Name - Moitrish Maity Roll-48 4th Sem CSE

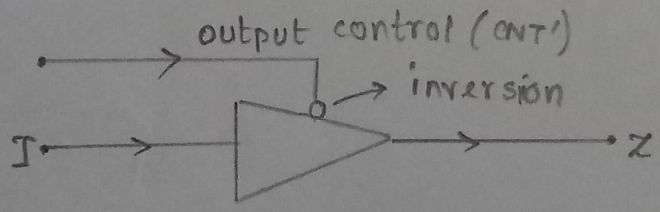
Experiment 2

- Title: Ternary Operator.
- Objective: Introduction to Ternary Operator
- Course Outcome: CO6
- Bloom's Level: Comprehension

(2.1) TRISTATE BUFFER

- Problem statement: Write a verilog program to design, simulate and test a circuit of tristate buffer having following properties: Take 1 bit input as I, Control input (CNT'), output is Z.
- Theory: A tristate buffer can be thought as an input control switch with an output that can be electronically turned 'ON' or 'OFF' by means of an external control signal input.

  Here, we are designing an tristate buffer with active low control signal and non-inverted output port terminals.
- Logic Diagram:



Truth Table:

OUTPUT CONTROL (CNT')	I	Z
L	L	L
L	H	H
H	X	Hi-Z

I : input
Z : output

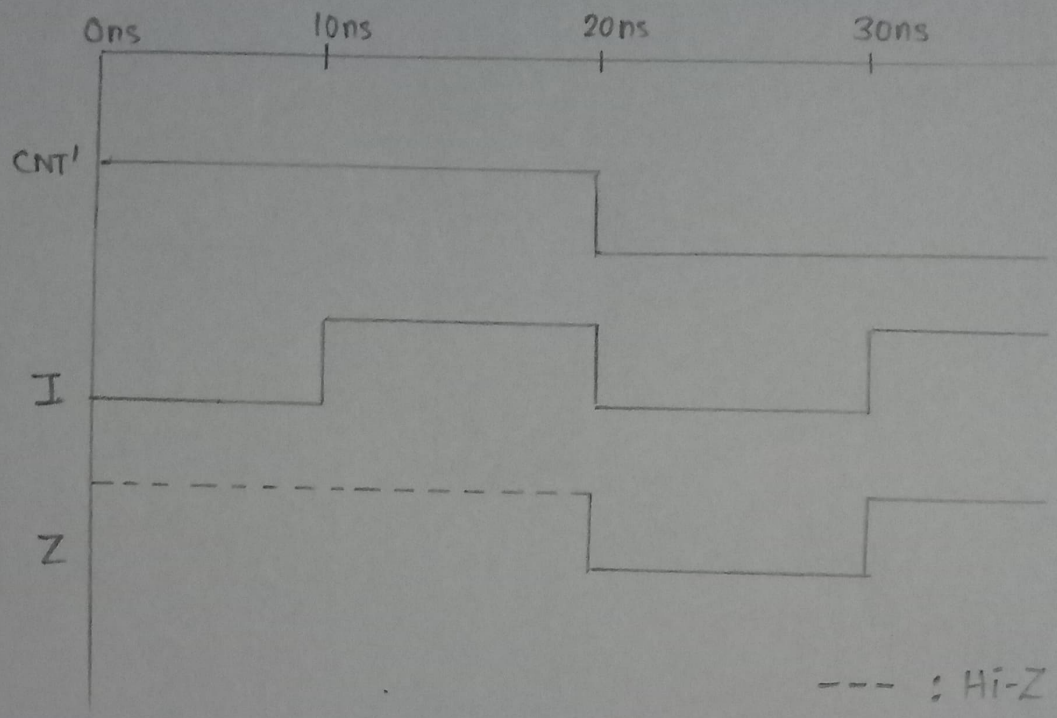
Design Code:

```
module tristate_buffer (i, cnt, z)
    input i, cnt;
    output z;
    assign z = (~cnt)? i : 1'bz;   // active low o/p control
endmodule                          // non-inverted o/p
```

Testbench Code:

```
module tristate_buffer_tb();
    reg i1, cnt1;
    wire z1;
tristate_buffer uut (.i(i1), .cnt(cnt1), .z(z1));
  initial begin
      $dumpfile ("dump.vcd"); $dumpvars;
      i1 = 0; cnt1 = 0; #10;
      i1 = 1; cnt1 = 0; #10;
      i1 = 0; cnt1 = 1; #10;
      i1 = 1; cnt1 = 1; #10;
  end
endmodule
```

