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) Encoder Circuit

Theory/Description:

In this circuit, a CRC (Cyclic Redundancy Check) encoder has been simulated, with data word size (k) = 4 bits, and code word size (n) = 7 bits. This circuit can be used to encode serially generated 4-bit data before transmitting it over an error prone channel. At the receiver side, this encoded code word has to be decoded 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.

The encoder generates a remainder of size (n-k) = 3 bits and appends it to the data word before transmission. This transmitted data is the code word. On the receiver side, the encoder generates a syndrome of size (n-k) = 3 bits. 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.

In this encoder, the remainder is generated using a shift register with three D flip-flops. The 7-bit augmented data 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 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)

This method is illustrated in Figures 1 and 2. The augmented data word, i.e., serial_in, is 1001_000, and the divisor is 1011. This same divisor is to be used in the decoder as well. The 3-bit remainder, 110 is appended to the data word to generate the 7-bit code word.

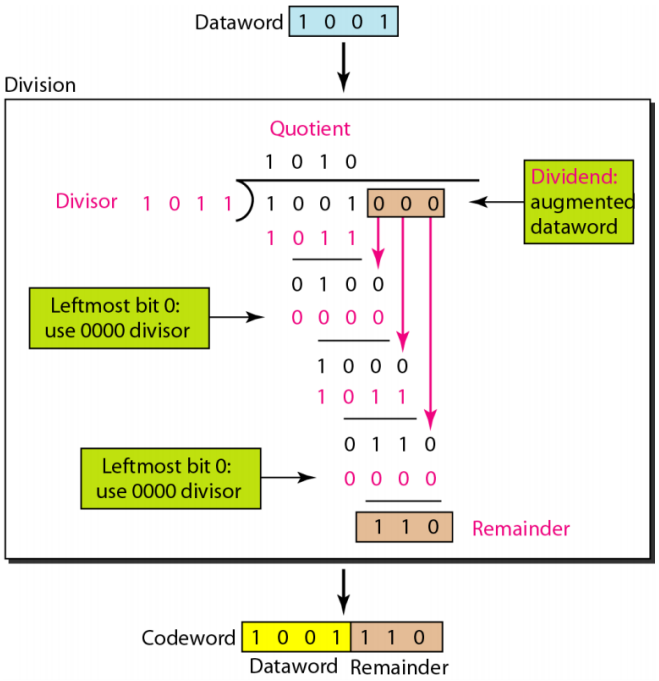

Figure 1: CRC (7, 4) Encoding Using Modulo 2 Division

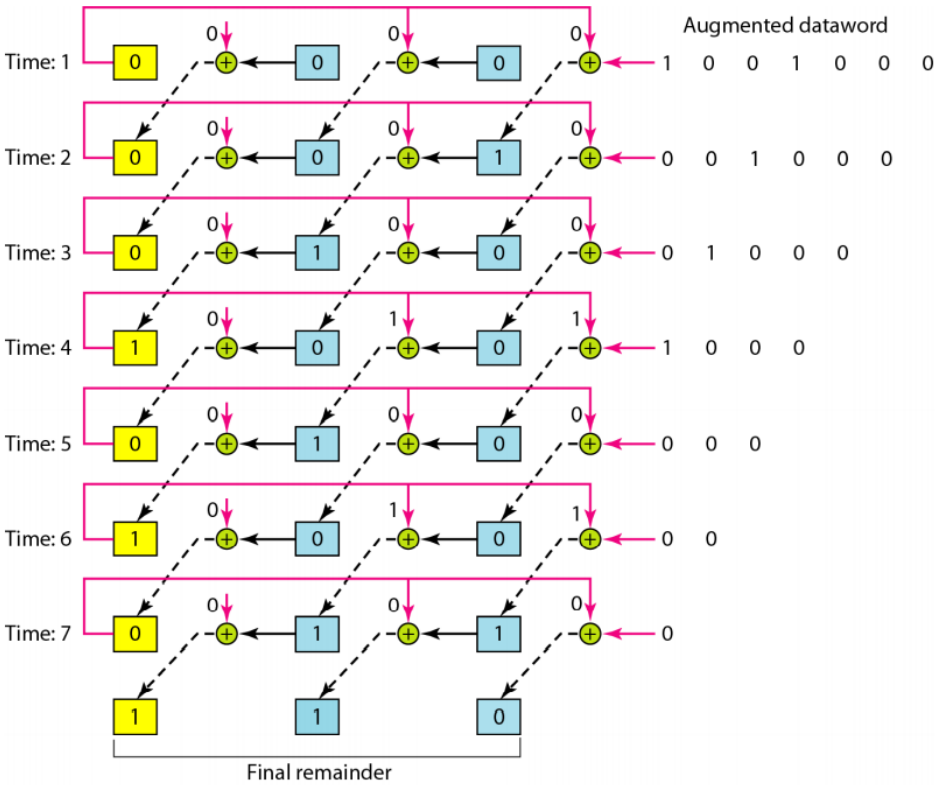

Figure 2: CRC (7, 4) Encoding with Serial Input – Working of the Circuit

Figure 3 shows the codebook for all possible 4-bit data words.

