REPORT

TOPIC:

A dynamic current mode design approach of 2/3 prescaler for phase locked loop application.

Submitted By:

1) Saransh Bansal
2) Mukund Sikarwar
3) Vedant Drona
4) Shubham Gupta

Submitted To:

Dr. Bharat Choudhary

Department Of ECE

MNIT, Jaipur

What we have understood from the paper.

What is this Paper About?

This paper presents a **low-power, high-speed frequency divider circuit**—a **2/3 dual-modulus prescaler**—designed using **Dynamic Current Mode Logic (DyCML)**. This circuit is intended for use in **Phase-Locked Loops (PLLs)** and **frequency synthesizers**, which are essential components in communication systems such as radios, wireless transceivers, and clock recovery modules.

Why is this Important?

In modern high-speed communication systems, accurate frequency division is critical for managing and distributing clock signals.

However, traditional **Current Mode Logic (CML)** designs:

- Consume high **static power**
- Are **not suitable for power-down modes**
- Involve **complex load designs**

The proposed DyCML-based approach:

- **Reduces static power dissipation**
- **Improves speed**
- Is **scalable** to advanced CMOS nodes (e.g., **28 nm**)

Main Contributions of the Paper

- Implementation of a **2/3 dual-modulus prescaler** using **Dynamic Current Mode Logic (DyCML)**
- **Low power consumption**: ~2.5 mW at 2 GHz input
- **Wide input frequency range**: 2.4 to 10 GHz
- **Low phase noise**: –147 dBc/Hz (ensures better signal quality)
- **Scalability**: Tested on both **90 nm** and **28 nm** CMOS technologies
- **Robustness**: Verified under **5% process variation** scenarios

How Does the 2/3 Prescaler Work?

A **prescaler** divides the input clock frequency into two modes based on the **Modulus Control (MC)** signal:

- **Divide-by-2 Mode (MC = 1)**:
  The circuit behaves like a **T flip-flop**, giving:
  f_out = f_in / 2
- **Divide-by-3 Mode (MC = 0)**:
  Uses delayed feedback through **NOR gates** to create a loop that results in:
  f_out = f_in / 3

Architecture Details

- Constructed using **two dynamic D flip-flops** and **two NOR gates**
- Implements **DyCML logic** which:
  - Eliminates static current paths
  - Uses **precharge and evaluation phases** controlled by the clock
  - Ensures **low power and high-speed switching**

Performance Highlights

Parameter	**Result**
Power Consumption	~2.5 mW at 2 GHz
Phase Noise	–147 dBc/Hz
Output Noise	–181.7 dB
Max Operating Frequency	10 GHz
Self-Oscillation Frequency	5 GHz
Process Variation Tolerance	Yes (5% skew tested)
Technology Nodes Tested	90 nm & 28 nm CMOS

Comparison with Previous Works

Compared to earlier designs:

- Consumes the **lowest power** at high frequencies
- Achieves **better phase noise performance**
- Uses **moderate chip area**
- Offers **broader frequency range**
- Is **more scalable** to advanced technology nodes

Validation at 28 nm Technology

To confirm its **scalability**, the design was implemented on **28 nm UMC CMOS** with a 1V supply:

Metric	Value
Power Consumption	2.86 mW
Phase Noise	–152.4 dBc/Hz
Output Noise	–183.2 dB
Frequency Range	1.5 – 8.5 GHz

