Circuit Simulation Project

https://esim.fossee.in/circuit-simulation-project

Name of the participant: Arjun Bathla

Project Guide: Dr R. Maheswari

Title of the circuit: Cyclic Redundancy Check (7, 4) Decoder Circuit for Serial Data

Theory/Description:

In this circuit, a CRC (Cyclic Redundancy Check) decoder has been simulated, with data word size (k) = 4 bits, and code word size (n) = 7 bits. This circuit can be used to decode serially generated 4-bit data after transmitting it over an error prone channel. At the sender side, the data word has to be encoded to detect any errors occurred during the transmission. On both sender and receiver sides, a common divisor of size (n-k+1) = 4 bits is used for encoding and decoding.

On the receiver side, the decoder generates a syndrome (remainder) of size (n-k) = 3 bits, from the received code word. If this syndrome is zero, no error is detected in the received code word and the extracted data word is accepted, otherwise error is detected and the extracted data word is discarded. The extracted data word is essentially the first four bits of the code word, starting from MSB.

In this decoder, the syndrome is generated using a shift register with three D flip-flops. The 7-bit code word is entered serially. Let this input be called 'serial_in'. Let the outputs of the first to last flip-flops be rm0 to rm2, which are the syndrome (remainder) bits. Let the divisor bits be dv3, dv2, d1 and dv0, where dv3 will always be 1. Now the states of the flip-flops – rm2, rm1 and rm0 – can be defined for each clock cycle by the following equations, where '&' implies the AND operation, and '^' implies the XOR operation:

- rm0 (t+1) = [rm2 (t) & dv0 (t)] ^ serial_in (t)
- rm1 (t+1) = [rm2 (t) & dv1 (t)] ^ rm0 (t)
- rm2 (t+1) = [rm2 (t) & dv2 (t)] ^ rm1 (t)

At the 7^{th} posedge of the clock, if all the syndrome bits are 0, the 'accept' bit is 1, otherwise it is 0, so we use a 3-input NOR gate. This NOR output undergoes the AND operation with a

square pulse, which is 0 until the 7th posedge, and 1 for the next clock cycle, so that the 'accept' bit can be 1 only in the 7th cycle.

This method is illustrated in Figures 1 and 2. The code words for the two cases are 1001_110 and 1000_110, and the divisor is 1011, same as that in the encoder. The 3-bit remainders in both the cases are called syndromes. In case 1, since the syndrome is 000, the extracted data word is accepted, while in case 2 it is discarded, since the syndrome is not 000.

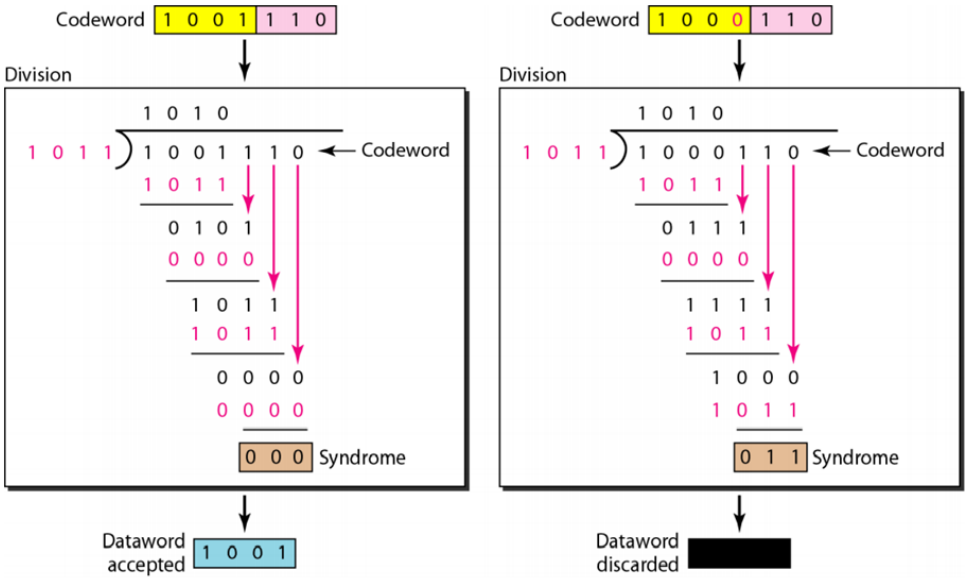

Figure 1: CRC (7, 4) Decoding Using Modulo 2 Division

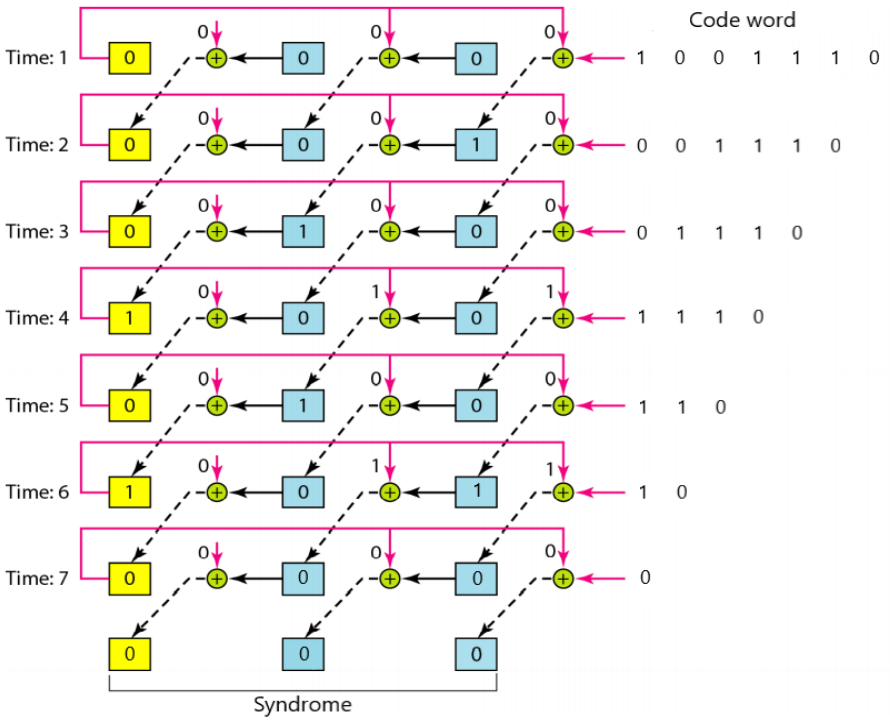

Figure 2: CRC (7, 4) Decoding with Serial Input – Working of the Circuit

Figure 3 shows the codebook for all possible 4-bit data words. Since the minimum Hamming distance between any two code words is 3, this technique can detect errors of up to 2 bits.

Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Figure 3: CRC (7, 4) Codebook

Circuit Diagrams:

A 3-input NOR gate subcircuit is used, for which the schematic and symbol are shown in Figures 4a and 4b respectively. Figure 5 shows the schematic of the main circuit.

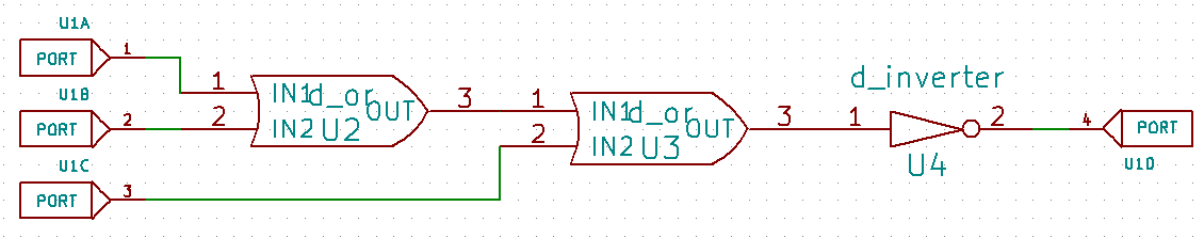

Figure 4a: 3-input NOR gate subcircuit schematic

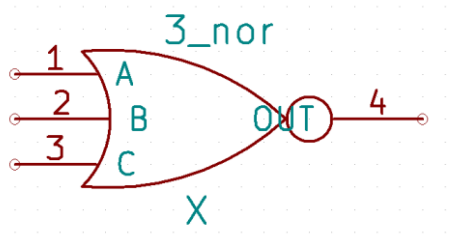

Figure 4b: 3-input NOR gate subcircuit symbol

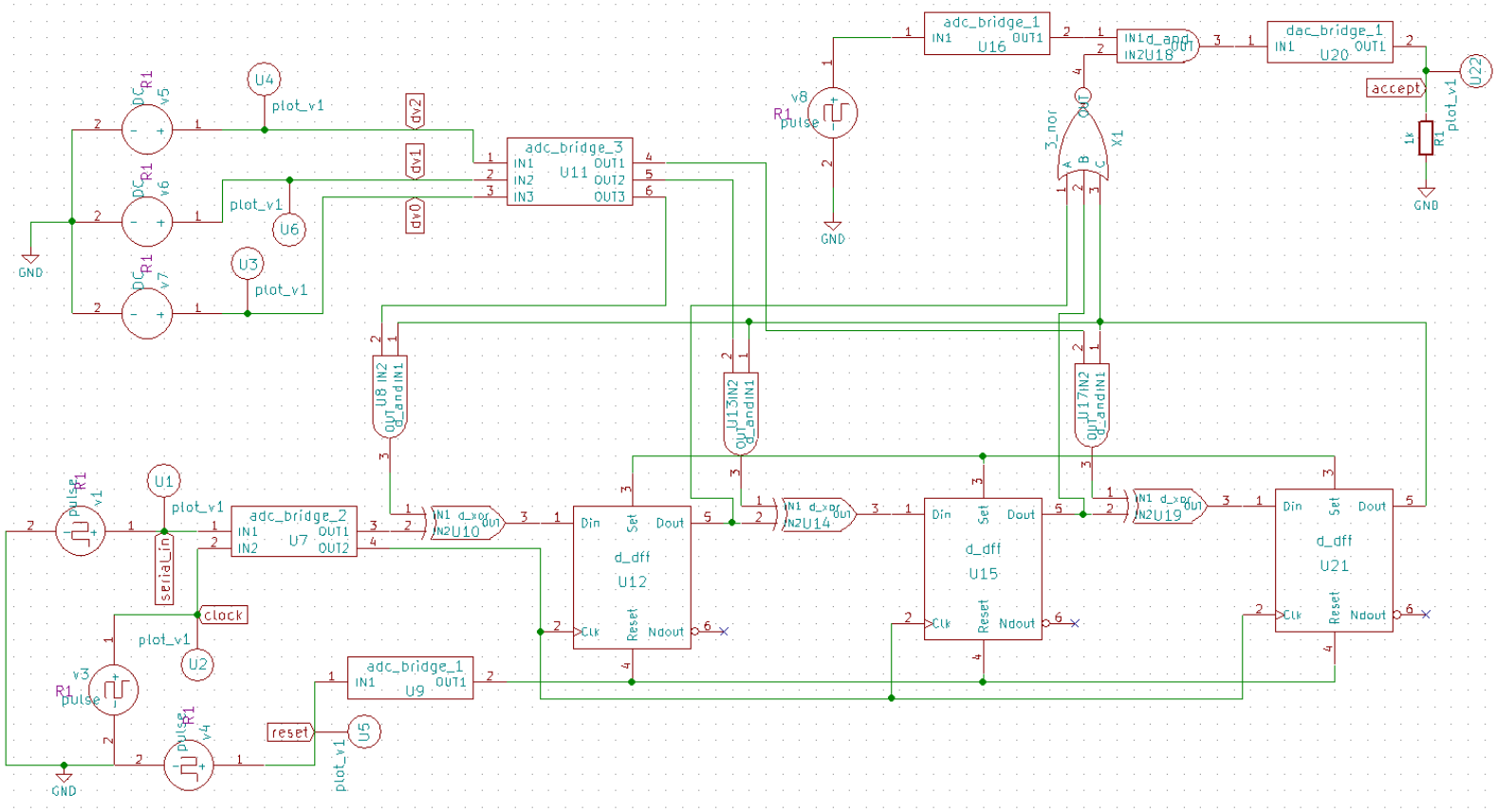

Figure 5: CRC (7, 4) Decoder for Serial Data - Schematic

Results (Input, Output waveforms):