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 Diagram:

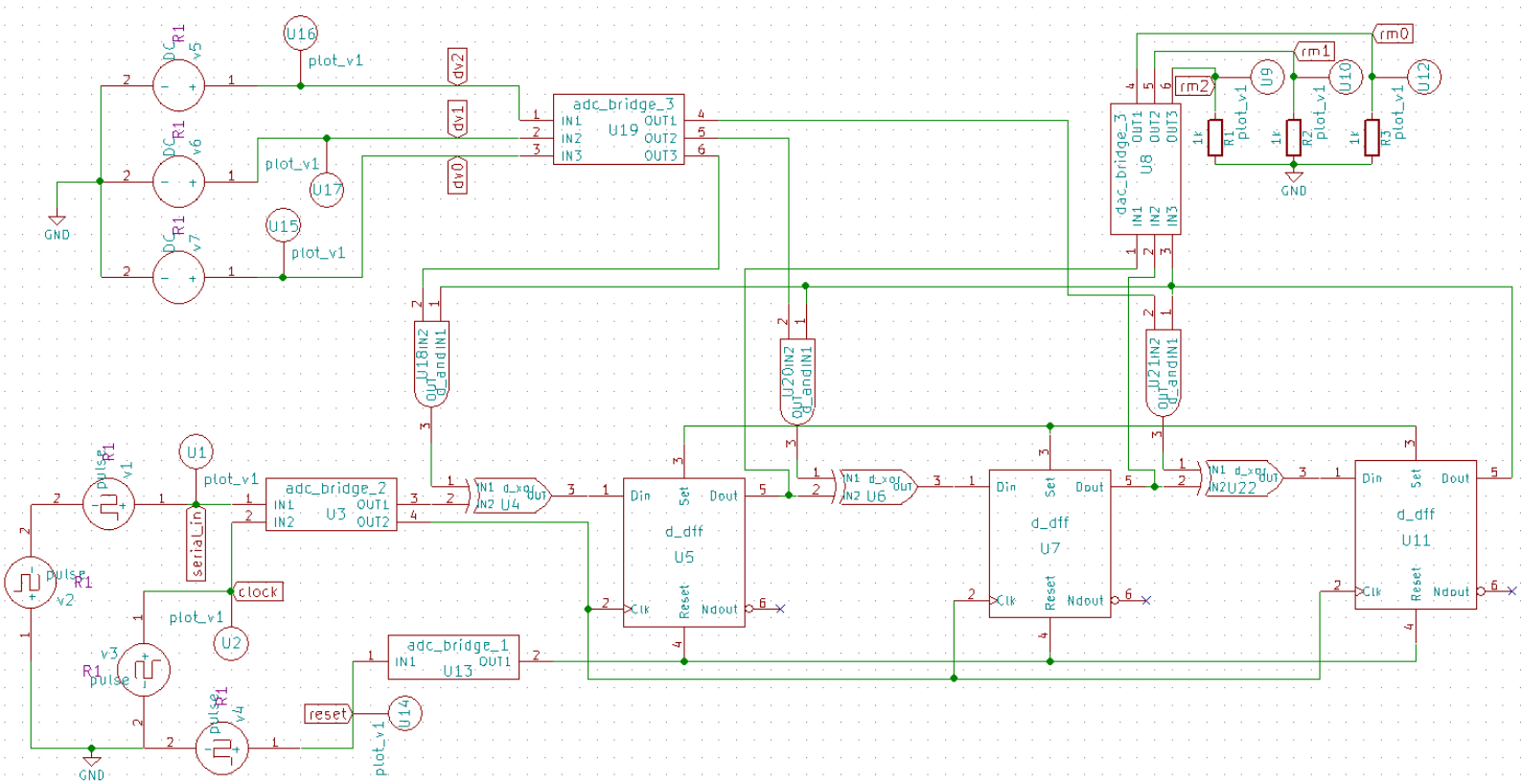

Figure 4: CRC (7, 4) Encoder Schematic

Results (Input, Output waveforms):

The working of the circuit in Figure 2 has been demonstrated in the following results. 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 signal 'serial_in' is the serial input data word 1001 with three 0s in the end, forming the augmented data word 1001_000. The signals rm2, rm1 and rm0 are the remainder bits obtained from the outputs of the flip-flops. The final remainder is obtained at the rising edge of the 7^{th} clock pulse, i.e., at the 130^{th} second. This can be sampled anytime from 130 to 150 seconds.

The remainders bits at each clock cycle can be verified from Figure 2.

