Digital Systems 1 CA1 Report

Soroush Mesforush Mashhad(810198472)

Project Generated Files:

A view of all my generated and compiled files can be seen in the following attachment:

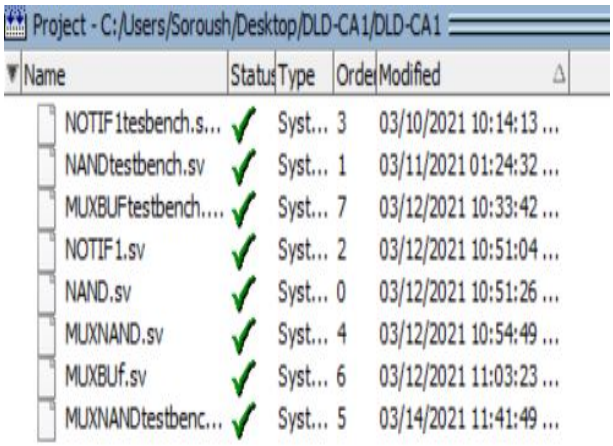

Figure 1: Generated and Compiled .sv files

Question 1:

- **Hand Simulation:**

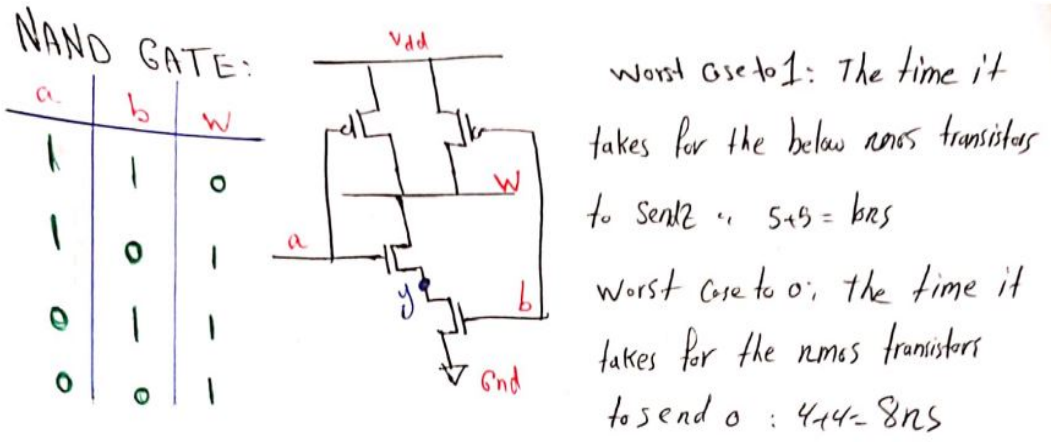

Figure 2: Hand Simulation

- **Verilog Code & Testbench:**

```
1   //In this module I have created a NAND ga
2     `timescale 1ns/1ns
3   module NAND (input i1,i2,output out);
4   //Here I have defined the needed wires ar
5     wire y;
6     supply1 Vdd;
7     supply0 Gnd;
8   //Here we connect the pmos and nmos trans
9     pmos #(5,6,7) T1(out,Vdd,i1);
10    pmos #(5,6,7) T2(out,Vdd,i2);
11    nmos #(3,4,5) T3(y,Gnd,i2);
12    nmos #(3,4,5) T4(out,y,i1);
13  endmodule
14
```

```
1   `timescale 1ns/1ns
2   module NANDtestbench();
3     logic aa=0;
4     logic bb=0;
5     wire ww;
6     NAND UUT(aa,bb,ww);
7     initial begin
8     #30 aa=1;
9     #30 bb=1;
10    #30 aa=0;bb=0;
11    #30 aa=1;bb=1;
12    #30 bb=0;
13    #40 $stop;
14    end
15  endmodule
```

Figure 3: Verilog Code & Testbench

- **Waveform:**

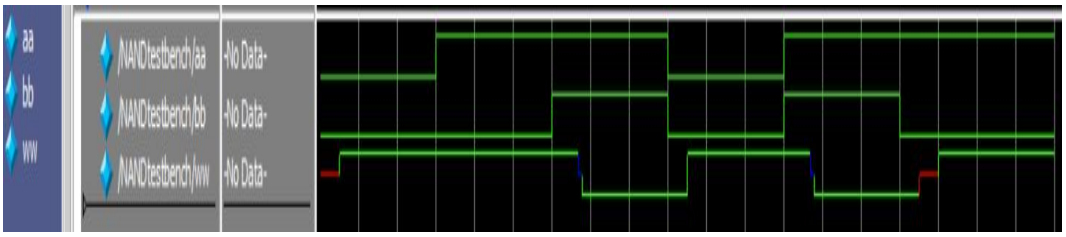

Figure 4: NAND Waveform

Question 2:

- **Hand Simulation:**

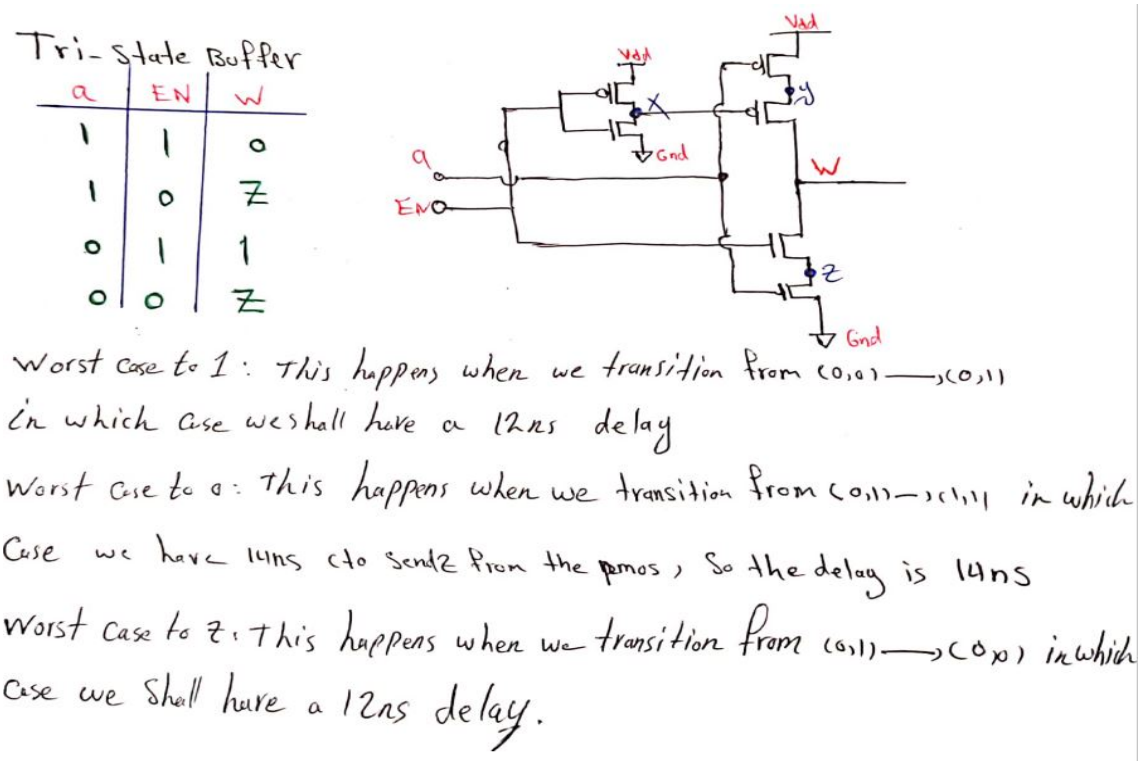

Figure 5: Hand Simulation

- **Verilog Code & Testbench:**

```
//In this module I have created a tri-state buffer
`timescale 1ns/1ns
module NOTIF1(input a,En,output w);
//Here I have defined the needed wires and supplies
  wire x,y,z;
  supply1 Vdd;
  supply0 Gnd;
//Here we connect the pmos and nmos transistors accordingly.
  pmos #(5,6,7) T1(x,Vdd,En);
  nmos #(3,4,5) T2(x,Gnd,En);
  pmos #(5,6,7) T3(y,Vdd,a);
  pmos #(5,6,7) T4(w,y,x);
  nmos #(3,4,5) T5(w,z,En);
  nmos #(3,4,5) T6(z,Gnd,a);
endmodule
```

```
`timescale 1ns/1ns
module NOTIF1testbench();
  logic aa=1;
  logic EN=1;
  wire ww;
  NOTIF1 UUT(aa,EN,ww);
  initial begin
  #30 aa=0;
  #30 EN=0;
  #30 EN=1;
  #30 aa=1;
  #50 $stop;
  end
endmodule
```