The working of the circuit in Figures 1 and 2 has been demonstrated in the following results. Both the cases in Figure 1 have been simulated. The divisor bits dv2, dv1 and dv0 have been set to 0V, 5V and 5V respectively, so that the divisor becomes 1011. A reset pulse for the first 5 seconds is used to reset the flip-flops initially. The clock signal is a square wave of period 20s with a 50% duty cycle. The signals 'serial_in' are the serial input code words 1001_110 and 1000_110. The syndrome bits obtained from the outputs of the flip-flops undergo the NOR operation. The final 'accept' bit is obtained at the rising edge of the 7th clock pulse, i.e., at the 130th second. This can be sampled anytime from 130 to 150 seconds. The syndrome bits for case 1 at each clock cycle can be verified from Figure 2, and the 'accept' bit after the 7th posedge can be verified for both cases from Figure 1.

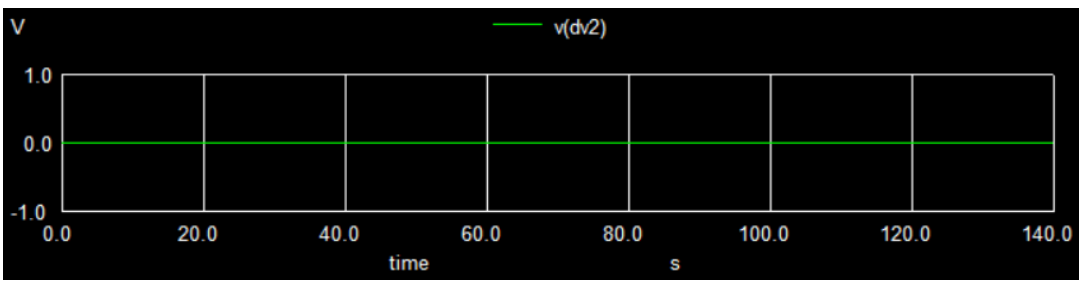

Figure 6a: Analog signal for dv2

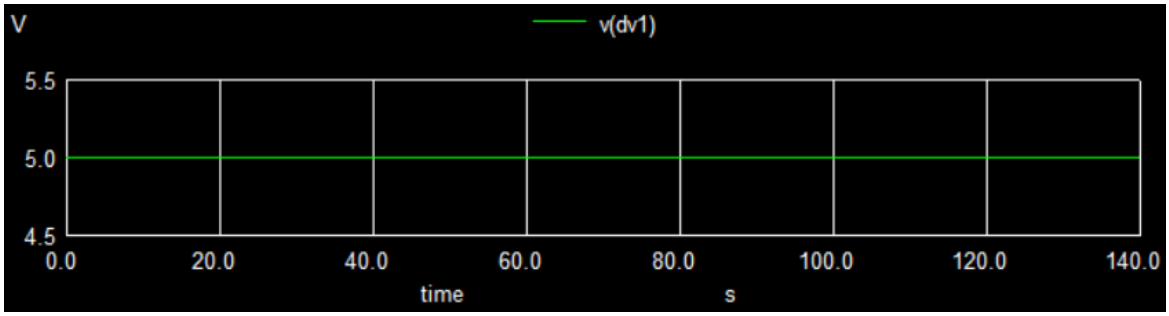

Figure 6b: Analog signal for dv1

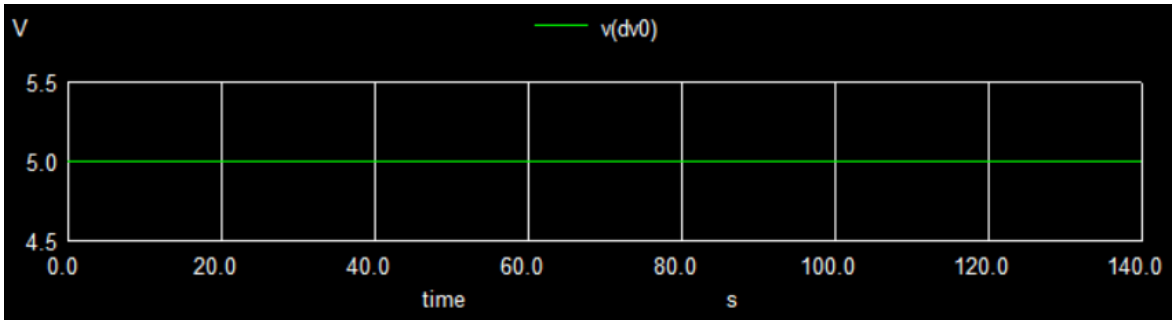

Figure 6c: Analog signal for dv0

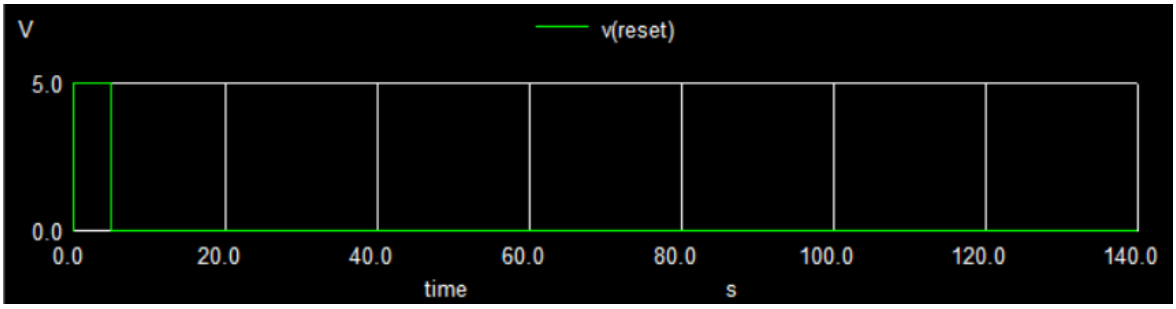

Figure 7: Analog signal for reset pulse

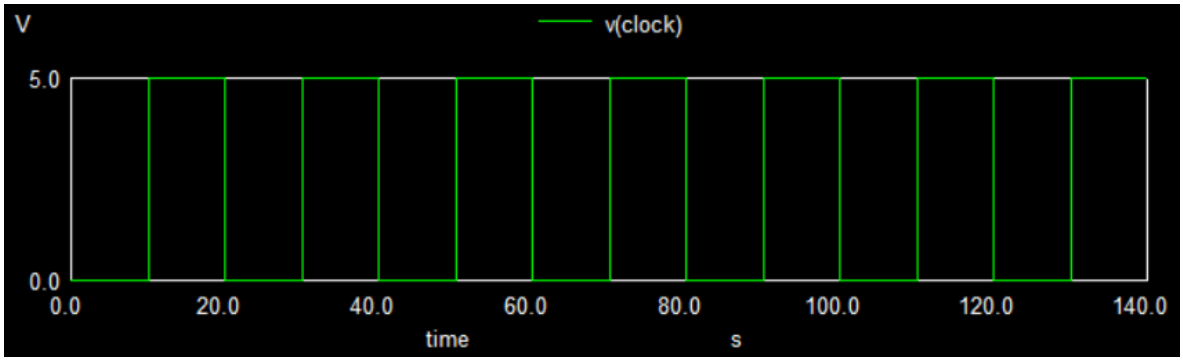

Figure 8: Analog signal for clock

CASE 1: Code word = 1001_110, data word is accepted, ‘accept’ bit is 1 at 7th posedge.

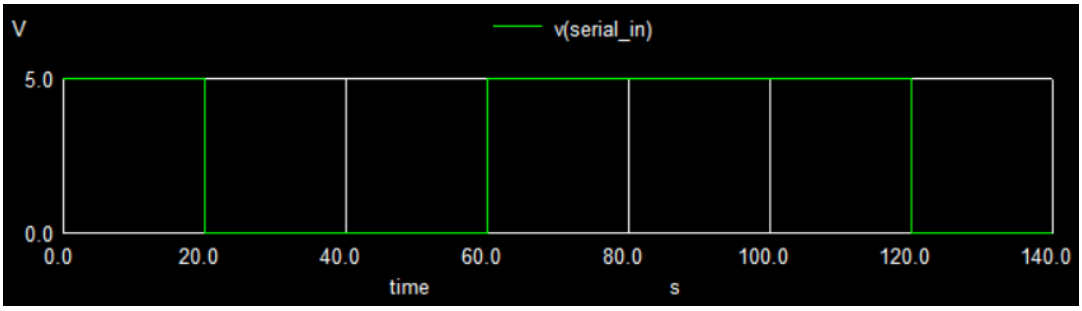

Figure 9: Analog signal for serial_in (1001_110)

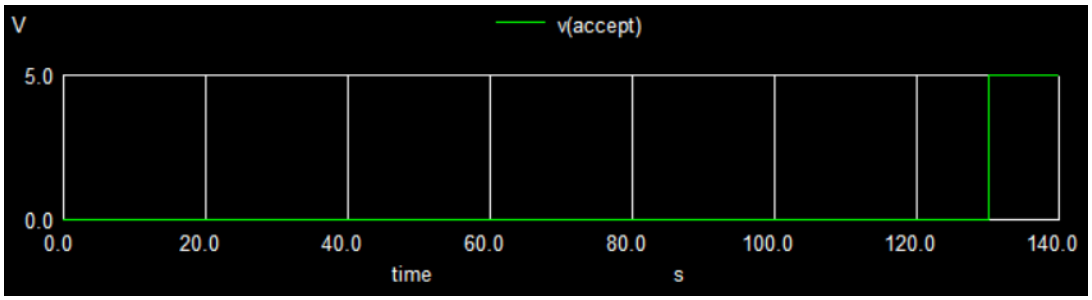

Figure 10: Analog signal for ‘accept’

CASE 2: Code word = 1000_110, data word is discarded, ‘accept’ bit is 0 at 7th posedge.

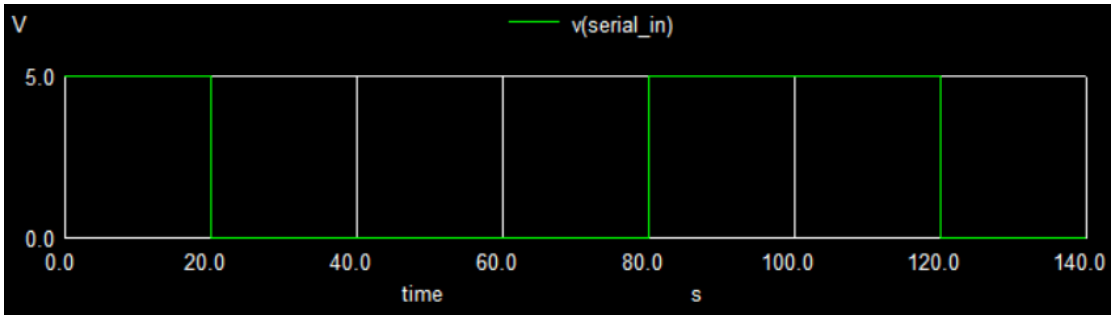

Figure 11: Analog signal for serial_in (1000_110)

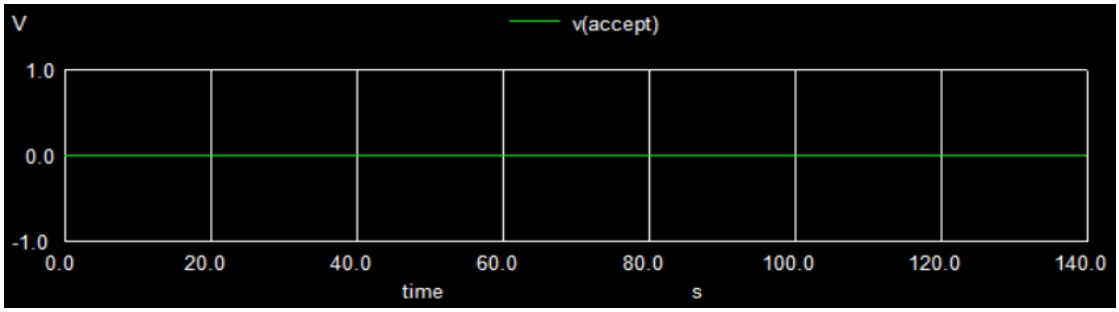

Figure 12: Analog signal for ‘accept’

