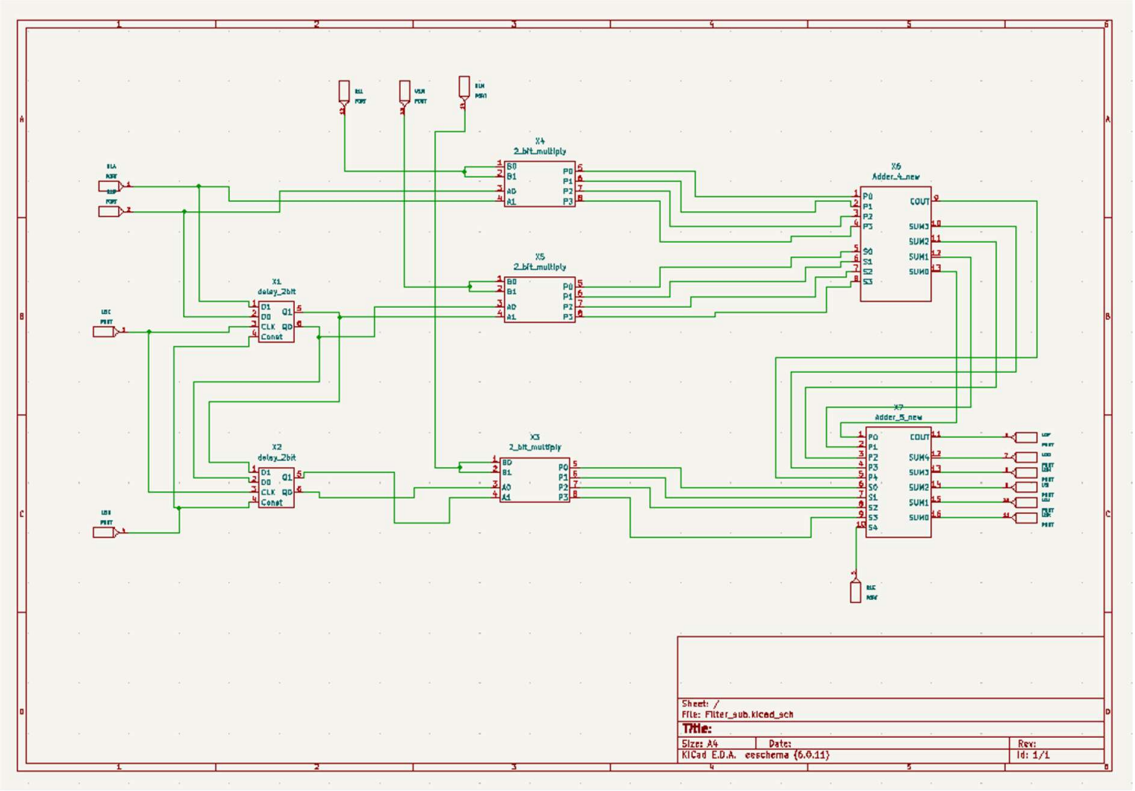


Fig 3 : 3-tap FIR Filter sub-circuit model with 2 delay units , 3 multiplier units and 4-bit adder and 5-bit adders for accumulation.

