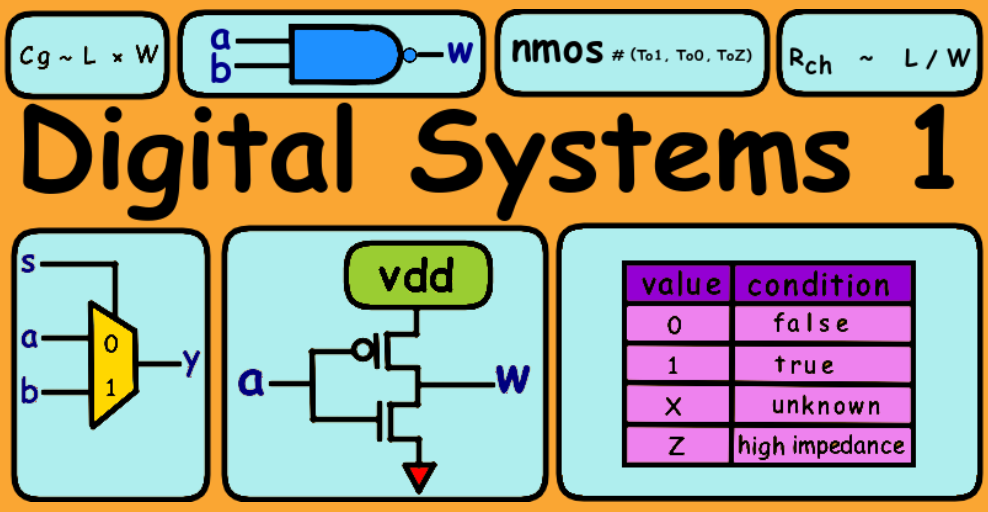
Digital Systems 1 UT

2021-3-15



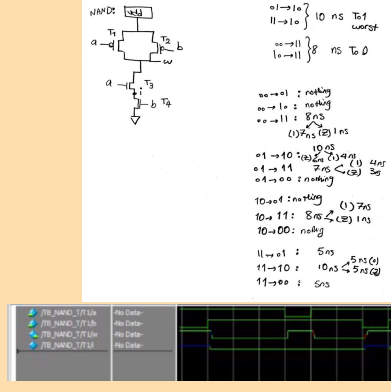
CA 1

Melika Minaei Bidgoli

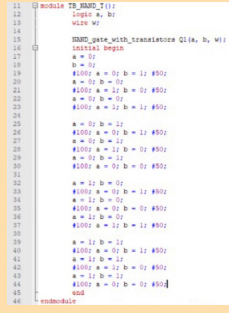
problem 1

Generate a 2-input CMOS NAND gate and verify its timing and functionality. Write the SystemVerilog description of this structure using NMOS and PMOS transistors. Use $\#(3,4,5)$ delay for the NMOS transistors and $\#(5,6,7)$ for the PMOS transistors. Generate a testbench for this circuit in SystemVerilog and examine it for various input changes. Among the various input changes, make sure you test the circuit for the worst-case delay of its output making To1 and To0 transitions. Make sure the time distance between your input changes is much larger than the gate delay values.

solution



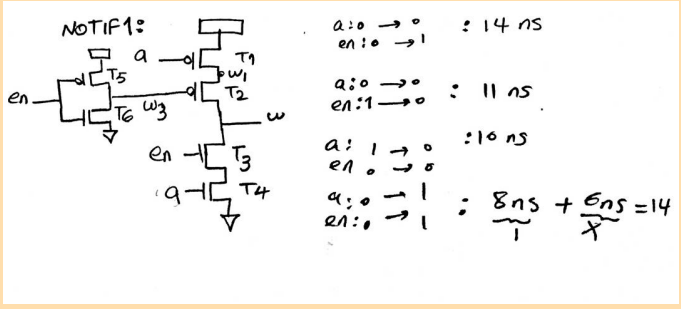
```
1  `timescale 1ns/1ns
2  module NAND_gate_with_transistors(input a, b, output w);
3          supply1 VDD;
4          supply0 GND;
5          wire i;
6          pmos#(5, 6, 7)T1(w, VDD, a);
7          pmos#(5, 6, 7)T2(w, VDD, b);
8          nmos#(3, 4, 5)T3(w, i, a);
9          nmos#(3, 4, 5)T4(i, GND, b);
10 endmodule
```



problem 2

Generate a CMOS Tri-State Buffer (functionality like NOTIF1 of SystemVerilog) using four transistors for the buffer and two for its inverter. Generate a testbench for this circuit in SystemVerilog and examine it for various input changes. You are allowed to use inverts as needed. Among the various input changes, make sure you test the circuit for the worst-case delay of its output making To1, To0, and ToZ transitions. Make sure the time distance between your input changes is much larger than the gate delay values.