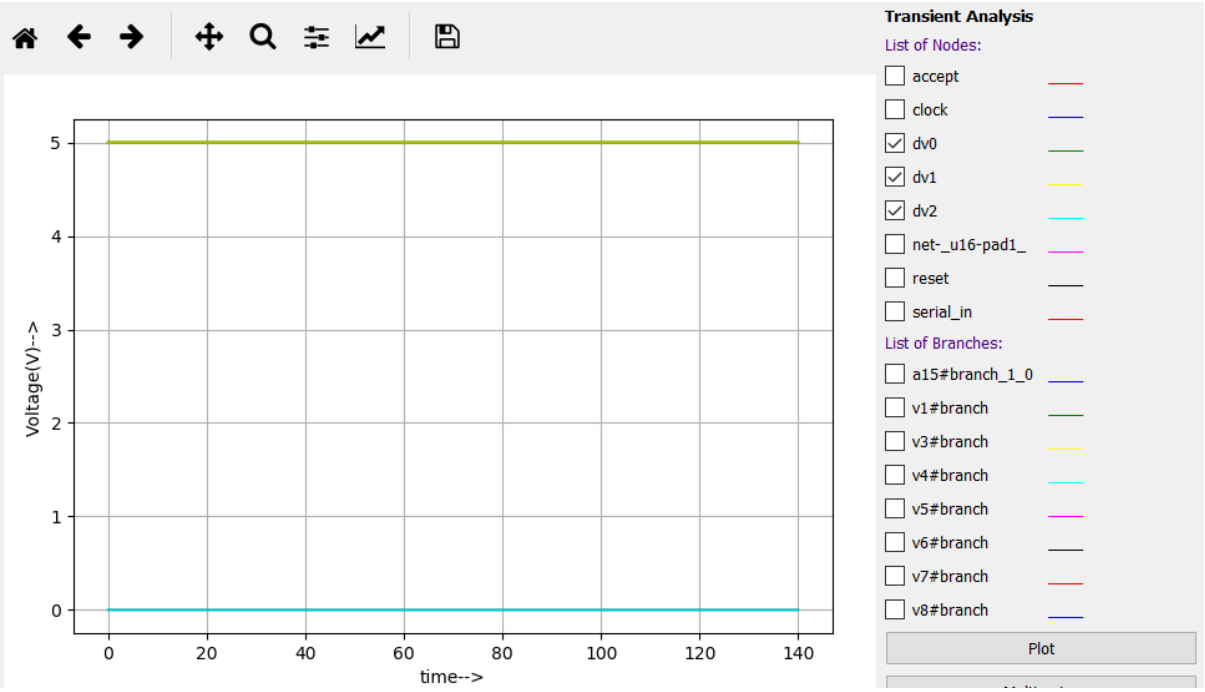

Figure 13: Analog signals for dv2 to dv0

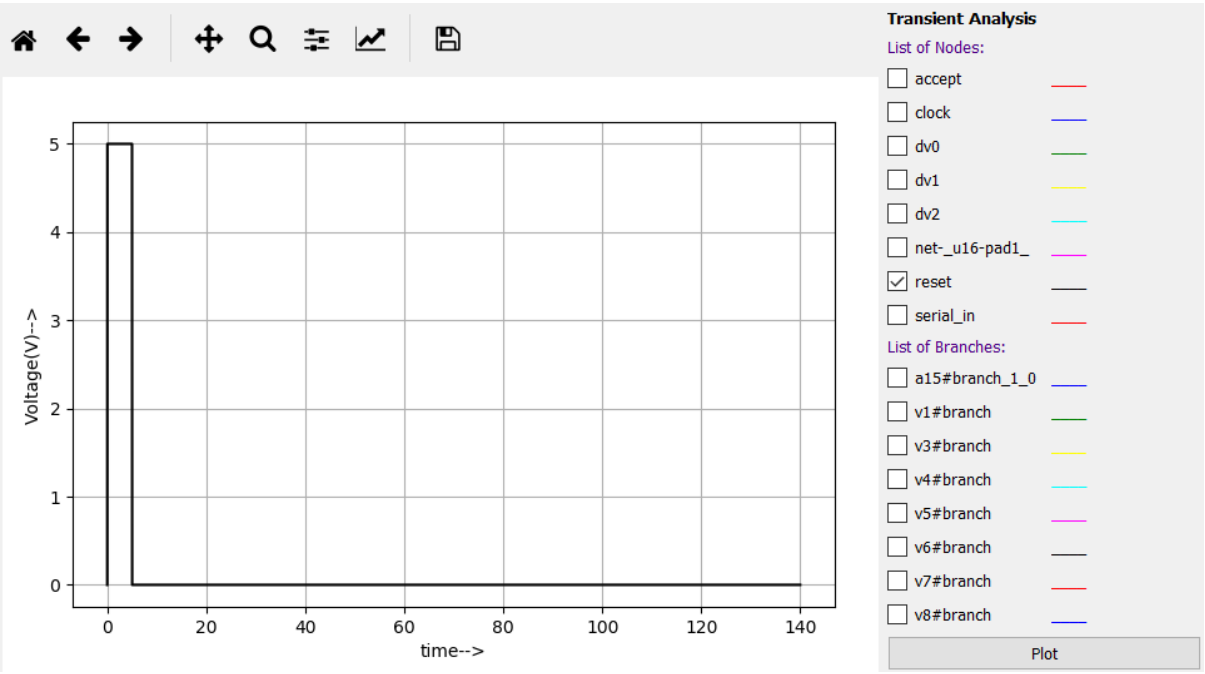

Figure 14: Analog signal for reset pulse

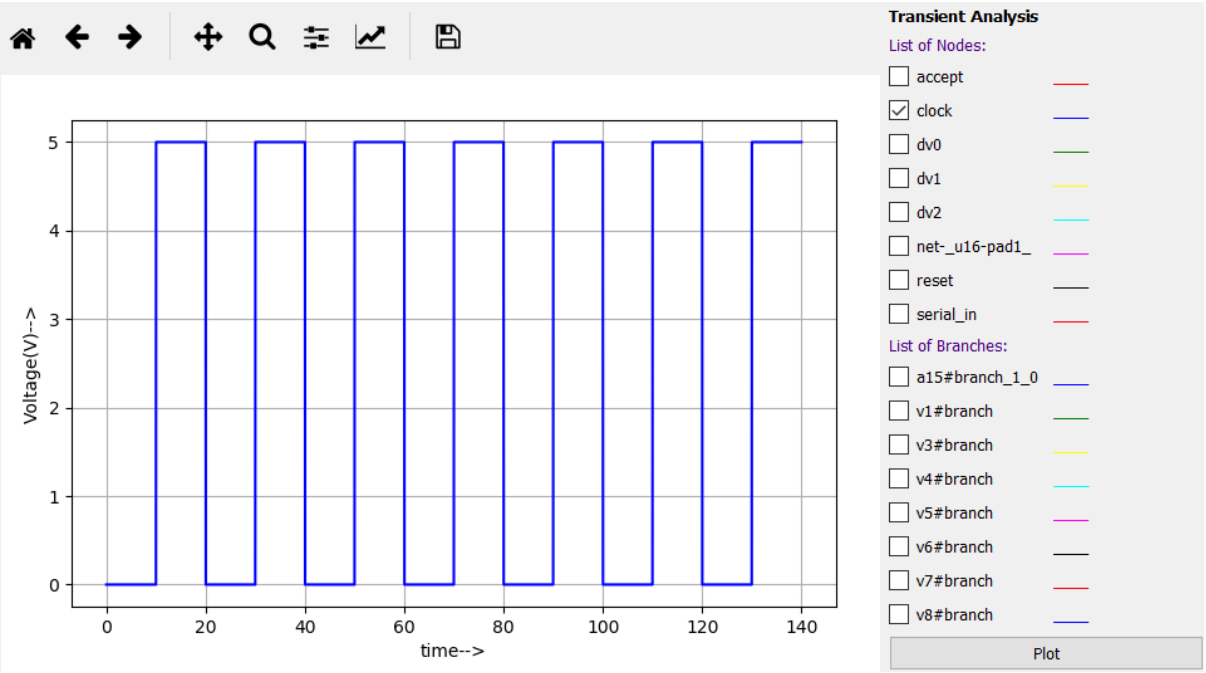