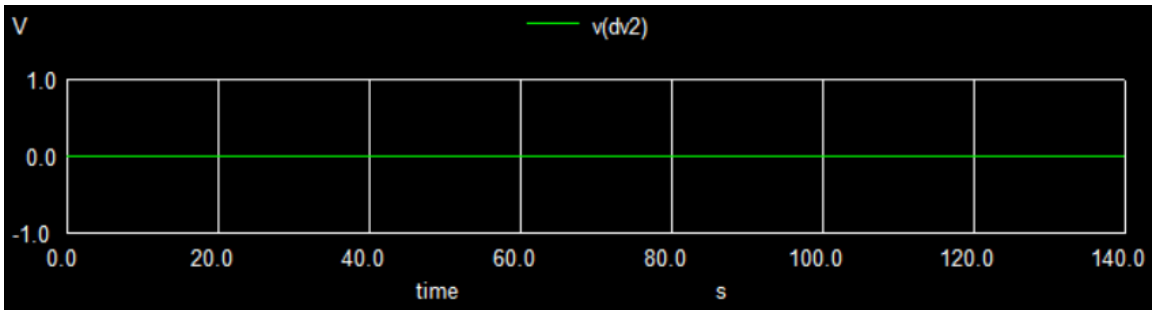

Figure 5a: Analog signal for dv2

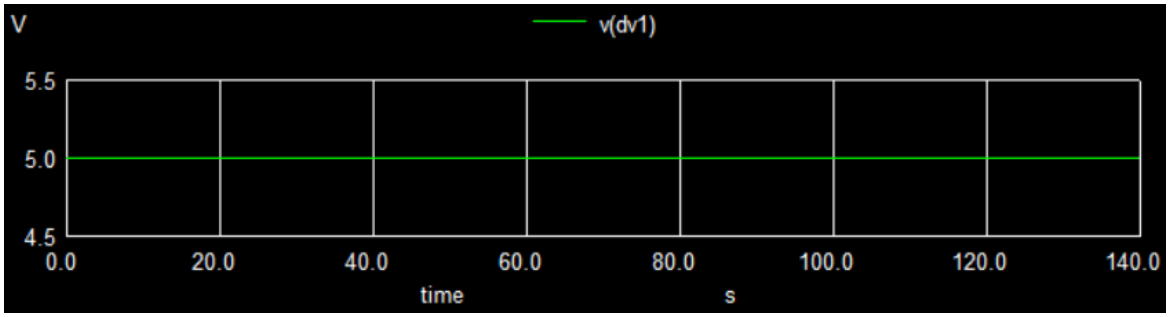

Figure 5b: Analog signal for dv1

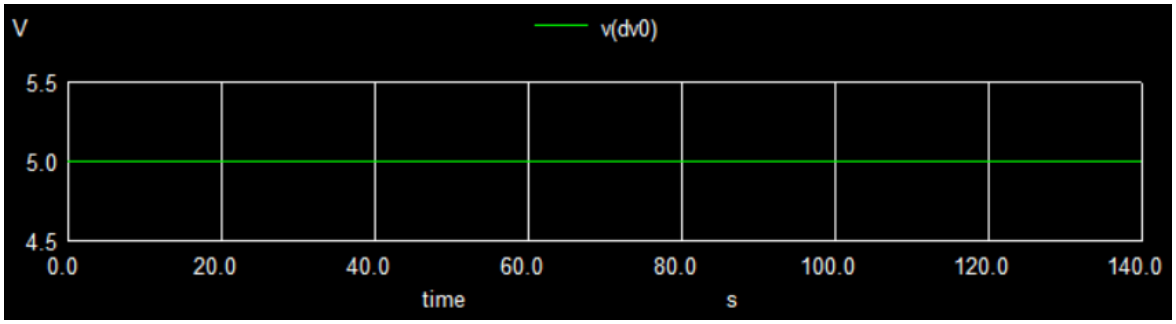

Figure 5c: Analog signal for dv0

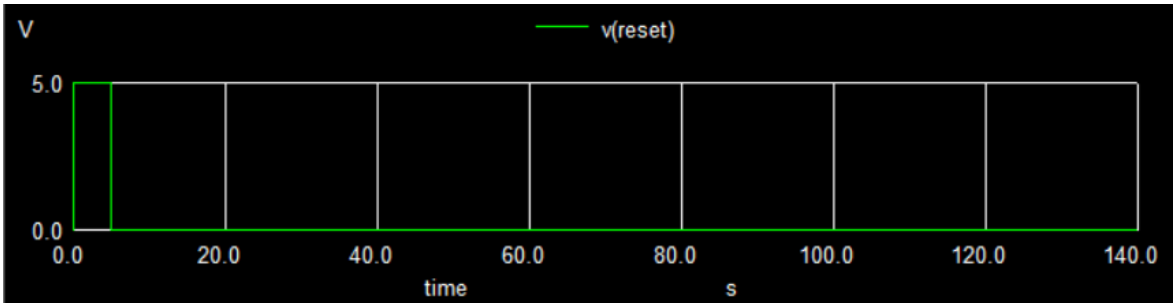

Figure 6: Analog signal for reset pulse

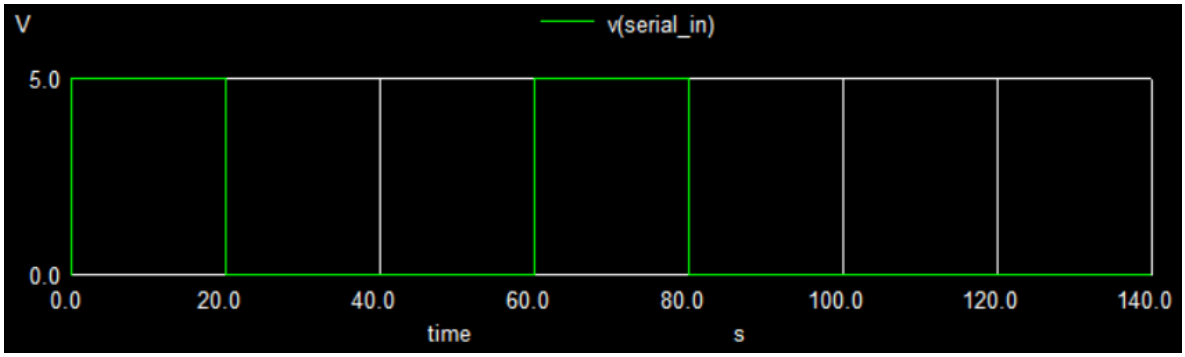

Figure 7: Analog signal for serial_in (1001_000)

