Summer Fellowship Report

On

Integrated Circuit Design using Subcircuit feature of eSim

Submitted by

Karthikeyan S
B.E (Electronics and Communication Engineering)
Loyola ICAM College of Engineering and Technology, Chennai

Under the guidance of

Mr. Sumanto Kar
Assistant Project Manager, FOSSEE, IIT Bombay

Prof. Kannan M. Moudgalya
Chemical Engineering Department, IIT Bombay

July 14, 2025

Acknowledgment

I take this opportunity to express my deepest gratitude to the entire FOSSEE team at IIT Bombay for providing me with this wonderful opportunity to work on the design and integration of sub-circuits using eSim and Device Modelling. The internship has given me valuable exposure to open-source tools, their real-world applications, and a deeper understanding of electronic design and simulation.

I am especially grateful to Prof. Kannan M. Moudgalya for his constant inspiration and vision behind the FOSSEE initiative. His commitment towards promoting open-source software for education and research has created numerous opportunities for students like me to gain hands-on experience and contribute to the open-source community. His farsighted leadership and encouragement have been truly motivating throughout this journey.

I would like to convey my heartfelt thanks to Mr. Sumanto Kar for his constant support, guidance, and encouragement throughout the internship. His valuable insights, timely feedback, and patient mentoring have been instrumental in helping me overcome challenges and successfully complete my project objectives. His dedication towards guiding interns like me has made this journey both enlightening and enjoyable.

I also sincerely appreciate the entire FOSSEE initiative for its efforts in empowering students and researchers by making tools freely accessible, thus removing the barriers imposed by costly commercial software. This mission not only fosters innovation but also strengthens the open-source ecosystem, allowing young learners to actively contribute and grow.

This internship has been a memorable and rewarding phase of my academic journey. The knowledge, skills, and experiences I have gained will undoubtedly serve as a strong foundation as I move forward in my career in the semiconductor and VLSI domain.

Contents

Chapter 1

Introduction

The advancement of Electronic Design Automation (EDA) tools has revolutionized the way electronic circuits are designed, simulated, and verified. Among these, open-source tools play a crucial role in making advanced EDA capabilities accessible to students, educators, and researchers without the burden of expensive licenses. The FOSSEE project at IIT Bombay has taken significant steps to promote such open-source tools globally.

One of the key open-source tools developed and promoted under FOSSEE is **eSim**, which integrates multiple open-source EDA tools to provide a unified platform for circuit design, simulation, and PCB layout. Various tools like NgSpice, Makerchip, and KiCad work together under eSim to offer a complete design environment.

1.1 eSim

eSim is an open-source Electronic Design Automation (EDA) tool developed by FOSSEE, IIT Bombay. The primary objective of eSim is to provide a completely free and open-source alternative to commercial circuit design and simulation software, thereby reducing dependency on expensive proprietary tools in academia and research.

eSim integrates multiple open-source tools to provide a comprehensive design and simulation environment. **KiCad** is used for schematic capture and PCB design, while **NgSpice** performs analog circuit simulation. For digital simulation, **GHDL** is integrated, supporting VHDL-based designs, and **Makerchip** enables online Transaction-Level Verilog (TL-Verilog) based digital design and verification. Python support is also available in eSim, allowing users to perform customized simulations and generate netlists.

One of the key features of eSim is its **Subcircuit feature**, which allows complex designs to be modularized into reusable blocks, simplifying the design of large systems. Additionally, eSim supports device modeling, enabling users to define and simulate custom semiconductor devices using real-world parameters. As part of its continuous development, eSim also supports **SkyWater SKY130 PDK**, allowing users to perform simulations using an open-source 130nm Process Design Kit, which is widely used for analog, mixed-signal, and digital IC design research.

By supporting a fully open-source toolchain, eSim empowers students, educators, and researchers to gain hands-on experience with industry-relevant EDA tools and contribute actively to the open-source hardware ecosystem.

1.1.1 NgSpice

NgSpice is an open-source circuit simulator integrated into eSim for performing analog and mixed-signal simulations. Based on the SPICE simulation engine, NgSpice supports DC, AC, transient, and parametric analyses, making it suitable for analyzing a wide range of circuits.

It allows users to simulate circuit behavior using industry-standard SPICE models, including support for subcircuits and behavioral modeling. In eSim, NgSpice works as the backend simulator for schematics created using KiCad, providing waveform outputs for voltage, current, and frequency responses. Its flexibility and accuracy make it a powerful tool for verifying designs before hardware implementation.

1.1.2 Makerchip

Makerchip is an integrated platform designed to simplify digital circuit design by offering both browser-based and desktop-based environments for coding, simulating, and debugging digital hardware designs. It supports multiple hardware description languages including Verilog, SystemVerilog, and Transaction-Level Verilog (TL-Verilog), allowing flexibility for users at various levels of expertise.

In eSim, Makerchip is interfaced through a Python-based application called Makerchip-App, which seamlessly launches the Makerchip IDE for digital design and verification. The platform integrates several open-source and proprietary tools to provide a rich set of features such as real-time simulation, waveform viewing, and code linting, thereby improving design accuracy and productivity.

1.1.3 KiCad

KiCad is an open-source PCB design and schematic capture tool integrated within eSim for creating circuit schematics and generating PCB layouts. It allows users to design multi-layer boards, define custom footprints, and perform design rule checks to ensure design correctness before fabrication.

In eSim, KiCad acts as the primary schematic editor, allowing users to graphically build circuits by placing and interconnecting components from extensive open-source libraries. The designed schematics can directly be used for both simulation and PCB layout generation. KiCad also supports features such as 3D visualization of PCBs, Gerber file export for manufacturing, and electrical rule checking to identify design issues early. Its seamless integration within eSim enables a smooth transition from schematic design to simulation and physical realization, offering a complete design-to-fabrication workflow entirely within an open-source environment.

1.1.4 GHDL

GHDL is an open-source simulator for VHDL, a hardware description language widely used in digital circuit design. It supports the complete IEEE VHDL standard, allowing designers to simulate, verify, and debug VHDL-based digital systems effectively.

In eSim, GHDL is integrated to enable digital simulation alongside analog simulation provided by NgSpice. Users can model digital circuits using VHDL, which are then simulated using GHDL to verify logical functionality and timing behavior. This integration allows eSim to handle mixed-signal designs, where the analog components are simulated by NgSpice and the digital components by GH022, providing a comprehensive platform for complex system design.

Chapter 2

Features of eSim

eSim offers a comprehensive set of features that make it a powerful open-source alternative for electronic design automation. Some of the key features include:

- **OpenOpen-source and Free**: eSim is fully open-source, allowing unrestricted access without licensing costs, making it ideal for academic and research purposes.
- **Integrated Toolchain**: Combines multiple open-source tools such as KiCad for schematic capture and PCB design, NgSpice for analog simulation, GHDL for digital simulation, and Makerchip for advanced digital design.
- **Mixed-Signal Simulation**: Supports both analog and digital simulation, allowing users to design and simulate mixed-signal circuits efficiently.
- **Subcircuit Feature**: Enables hierarchical and modular design by allowing complex circuits to be broken into reusable subcircuits.
- **Device Modeling**: Supports custom device modeling, allowing users to simulate real-world semiconductor devices using user-defined parameters.
- **SkyWater SKY130 PDK Support**: Provides access to open-source 130nm process design kit for IC design and simulation.
- **Python Integration**: Allows automation, scripting, and advanced analysis using Python interfaces.
- **User-Friendly Interface**: Offers an intuitive GUI that simplifies circuit creation, simulation setup, and result analysis, making it suitable for both beginners and advanced users.
- **Cross-Platform Support**: Compatible with major operating systems like Linux and Windows.
- **Active Community and Documentation**: Extensive documentation, tutorials, and community support provided through FOSSEE ensure smooth learning and troubleshooting.

Chapter 3

Problem Statement

To design, develop, and test various Analog and Digital Integrated Circuit Models using the sub-circuit feature in eSim and create Device Models using the Model Editor tool. These subcircuits are to be built using the device models already present in the eSim library. After successful testing of these IC models and their integration into the eSim subcircuit library, these models would be useful in the future for circuit designing purposes.

3.1 Subcircuit Design and Integration

Digital ICs are modeled as subcircuits based on datasheet specifications and integrated into eSim's subcircuit library.

3.1.1 Approach

- **Datasheet Analysis**: Browse through various analog and digital IC datasheets to identify suitable circuits not previously included in the eSim library. Verify the detailed schematic and truth table to finalize the IC for implementation.
- **Subcircuit Creation**: Model the selected IC as a subcircuit in eSim using only the model files from the eSim device model library, strictly adhering to the official datasheet specifications. This includes creating the Symbol/Pin diagram based on the datasheet's packaging and pin description.
- **Symbol Creation**: Create custom IC symbols according to package type and pin configuration for accurate circuit integration.
- **Test Circuit Design**: Build test circuits based on datasheet recommendations using the created IC component.
- **Simulation and Validation**: Simulate the test circuits to obtain output waveforms and plots using eSim's KiCad to NgSpice conversion and simulation feature. If the output does not meet expectations, revisit the IC or test circuit design to identify and correct errors, then repeat the testing process. Once the output matches the expected results, the IC is declared successfully working and verified.

Chapter 4

Digital ICs

The Subcircuit feature of eSim allows hierarchical modeling of complex Integrated Circuits by combining multiple basic components into a single reusable block. By utilizing datasheets and internal schematics of various ICs, accurate subcircuit models are created using eSim's schematic editor.

4.1 CD4013BC - Dual D-Type Flip-Flop

Description: Dual D-Type Flip-Flop.

General Description: The CD4013B dual D-type flip-flop is a monolithic complementary MOS (CMOS) integrated circuit constructed with N- and P-channel enhancement mode transistors. Each flip-flop has independent data, set, reset, and clock inputs and Q and Q outputs. These devices can be used for shift register applications, and by connecting Q output to the data input, for counter and toggle applications. The logic level present at the D input is transferred to the Q output during the positive-going transition of the clock pulse. Setting or resetting is independent of the clock and is accomplished by a high level on the set or reset line respectively.