Figure 15: Analog signal for clock

CASE 1: Code word = 1001_110, data word is accepted, 'accept' bit is 1 at 7th posedge.

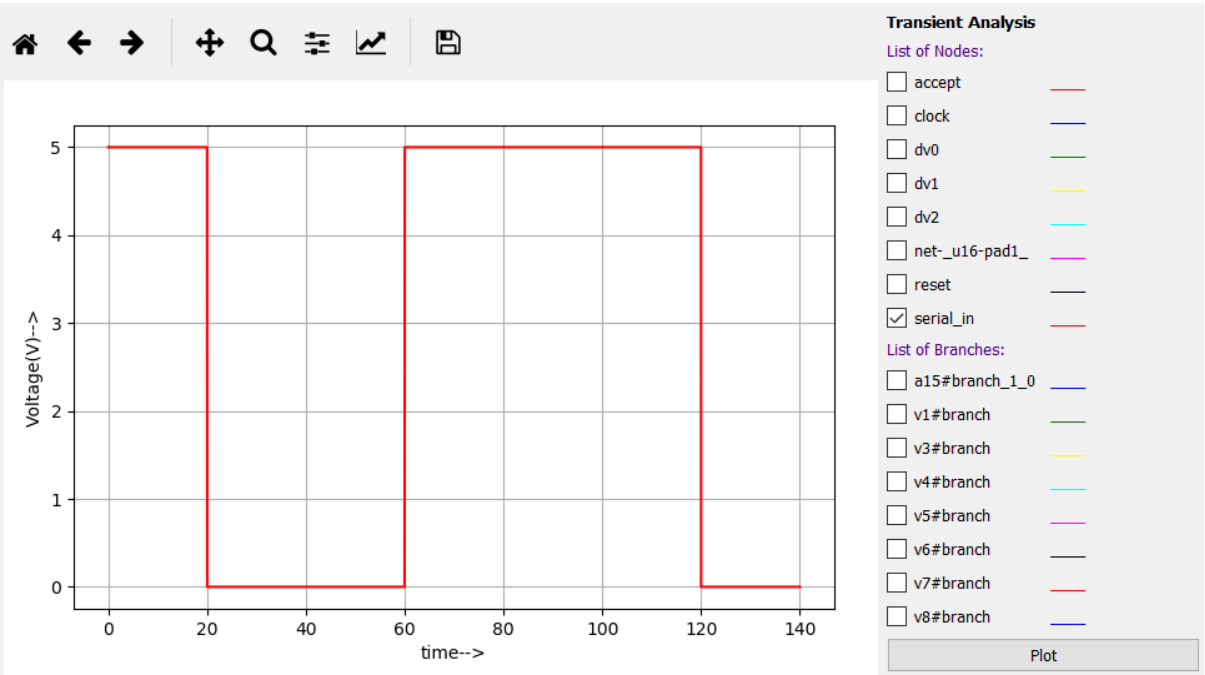

Figure 16: Analog signal for serial_in (1001_110)