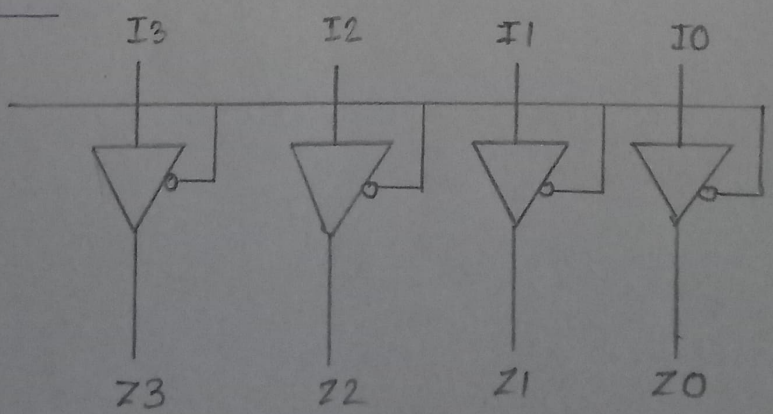
□ Timing Diagram:



(2.2) 4-BIT UNIDIRECTIONAL BUS:

□ Theory: A computer bus is used by CPU to communicate with devices contained within a computer. Here, we are doing an unidirectional bus circuit where data flows only in one direction. It is obtained by joining tristate buffers. When control signals is in active low state data flows through this unless it acts as an short circuited wire. So, no data can flow in reverse direction.

□ Logic Diagram:



Function Table :

OUTPUT CONTROL	I3	I2	I1	I0	Z3	Z2	Z1	Z0
L	L	L	L	L	L	L	L	L
L	L	H	H	L	L	H	H	L
L	H	L	H	L	H	L	H	L
L	L	L	H	H	L	L	H	H
H	X	X	X	X	HiZ	HiZ	HiZ	HiZ
L	H	L	L	L	H	L	L	L
H	X	X	X	X	HiZ	HiZ	HiZ	HiZ

Design Code :

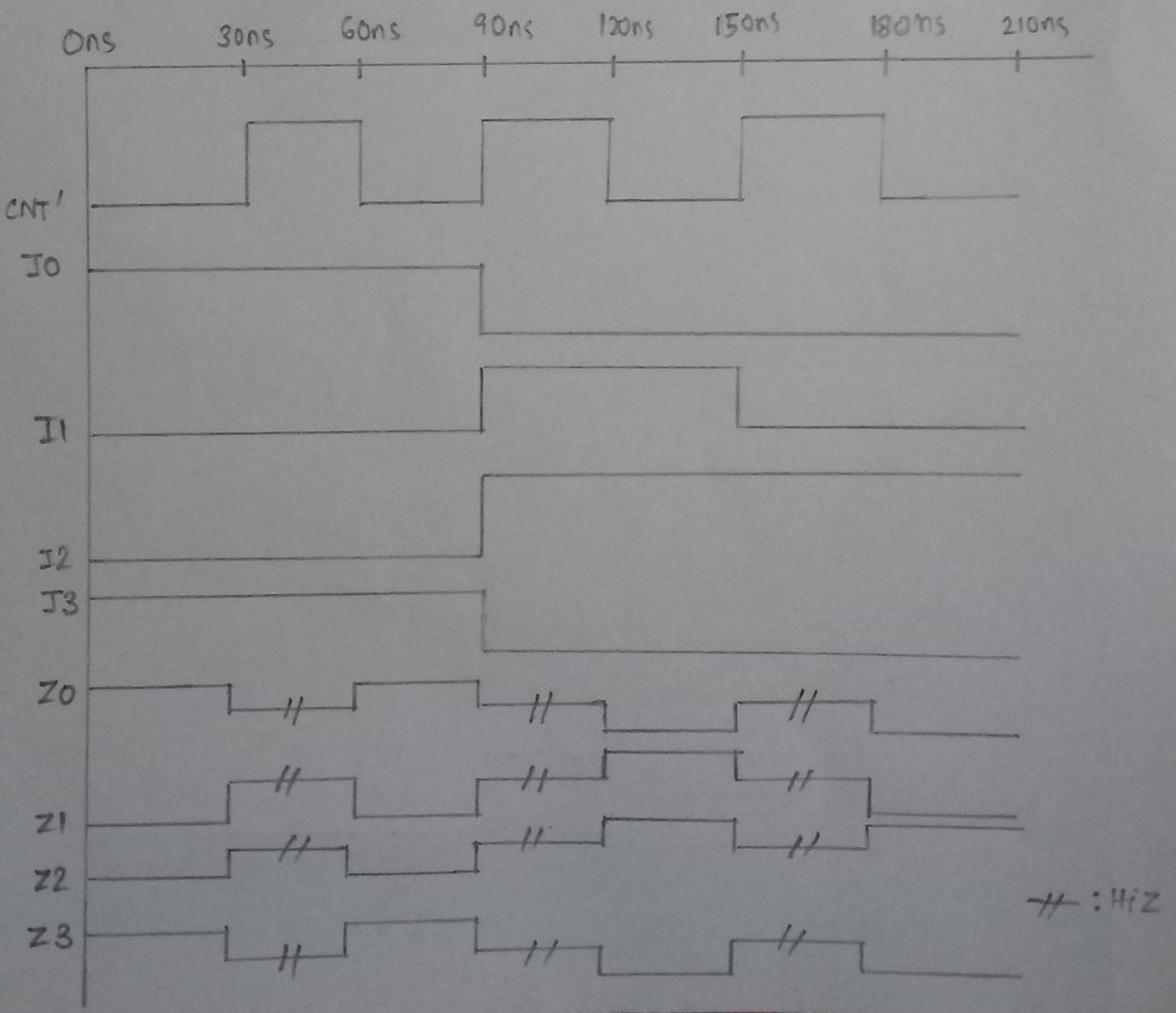
```
module tristate-buffer (i, cnt, z) ;
    input i, cnt ;
    output z ;
    assign z = (~cnt) ? a : 1'bz ;    // active low control i/p
endmodule
module four-bit-bus (input i0, i1, i2, i3, cnt1,
                     output z0, z1, z2, z3);
    tristate-buffer f1 (i0, cnt1, z0);
    tristate-buffer f2 (i1, cnt1, z1);
    tristate-buffer f3 (i2, cnt1, z2);
    tristate-buffer f4 (i3, cnt1, z3);
endmodule
```