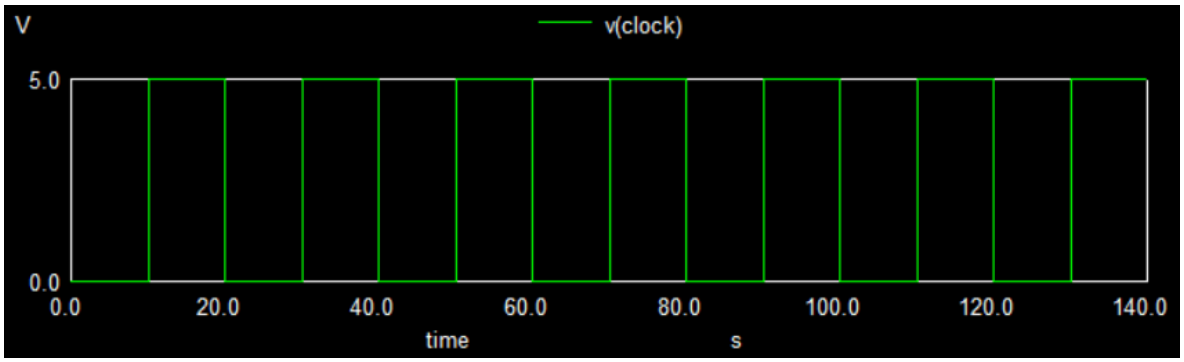

Figure 8: Analog signal for clock

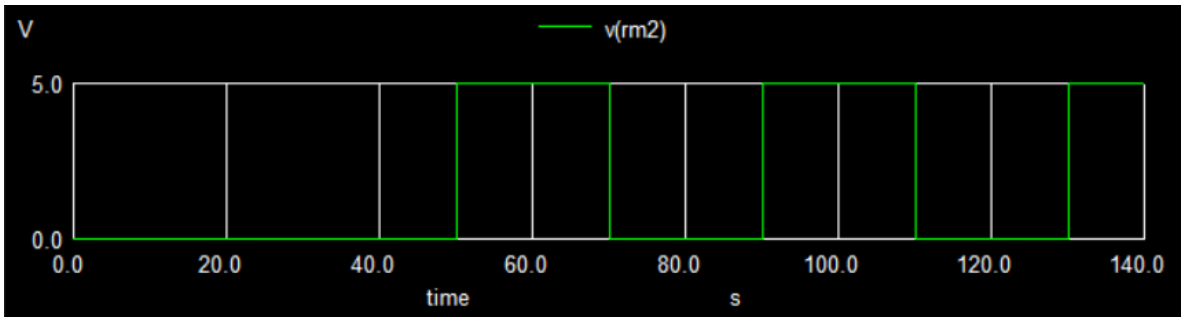

Figure 9a: Analog signal for rm2

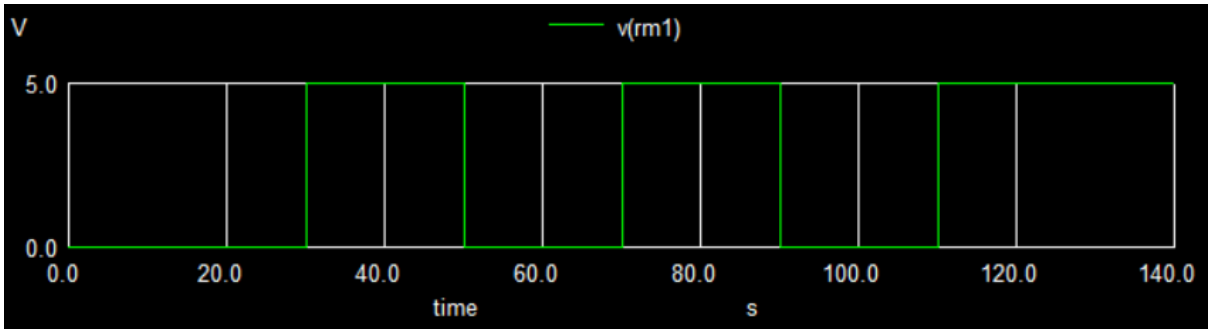

Figure 9b: Analog signal for rm1