solution

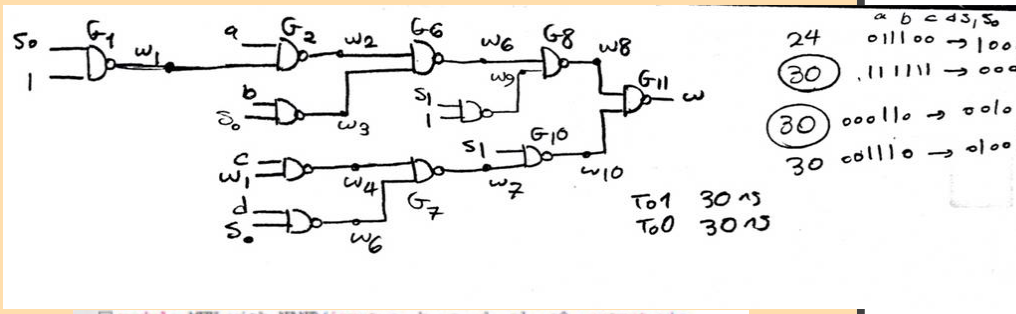



```
module Tristatebuffer_with_transistors(input a, en, output w);
        supply1 VDD;
        supply0 GND;
        wire w1, w2, w3;
        pmos#(5, 6, 7)T1(w1, VDD, a);
        pmos#(5, 6, 7)T2(w, w1, w3);
        nmos#(3, 4, 5)T3(w, w2, en);
        nmos#(3, 4, 5)T4(w2, GND, a);
        nmos#(3, 4, 5)T5(w3, VDD, en);
        pmos#(5, 6, 7)T6(w3, GND, en);
endmodule
```

problem 3

Using NAND gates with as many inputs as needed based on that of Problem 1, generate a 4- to-1 MUX with two select inputs, s1 and s0, and four data inputs a, b, c, and d. Deduce gate delays based on your findings of Problem 1. Generate a testbench for this circuit in SystemVerilog and examine it for various input changes. Among the various input changes, make sure you test the circuit for the worst-case delay of its output making To1 and To0. Make sure the time distance between your input changes is much larger than the gate delay values.

solution



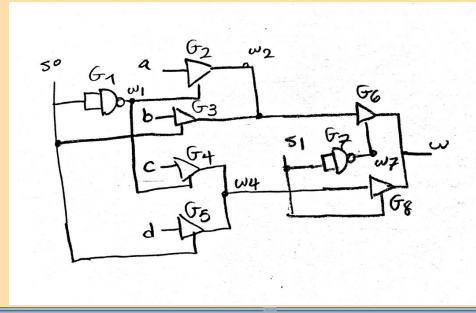

```
module MUX_with_NAND(input a, b, c, d, s1, s0, output w);
        wire w1, w2, w3, w4, w5, w6, w7, w8, w9, w10;
        NAND_gate_with_transistors G1(s0, 1'b1, w1);
        NAND_gate_with_transistors G2(a, w1, w2);
        NAND_gate_with_transistors G3(b, s0, w3);
        NAND_gate_with_transistors G4(c, w1, w4);
        NAND_gate_with_transistors G5(d, s0, w5);
        NAND_gate_with_transistors G6(w2, w3, w6);
        NAND_gate_with_transistors G7(w4, w5, w7);
        NAND_gate_with_transistors G8(w6, w9, w8);
        NAND_gate_with_transistors G9(s1, 1'b1, w9);
        NAND_gate_with_transistors G10(s1, w7, w10);
        NAND_gate_with_transistors G11(w8, w10, w);
endmodule
```

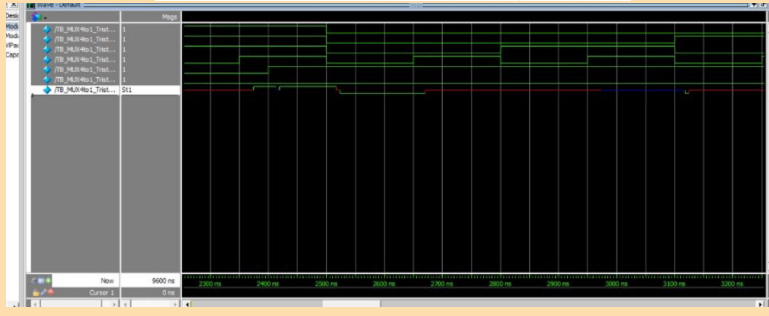
problem 4

Using the Tri-State Buffer of Problem 2, generate a 4-to-1 MUX with two select inputs, s1 and s0, and four data inputs a, b, c, and d. Generate a testbench for this circuit in SystemVerilog and examine it for various input changes.

Among the various input changes, make sure you test the circuit for the worst-case delay of its output making To1 and To0 transitions. Make sure the time distance between your input changes is much larger than the gate delay values.

solution





problem 5

In a testbench, instantiate the MUX circuits of Part 3 and Part 4 and compare the timing of these circuits. Explain the differences between these two circuits as far as the number of transistors and other physical parameters such as power consumption.

solution