Testbench Code :

```
module four-bit-bus-tb();
    reg I0, I1, I2, I3, CNT;
    wire z0, z1, z2, z3 ;
```

```
four_bit_bus uut (.i0(I0), .i1(I1), .i2(I2), .i3(I3),
                  .cnt1(CNT1), .z0(Z0), .z1(Z1), .z2(Z2),
                  .z3(Z3));
initial begin;
$dumpfile ("dump.vcd"); $dumpvars;
I0=1; I1=0; I2=0; I3=1; CNT=0; #30;
I0=1; I1=0; I2=0; I3=1; CNT=1; #30;
I0=1; I1=0; I2=0; I3=1; CNT=0; #30;
I0=0; I1=1; I2=1; I3=0; CNT=1; #30;
I0=0; I1=1; I2=1; I3=0; CNT=0; #30;
I0=0; I1=1; I2=1; I3=0; CNT=1; #30;
I0=0; I1=1; I2=1; I3=0; CNT=0; #30;
  end
endmodule
```

□ Timing Diagram:



□ Conclusion: In this experiment, we have implemented a tristate buffer and a 4 bit unidirectional bus in Verilog code. We also learned the use of ternary operators in Verilog code.

□ Questionairres:

(1.1) Difference between blocking and non-blocking assignments.

Ans> Blocking Assignments

- In blocking assignments, the evaluation of expression on the RHS is updated to the LHS variable autonomously based on the delay value. (Either 0 if delay specified or scheduled as a future event if a mono value is specified)
- Recommended to use within combinational always blocks.
- Represented by '=' operator sign b/w LHS and RHS.

Non-blocking Assignments

Non-blocking assignment to LHS is scheduled to occur when the next evaluation cycle occurs in simulation and not immediately updates are not available immediately within same time unit.

Recommended to use within sequential always blocks.

Represented by '<=' operator sign b/w LHS and RHS.

(1.2) Write a Verilog Code to swap contents of two registers with and without a temp register

Ans> With Temp register (using blocking assignment)

```
always @ (posedge clock) ;
  begin ;  temp=b ;
```

```
    b = a;
    a = temp;
end
```

Without temp register (using non-blocking assign)

```
always @ (posedge clock)
begin
    a <= b;
    b <= a;
end
```

(1.3) Difference between === and == ?

Ans)

===	==
• Used for comparison b/w 2 variables but this will check strict type, means it will check data types and compare the values.	Used for comparison b/w 2 variable irrespective of datatypes of variables.
• It tests 4 states logical equality (tests for 1, 0, z and x)	It tests for logical equality (tests for 1, 0 , all other in x)

□ Justification of CO:

In this experiment, we have implemented a tristate buffer and an unidirectional bus system by using Verilog code. We also learned about ternary operators in Verilog. So, CO6 is justified.