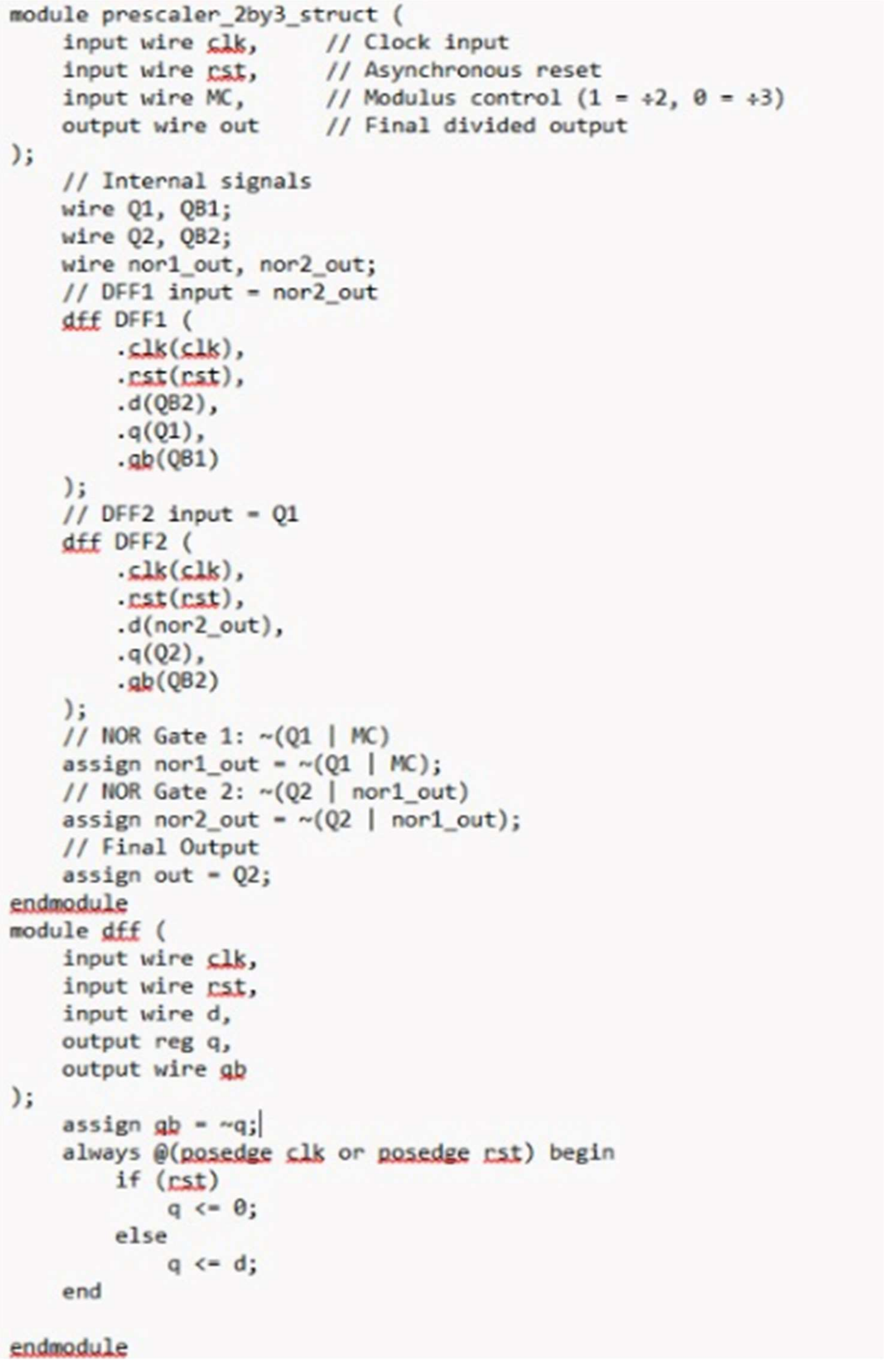
Conclusion

This **2/3 prescaler design**:

- Reduces power significantly using **DyCML**
- Supports **high-frequency** operation up to 10 GHz
- Is **robust to process variations**
- Is **scalable** to modern CMOS nodes like **28 nm**
- Is ideal for integration into **PLLs**, **frequency synthesizers**, and modern **RF communication systems**