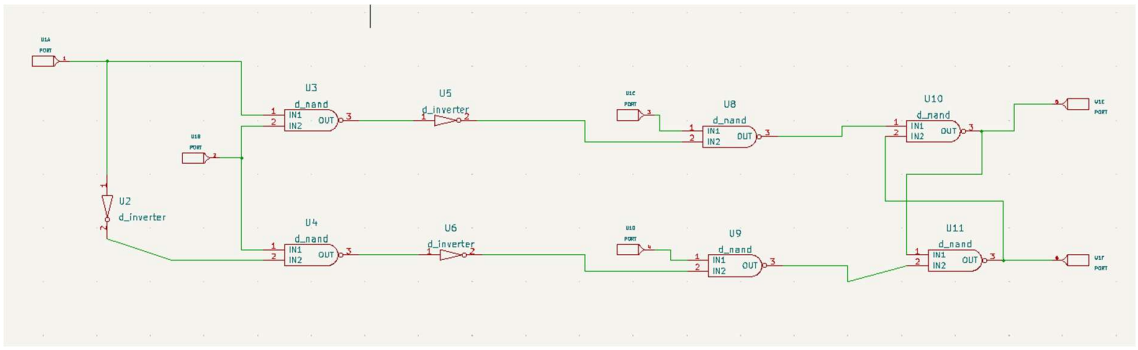


Fig 4 : Custom-made D flip flop subcircuit using NAND and Inverter logic gates.

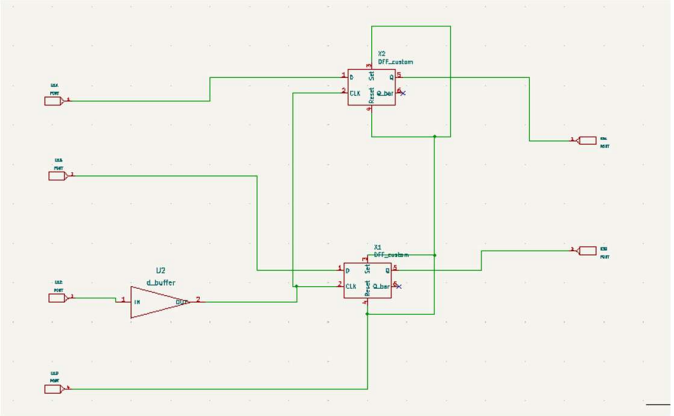


Fig 5: 2- bit Delay unit sub-circuit with 2 DFFs parallelly connected and buffered clock.

