# 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:** Programmable Frequency Divider

**Theory/Description:**

Advanced microcontrollers and microprocessors have different clock sources with different frequencies, but most of the times these frequencies are not desired, and they need to be divided before use. For this purpose, these devices use frequency divider systems. These systems use control bits to divide the clock frequencies by the desired integers.

In this project, an industry grade frequency divider is designed and simulated. This circuit uses 2 multiplexers, one 4:1 and one 8:1. The output of the first mux can be the input clock signal with its frequency divided by 1, 2, 4 or 8. Three D flip-flops are used in series for this operation, each divides the frequency by 2. In each flip-flop, the input is connected to the complemented output of the same flip-flop. The clock input for the first flip-flop is the original clock signal, and for the subsequent flip-flops it is the output of the previous flip-flop.

The output at this stage acts as the clock signal for the next stage, where 3 serial D flip-flops are used for division by 2, 4 and 8, and separate subcircuits are used for division by 3, 5, 6 and 7. For frequency division by 3, 5 and 7, modified mod-3, mod-5 and mod-7 counters are used, respectively. A mod-N counter divides the frequency by N, but the duty cycle is not necessarily 50%. Thus, extra circuitry is required to rectify that. One of the output bits of a counter is delayed by either one or half clock cycle, using a posedge or negedge D flip-flop respectively, and then the original and delayed signals undergo the OR operation to provide the output with 50% duty cycle. Frequency division by 6 is obtained with subsequent division by 2 and 3.

Five control signals are used to divide the frequency of the input clock signal. DIV_A1 and DIV_A0 control the first mux, and DIV_B2, DIV_B1 and DIV_B0 control the second mux. Tables 1 and 2 show the control signals and their respective operations. The output signal is

frequency divided version of the input clock signal, where the division factor is the multiple of division factors in mux 1 and mux 2.

| DIV_A1 | DIV_A0 | OPERATION |
|---|---|---|
| 0 | 0 | Frequency divided by 1 |
| 0 | 1 | Frequency divided by 2 |
| 1 | 0 | Frequency divided by 4 |
| 1 | 1 | Frequency divided by 8 |

**Table 1: Control Signals and Operation of Mux 1**

| DIV_B2 | DIV_B1 | DIV_B0 | OPERATION |
|---|---|---|---|
| 0 | 0 | 0 | Frequency divided by 1 |
| 0 | 0 | 1 | Frequency divided by 2 |
| 0 | 1 | 0 | Frequency divided by 3 |
| 0 | 1 | 1 | Frequency divided by 4 |
| 1 | 0 | 0 | Frequency divided by 5 |
| 1 | 0 | 1 | Frequency divided by 6 |
| 1 | 1 | 0 | Frequency divided by 7 |
| 1 | 1 | 1 | Frequency divided by 8 |