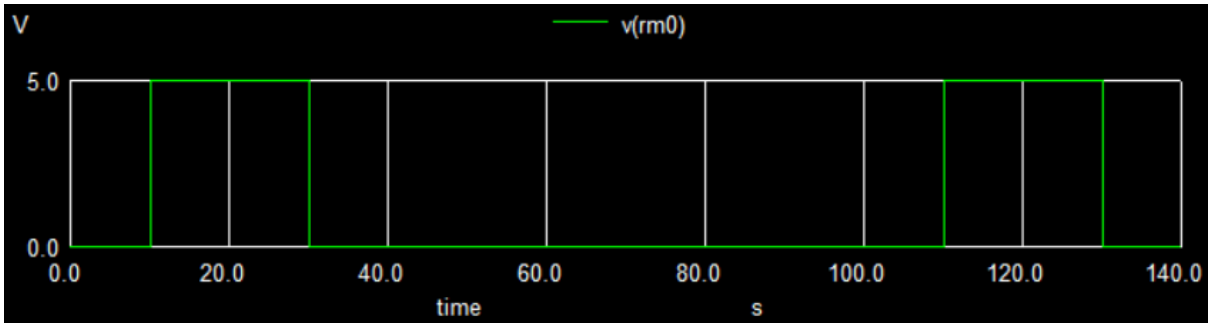

Figure 9c: Analog signal for rm0

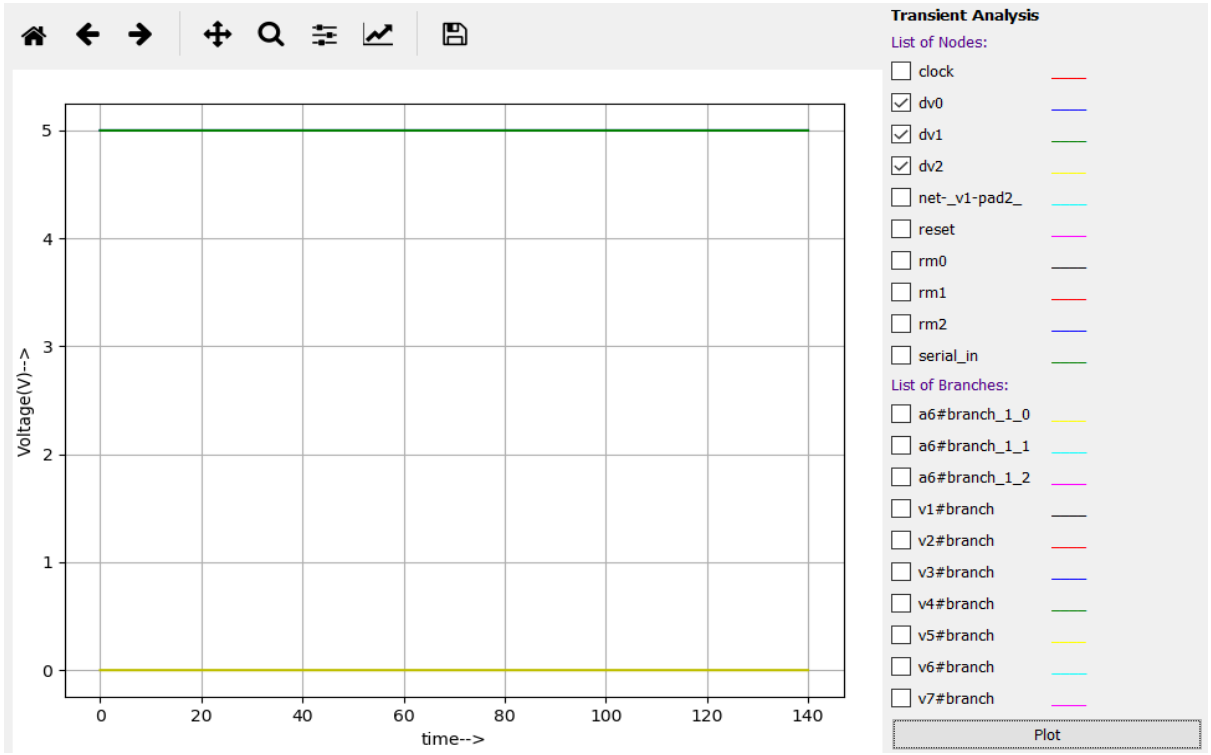

Figure 10: Analog signals for dv2 to dv0

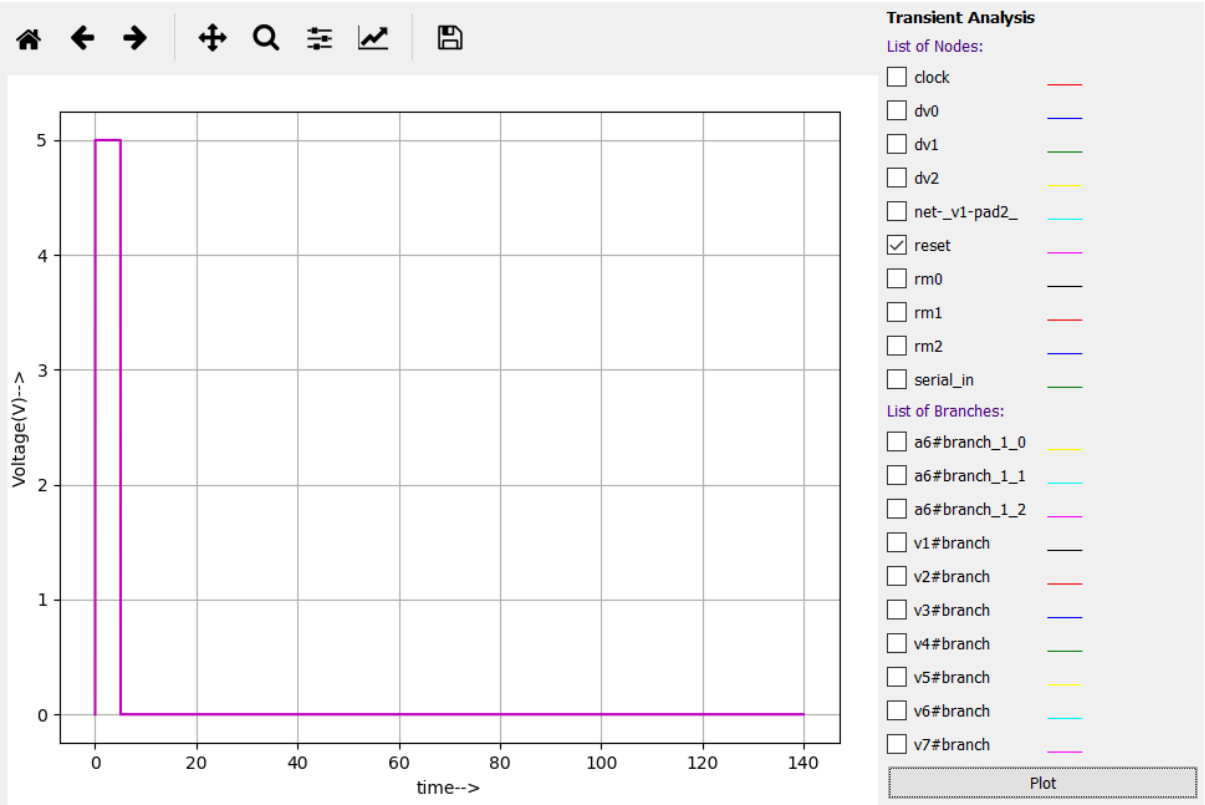

Figure 11: Analog signal for reset pulse