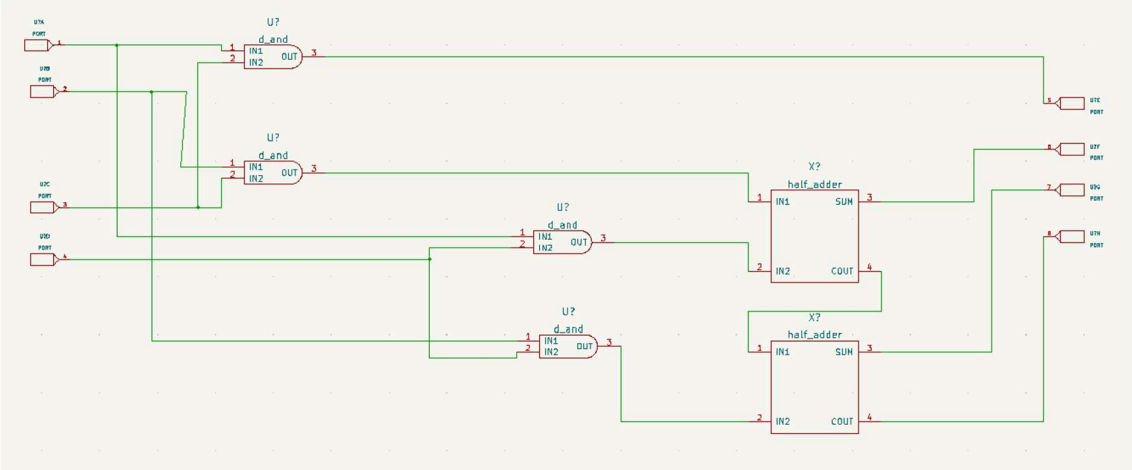


Fig 6: 2 bit multiplier sub-circuit using AND and Half_Adder gates

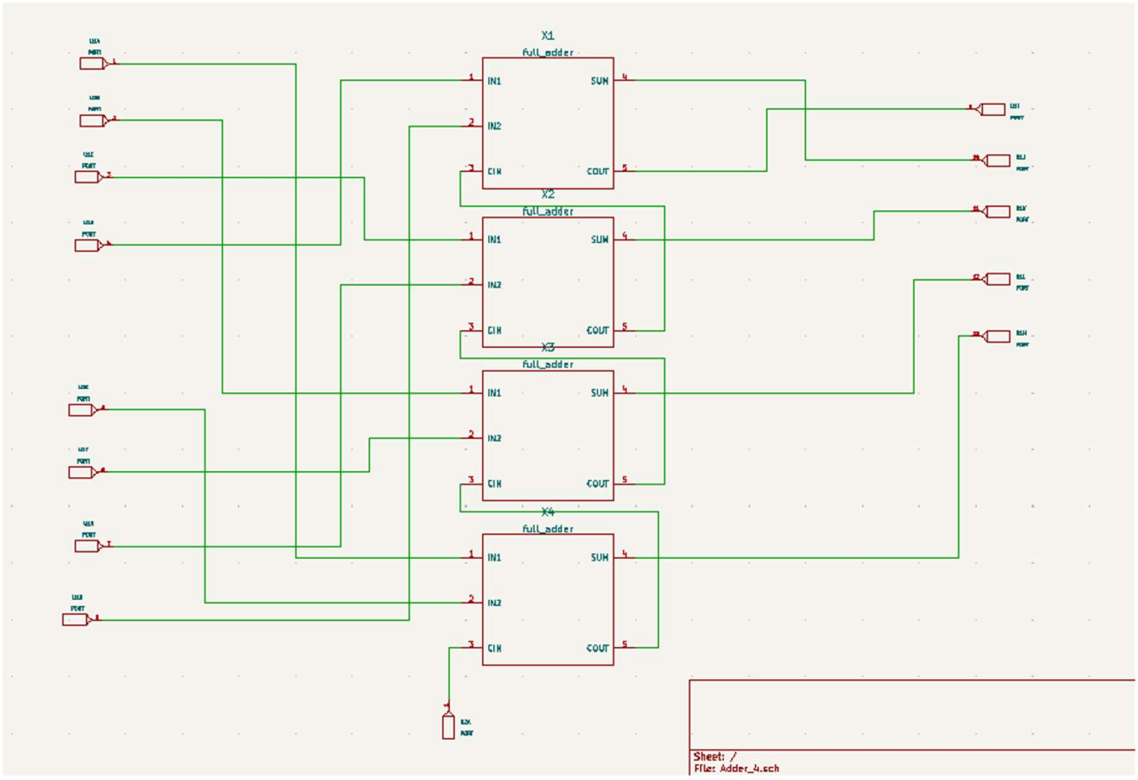
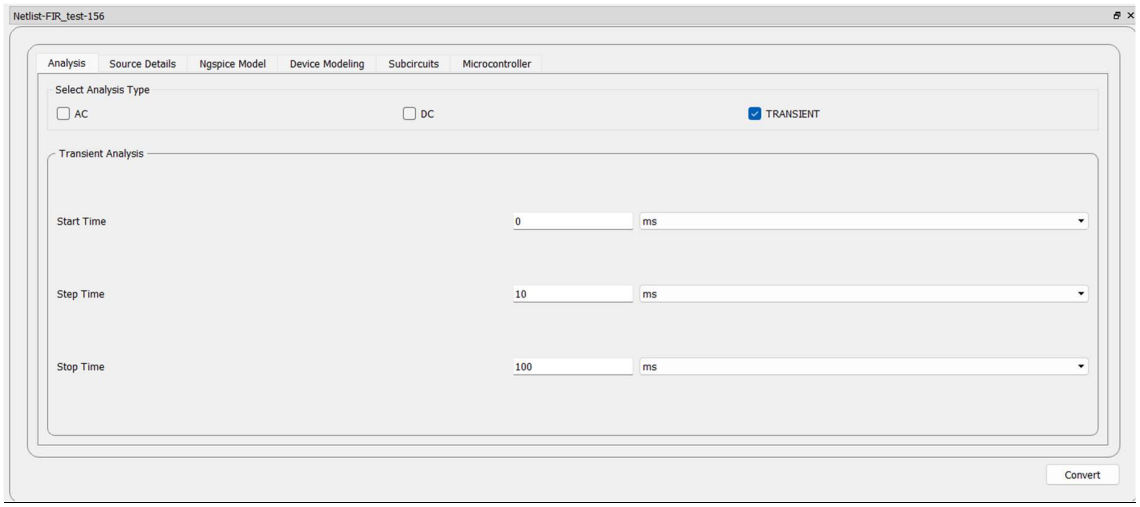
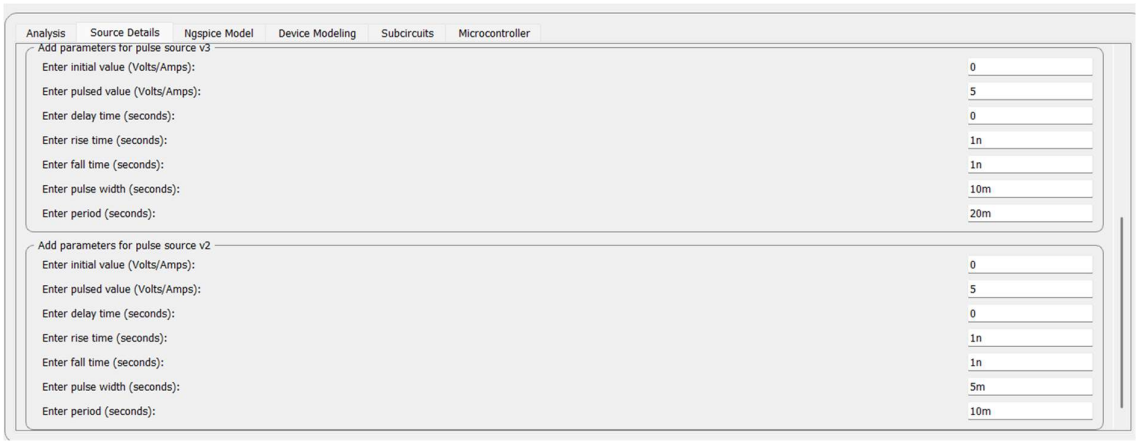