**Table 2: Control Signals and Operation of Mux 2**

**Circuit Diagrams:**

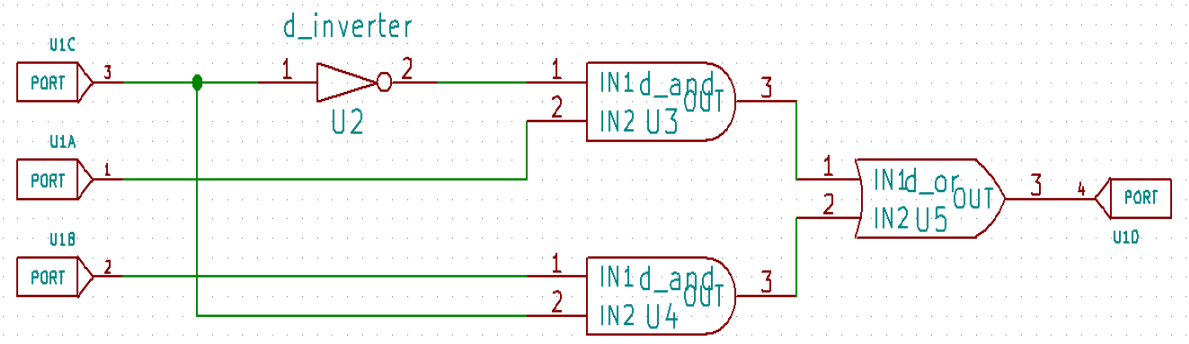


**Figure 1a: Subcircuit Schematic for 2:1 Mux**

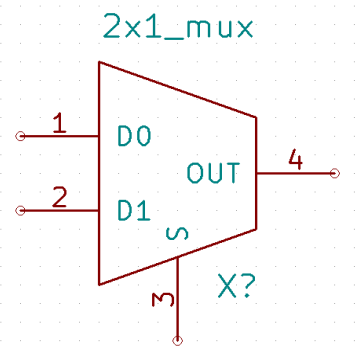


**Figure 1b: Subcircuit Symbol for 2:1 Mux**

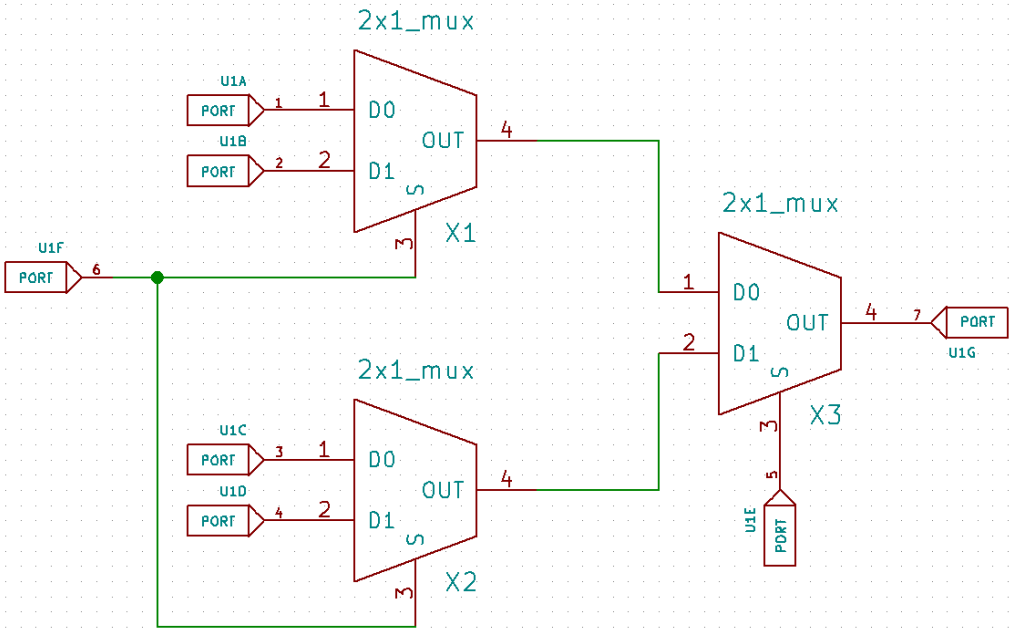


**Figure 2a: Subcircuit Schematic for 4:1 Mux**

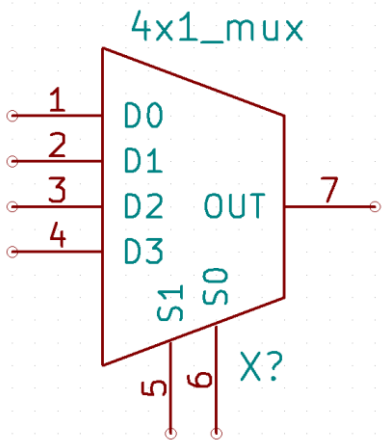


**Figure 2b: Subcircuit Symbol for 4:1 Mux**

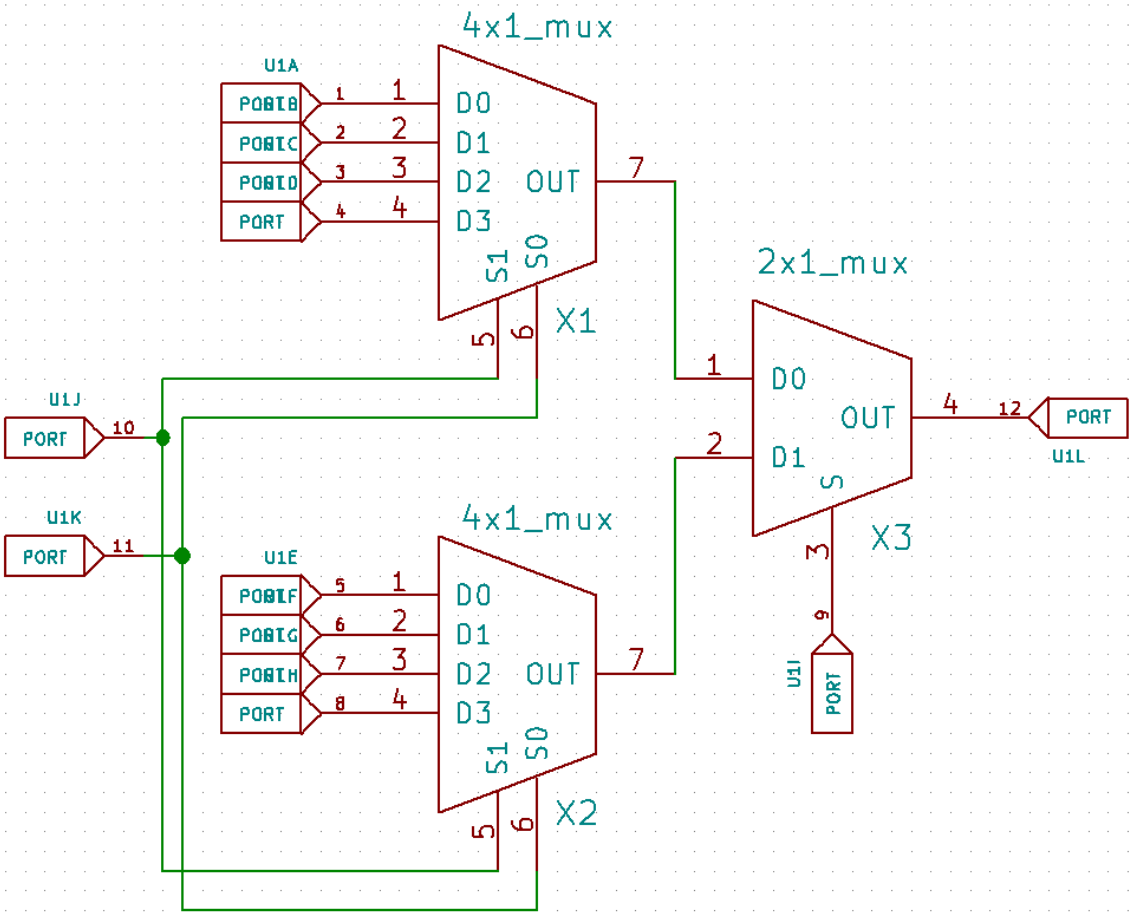