4.1.1 IC Layout

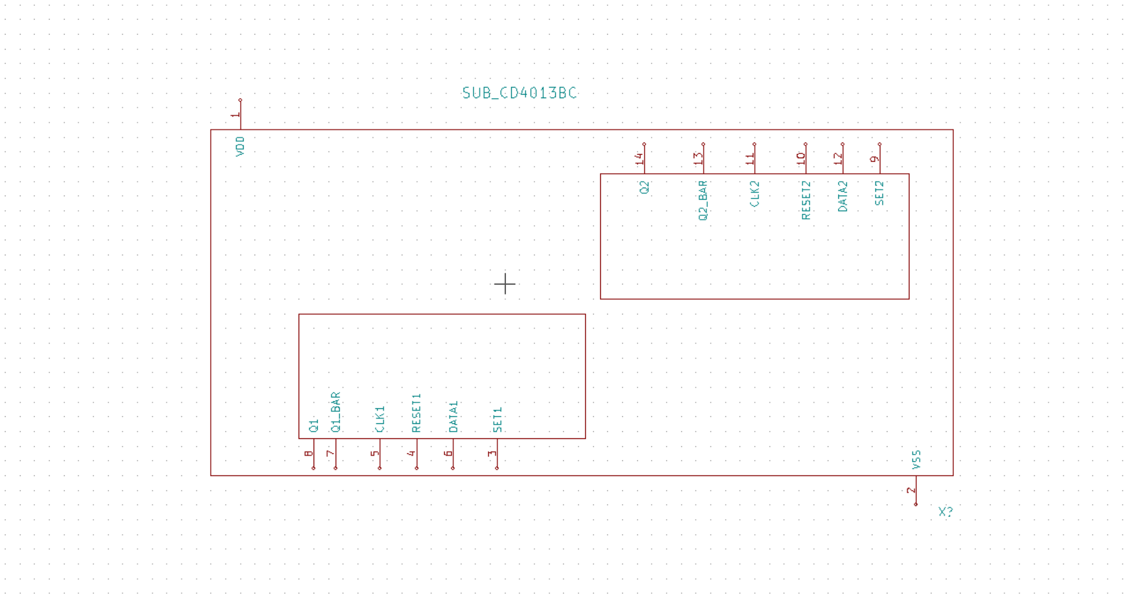


Figure 4.1: IC Layout of CD4013BC

4.1.2 Subcircuit Schematic Diagram of CD4013BC

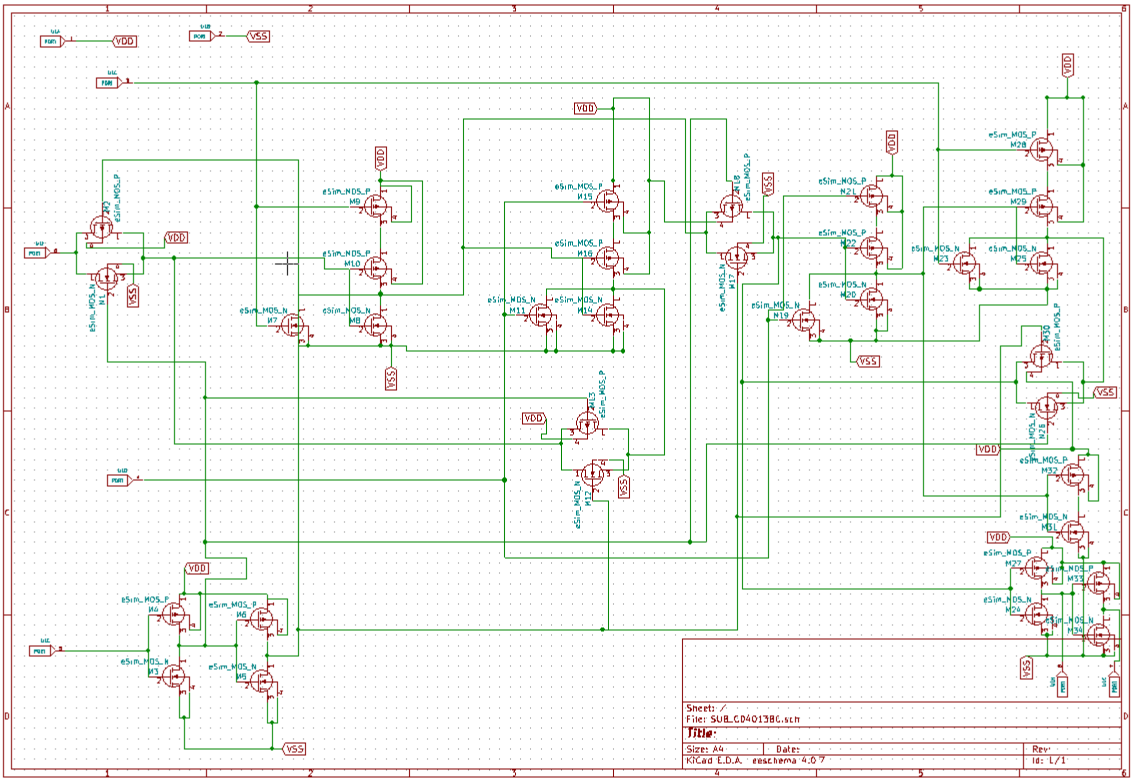


Figure 4.2: Subcircuit Schematic Diagram of CD4013BC

4.1.3 Test Circuit of CD4013BC

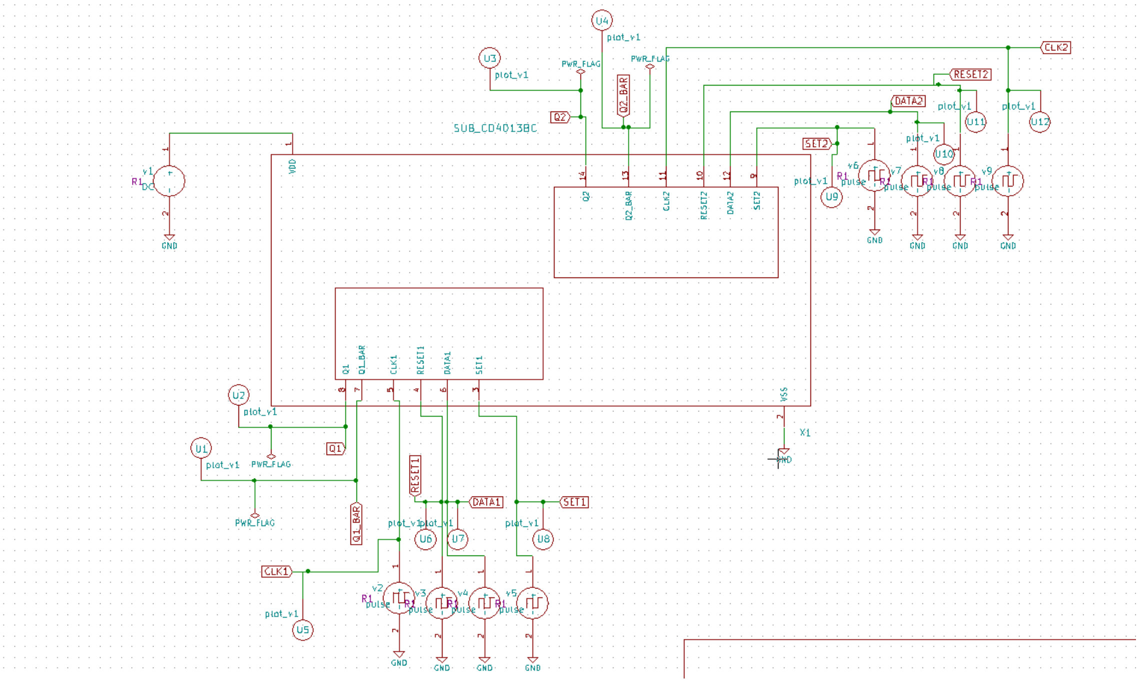


Figure 4.3: Test Circuit of CD4013BC

4.1.4 Input and Output Voltage Waveform of CD4013BC

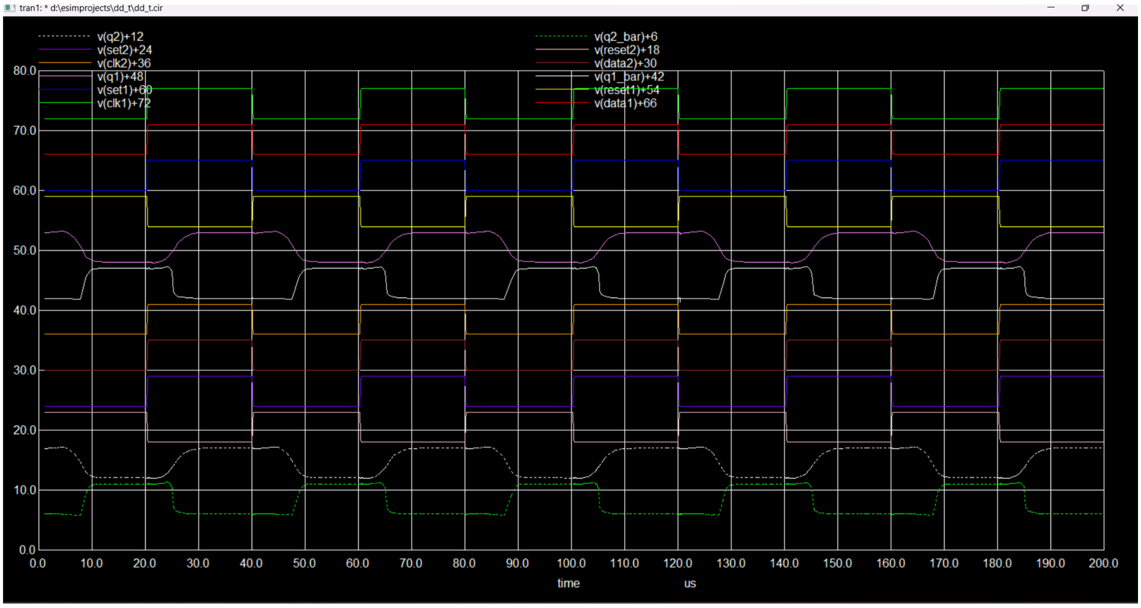


Figure 4.4: Input and Output Voltage Waveform of CD4013BC

4.2 HCC4076B - 4-Bit D-Type Register

Description: 4-Bit D-Type Register with Clocked Inputs and Three-State Outputs.

General Description: The HCC/HCF4076B types are four-bit registers consisting of D-type flip-flops that feature three-state outputs. Data Disable inputs control the entry of data into the flip-flops. When both Data Disable inputs are low, data at the inputs are loaded into their respective flip-flops on the next positive transition of the clock input. Output Disable inputs are provided, and when both are low, the normal logic states of the four outputs are available to the load. The outputs are disabled independently of the clock by a high logic level at either Output Disable input, presenting a high impedance.

4.2.1 IC Layout

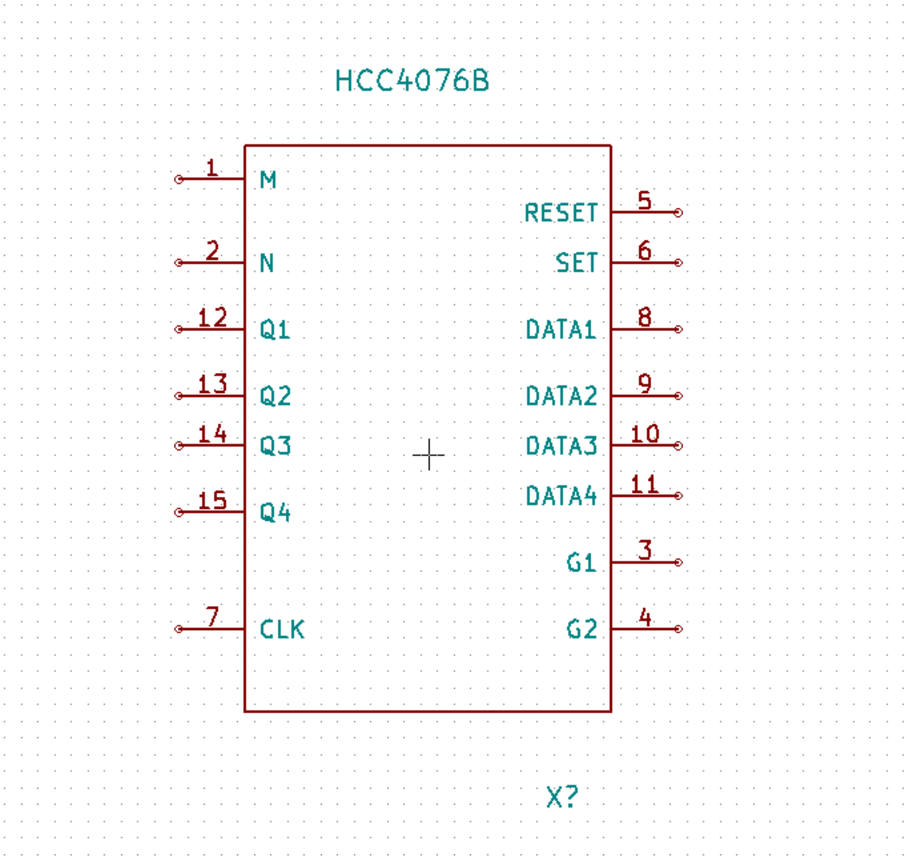


Figure 4.5: IC Layout of HCC4076B

4.2.2 Subcircuit Schematic Diagram of HCC4076B

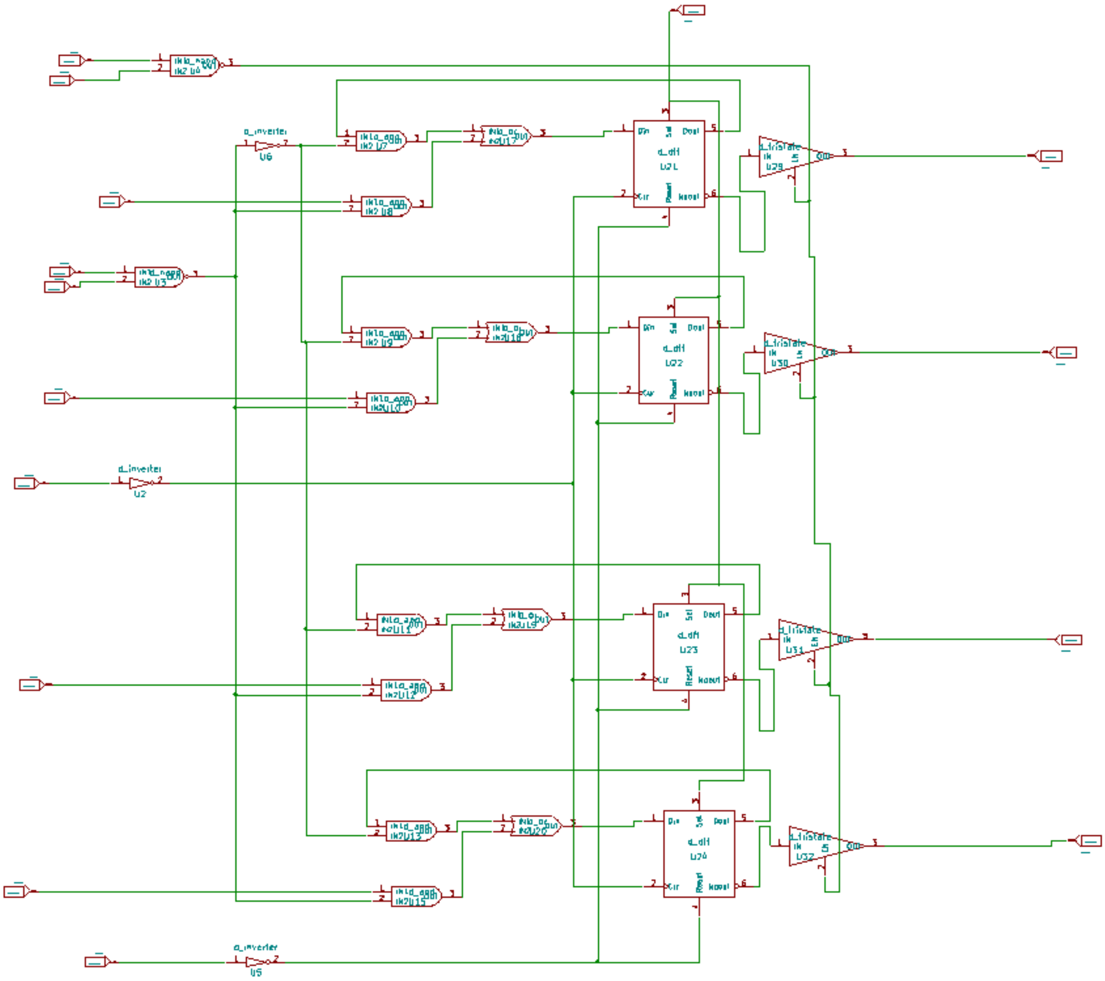


Figure 4.6: Subcircuit Schematic Diagram of HCC4076B

4.2.3 Test Circuit of HCC4076B

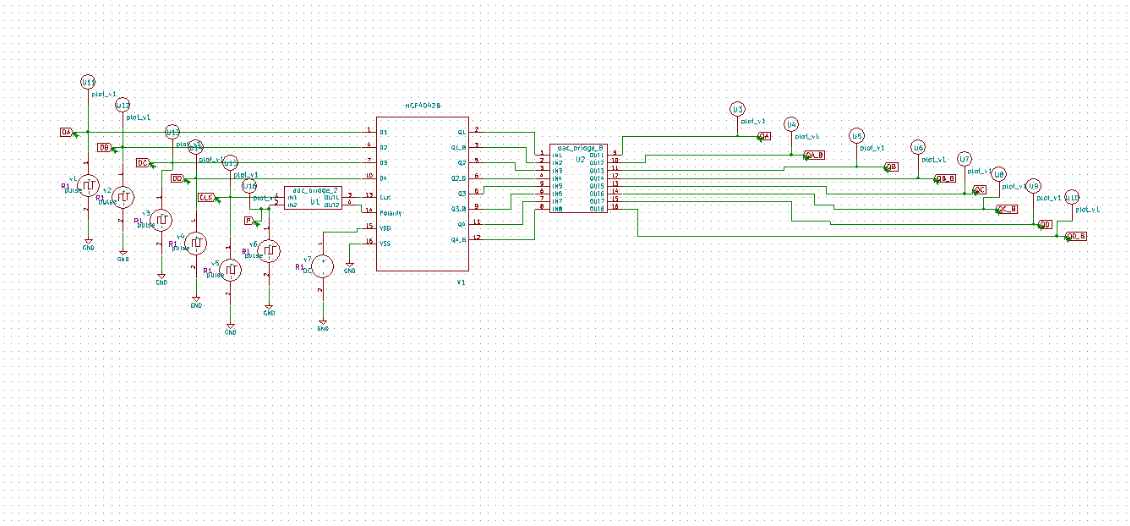


Figure 4.7: Test Circuit of HCC4076B

4.2.4 Input and Output Voltage Waveform of HCC4076B

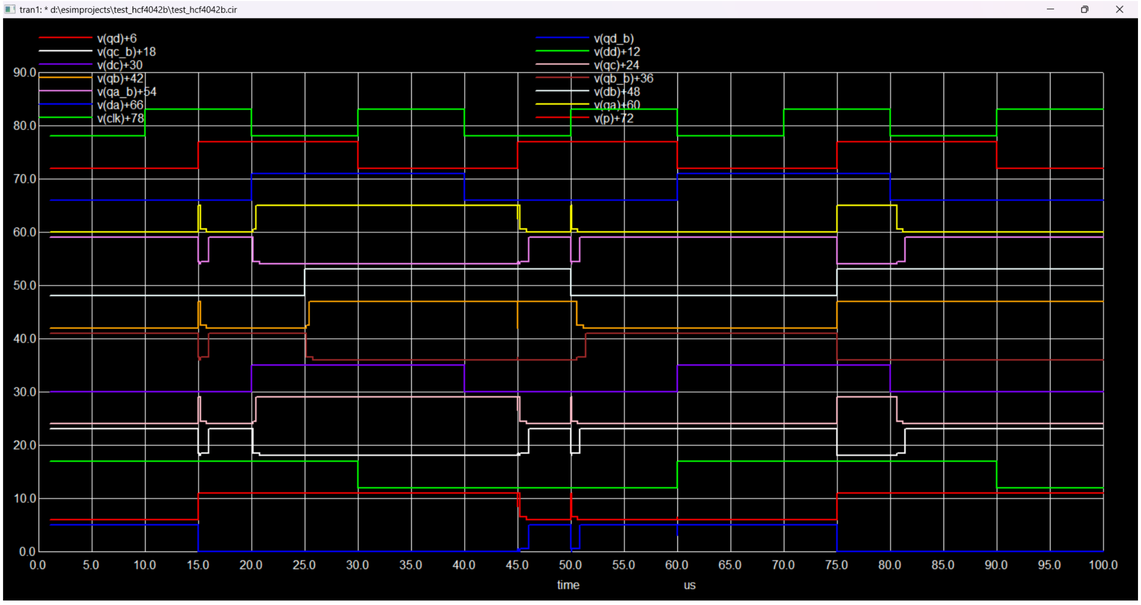


Figure 4.8: Input and Output Voltage Waveform of HCC4076B

4.3 CD4035BM - 4-Bit Parallel-In/Parallel-Out Shift Register

Description: 4-Bit Shift Register capable of parallel-in and parallel-out.

General Description: The CD4035B is a 4-bit parallel-in/parallel-out shift register built using CMOS technology with P- and N-channel enhancement-mode transistors. It features four D flip-flop stages with synchronous parallel and serial data entry. Serial input to the first stage is via JK logic, and data transfer occurs on the positive clock edge. Parallel data entry is enabled when the parallel/serial control is high. A true/complement control provides either the actual data or its complement at the outputs, operating asynchronously with the clock. The JK logic reduces external gate count in counting or sequence-generation applications, and a common asynchronous reset is included.