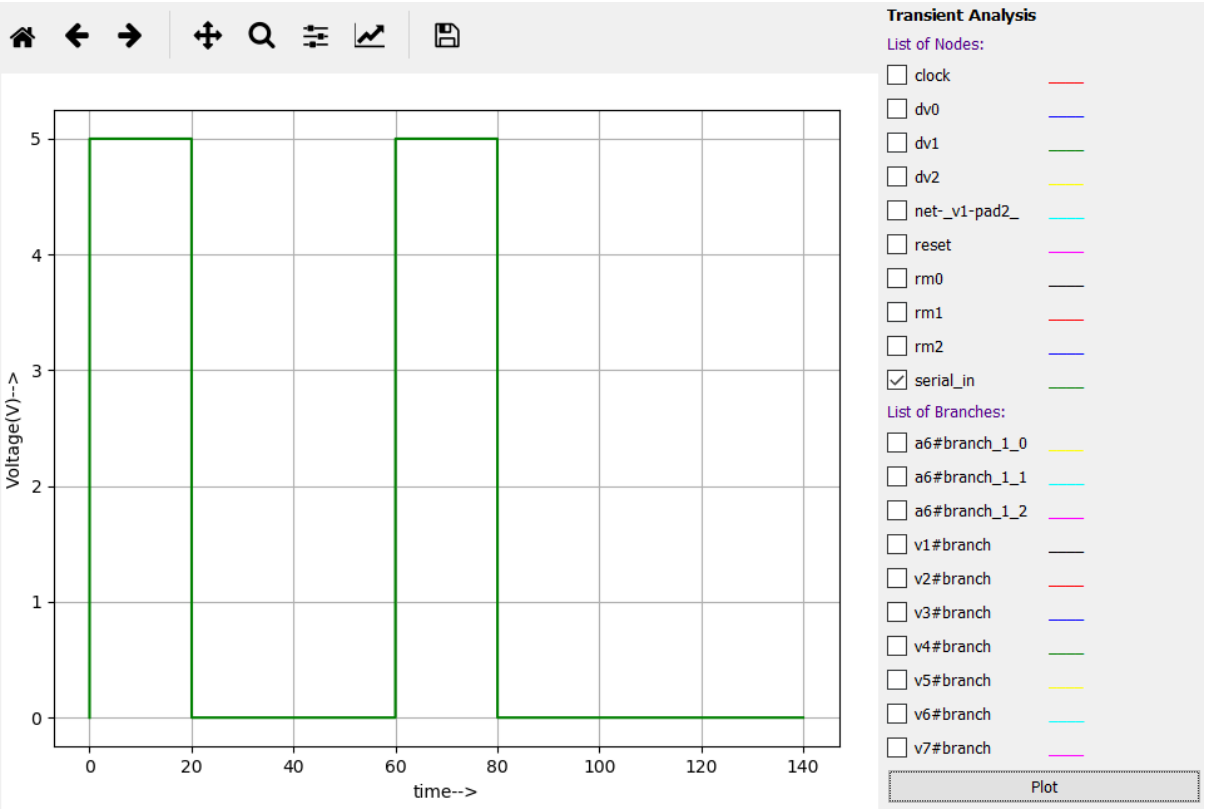

Figure 12: Analog signal for serial_in

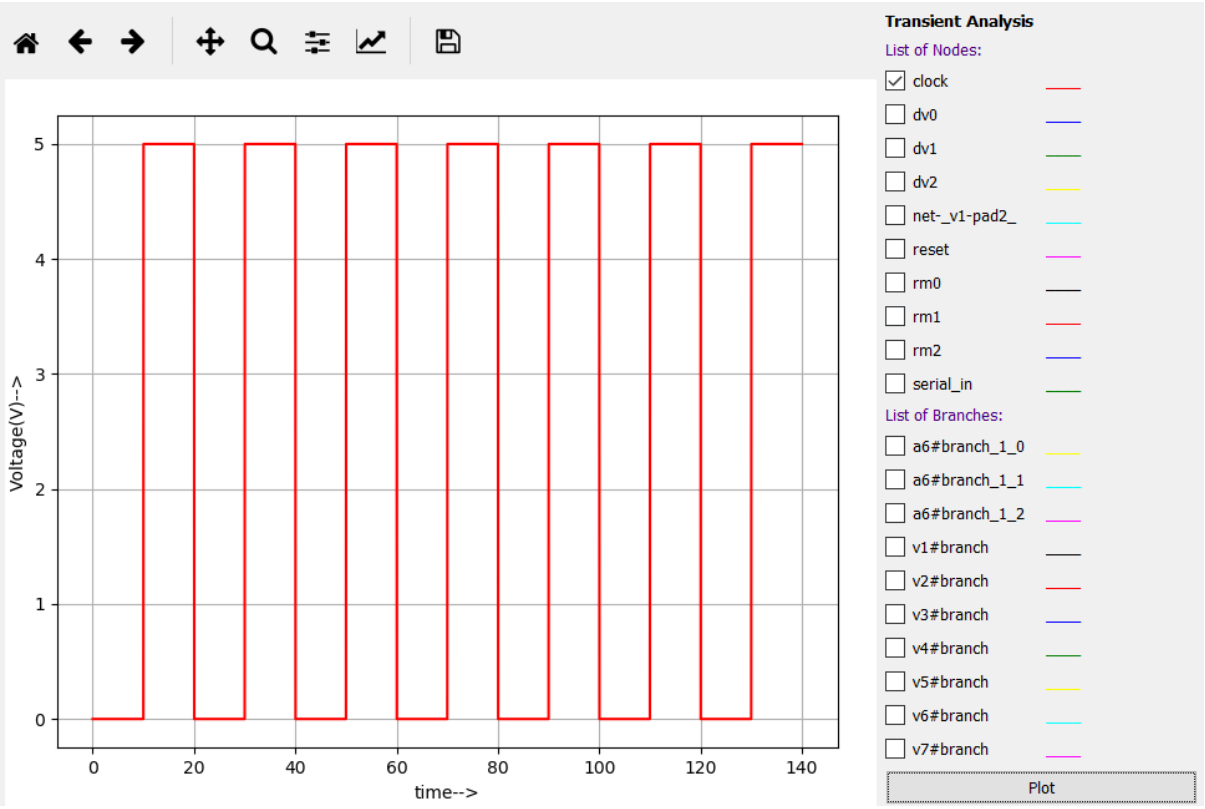

Figure 13: Analog signal for clock

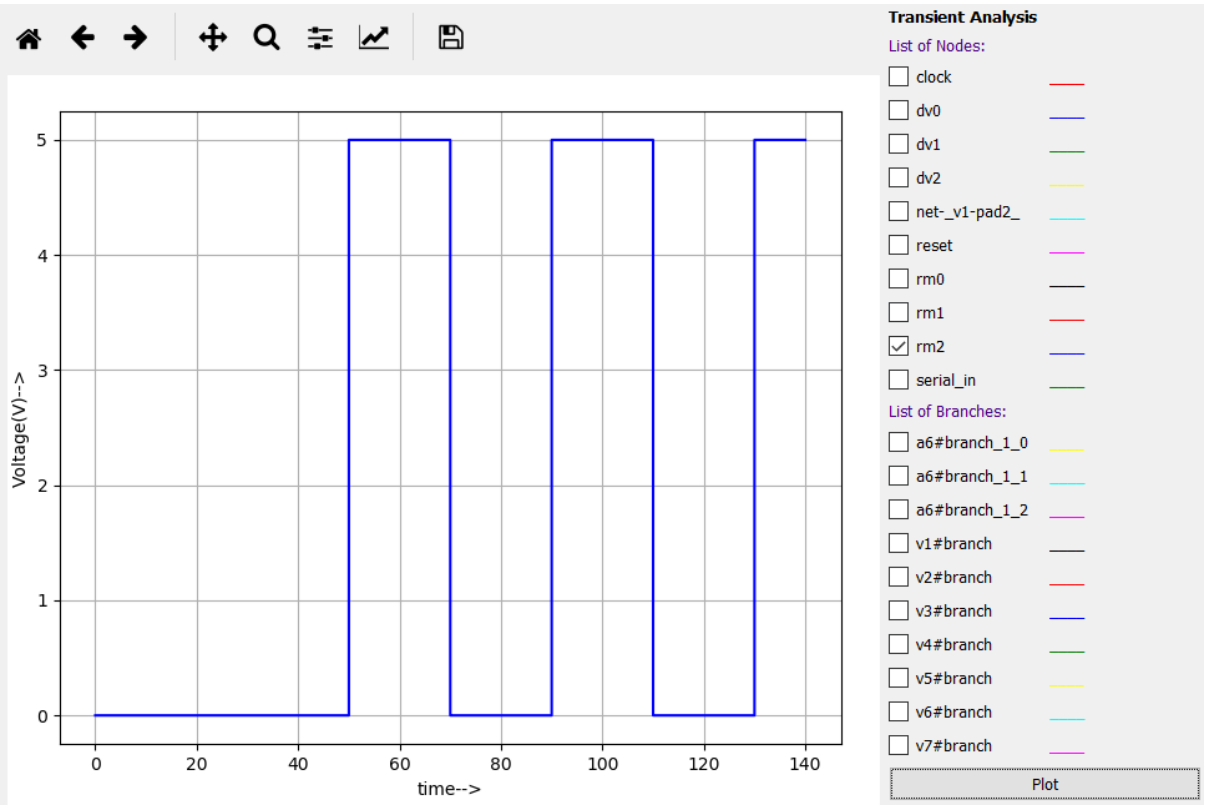