**Figure 3a: Subcircuit Schematic for 8:1 Mux**

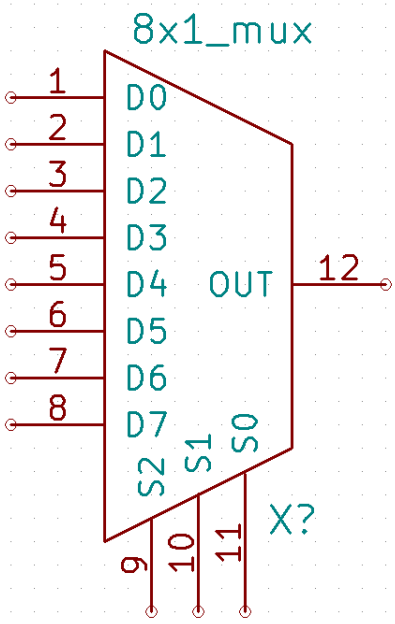


**Figure 3b: Subcircuit Symbol for 8:1 Mux**

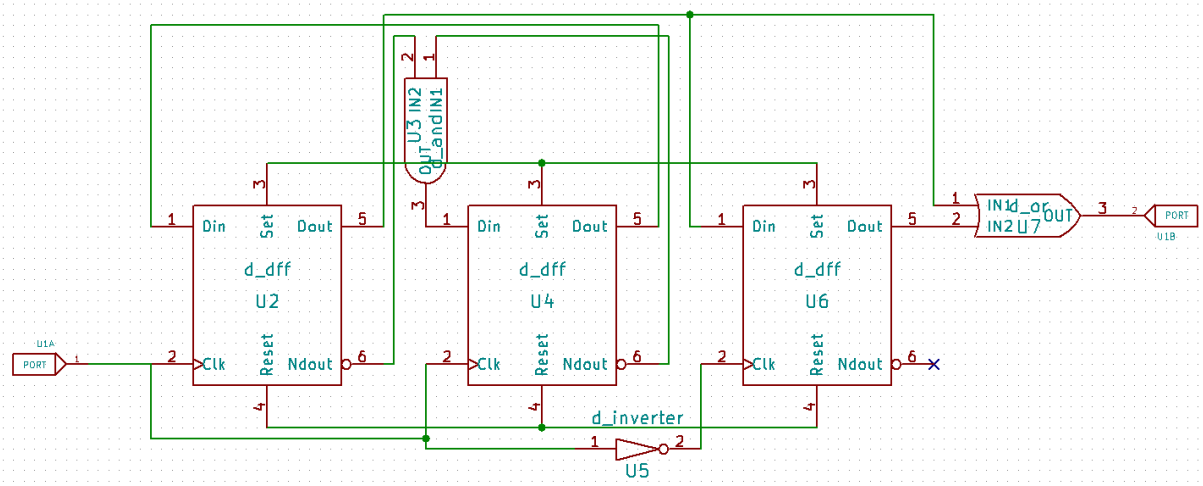


**Figure 4a: Subcircuit Schematic for Frequency Divider by 3**


**Figure 4b: Subcircuit Symbol for Frequency Divider by 3**

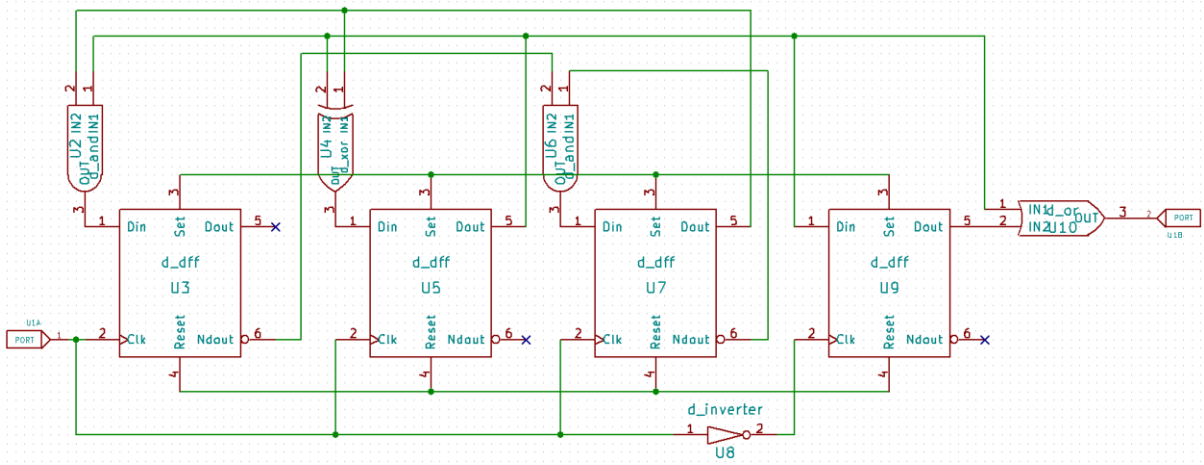


**Figure 5a: Subcircuit Schematic for Frequency Divider by 5**

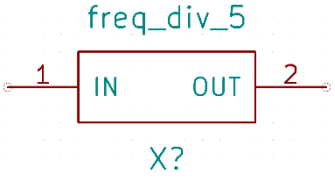


**Figure 5b: Subcircuit Symbol for Frequency Divider by 5**

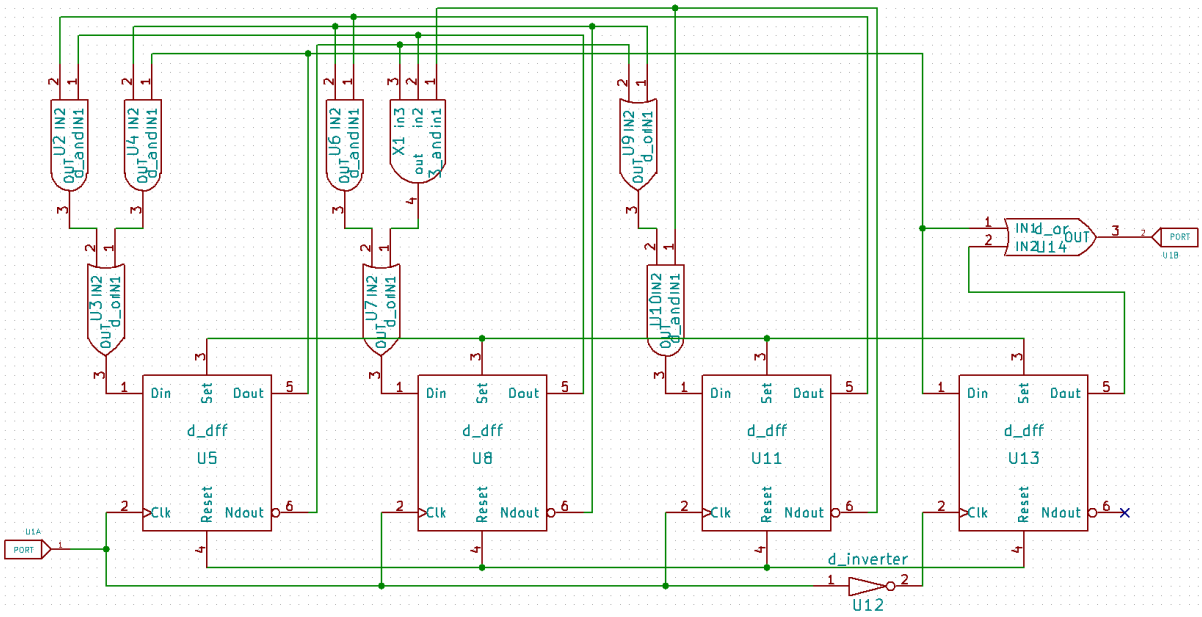