4.3.1 IC Layout

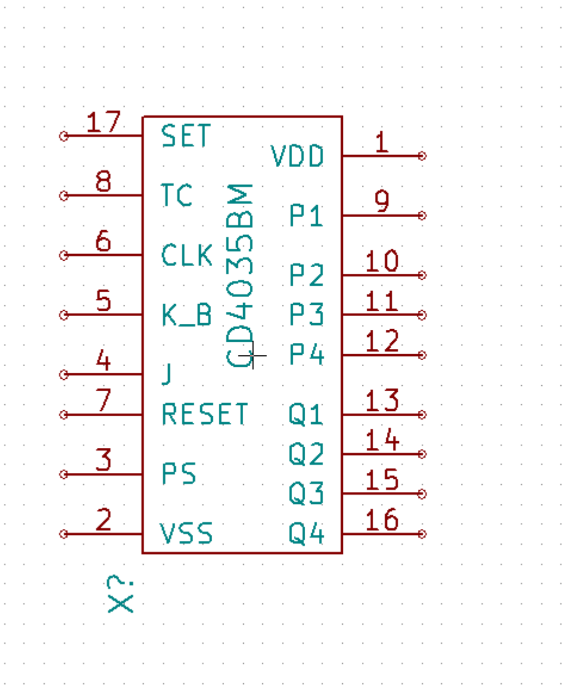


Figure 4.9: IC Layout of CD4035BM

4.3.2 Subcircuit Schematic Diagram of CD4035BM

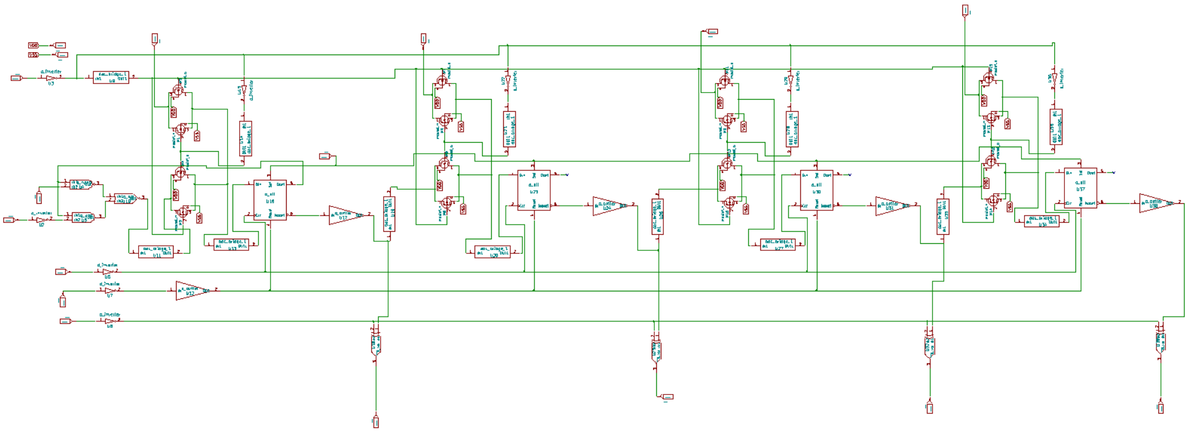


Figure 4.10: Subcircuit Schematic Diagram of CD4035BM

4.3.3 Test Circuit of CD4035BM

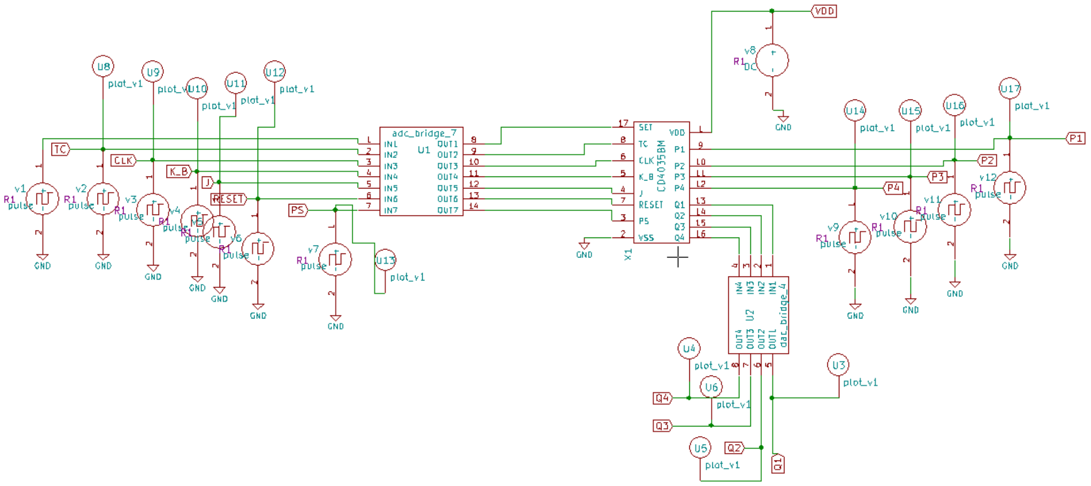


Figure 4.11: Test Circuit of CD4035BM

4.3.4 Input and Output Voltage Waveform of CD4035BM

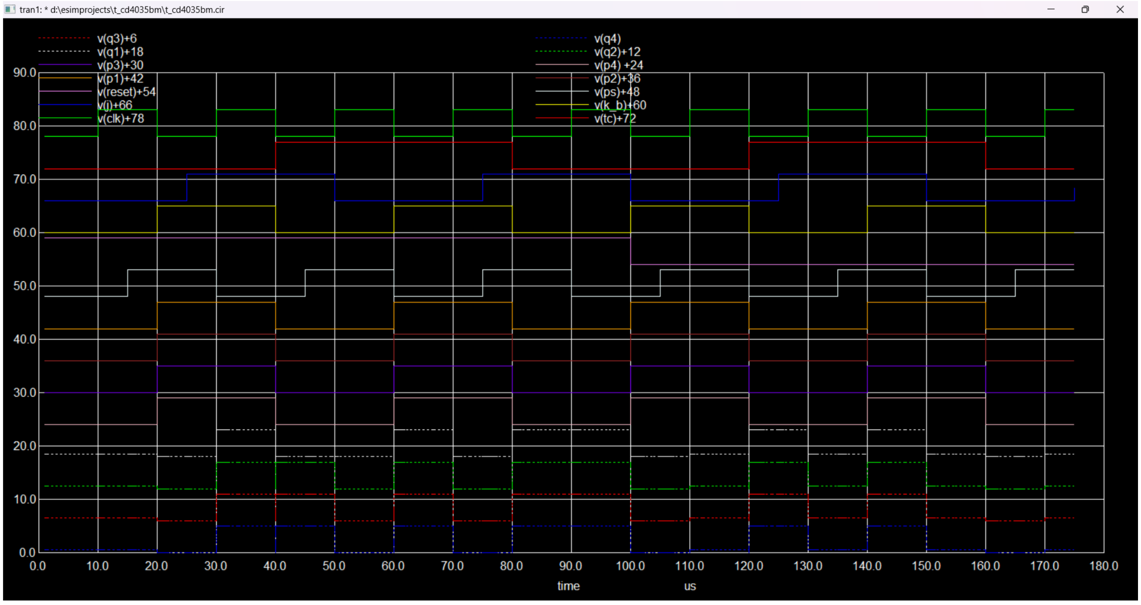


Figure 4.12: Input and Output Voltage Waveform of CD4035BM

4.4 CD4529BM - Dual 4-Channel or Single 8-Channel Analog Data Selector

Description: Dual 4-Channel or Single 8-Channel Analog Data Selector/Multiplexer.

General Description: The CD4529B is a dual 4-channel or single 8-channel analog data selector implemented using CMOS technology with N- and P-channel enhancement-mode transistors. Depending on input coding, it can operate in dual 4-channel or single 8-channel mode. For 8-channel mode, outputs Z and W are tied together. The device supports both digital and analog signal routing and is suitable for 1-of-4 or 1-of-8 data selector applications. Its bidirectional analog capability also allows use as a dual binary-to-1-of-4 or single 1-of-8 decoder.

4.4.1 IC Layout

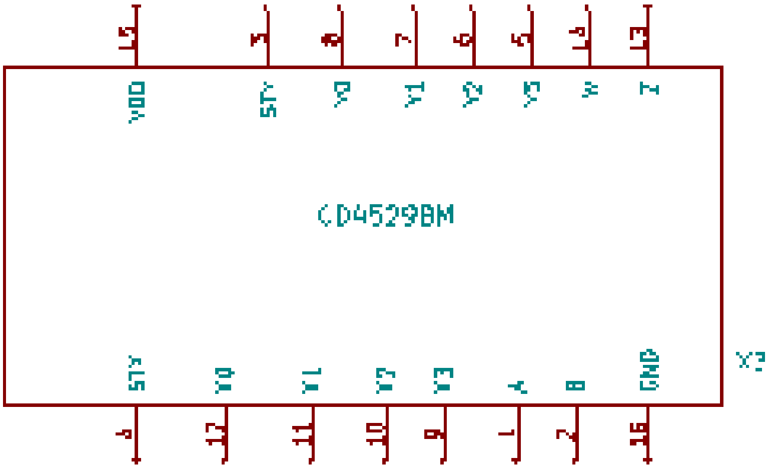


Figure 4.13: IC Layout of CD4529BM

4.4.2 Subcircuit Schematic Diagram of CD4529BM

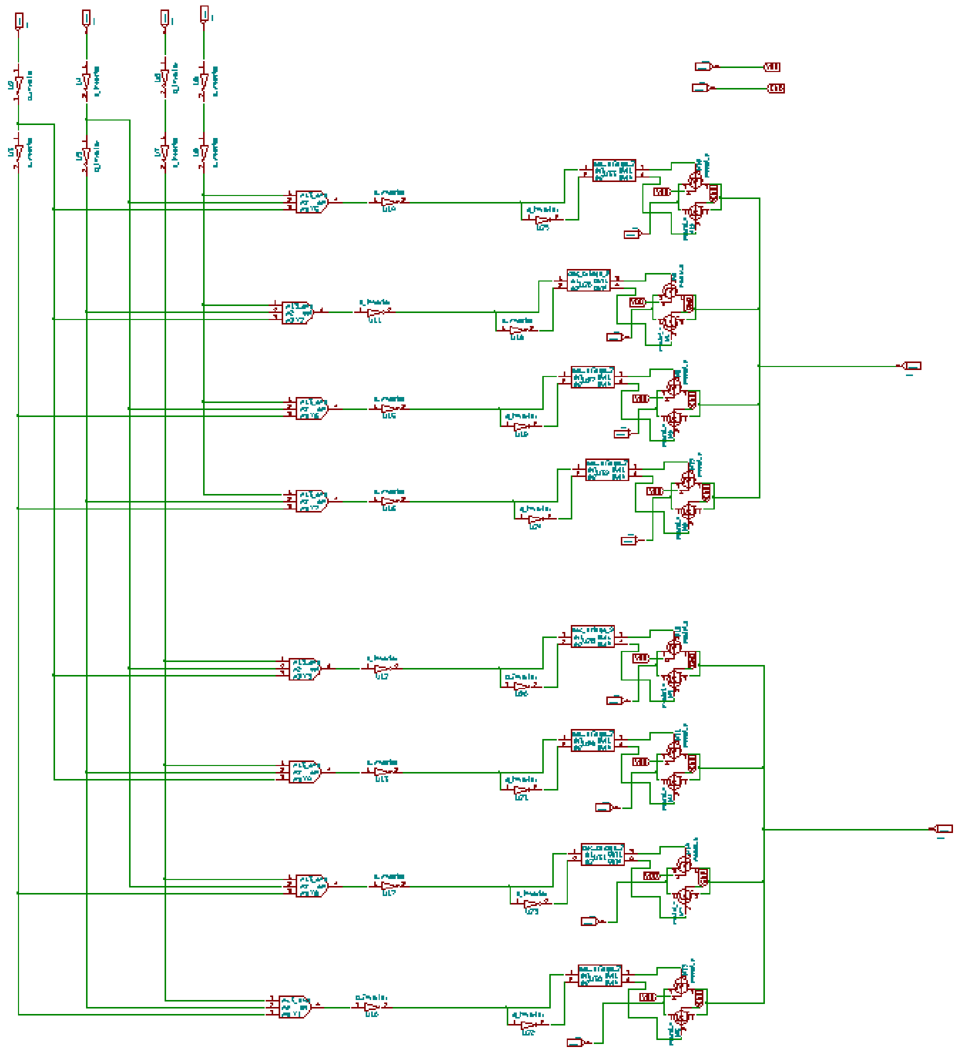


Figure 4.14: Subcircuit Schematic Diagram of CD4529BM

4.4.3 Test Circuit of CD4529BM

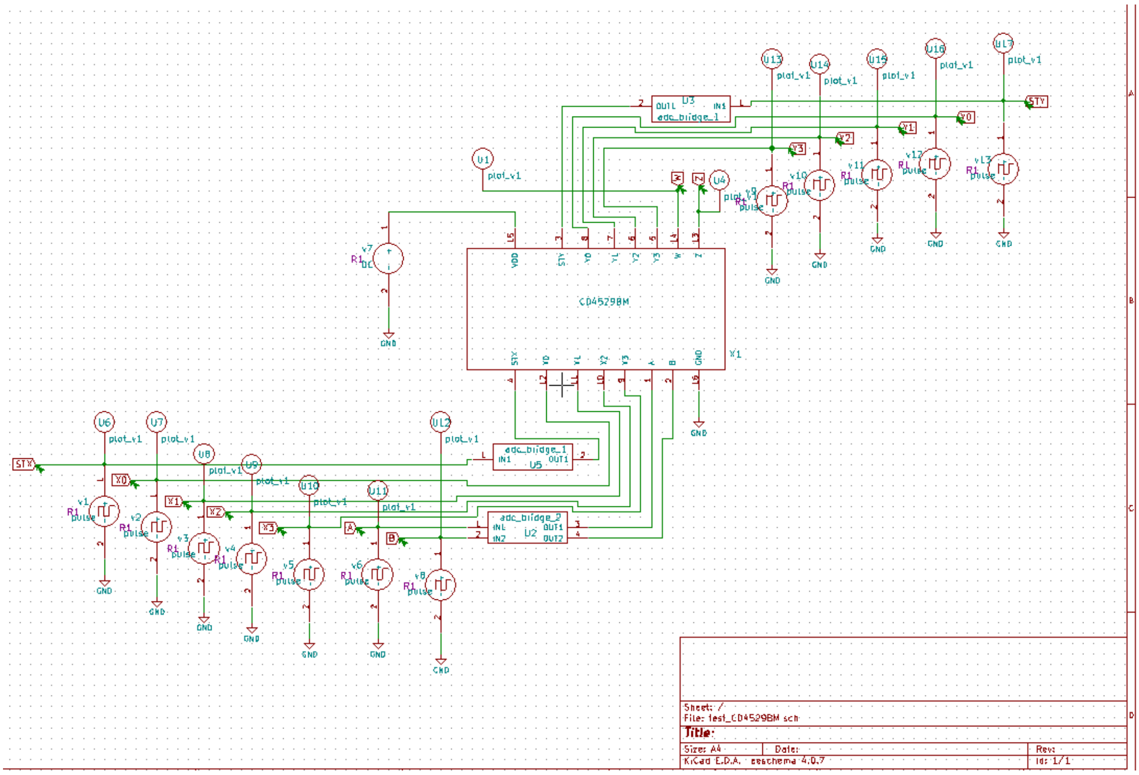


Figure 4.15: Test Circuit of CD4529BM

4.4.4 Input and Output Voltage Waveform of CD4529BM

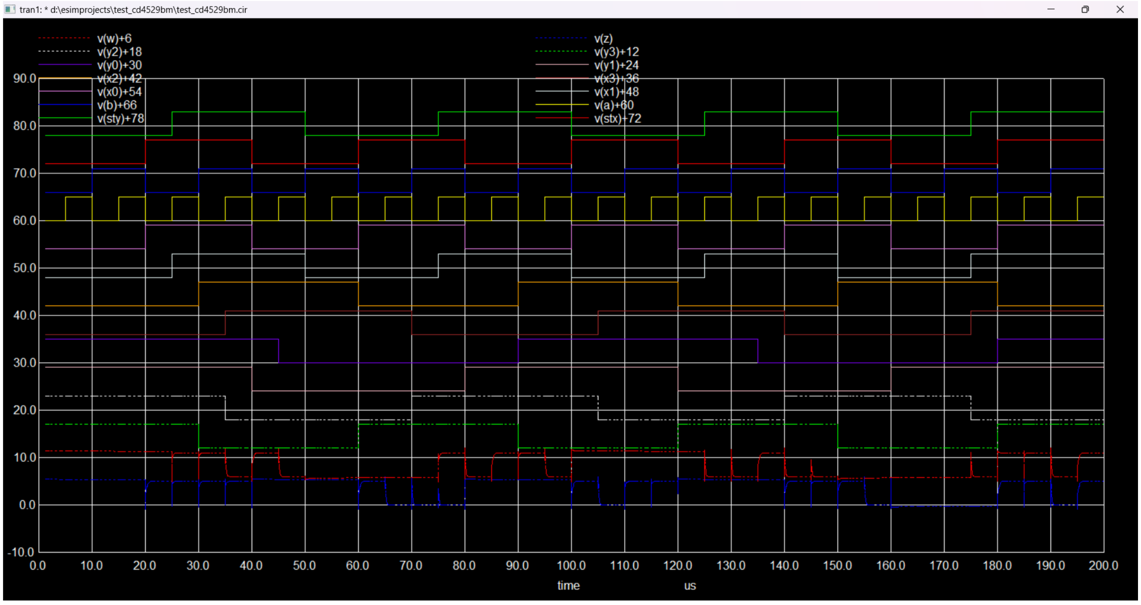


Figure 4.16: Input and Output Voltage Waveform of CD4529BM

4.5 CD4032B - CMOS Triple Serial Adders Positive Logic

Description: Triple Serial Adders designed with positive logic CMOS technology.