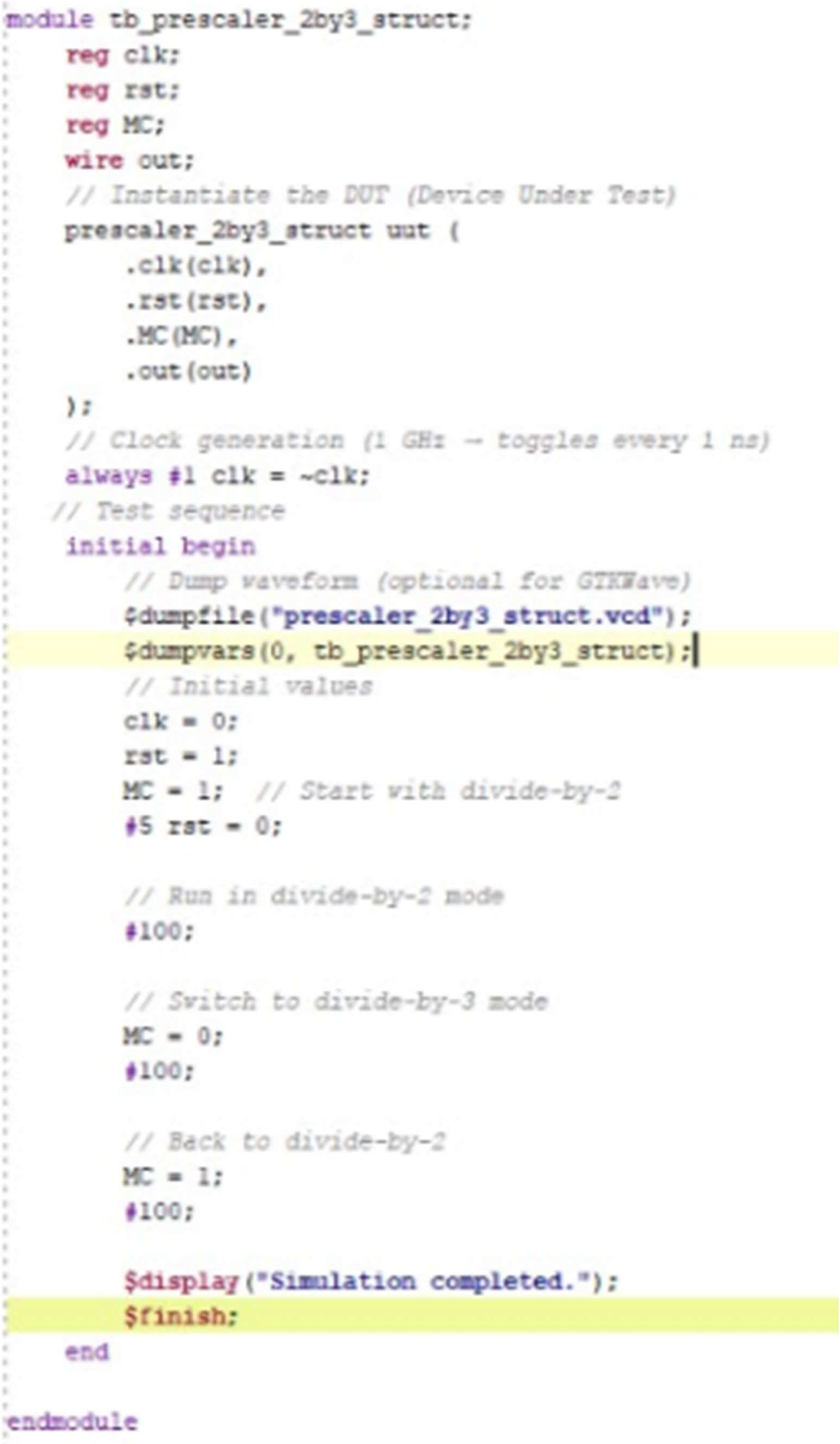
What we have analysed while implementing the same on Xilinx Vivado.

The Design we have written in Verilog is as follows:



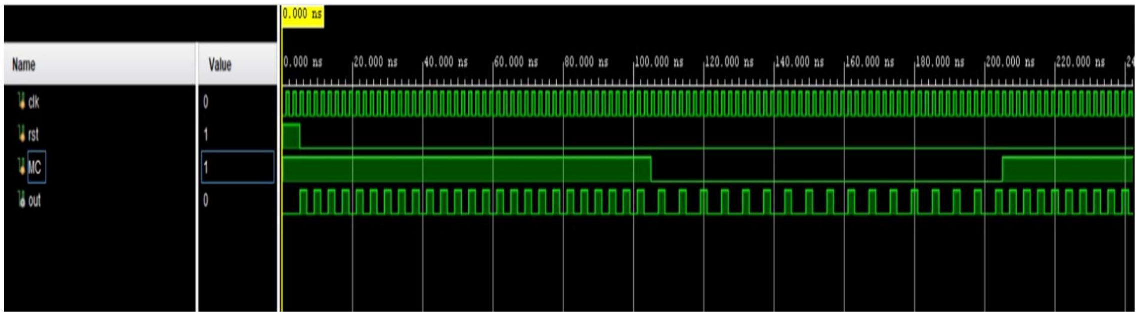
```
module prescaler_2by3_struct (
    input wire clk,      // Clock input
    input wire rst,      // Asynchronous reset
    input wire MC,       // Modulus control (1 = ÷2, 0 = ÷3)
    output wire out      // Final divided output
);
    // Internal signals
    wire Q1, QB1;
    wire Q2, QB2;
    wire nor1_out, nor2_out;
    // DFF1 input = nor2_out
    dff DFF1 (
        .clk(clk),
        .rst(rst),
        .d(QB2),
        .q(Q1),
        .qb(QB1)
    );
    // DFF2 input = Q1
    dff DFF2 (
        .clk(clk),
        .rst(rst),
        .d(nor2_out),
        .q(Q2),
        .qb(QB2)
    );
    // NOR Gate 1: ~(Q1 | MC)
    assign nor1_out = ~(Q1 | MC);
    // NOR Gate 2: ~(Q2 | nor1_out)
    assign nor2_out = ~(Q2 | nor1_out);
    // Final Output
    assign out = Q2;
endmodule
module dff (
    input wire clk,
    input wire rst,
    input wire d,
    output reg q,
    output wire qb
);
    assign qb = ~q;
    always @(posedge clk or posedge rst) begin
        if (rst)
            q <= 0;
        else
            q <= d;
    end

endmodule
```

The Testbench we have written in Verilog is as follows:



```
module tb_prescaler_2by3_struct;
    reg clk;
    reg rst;
    reg MC;
    wire out;
    // Instantiate the DUT (Device Under Test)
    prescaler_2by3_struct uut (
        .clk(clk),
        .rst(rst),
        .MC(MC),
        .out(out)
    );
    // Clock generation (1 GHz → toggles every 1 ns)
    always #1 clk = ~clk;
   // Test sequence
    initial begin
        // Dump waveform (optional for GTKWave)
        $dumpfile("prescaler_2by3_struct.vcd");
        $dumpvars(0, tb_prescaler_2by3_struct);
        // Initial values
        clk = 0;
        rst = 1;
        MC = 1;  // Start with divide-by-2
        #5 rst = 0;

        // Run in divide-by-2 mode
        #100;

        // Switch to divide-by-3 mode
        MC = 0;
        #100;

        // Back to divide-by-2
        MC = 1;
        #100;

        $display("Simulation completed.");
        $finish;
    end

endmodule
```

Finally, after running simulation the waveform output we have got is as follows:



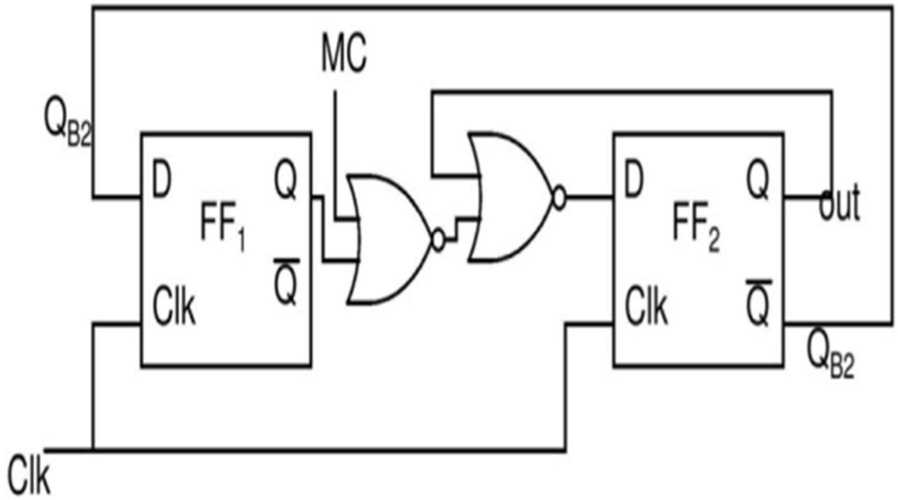
Name	Value
clk	0
rst	1
MC	1
out	0

Observation from the Simulation:

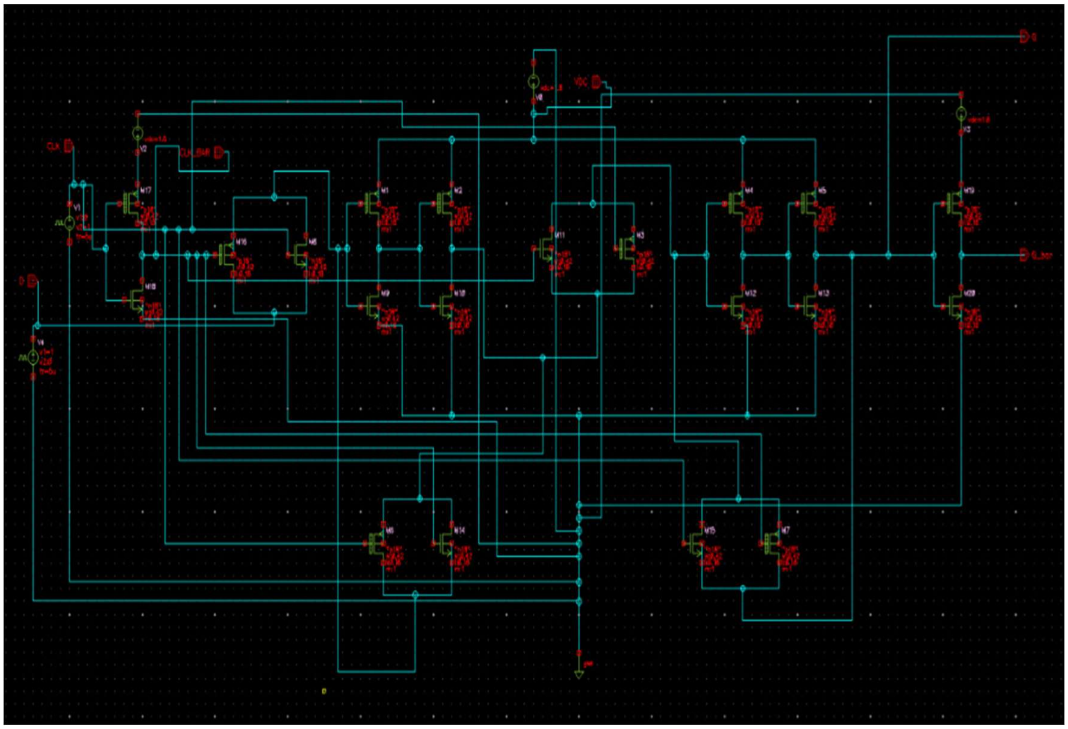
- Initially, the Modulus Control (MC) is set to 1, enabling divide-by-2 mode.
    - From 20 ns to 120 ns, the output (out) shows a frequency that is exactly half of the input clock (clk).
    - This confirms that the circuit correctly functions as a divide-by-2 prescaler in this interval.
- At 120 ns, the MC signal switches to 0, activating divide-by-3 mode.
    - From this point onward, the output waveform exhibits a lower frequency—specifically, the output toggles once every 3 input clock cycles.
    - This confirms successful transition to divide-by-3 operation

What we have analysed while implementing the same on Cadence Virtuoso.