Figure 6: Tri-State Buffer Code & Testbench

- **Waveform:**

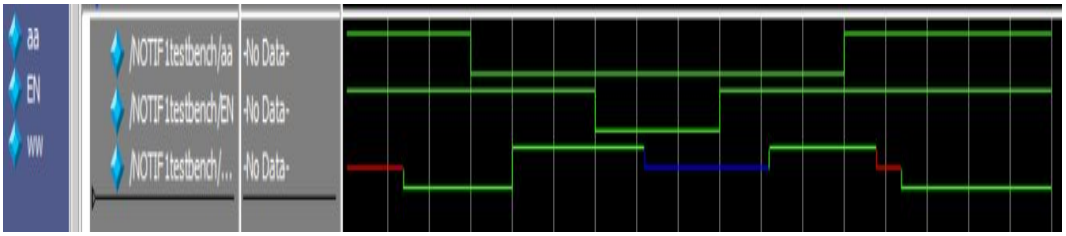

Figure 7: Tri-State Buffer Waveform

Question 3:

- **Hand Simulation:**

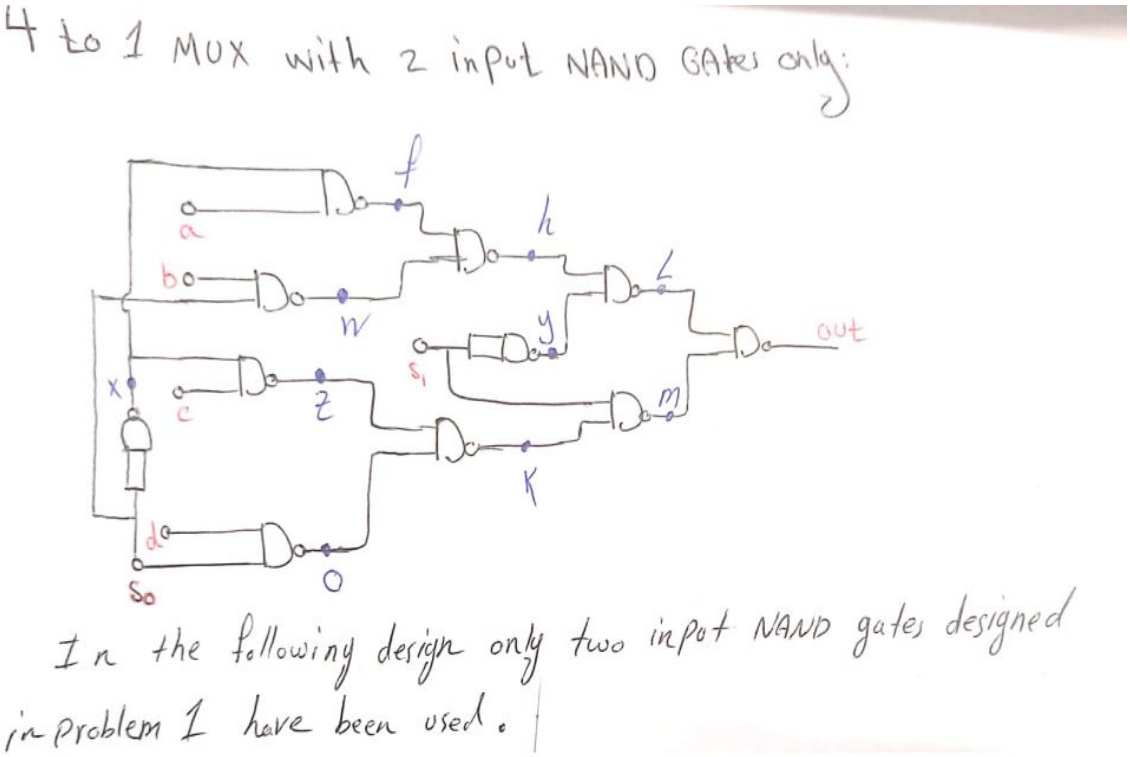

Figure 8: Hand Simulation

First we should note that the worst case delays to 1 and 0 for a NOT gate made with a NAND gate shall be 5ns and 8ns respectively,therefore we can have odd numbers during the simulation,whilst simulating I encountered a 41 ns delay when transitioning to 1 and also 0 and a weird 44ns delay which shall be depicted in the future parts.