General Description: The CD4032B consists of three serial adder circuits with a common CLOCK and CARRY-RESET input. Each adder has two serial DATA inputs and an INVERT control. When INVERT is high, the output sum is complemented. Data is shifted in with the least significant bit (LSB) first and sign bit last. The MOD-2 sum is calculated from both inputs and the carry from the previous stage. In the CD4032B, carry propagation occurs on the positive edge of the clock. CARRY is reset to zero by a logic high on the CARRY-RESET line, one bit before the next word begins.

4.5.1 IC Layout

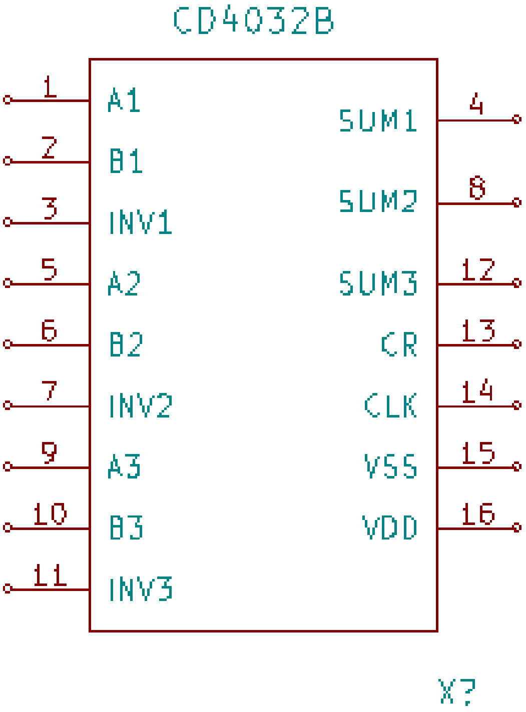


Figure 4.17: IC Layout of CD4032B

4.5.2 Subcircuit Schematic Diagram of CD4032B

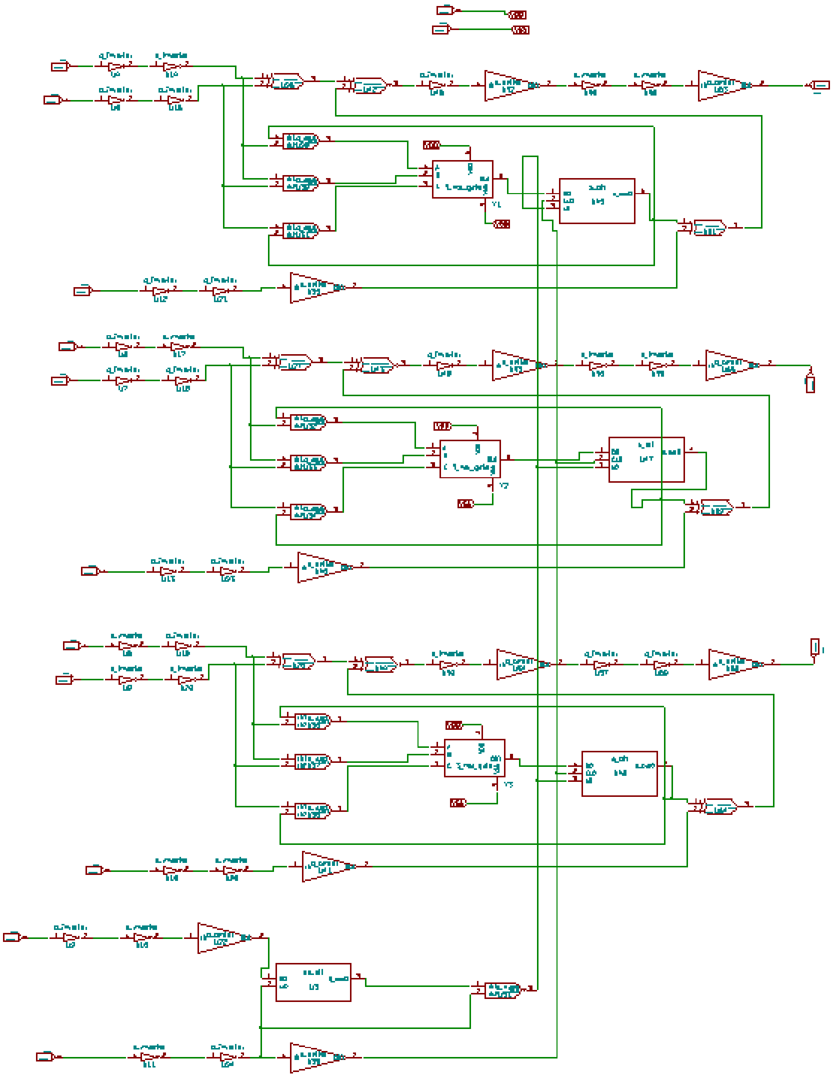


Figure 4.18: Subcircuit Schematic Diagram of CD4032B

4.5.3 Test Circuit of CD4032B

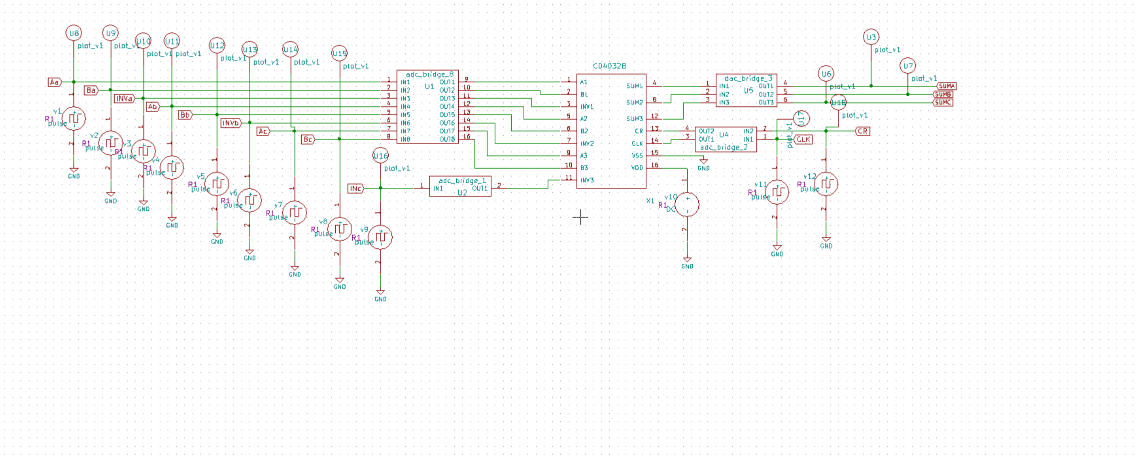


Figure 4.19: Test Circuit of CD4032B

4.5.4 Input and Output Voltage Waveform of CD4032B

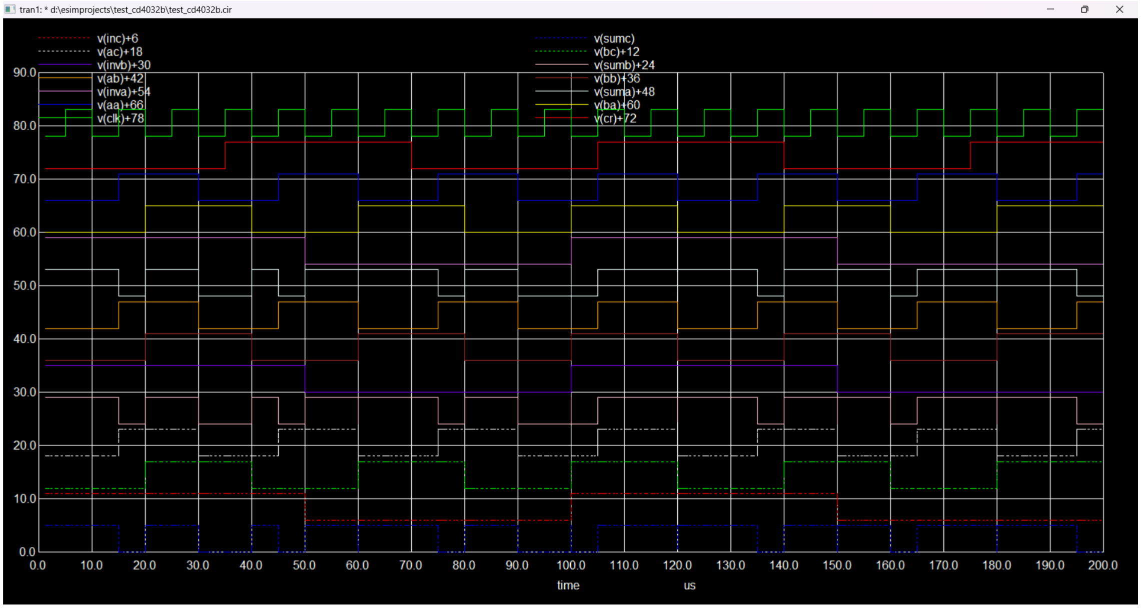


Figure 4.20: Input and Output Voltage Waveform of CD4032B

4.6 74HC20 - Dual 4-Input NAND Gate

Description: High-speed CMOS Dual 4-Input NAND Gate.

General Description: The 74HC20 is a dual 4-input NAND gate. Inputs include clamp diodes, enabling the use of current-limiting resistors to interface inputs to voltages in excess of VCC.