**Figure 6a: Subcircuit Schematic for Frequency Divider by 7**

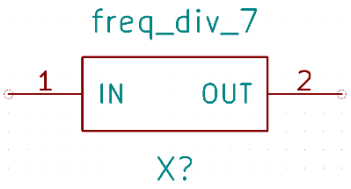


**Figure 6b: Subcircuit Symbol for Frequency Divider by 7**

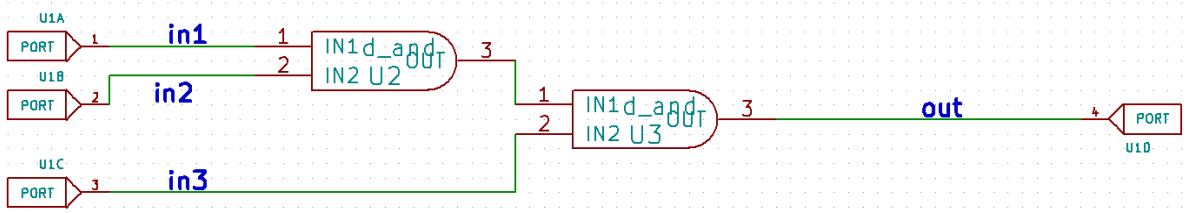


**Figure 7a: Subcircuit Schematic for 3-Input AND Gate (already exists in eSim_Subckt)**

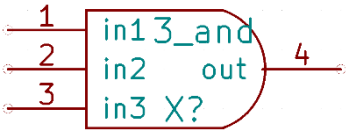


**Figure 7b: Subcircuit Symbol for 3-Input AND Gate (already exists in eSim_Subckt)**

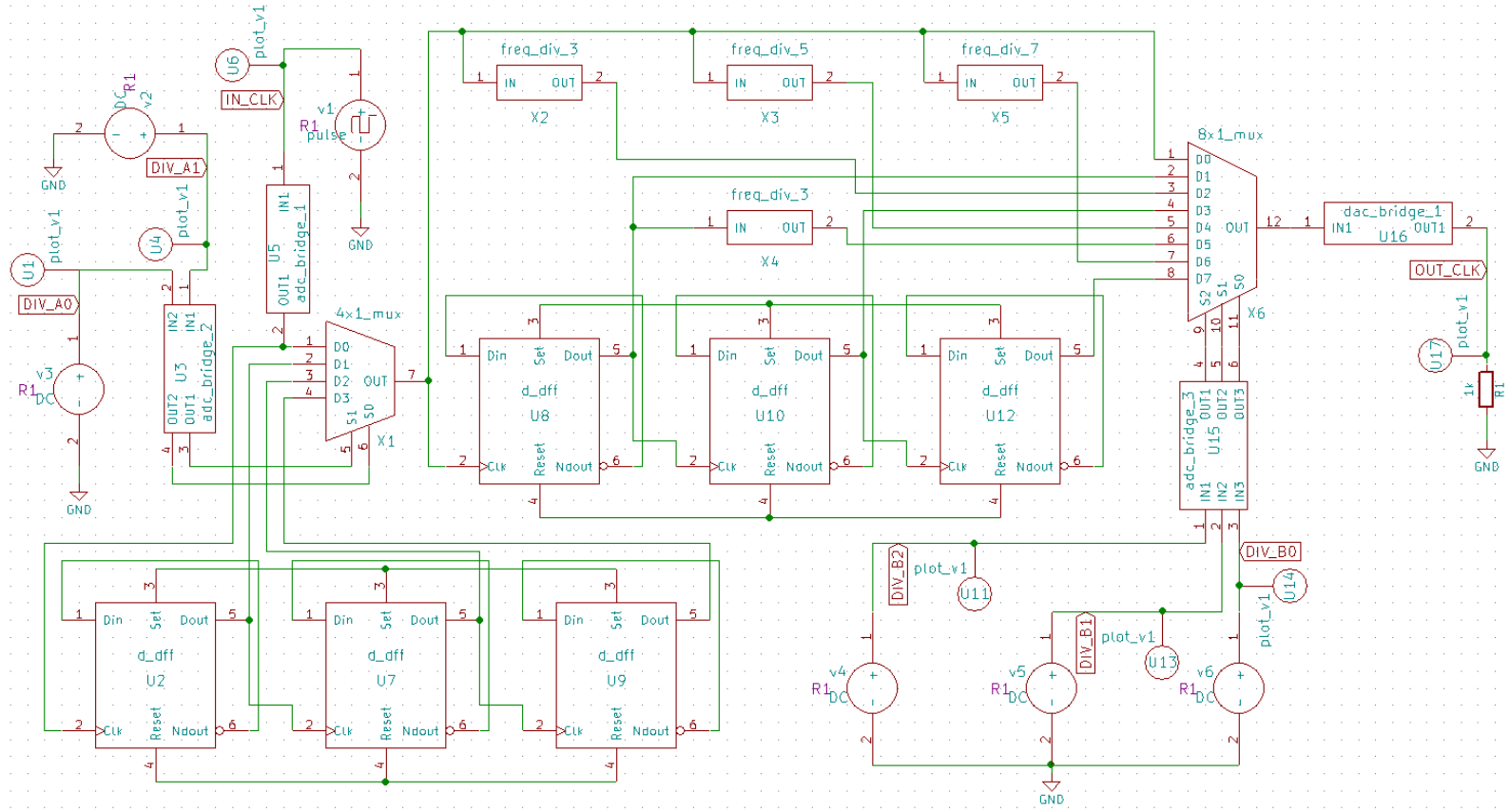