- **Verilog Code & Testbench:**

```
//In this Module I've created a 4 to 1
//MUX with 2 input NAND gates only.
  `timescale 1ns/1ns
module MUXNAND(input a,b,c,d,s0,s1, output out);
//Here I have defined the needed wires.
  wire x,y,z,w,f,h,k,l,m,o;
/*Based on the design for a 4 to 1 MUX
with 2 input NAND gates,I've used the
NAND gate from problem 1 (11 gates) and connected
them in the proper manner.*/
  NAND T1 (.i1(s0),.i2(s0),.out(x));
  NAND T2 (.i1(s0),.i2(d),.out(o));
  NAND T3 (.i1(s1),.i2(k),.out(m));
  NAND T4 (.i1(s1),.i2(s1),.out(y));
  NAND T5 (.i1(b),.i2(x),.out(z));
  NAND T6 (.i1(o),.i2(z),.out(k));
  NAND T7 (.i1(s0),.i2(c),.out(w));
  NAND T8 (.i1(x),.i2(a),.out(f));
  NAND T9 (.i1(w),.i2(f),.out(h));
  NAND T10 (.i1(h),.i2(y),.out(l));
  NAND T11 (.i1(l),.i2(m),.out(out));
endmodule
```

```
  `timescale 1ns/1ns
module MUXNANDtestbench();
  logic aa,bb,cc,dd,ss0,ss1;
  wire oout;
  integer i=0;
  MUXNAND UUT(aa,bb,cc,dd,ss0,ss1,oout);
  initial begin
  while(i<64) begin
  #100;
  {aa,bb,cc,dd,ss0,ss1}=64-i;
  #100;
  {aa,bb,cc,dd,ss0,ss1}=i;
  i=i+1;
  end
  #100 $stop;
  end
endmodule
```

Figure 9: 4to1 MUX(With NAND) Code & Testbench

- **Waveform:**

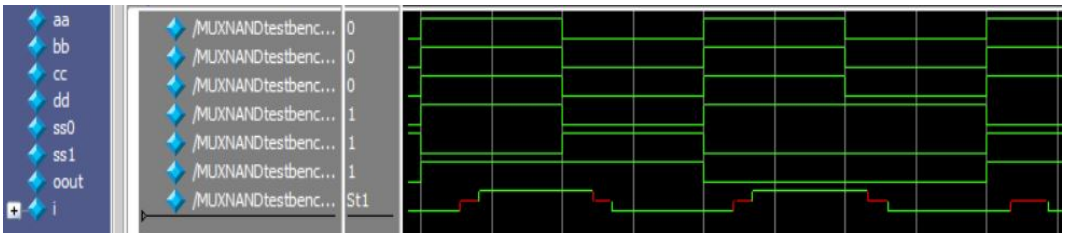

Figure 10: 4to1 MUX(With NAND) Waveform

Question 4:

- **Hand Simulation:**

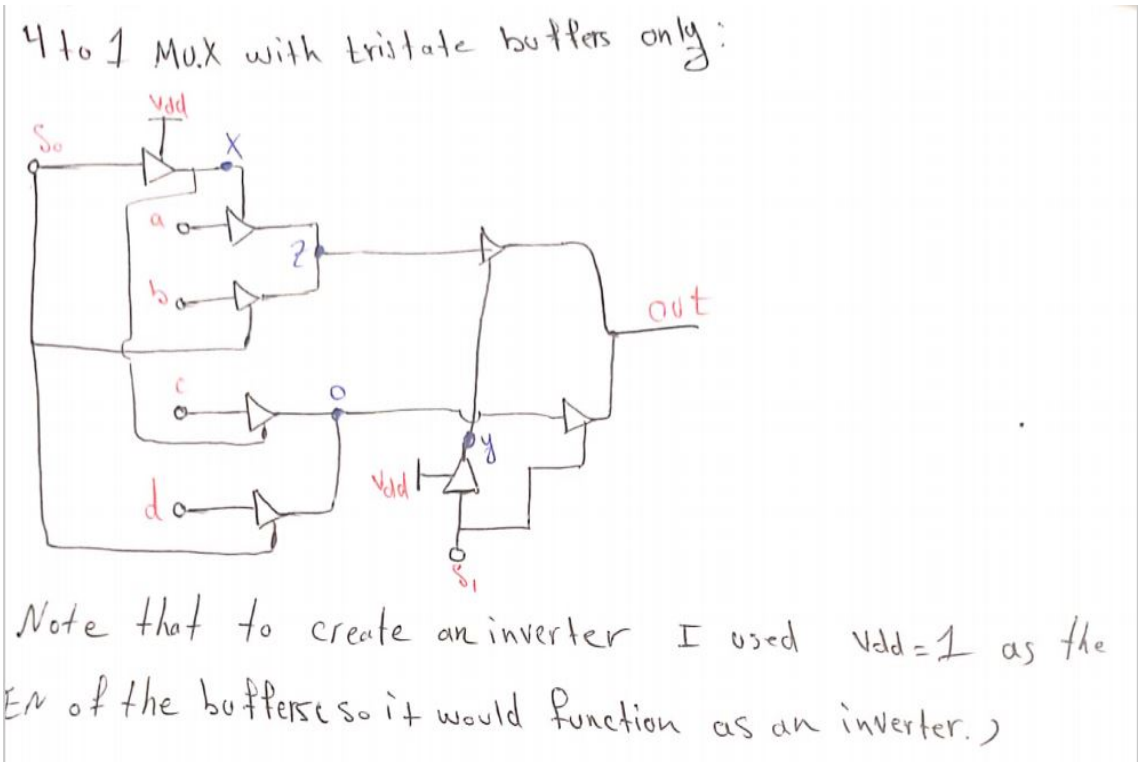

Figure 11: Hand Simulation

In this design I have only used Tri-State buffers(even to implement the not gate),whilst simulating I encountered a 36ns delay for transitioning to 1 and 0 respectively.

- **Verilog Code:**

```
/*In this module I've created a 4 to 1
MUX using a Tristate buffer only.*/
 `timescale 1ns/1ns
module MUXBUf(input a2,b2,c2,d2,s02,s12,output out2);
//Here I've defined the needed wires.
  wire x,y,z,o;
/*The supply is used to connect
to the En of the buffers to create an inverter*/
  supply1 Vdd;
/*Based on the design for a 4 to 1 MUX
with tri-state buffers,I've used the
Buffer from problem 2 (8 gates) and connected
them in the proper manner.*/
  NOTIF1 G1 (.a(s02),.En(Vdd),.w(x));
  NOTIF1 G2 (.a(a2),.En(x),.w(z));
  NOTIF1 G3 (.a(c2),.En(s02),.w(z));
  NOTIF1 G4 (.a(b2),.En(x),.w(o));
  NOTIF1 G5 (.a(d2),.En(s02),.w(o));
  NOTIF1 G6 (.a(s12),.En(Vdd),.w(y));
  NOTIF1 G7 (.a(o),.En(s12),.w(out2));
  NOTIF1 G8 (.a(z),.En(y),.w(out2));
endmodule
```

```
  `timescale 1ns/1ns
module MUXBUFtestbench();
  logic aa,bb,cc,dd,ss0,ss1;
  wire oout;
  integer i=0;
  MUXBUf UUT(aa,bb,cc,dd,ss0,ss1,oout);
  initial begin
  while(i<64) begin
  #100;
  {aa,bb,cc,dd,ss0,ss1}=64-i;
  #100;
  {aa,bb,cc,dd,ss0,ss1}=i;
  i=i+1;
  end
  #100 $stop;
  end
endmodule
```

Figure 12: 4to1 MUX(With Tri-State Buffer) Code and Testbench

- **Waveform:**

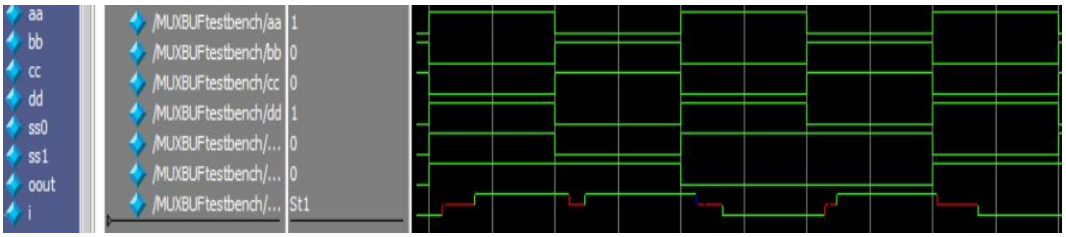

Figure 13: 4to1 MUX(With Tri-State Buffer) Waveform

Question 5:

The delays are less for the MUX made with Tri-state buffers compared to the one made with NAND gates,this is because of the structural layout of transistors in the said designs. The design for problem 3 has 11 NAND gates which consists of 44 PMOS and NMOS transistors and the design for problem 4 has 8 Tri-State buffers which is made up of 48 PMOS and NMOS transistors therefore I think it has a greater power consumption.