Fig 7 : 4-bit adder sub-circuit using Ripple Carry Full_adders (correction: for LSB, half adder was used originally in the main project)

KiCAD TO NGSPICE CONVERSION DETAILS:





RESULTS / OUTPUTS:

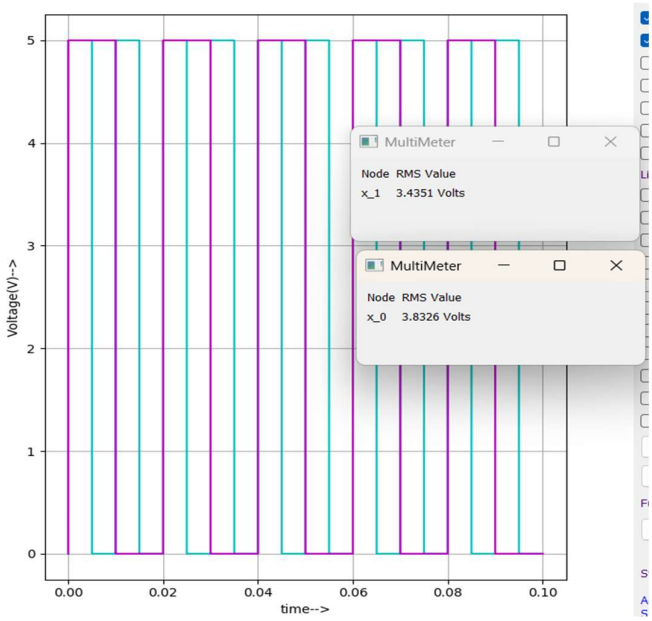


Fig 1 (INPUT)