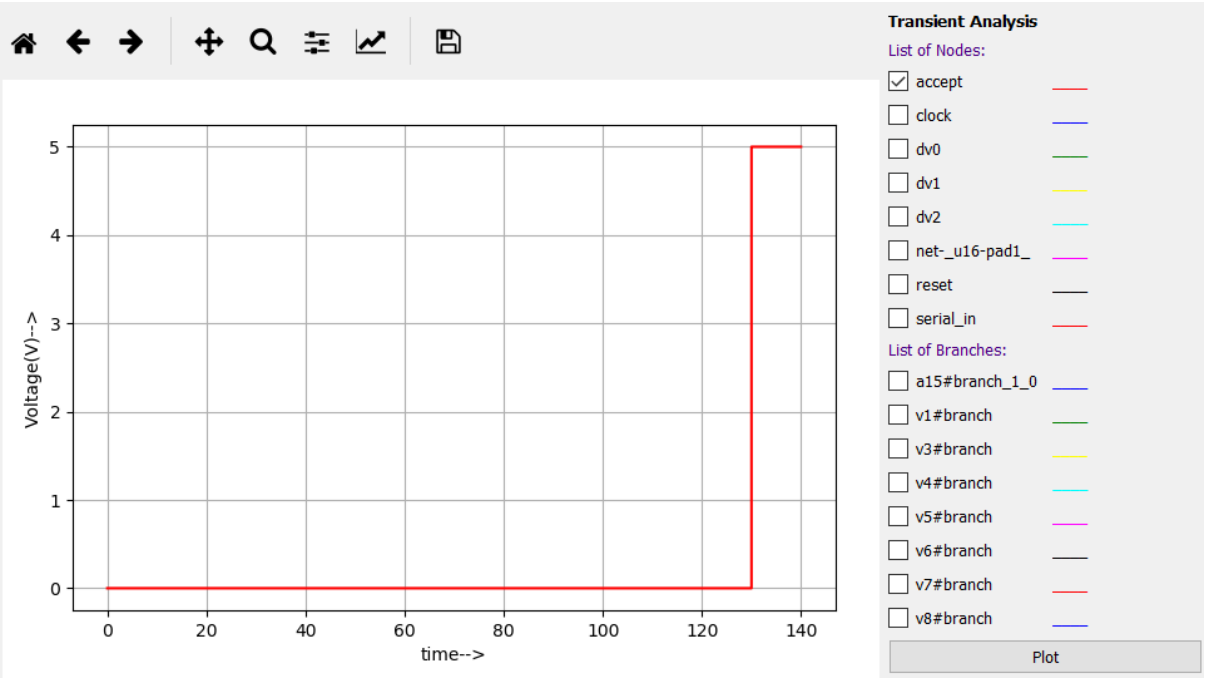

Figure 17: Analog signal for 'accept'

CASE 2: Code word = 1000_110, data word is discarded, 'accept' bit is 0 at 7th posedge.

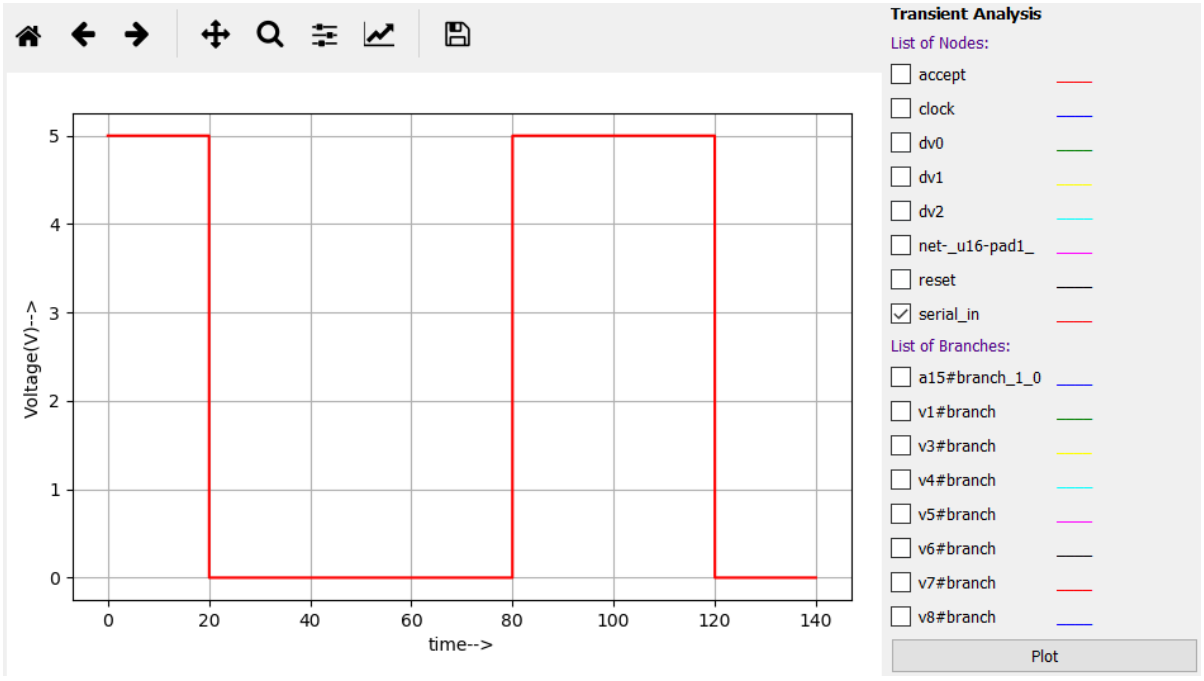

Figure 18: Analog signal for serial_in (1000_110)