4.6.1 IC Layout

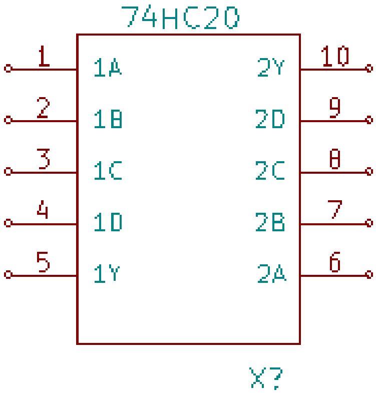


Figure 4.21: IC Layout of 74HC20

4.6.2 Subcircuit Schematic Diagram of 74HC20

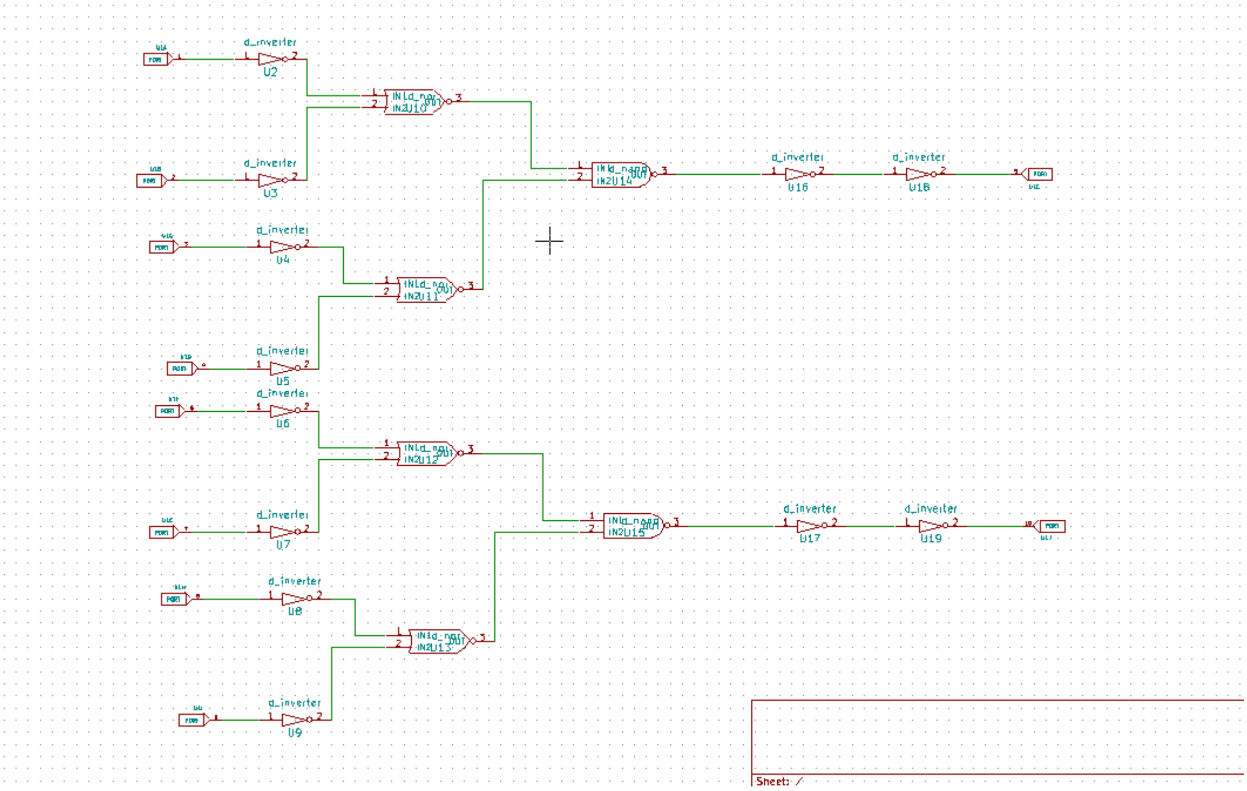


Figure 4.22: Subcircuit Schematic Diagram of 74HC20

4.6.3 Test Circuit of 74HC20

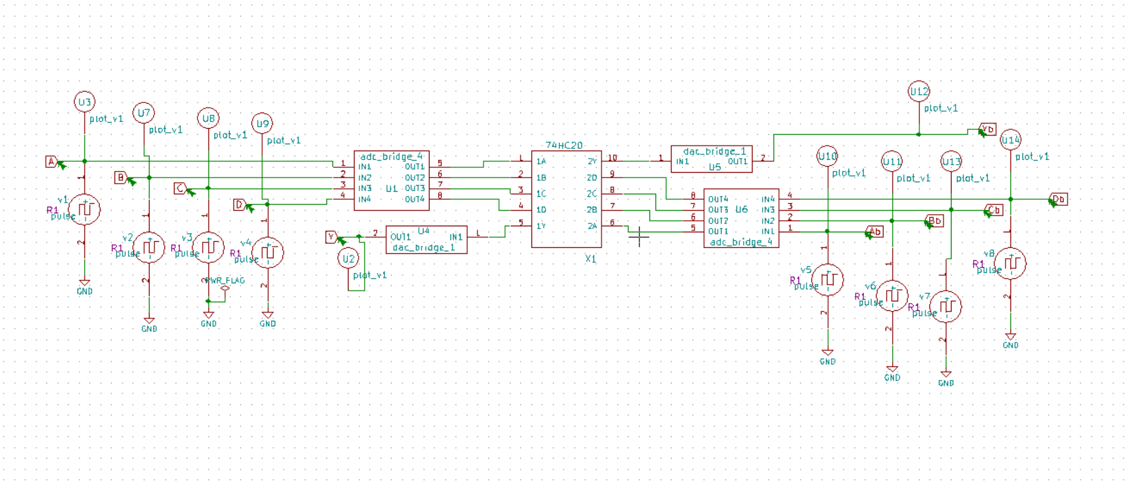


Figure 4.23: Test Circuit of 74HC20

4.6.4 Input and Output Voltage Waveform of 74HC20

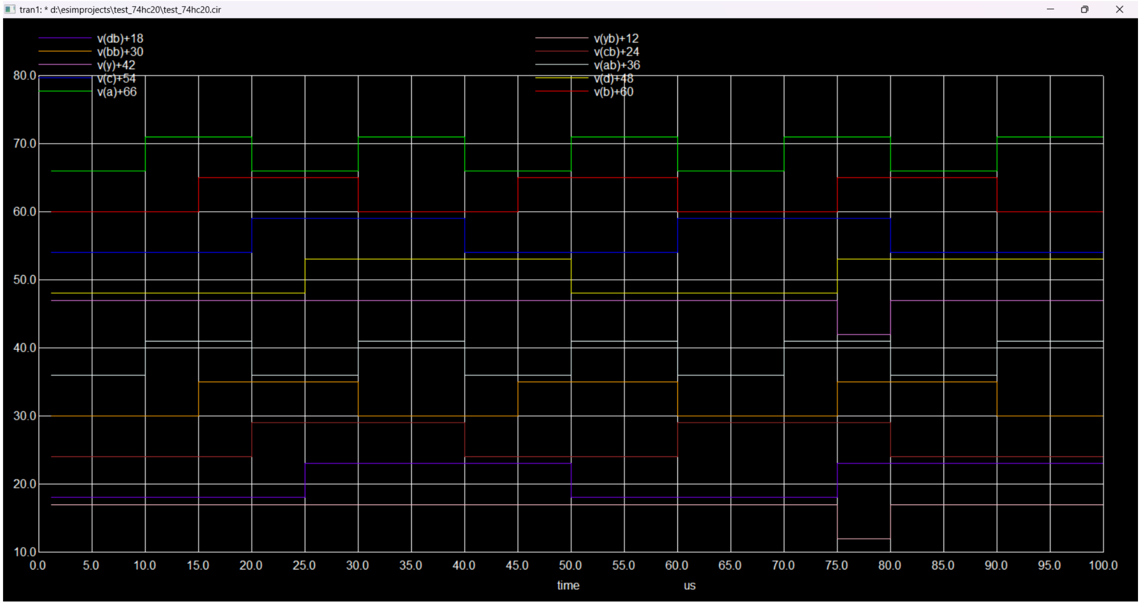


Figure 4.24: Input and Output Voltage Waveform of 74HC20

4.7 CD4010B-Q1 - CMOS Hex Buffer/Converter

Description: Hex Non-Inverting Buffer/Converter.

General Description: The CD4010B-Q1 consists of six non-inverting buffer-/converter circuits. Each buffer provides high input impedance and low output impedance, making it ideal for level shifting and signal buffering applications. The device can operate with different supply voltages on input and output sides, enabling voltage level conversion between different logic families. The buffers feature CMOS technology providing low power consumption and high noise immunity.

4.7.1 IC Layout

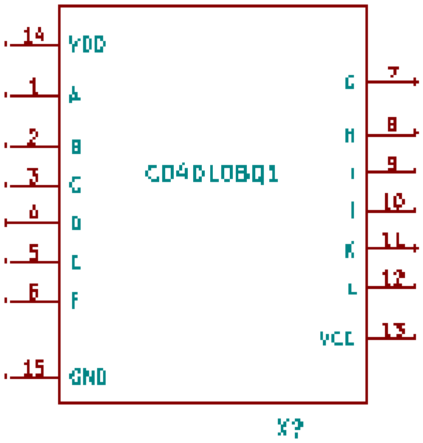


Figure 4.25: IC Layout of CD4010B-Q1

4.7.2 Schematic of One of Six Identical Stages

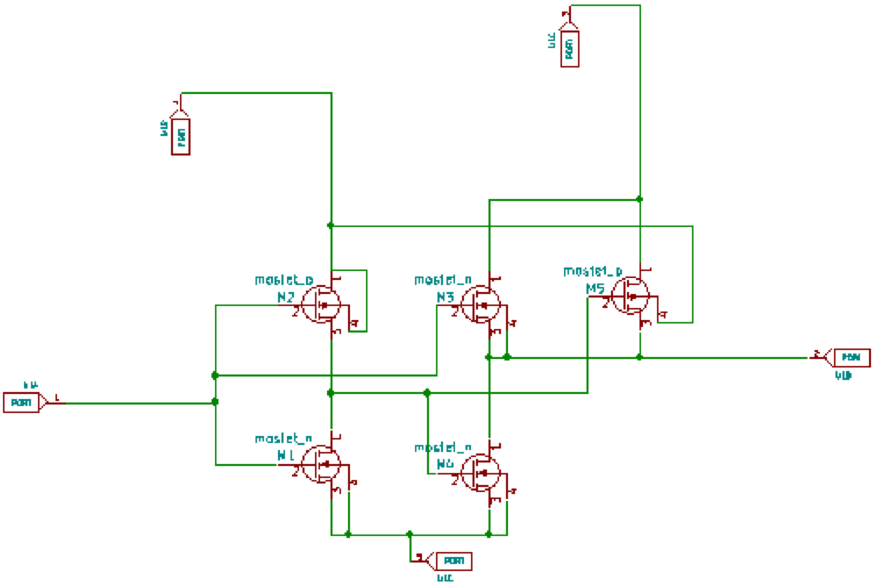


Figure 4.26: Schematic Diagram – One of Six Identical Stages

4.7.3 Subcircuit Schematic Diagram of CD4010B-Q1

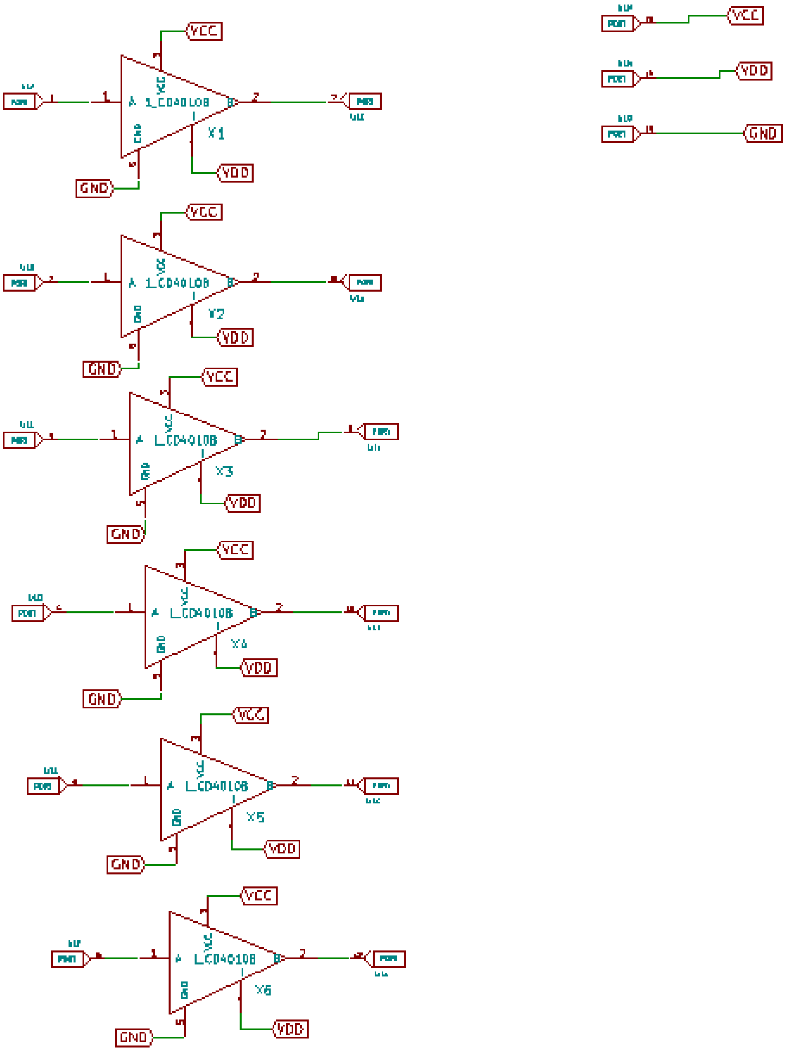


Figure 4.27: Subcircuit Schematic Diagram of CD4010B-Q1

4.7.4 Test Circuit of CD4010B-Q1

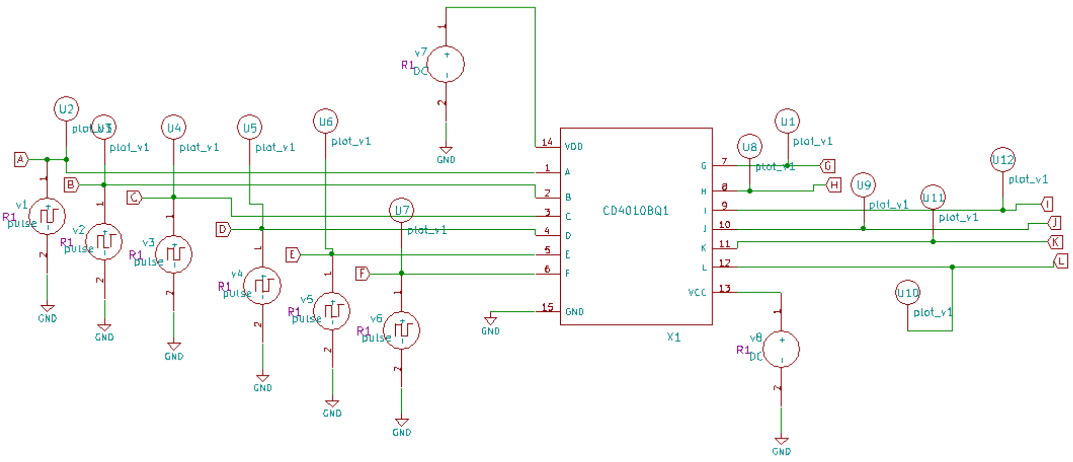


Figure 4.28: Test Circuit of CD4010B-Q1

4.7.5 Input and Output Voltage Waveform of CD4010B-Q1

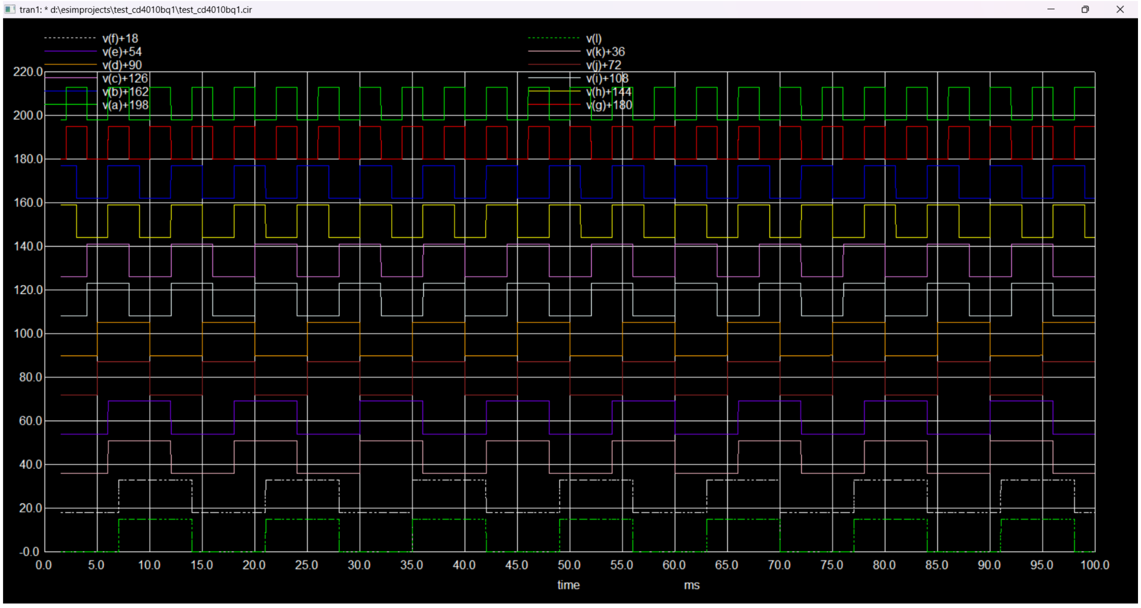


Figure 4.29: Input and Output Voltage Waveform of CD4010B-Q1

4.8 CD4038B - CMOS Triple Serial Adders Negative Logic

Description: Triple Serial Adders using CMOS technology with negative logic.

General Description: The CD4038B features three serial adder circuits sharing a common CLOCK and CARRY-RESET input. Each circuit accepts two DATA inputs and an INVERT command. A high INVERT signal complements the sum output. Data is input with LSB first and sign bit last. The adder produces a MOD-2 sum with carry from the prior bit. For CD4038B, carry addition occurs on the negative edge of the clock. Input transitions should follow the triggering edge for glitch-free output. CARRY is reset using a logic high signal applied just before the first bit of the next word.