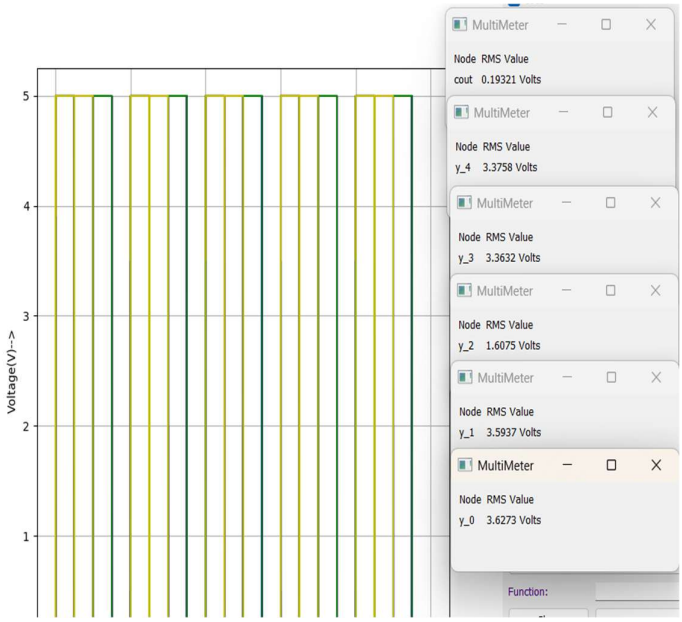


Fig 2 (OUTPUT)

Fig. 1 shows the Python plots and multimeter readings of input bits x[1] & x[0] while Fig. 2 displays the python plots of output bits y[4:0] and cout along with multimeter readings. The observed input and output waveforms are squared – wave pulses and constants as expected from a pure digital circuit.

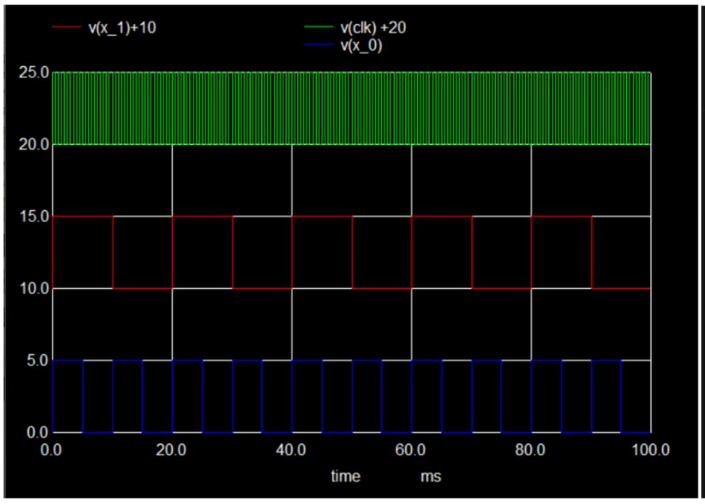


Fig 3 (INPUT)

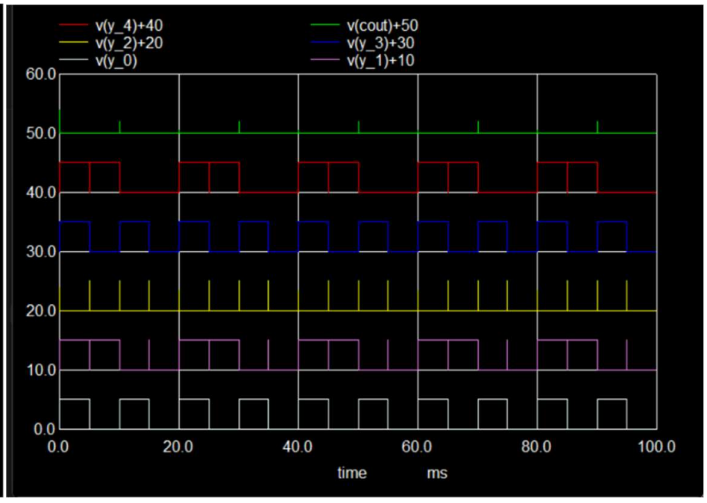


Fig 4 (OUTPUT)

Fig. 3 displays the Ngspice plot for the 2 input bits and clk where the circuit is fed pulsating inputs varying with clock . Meanwhile Fig. 4 shows its respective output 5-bit y[4 :0] along with cout with the accumulated result.