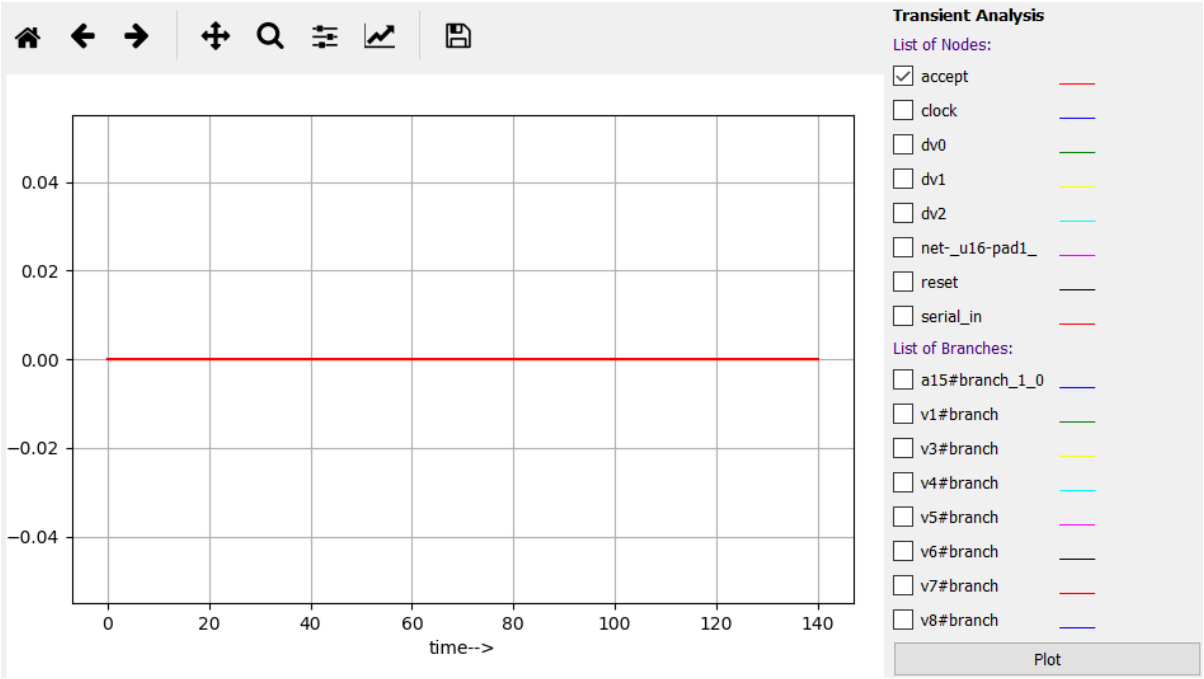

Figure 19: Analog signal for 'accept'

Simulation Parameters for reference:

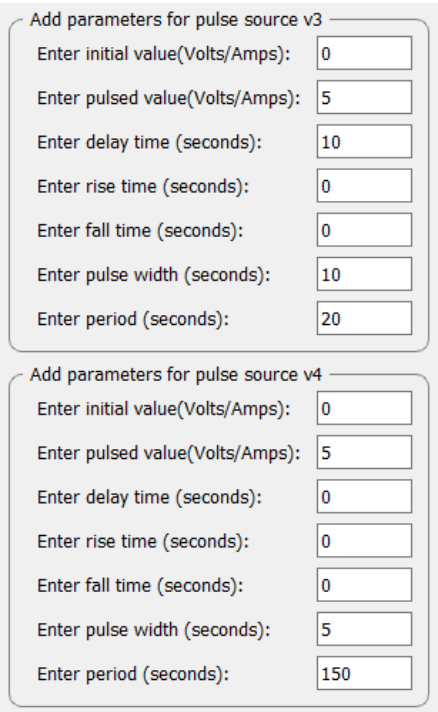

Figure 20a

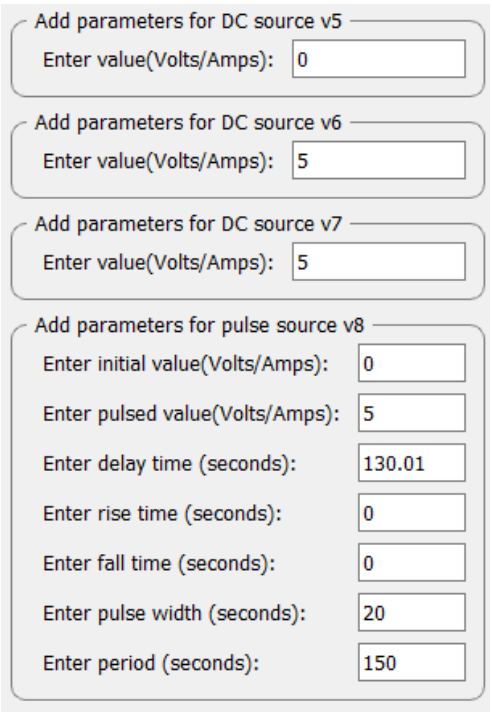

Figure 20b

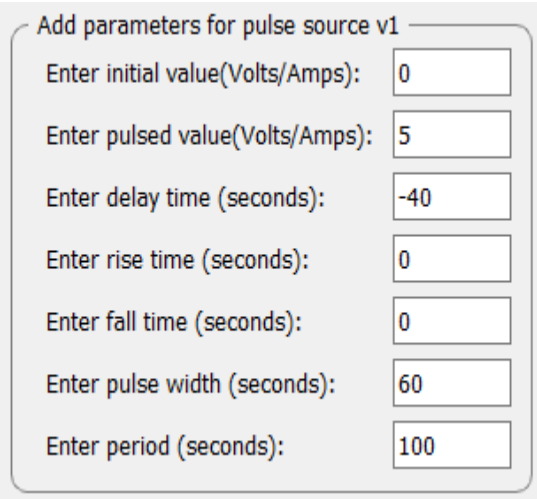

Figure 20c: CASE 1

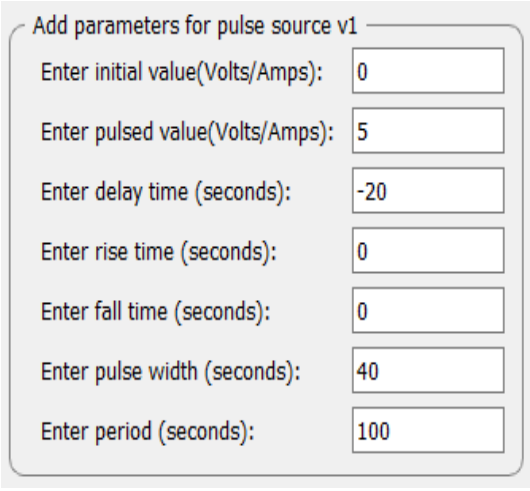

Figure 20d: CASE 2

Source/Reference(s):

https://cse.iitkgp.ac.in/~ksrao/pdf/iti-18/slide-3.pdf

Pages 58-65