Figure 14a: Analog signal for rm2

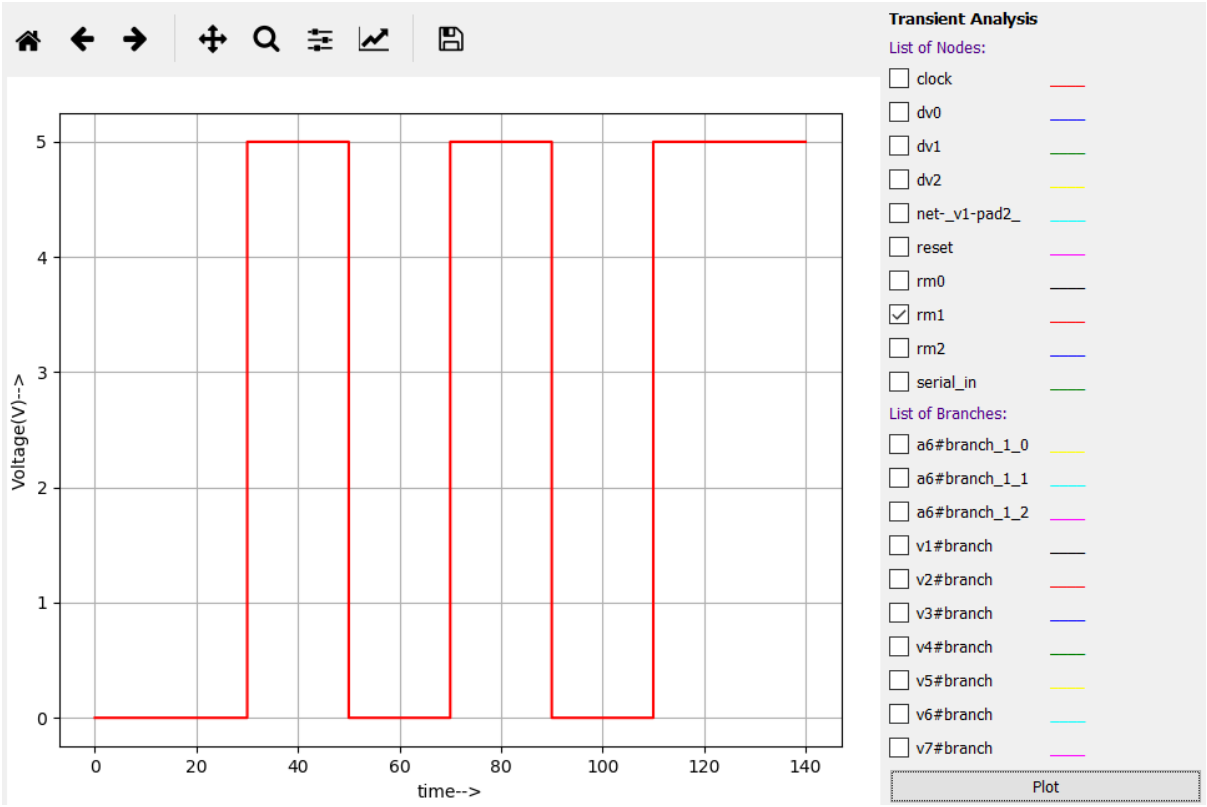

Figure 14b: Analog signal for rm1

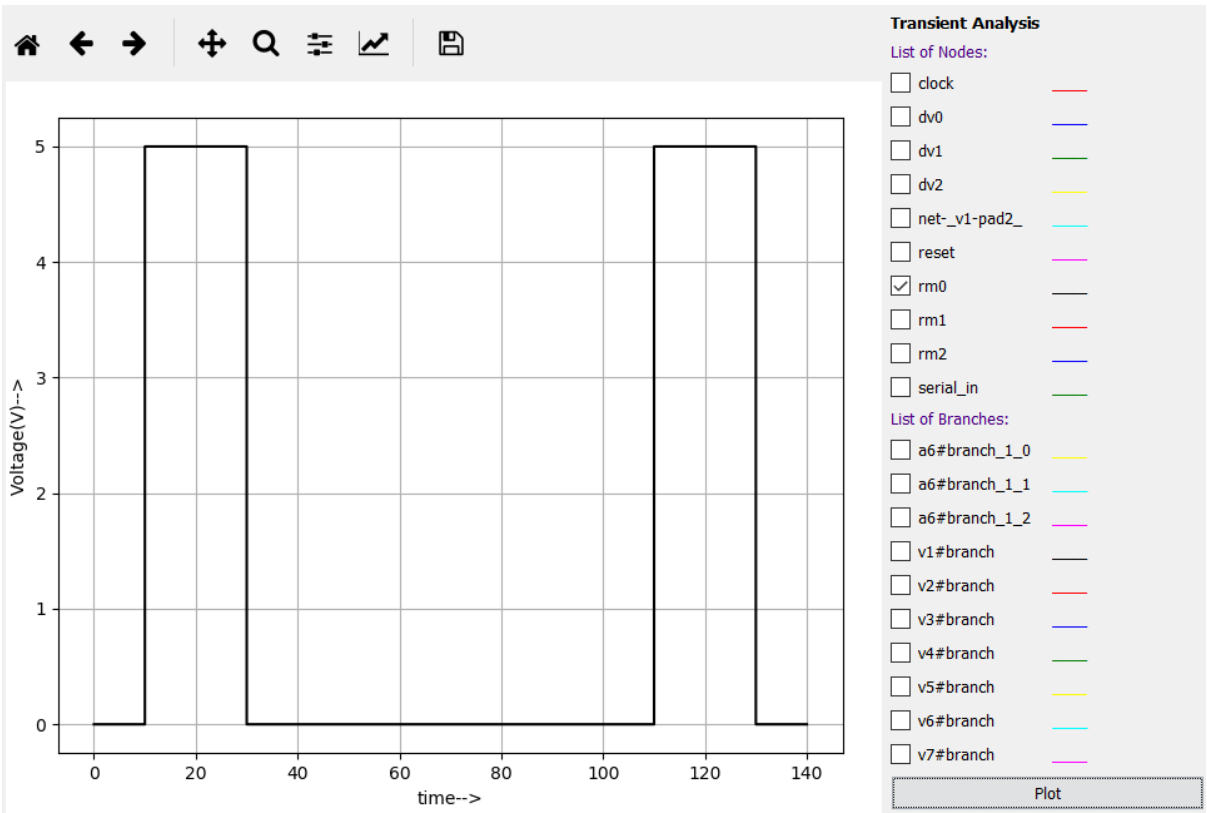

Figure 14c: Analog signal for rm0

Simulation Parameters for reference:

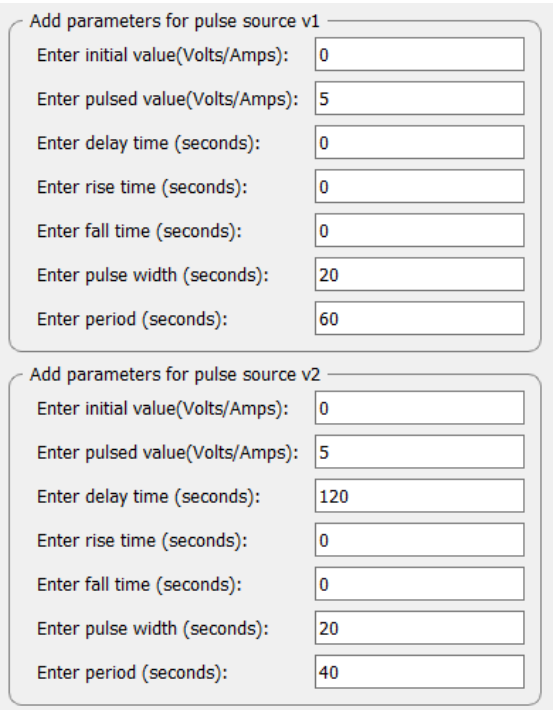

Figure 15a

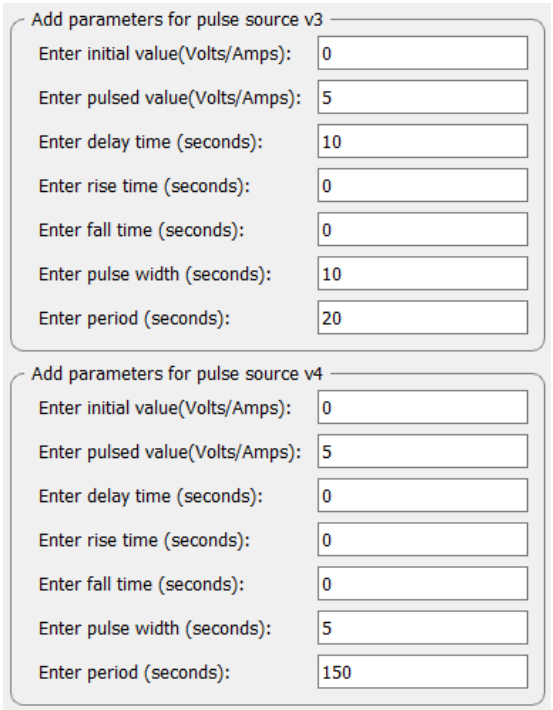

Figure 15b

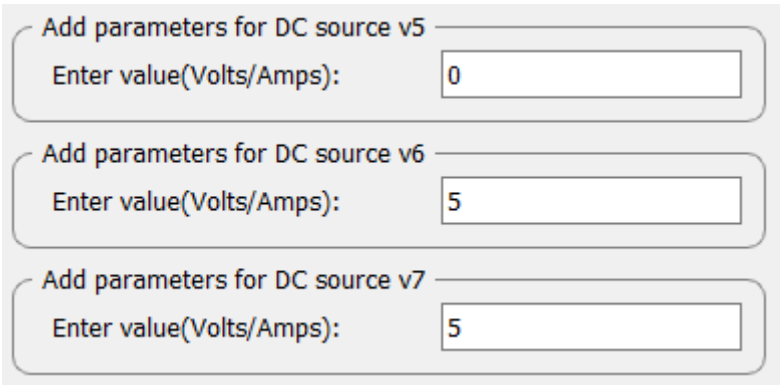

Figure 15c

Source/Reference(s):

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

Pages 58-64