4.8.1 IC Layout

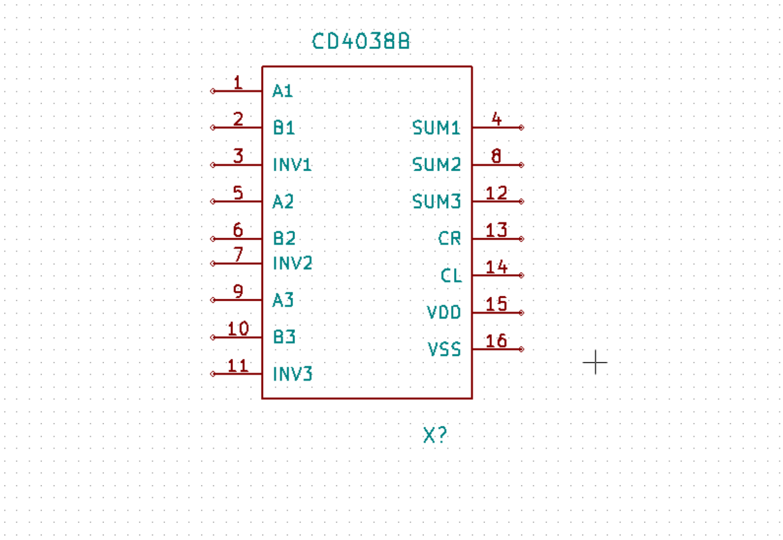


Figure 4.30: IC Layout of CD4038B

4.8.2 Subcircuit Schematic Diagram of CD4038B

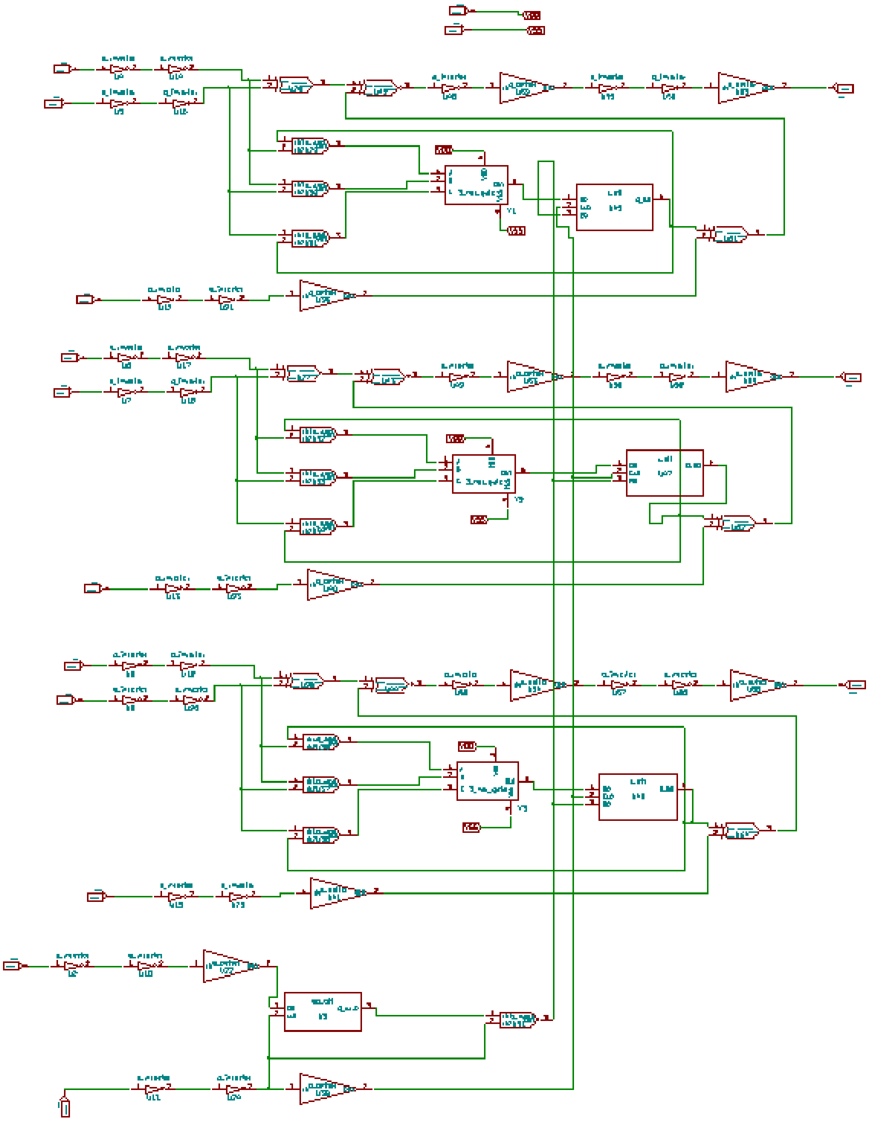


Figure 4.31: Subcircuit Schematic Diagram of CD4038B

4.8.3 Test Circuit of CD4038B

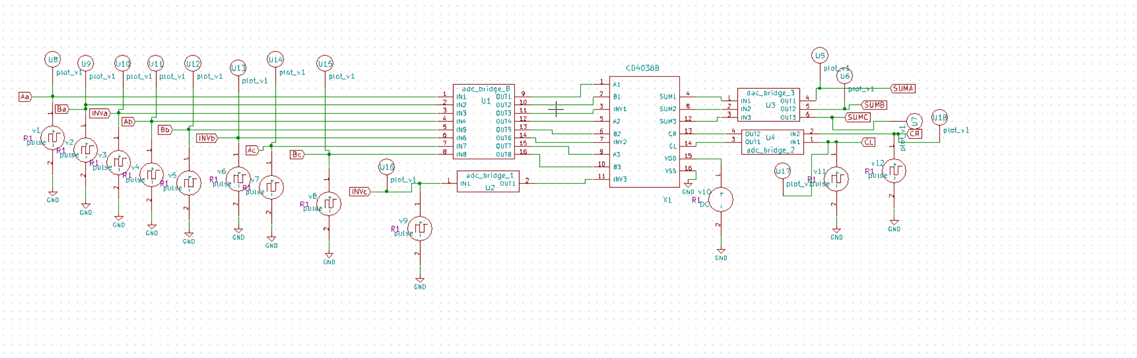


Figure 4.32: Test Circuit of CD4038B

4.8.4 Input and Output Voltage Waveform of CD4038B

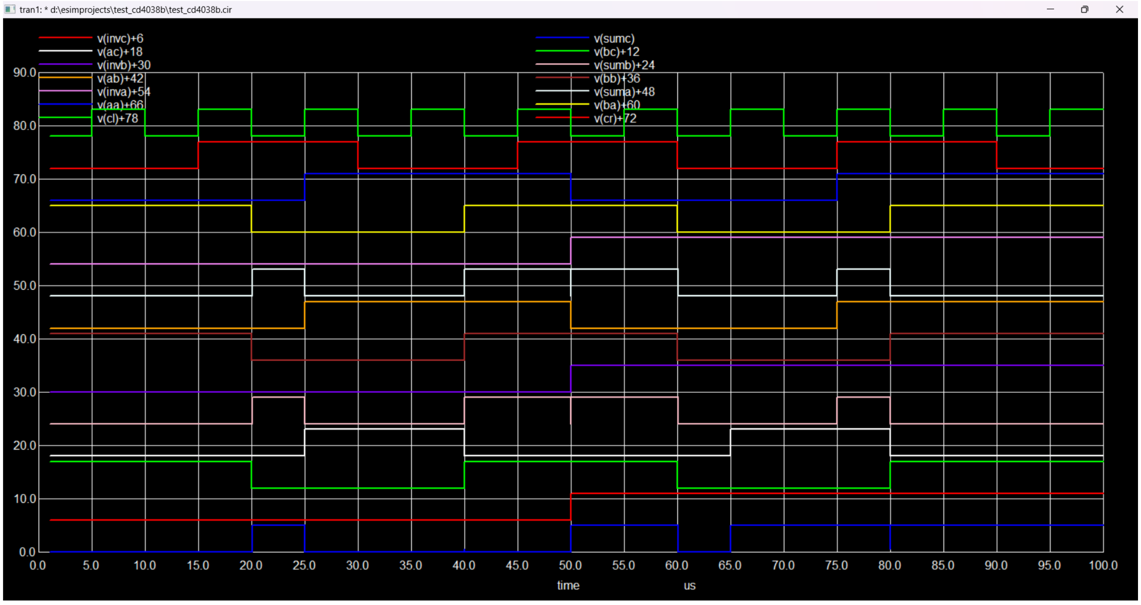


Figure 4.33: Input and Output Voltage Waveform of CD4038B

4.9 HCF4042B - Quad Clocked D Latch

Description: Quad Clocked D Latch with 3-state Outputs.

General Description: The HCC/HCF4042B types contain four latch circuits, each strobed by a common clock. Complementary buffered outputs are available from each circuit. The impedance of the n- and p-channel output devices is balanced, and all outputs are electrically identical. Information present at the data input is transferred to outputs Q and Q during the CLOCK level programmed by the POLARITY input. For POLARITY=0, the transfer occurs during the 0 CLOCK level, and for POLARITY=1, during the 1 CLOCK level. The outputs follow the data input provided the CLOCK and POLARITY levels are present. When a CLOCK transition occurs (positive for POLARITY=0, negative for POLARITY=1), the information present at the input during the CLOCK transition is retained at the outputs until an opposite CLOCK transition occurs.

4.9.1 IC Layout

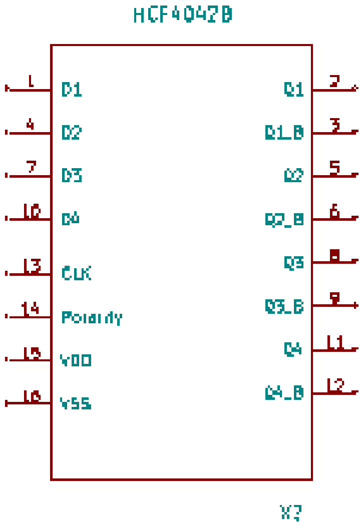


Figure 4.34: IC Layout of HCF4042B

4.9.2 Subcircuit Schematic Diagram of HCF4042B

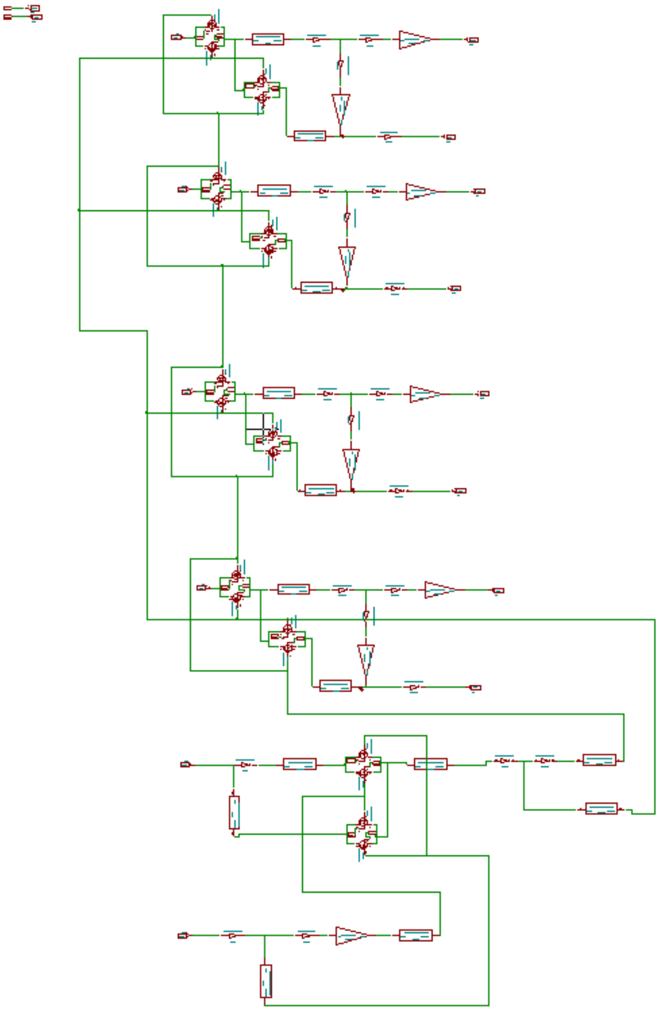


Figure 4.35: Subcircuit Schematic Diagram of HCF4042B

4.9.3 Test Circuit of HCF4042B

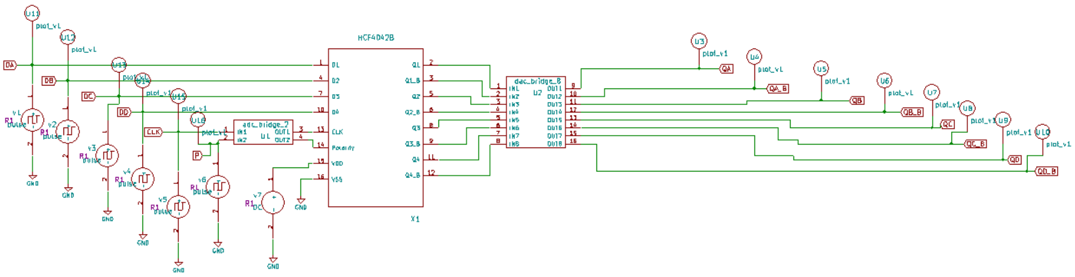


Figure 4.36: Test Circuit of HCF4042B

4.9.4 Input and Output Voltage Waveform of HCF4042B

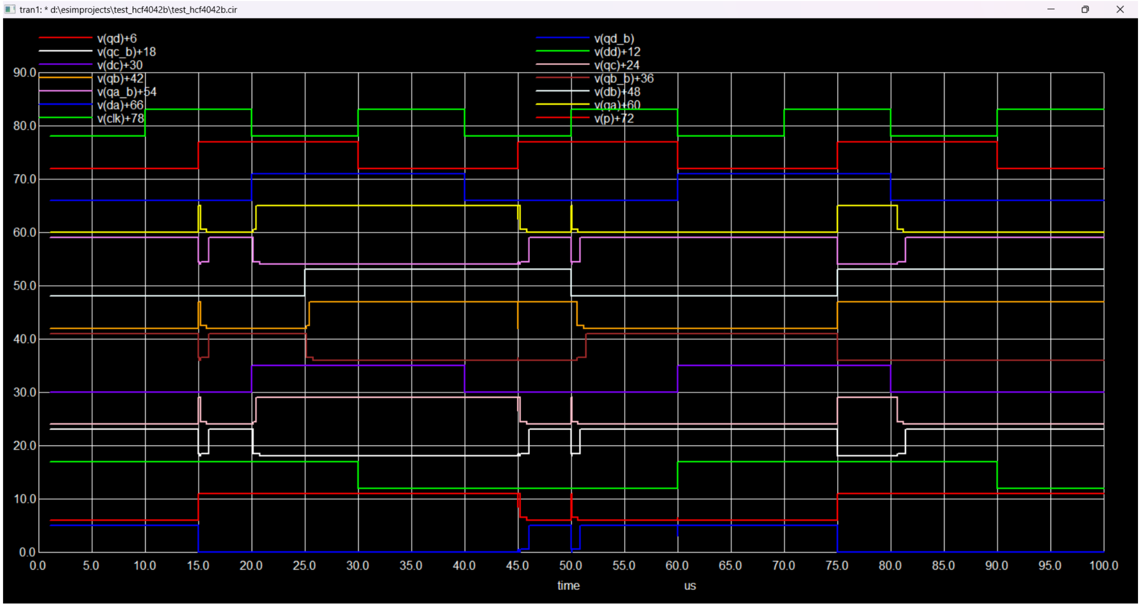


Figure 4.37: Input and Output Voltage Waveform of HCF4042B

4.10 SN54LVC157A - Quadruple 2-Line to 1-Line Data Selectors/Multiplexers

Description: High-Speed CMOS Quad 2-Line to 1-Line Data Selector/Multiplexer.

General Description: These quadruple 2-line to 1-line data selectors/multiplexers are designed for 1.65V to 3.6V VCC operation.