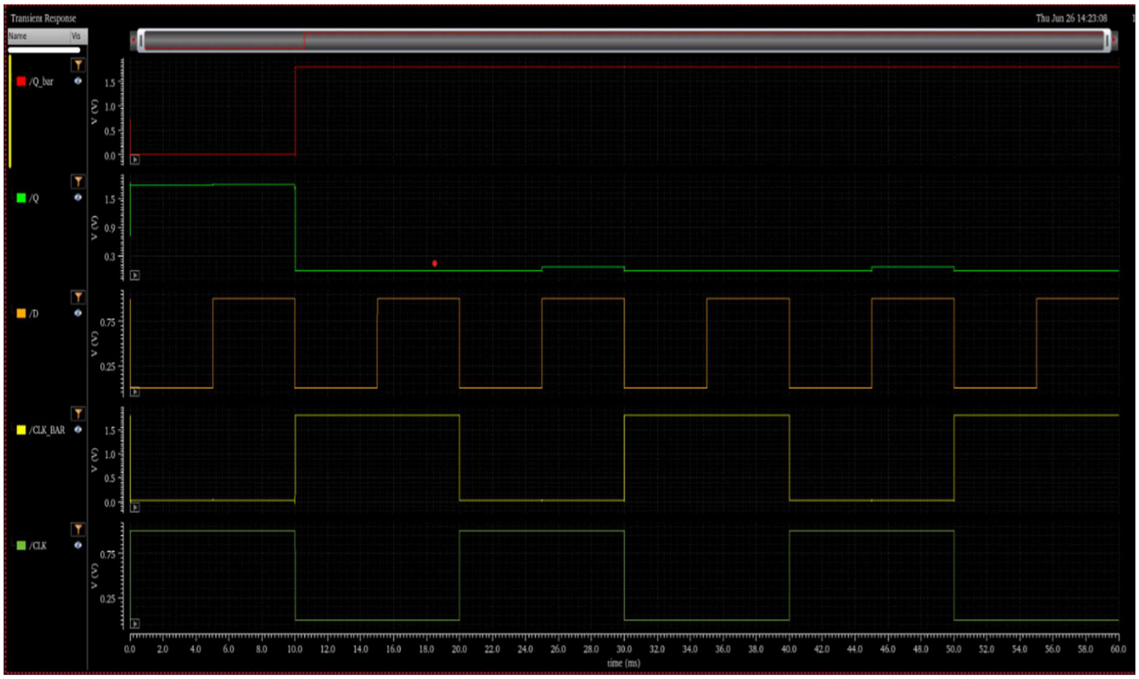
The Circuit that we used for our analysis:



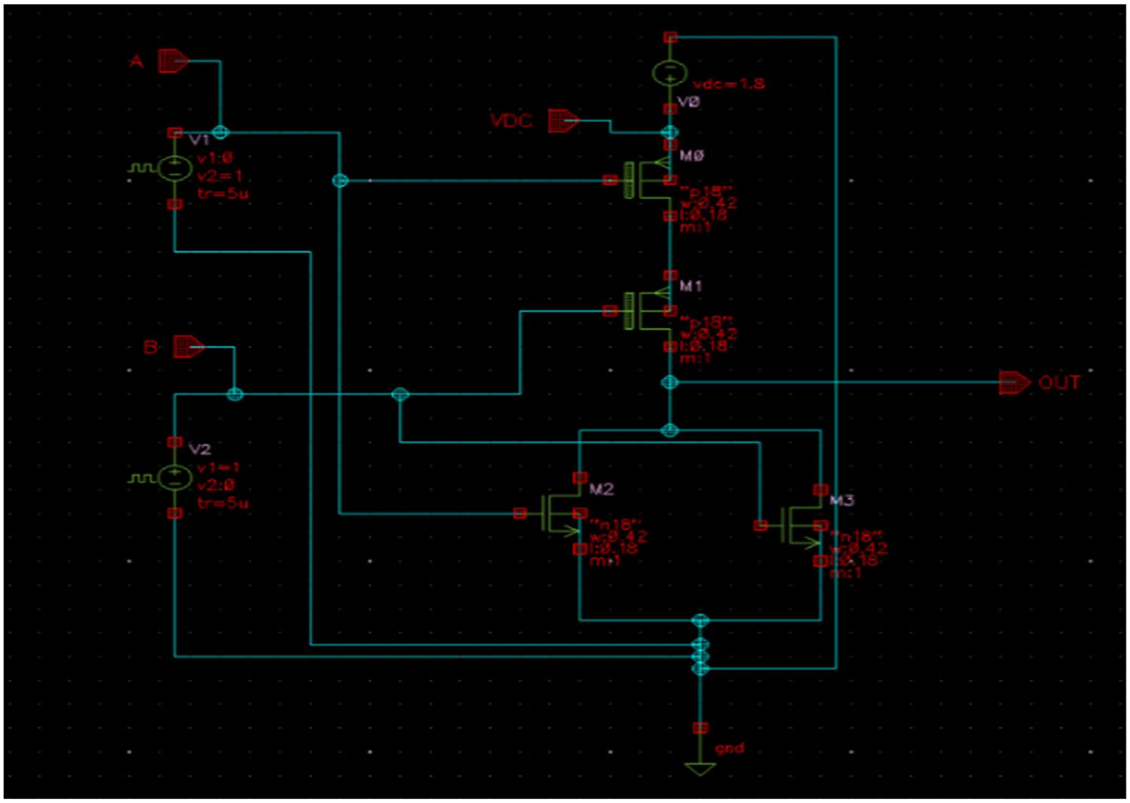
For this firstly we have designed an D Flip Flop (Master Slave Configuration) Using CMOS.