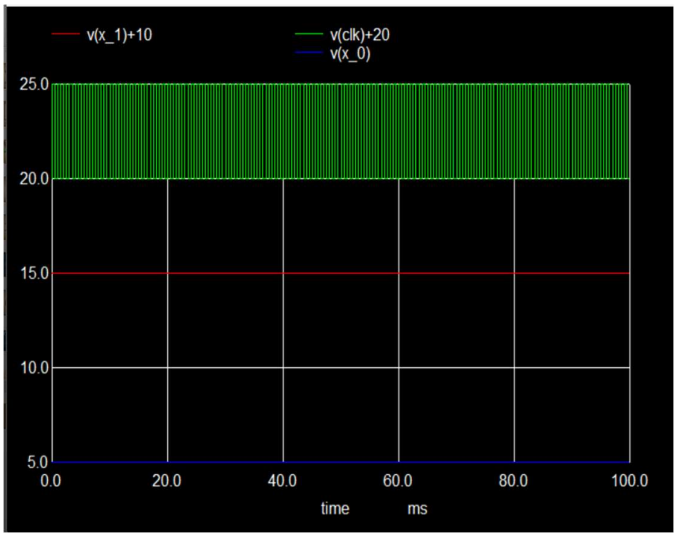


Fig 5 (INPUT)

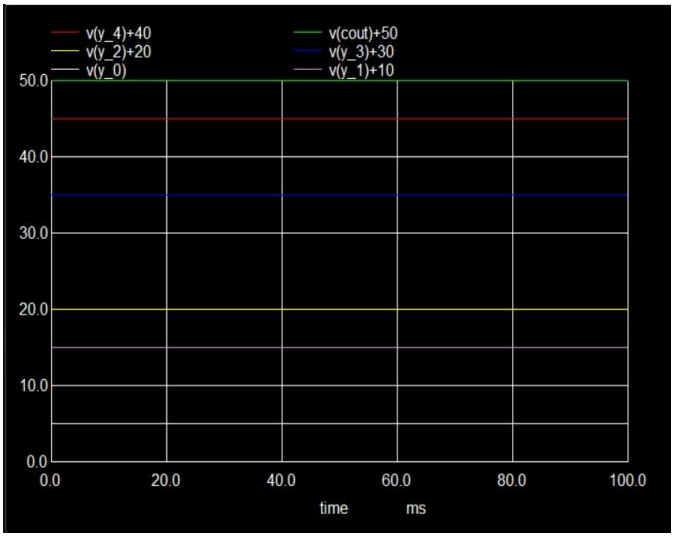


Fig 6(OUTPUT)

Fig. 5 displays the Ngspice plot for the 2 input bits and clk where the circuit is fed with constant input, each bit of x_in representing a binary bit. Meanwhile Fig. 6 shows its respective output 5-bit y[4 :0] along with cout with the accumulated result.

The FIR filter was tested using both constant and time-varying inputs. For constant inputs, the output matched the expected theoretical values, confirming correct arithmetic operation.

For time-varying inputs, the output exhibited sequential behavior and dependency on previous samples, validating the functionality of delay elements.

REFERENCES/PAPERS:

Title : Design and Implementation of FIR Filter using Low Power and High-Speed Multiplier and Adders

Author : O. Venkata Krishna

Journal: CVR Journal of Science and Technology, 2019

Page No. : (Approx.) 80–84

Link : https://www.researchgate.net/publication/330640108_Design_and_Implementation_of_FIR_Filter_usingLow_Power_and_High-Speed_Multiplier_and_Adders

1) https://www.geeksforgeeks.org/electronics-engineering/difference-between-fir-filter-and-iir-filter/
2) https://analogcircuitdesign.com/FIR-filter/