4.10.1 IC Layout

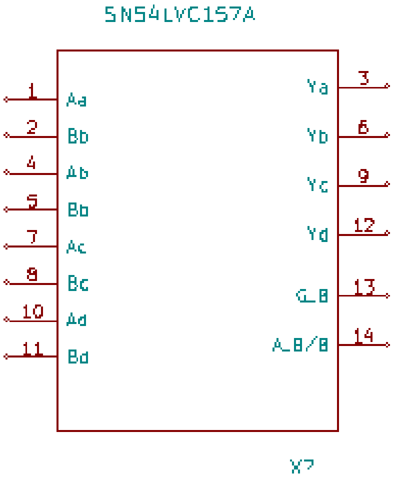


Figure 4.38: IC Layout of SN54LVC157A

4.10.2 Subcircuit Schematic Diagram of SN54LVC157A

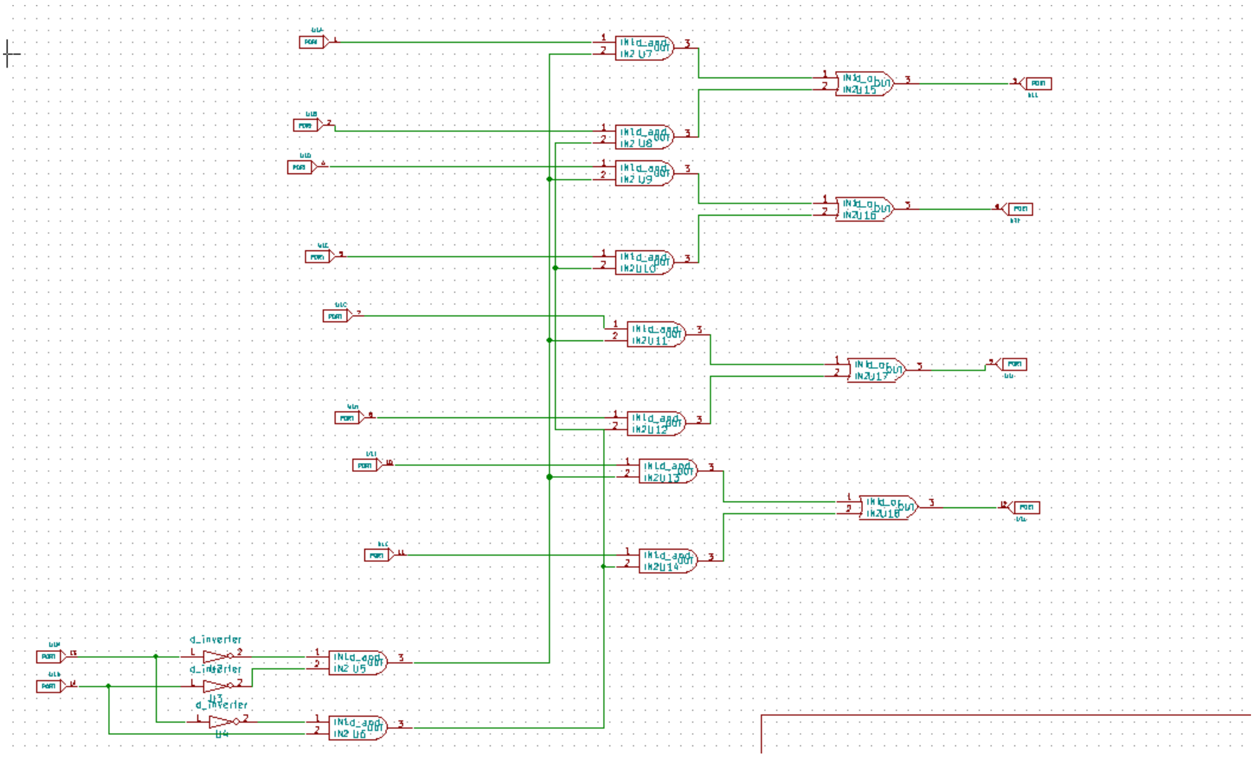


Figure 4.39: Subcircuit Schematic Diagram of SN54LVC157A

4.10.3 Test Circuit of SN54LVC157A

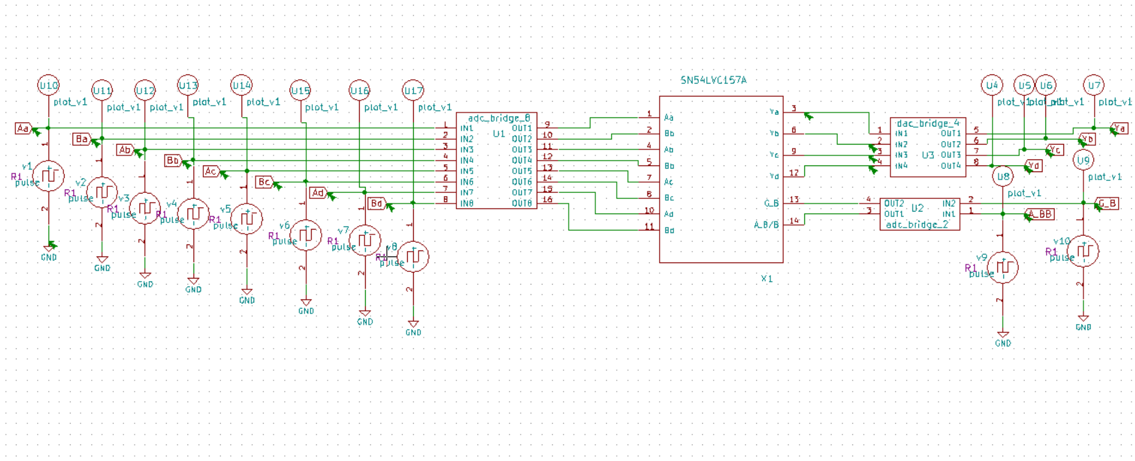


Figure 4.40: Test Circuit of SN54LVC157A

4.10.4 Input and Output Voltage Waveform of SN54LVC157A

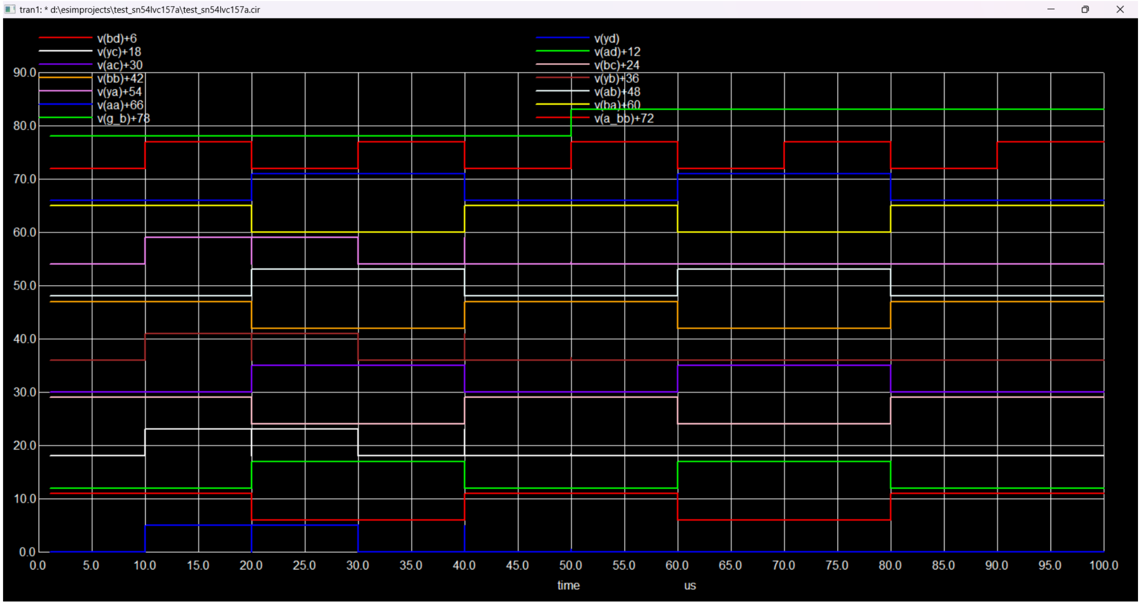


Figure 4.41: Input and Output Voltage Waveform of SN54LVC157A

4.11 Digital Design & Simulation using Icarus Verilog and eSim

Developed Verilog modules for a Configurable Duty Cycle PWM Generator and Ideal Vending Machine. Functional simulation and waveform analysis were performed using Icarus Verilog and GTKWave. The designs were then implemented in eSim, where schematic design and symbol creation were carried out based on the Verilog modules, followed by digital simulation.

4.12 PWM Generator with Configurable Duty Cycle

4.12.1 Description

This project implements a Pulse Width Modulation (PWM) signal generator using Verilog HDL, where the duty cycle can be dynamically adjusted using two control inputs — `increase_duty` and `decrease_duty`. The system is designed to generate a consistent PWM output frequency of 10 MHz.

4.12.2 Simulation using Icarus Verilog

Verilog Module

```
// PWM Generator with two debounced buttons (10% step duty cycle control)
module PWM_Generator_Verilog (
  input clk,                 // 100MHz clock input
  input increase_duty,       // Button to increase duty by 10%
  input decrease_duty,       // Button to decrease duty by 10%
  output PWM_OUT             // 10MHz PWM output
);
  reg [27:0] counter_debounce = 0;
  wire slow_clk_enable = (counter_debounce == 1); // For simulation only

  // Debounce signals
  wire tmp1, tmp2, duty_inc;
  wire tmp3, tmp4, duty_dec;

  // PWM generation
  reg [3:0] counter_PWM = 0;
  reg [3:0] DUTY_CYCLE = 5;

  // Slow clock for debounce
  always @(posedge clk) begin
    counter_debounce <= (counter_debounce >= 1) ? 0 : counter_debounce + 1;
  end

  // Debounce FFs
  DFF_PWM D1(clk, slow_clk_enable, increase_duty, tmp1);
  DFF_PWM D2(clk, slow_clk_enable, tmp1, tmp2);
  assign duty_inc = tmp1 & ~tmp2 & slow_clk_enable;

  DFF_PWM D3(clk, slow_clk_enable, decrease_duty, tmp3);
  DFF_PWM D4(clk, slow_clk_enable, tmp3, tmp4);
  assign duty_dec = tmp3 & ~tmp4 & slow_clk_enable;

  // Update duty cycle
  always @(posedge clk) begin
    if (duty_inc && DUTY_CYCLE < 10) DUTY_CYCLE <= DUTY_CYCLE + 1;
    else if (duty_dec && DUTY_CYCLE > 0) DUTY_CYCLE <= DUTY_CYCLE - 1;
  end

  // Generate 10MHz PWM signal
  always @(posedge clk) begin
    counter_PWM <= (counter_PWM >= 9) ? 0 : counter_PWM + 1;
  end

  assign PWM_OUT = (counter_PWM < DUTY_CYCLE);

endmodule

// 1-bit D Flip-Flop with enable (used for button debounce)
module DFF_PWM(input clk, input en, input D, output reg Q);
  always @(posedge clk) if (en) Q <= D;
endmodule
```

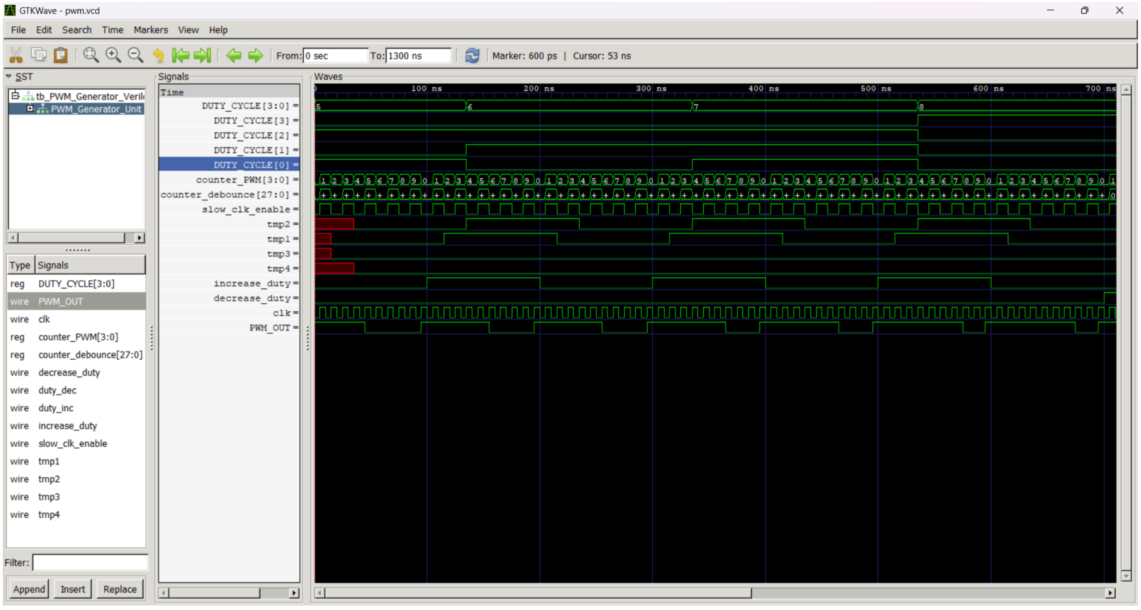
Verilog Testbench

```
`timescale 1ns / 1ps

module tb_PWM_Generator_Verilog;

  reg clk;
  reg increase_duty;
  reg decrease_duty;
  wire PWM_OUT;

  PWM_Generator_Verilog PWM_Generator_Unit(
    .clk(clk),
    .increase_duty(increase_duty),
    .decrease_duty(decrease_duty),
    .PWM_OUT(PWM_OUT)
  );

  // Generate 100 MHz clock
  initial begin
    clk = 0;
    forever #5 clk = ~clk;
  end
  // Add VCD dump
  initial begin
    $dumpfile("pwm.vcd");
    $dumpvars(0, tb_PWM_Generator_Verilog);
  end

  // Apply stimulus
  initial begin
    increase_duty = 0;
    decrease_duty = 0;

    #100; increase_duty = 1;
    #100; increase_duty = 0;

    #100; increase_duty = 1;
    #100; increase_duty = 0;

    #100; increase_duty = 1;
    #100; increase_duty = 0;

    #100; decrease_duty = 1;
    #100; decrease_duty = 0;

    #100; decrease_duty = 1;
    #100; decrease_duty = 0;

    #100; decrease_duty = 1;
    #100; decrease_duty = 0;

    #100; $finish;
  end
endmodule
```

The module and testbench were compiled and simulated using the following Icarus Verilog commands:

```
iverilog -o pwm_sim .\PWM_Generator_Verilog.v .\tb_PWM_Generator_Verilog.v
vvp .\pwm_sim
gtkwave pwm.vcd
```

4.12.3 Waveform Analysis using GTKWave



Type	Signals
reg	DUTY_CYCLE[3:0]
wire	PWM_OUT
wire	clk
reg	counter_PWM[3:0]
reg	counter_debounce[27:0]
wire	decrease_duty
wire	duty_dec
wire	duty_inc
wire	increase_duty
wire	slow_clk_enable
wire	tmp1
wire	tmp2
wire	tmp3
wire	tmp4

Figure 4.42: PWM Output Waveform observed in GTKWave

4.12.4 Implementation and Simulation in eSim

Verilog Module and Symbol Generation

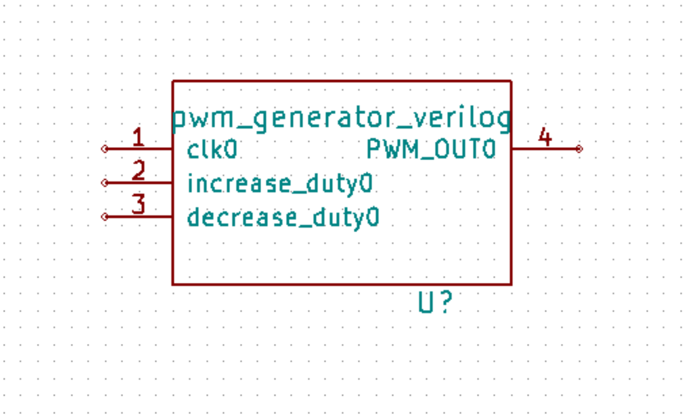


Figure 4.43: Generated Symbol of PWM

Schematic Design

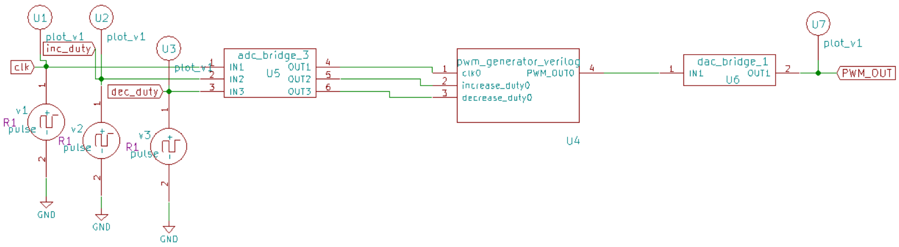


Figure 4.44: Schematic Design of PWM

Output Waveform

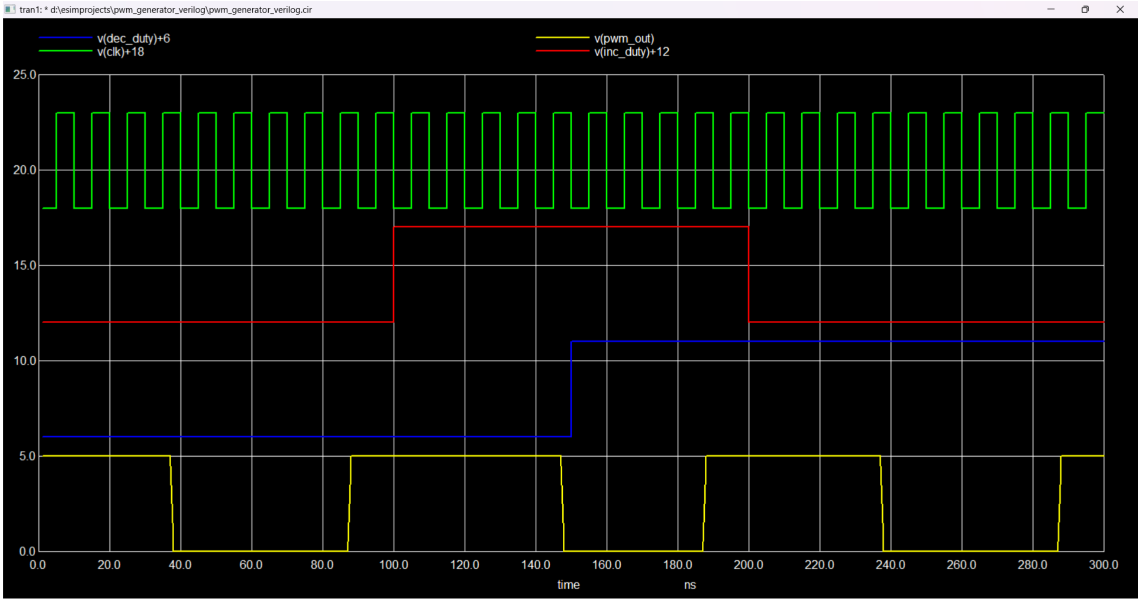


Figure 4.45: PWM Output Waveform from eSim Simulation

4.13 Ideal Vending Machine

4.13.1 Description

This is an ideal vending machine modeled using a finite state machine with three states: **S0**, **S1**, and **S2**.