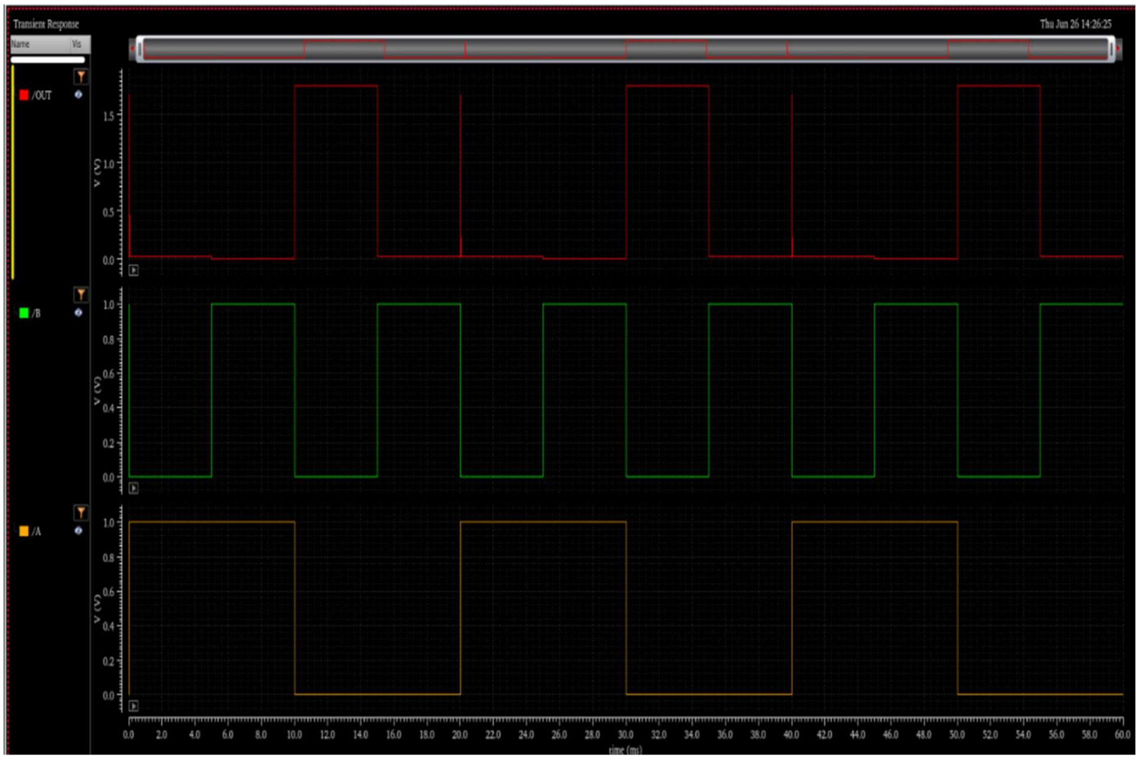
Also we have verified waveform outputs from the truth table:



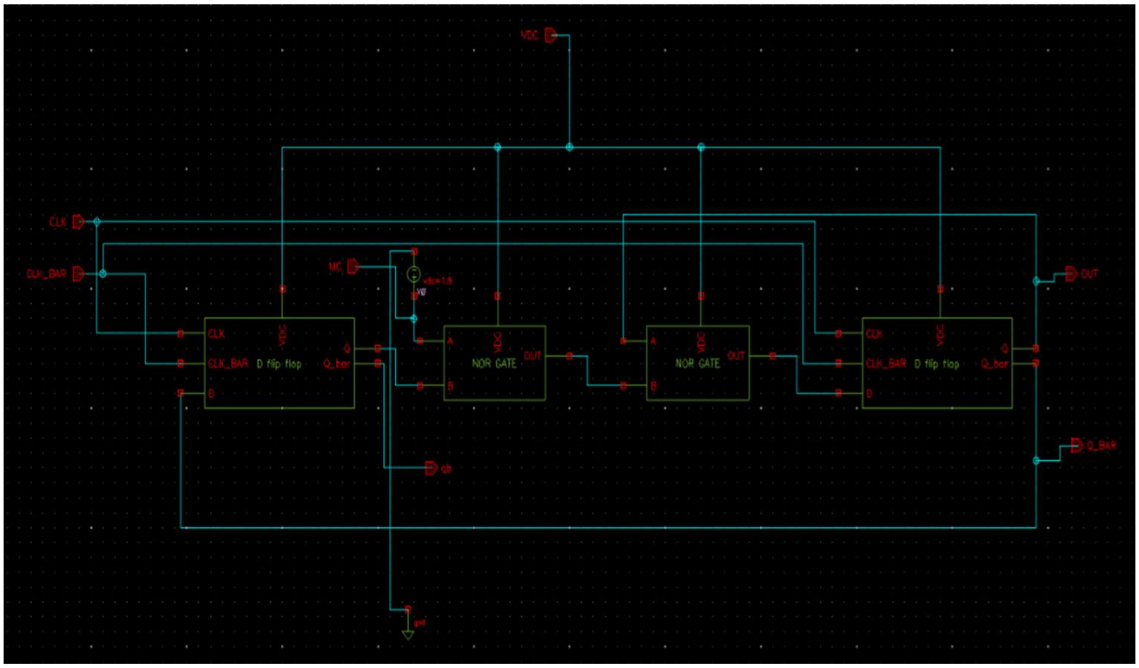
Then we have designed an NOR Gate Using CMOS.



Also we have verified waveform outputs from the truth table:



Final Circuit using CMOS:



This above circuit is at MC= 1

Conclusion (CMOS-Based 2/3 Prescaler on Cadence Virtuoso)

The 2/3 dual-modulus prescaler was successfully designed and implemented at the transistor level using CMOS technology in Cadence Virtuoso. The circuit was built using two D flip-flops and two NOR gates, following the proposed Dynamic Current Mode Logic (DyCML)-inspired architecture.

Simulation Results:

- The input clock (clk) and modulus control signal (MC) were applied during transient simulation.
- When MC = 1, the circuit output (out) exhibited a frequency equal to half of the input clock, confirming divide-by-2 operation.
- When MC = 0, the output frequency reduced to one-third of the input clock, indicating divide-by-3 operation.

This behavior was verified through transient waveform analysis in Cadence Virtuoso, where:

- From 20 ns to 120 ns → Output clearly demonstrated f/2 behavior
- After 120 ns → The output waveform transitioned and stabilized into f/3 mode

Key Takeaways:

- The manual transistor-level design in Cadence provided complete control over sizing, loading, and connectivity of CMOS devices.
- The prescaler functioned accurately across both modes, without glitches during MC switching.
- This validates the robustness of the design and its suitability for low-power, high-frequency digital applications such as:
    - Phase-Locked Loops (PLLs)
    - Clock and Data Recovery (CDR) circuits
    - Frequency synthesizers

Final Statement:

The successful transistor-level implementation and simulation in Cadence Virtuoso demonstrates that the designed CMOS-based 2/3 prescaler is functionally correct, frequency-responsive, and ready to be scaled or integrated into larger high-speed communication systems.