**Figure 8: Main Circuit Schematic – Programmable Frequency Divider**

**Results (Input, Output waveforms):**

**EXAMPLE 1: DIV_A = 01, DIV_B = 110 (Frequency divided by 2x7 = 14)**

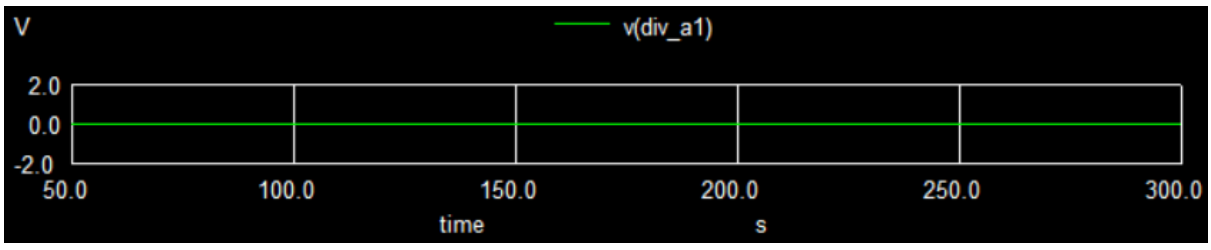


**Figure 9a: Analog Signal for DIV_A1**

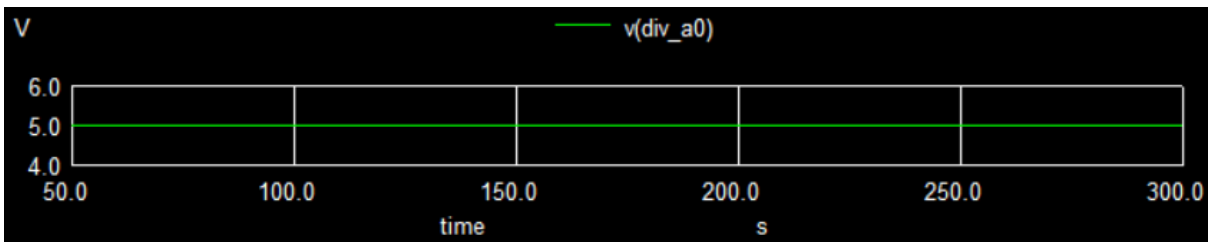


**Figure 9b: Analog Signal for DIV_A0**

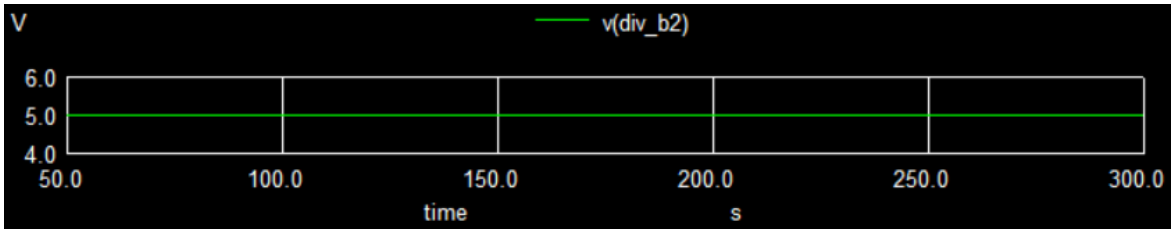


**Figure 9c: Analog Signal for DIV_B2**

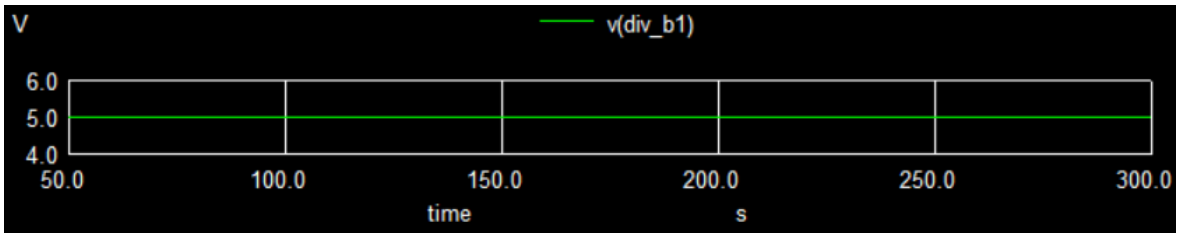