- **S0** represents Rs. 0 inserted.
- **S1** represents Rs. 5 inserted.
- **S2** represents Rs. 10 inserted.

The product costs Rs. 15. When the total inserted amount reaches Rs. 15, the vending machine sets the output signal `out` high to indicate that the product is dispensed, and then returns to the initial state **S0**. If the transaction is incomplete or the amount inserted exceeds Rs. 15, the excess amount (change) is returned to the user.

4.13.2 Simulation using Icarus Verilog

Verilog Module

```
module ideal_vending_machine(
  input clk,
  input rst,
  input in_5,   // Pulse input for 5 Rs
  input in_10,  // Pulse input for 10 Rs
  output reg out,
  output reg [1:0] change
);

  parameter s0 = 2'b00;
  parameter s1 = 2'b01;
  parameter s2 = 2'b10;

  reg [1:0] c_state, n_state;

  always @(posedge clk) begin
    if (rst) begin
      c_state <= 0;
      n_state <= 0;
      change <= 2'b00;
      out <= 0;
    end else begin
      c_state <= n_state;
      case (c_state)
        s0: begin
          if (in_5) begin
            n_state <= s1;
            out <= 0;
            change <= 2'b00;
          end else if (in_10) begin
            n_state <= s2;
            out <= 0;
            change <= 2'b00;
          end else begin
            n_state <= s0;
            out <= 0;
            change <= 2'b00;
          end
        end
        s1: begin
          if (in_5) begin
            n_state <= s2;
            out <= 0;
            change <= 2'b00;
          end else if (in_10) begin
            n_state <= s0;
            out <= 1;
            change <= 2'b00;
          end else begin
            n_state <= s0;
            out <= 0;
            change <= 2'b01; // Return 5 Rs
          end
        end
        s2: begin
          if (in_5) begin
            n_state <= s0;
            out <= 1;
            change <= 2'b00;
          end else if (in_10) begin
            n_state <= s0;
            out <= 1;
            change <= 2'b01; // Return 5 Rs
          end else begin
            n_state <= s0;
            out <= 0;
            change <= 2'b10; // Return 10 Rs
          end
        end
      endcase
    end
  end
endmodule
```

Verilog Testbench

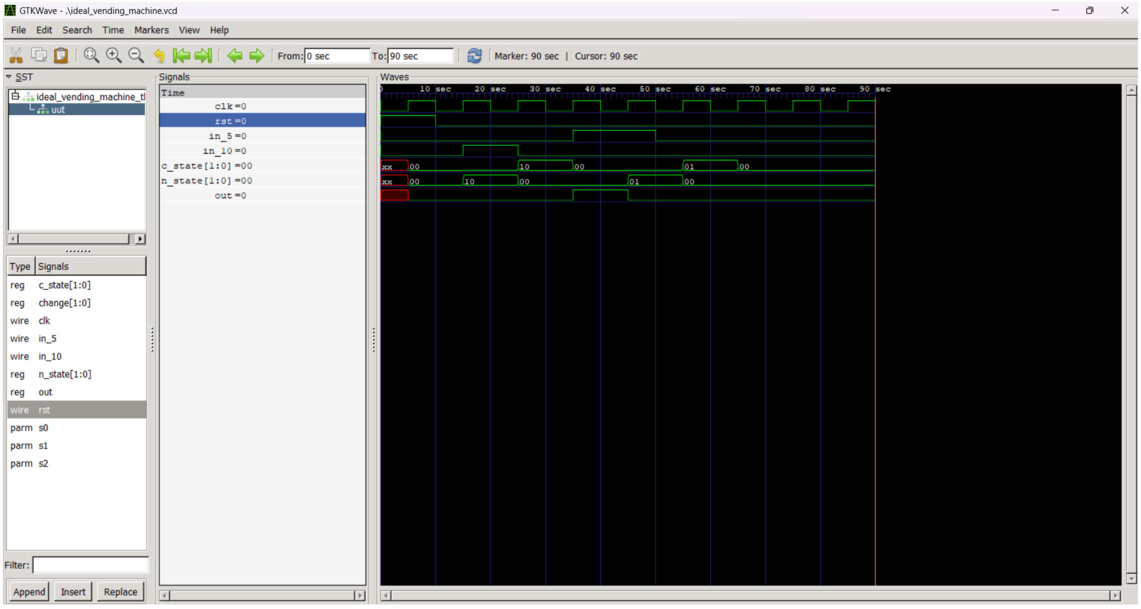
```
module ideal_vending_machine_tb;
```

```
  // Inputs
  reg clk;
  reg in_5;
  reg in_10;
  reg rst;

  // Outputs
  wire out;
  wire [1:0] change;

  // Instantiate the Unit Under Test (UUT)
  ideal_vending_machine uut (
    .clk(clk),
    .rst(rst),
    .in_5(in_5),
    .in_10(in_10),
    .out(out),
    .change(change)
  );

  initial begin
    $dumpfile("ideal_vending_machine.vcd");
    $dumpvars(0, ideal_vending_machine_tb);

    // Initialize inputs
    clk = 0;
    rst = 1;
    in_5 = 0;
    in_10 = 0;

    // Reset release
    #10 rst = 0;

    // Insert 10 Rs (pulse for 1 clk cycle)
    #5 in_10 = 1; #10 in_10 = 0; // Covers clk edge at t=15

    // Insert 5 Rs
    #10 in_5 = 1; #15 in_5 = 0;

    #40 $finish;
  end
  always #5 clk = ~clk; // Clock with 10 time units period
endmodule
```

The module and testbench were compiled and simulated using the following Icarus Verilog commands:

```
iverilog -o vm_sim .\ideal_vending_machine.v .\ideal_vending_machine_tb.v
vvp .\vm_sim
gtkwave .\ideal_vending_machine.vcd
```

4.13.3 Waveform Analysis using GTKWave


GTKWave - .\ideal_vending_machine.vcd
File Edit Search Time Markers View Help
From: 0 sec To: 90 sec
Marker: 90 sec | Cursor: 90 sec
SST
ideal_vending_machine_t
uut
Signals
Waves
Time
10 sec 20 sec 30 sec 40 sec 50 sec 60 sec 70 sec 80 sec 90 sec
Filter:
Append Insert Replace

```
clk =0
rst =0
in_5 =0
in_10 =0
c_state[1:0] =00
n_state[1:0] =00
out =0
```

Type	Signals
reg	c_state[1:0]
reg	change[1:0]
wire	clk
wire	in_5
wire	in_10
reg	n_state[1:0]
reg	out
wire	rst
parm	s0
parm	s1
parm	s2

Figure 4.46: Ideal Vending Machine Output Waveform observed in GTKWave

4.13.4 Implementation and Simulation in eSim

Verilog Module and Symbol Generation

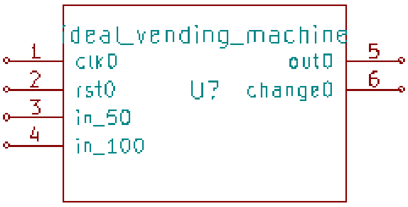


Figure 4.47: Generated Symbol of Ideal Vending Machine

Schematic Design

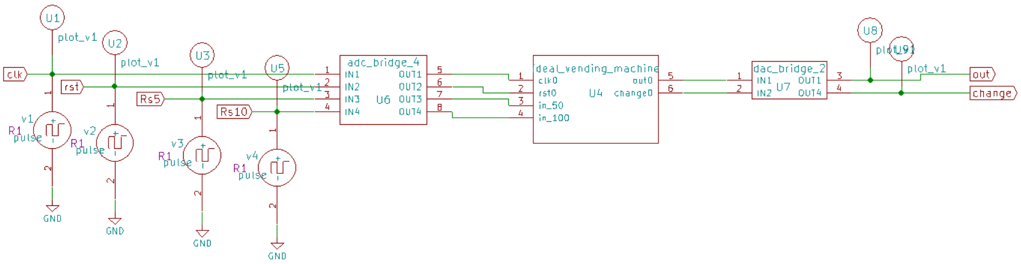


Figure 4.48: Schematic Design of Ideal Vending Machine

Output Waveform

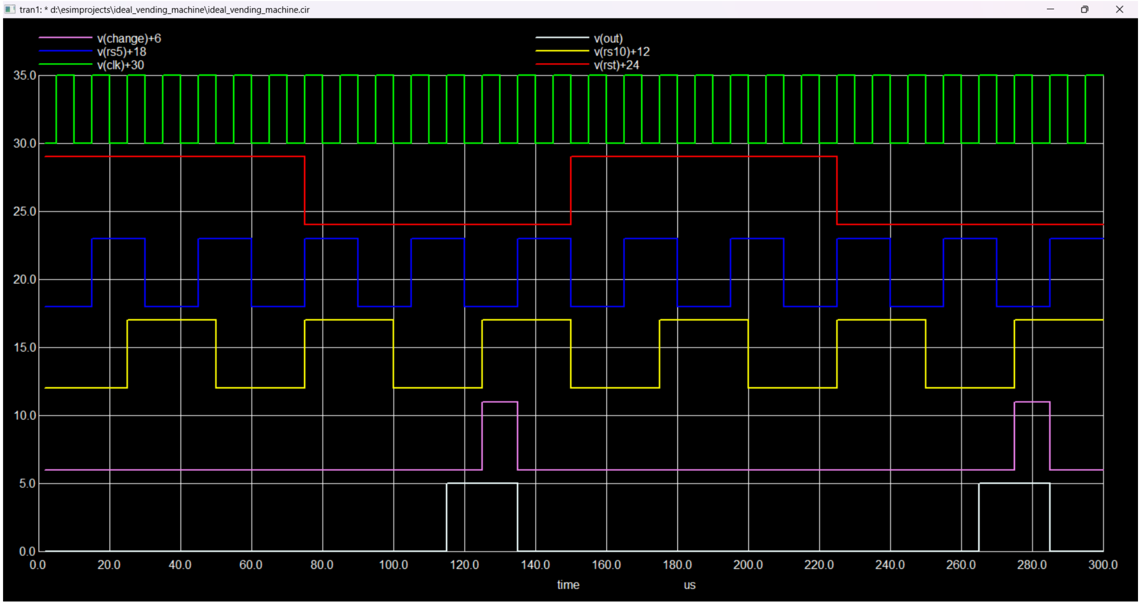


Figure 4.49: Ideal Vending Machine Output Waveform from eSim Simulation

4.14 Failed ICs

4.14.1 Overview

In this section, we discuss circuits that did not perform as expected during testing. Understanding the reasons for these failures helps in diagnosing issues and improving circuit design.

4.14.2 CD4724BC

There is an issue with this circuit showing a zero-length vector error in NGSPICE. During simulation, the circuit fails to generate any output data, and NGSPICE reports that the vector length is zero, preventing waveform plots or numerical results.

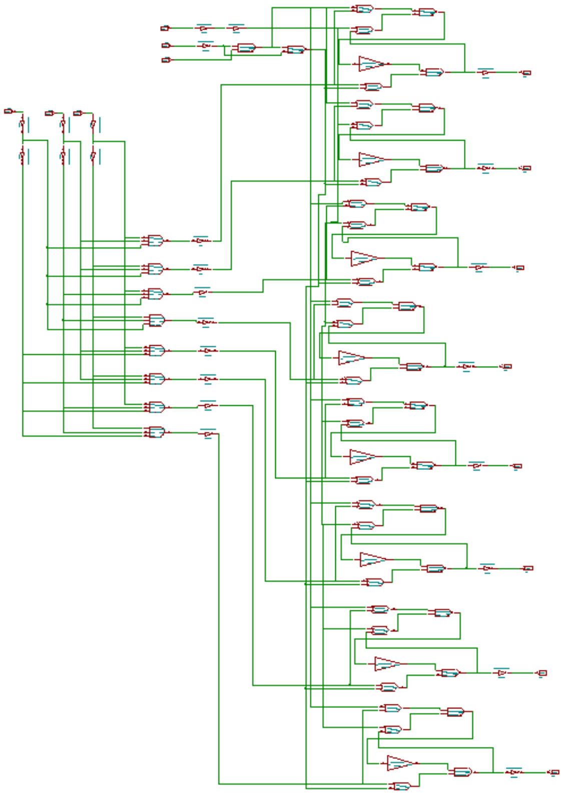


Figure 4.50: Subcircuit of CD4724BC

4.14.3 CD4024BC

The input is receiving millivolts (mV) instead of volts (V) as expected. This affects the circuit's ability to process digital signals properly, as the input levels are insufficient to trigger the internal logic gates.

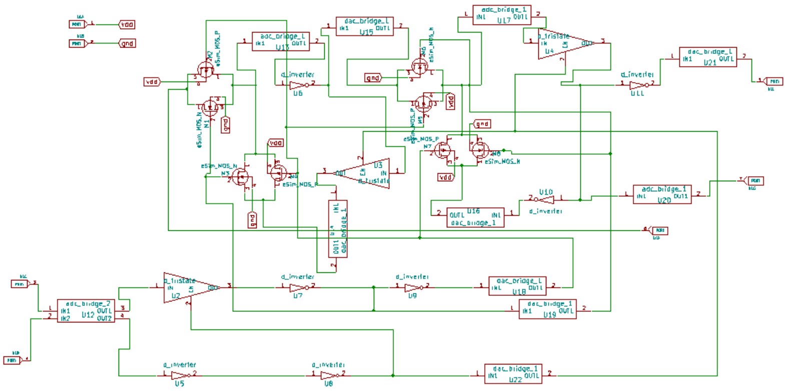


Figure 4.51: Subcircuit of CD4024BC

4.15 Conclusion and Future Scope

The project achieved its objective of developing a wide range of subcircuits for Digital Integrated Circuits, with each IC model meticulously crafted based on the specifications provided in their official datasheets. Each IC model was designed strictly in accordance with its respective datasheet. The functionality of each IC was verified using standard test circuits. Additionally, several digital IC models were implemented as Verilog modules using eSim. All developed ICs are now ready to be integrated into the subcircuit library of eSim. These models can serve as functional units for researchers and students in their circuit designs. With the continued growth of eSim's device model library, more such ready-to-use digital IC models are expected to be developed in the future.

Bibliography

[1] FOSSEE Official Website. 2020. [Online]. Available: https://fossee.in/about

[2] All Datasheet. *Electronics Component Database and Datasheets*. 2025. [Online]. Available: https://www.alldatasheet.com/

[3] ON Semiconductor. *ON Semiconductor Official Website (formerly Fairchild Semiconductor)*. 2025. [Online]. Available: https://www.onsemi.com/

[4] STMicroelectronics. *STMicroelectronics Official Website*. 2025. [Online]. Available: https://www.st.com/

[5] Texas Instruments. *Texas Instruments Official Website (includes former National Semiconductor products)*. 2025. [Online]. Available: https://www.ti.com/

[6] Nexperia. *Nexperia Official Website*. 2025. [Online]. Available: https://www.nexperia.com/

[7] Datasheet Catalog. *Electronic Components Datasheet Archive*. 2025. [Online]. Available: https://www.datasheetcatalog.com/

[8] FPGA4Student. "Verilog Code for PWM Generator," *FPGA4Student*, 2017. [Online]. Available: https://www.fpga4student.com/2017/08/verilog-code-for-pwm-generator.html