**Figure 9d: Analog Signal for DIV_B1**

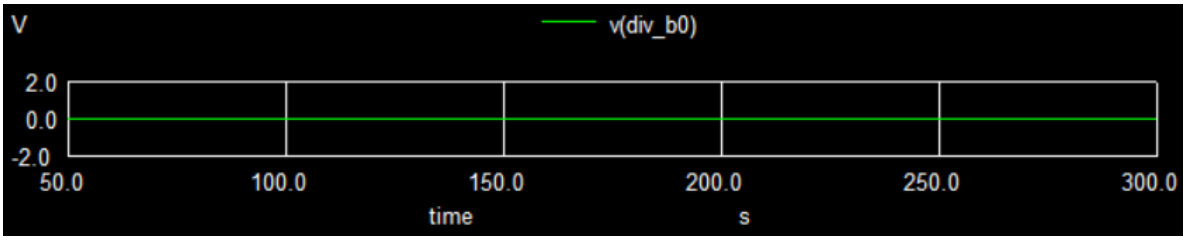


**Figure 9e: Analog Signal for DIV_B0**

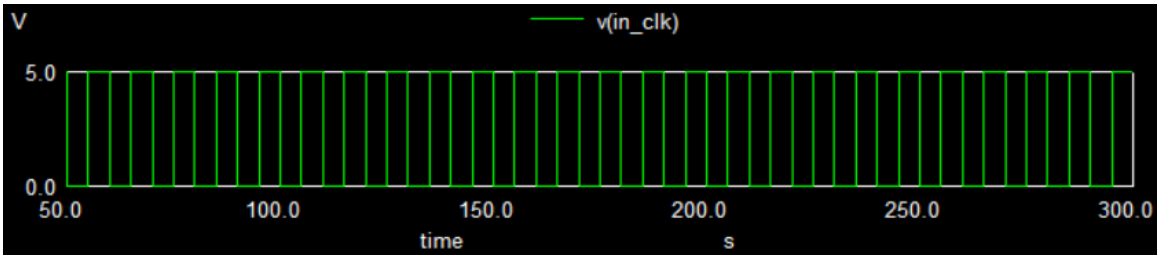


**Figure 10a: Analog Signal for in_clk (Input Clock Signal)**

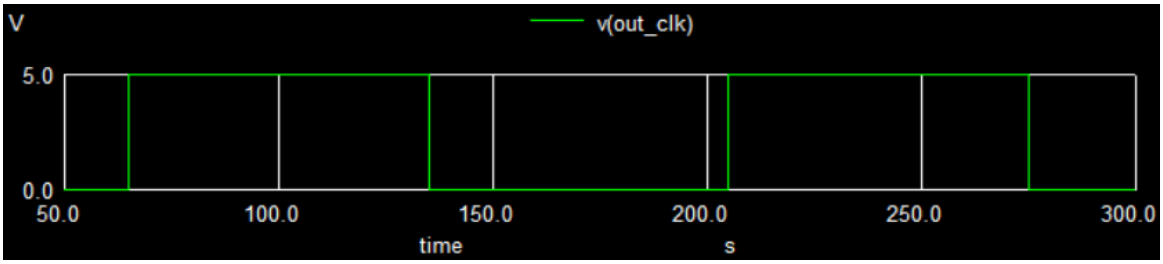


**Figure 10b: Analog Signal for out_clk (Output Clock Signal)**

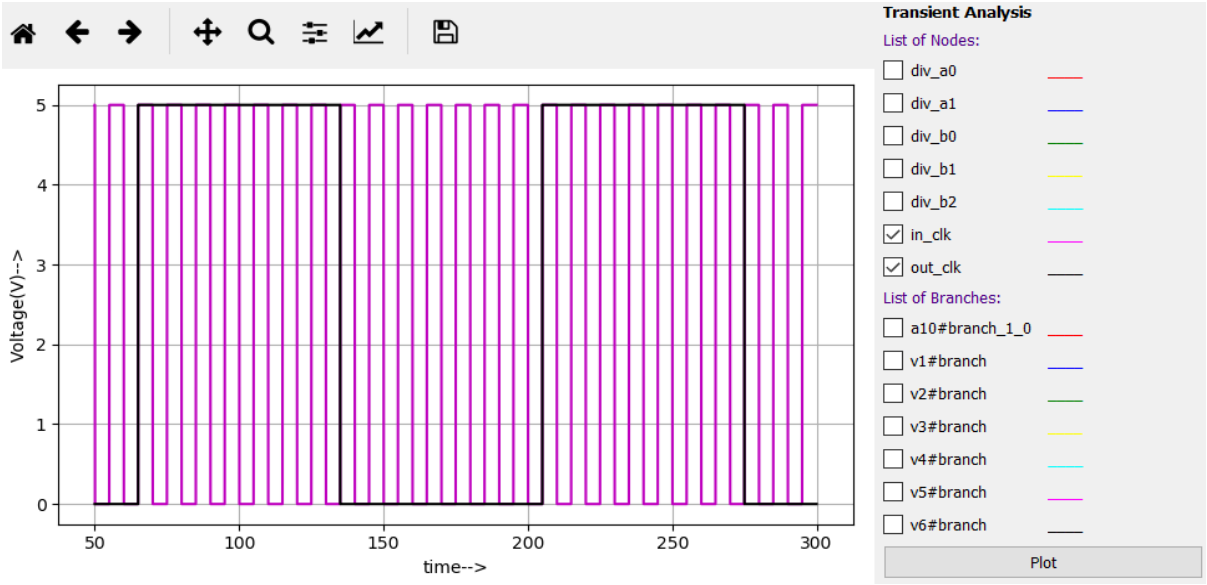


**Figure 10c: Analog Signals for in_clk vs out_clk – Python Plot**

**EXAMPLE 2: DIV_A = 10, DIV_B = 010 (Frequency divided by 4x3 = 12)**

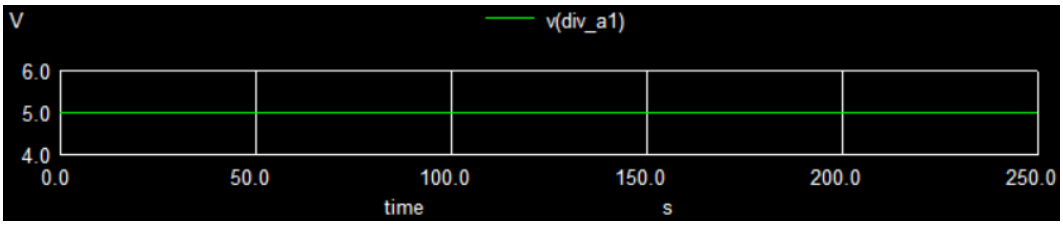


**Figure 11a: Analog Signal for DIV_A1**


**Figure 11b: Analog Signal for DIV_A0**


**Figure 11c: Analog Signal for DIV_B2**

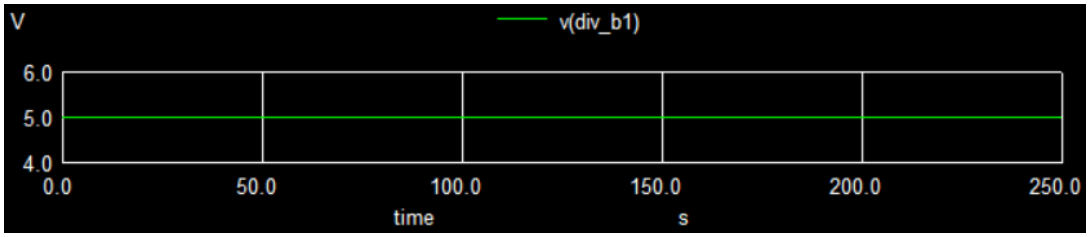


**Figure 11d: Analog Signal for DIV_B1**

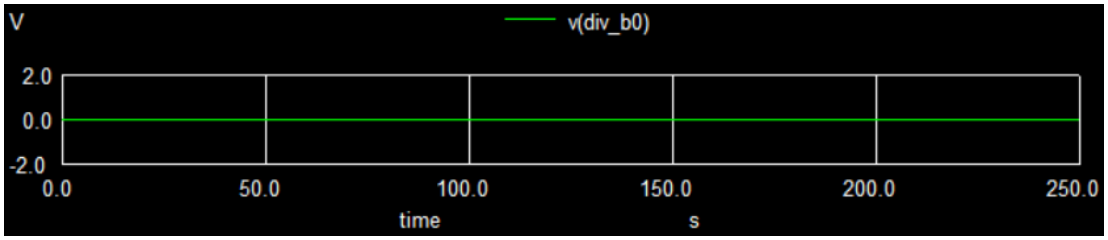


**Figure 11e: Analog Signal for DIV_B0**

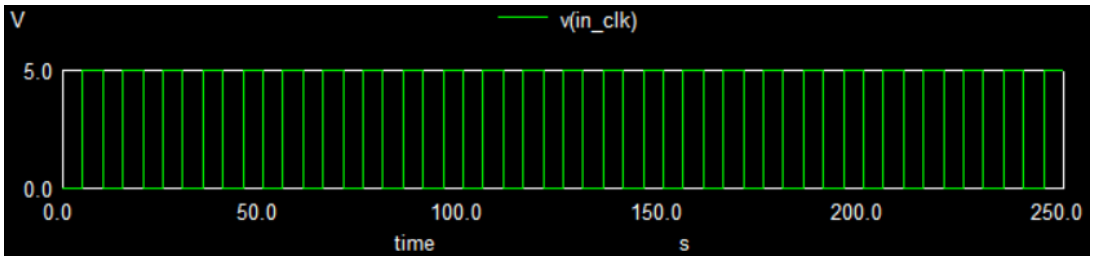


**Figure 12a: Analog Signal for in_clk (Input Clock Signal)**

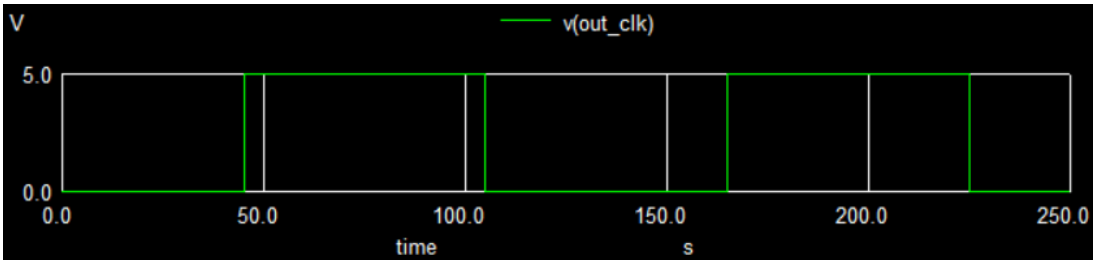


**Figure 12b: Analog Signal for out_clk (Output Clock Signal)**

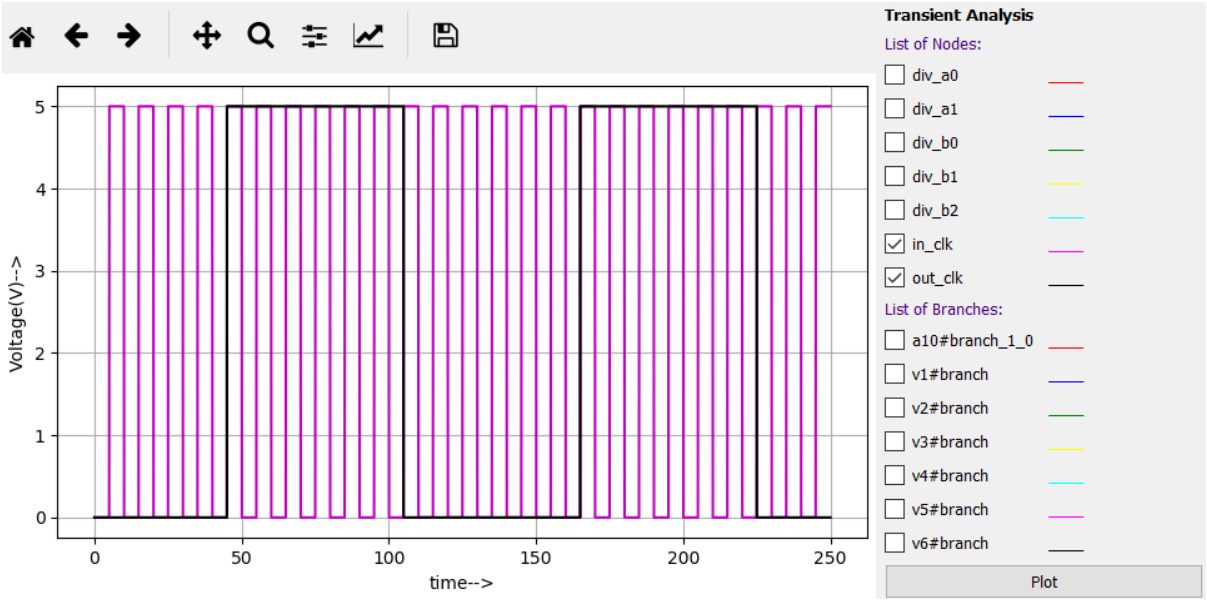


**Figure 12c: Analog Signals for in_clk vs out_clk – Python Plot**

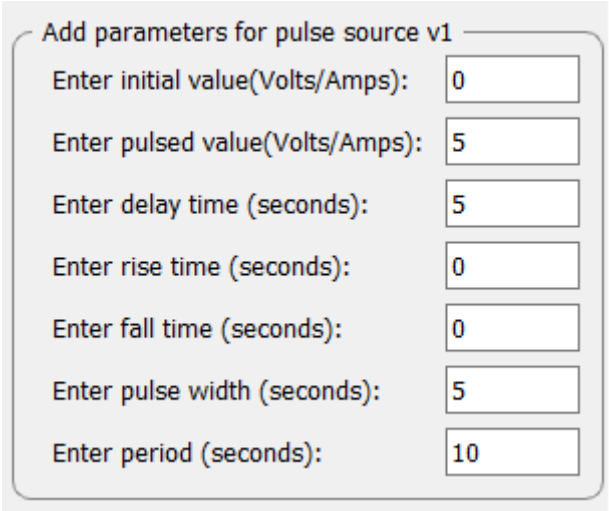


**Figure 13: Input Clock Signal in_clk Parameters**

**Source/Reference(s):**


VLSICoding: Implement Divide by 2, 4, 8 and 16 Counter using Flip-Flop

digital logic - Divide clock frequency by 3 with 50% duty cycle by using a Karnaugh Map? - Electrical Engineering Stack Exchange

Divide by N clock (slideshare.net)