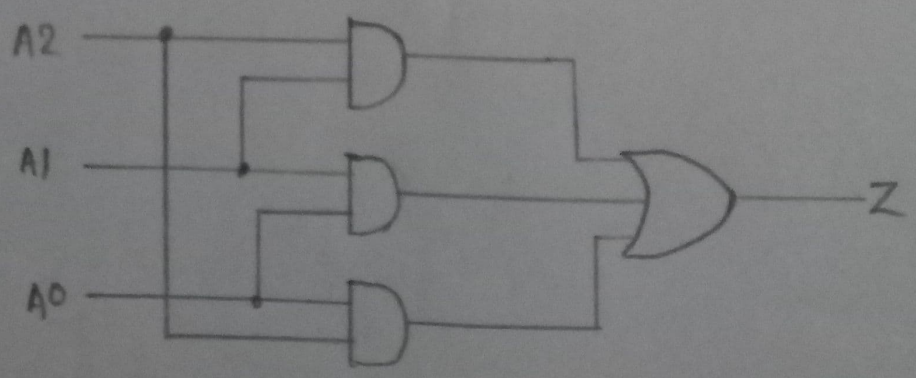
Name - MOITRISH MAITY Roll - 48 Dept - CSE 4th Sem

EXPERIMENT - 3

- TITLE : Minimization techniques.
- Objective : Implementation of Boolean function and logic equations.
- COURSE OUTCOME : CO6
- BLOOM'S LEVEL : COMPREHENSION
- PROBLEM STATEMENT : Write a Verilog program to design, simulate and test a circuit of 3 input majority circuit having following properties : Take 3- bit input as A2A1A0. Output as z.
- THEORY : A majority circuit is a combinational circuit whose output is equal to 1 if input variables have more 1's than 0's and output is equal to 0 if input variable have more 0's. Suppose input is '011', here no. of 1's is greater than no. of 0's. So, corresponding output will be '1' and vice versa. Here, we will implement a majority circuit in Verilog Code.
- LOGIC DIAGRAM :



FUNCTION TABLE :

A2	A1	A0	Z
L	L	L	L
L	L	H	L
L	H	L	L
L	H	H	H
H	L	L	L
H	L	H	H
H	H	H	H

DESIGN CODE:

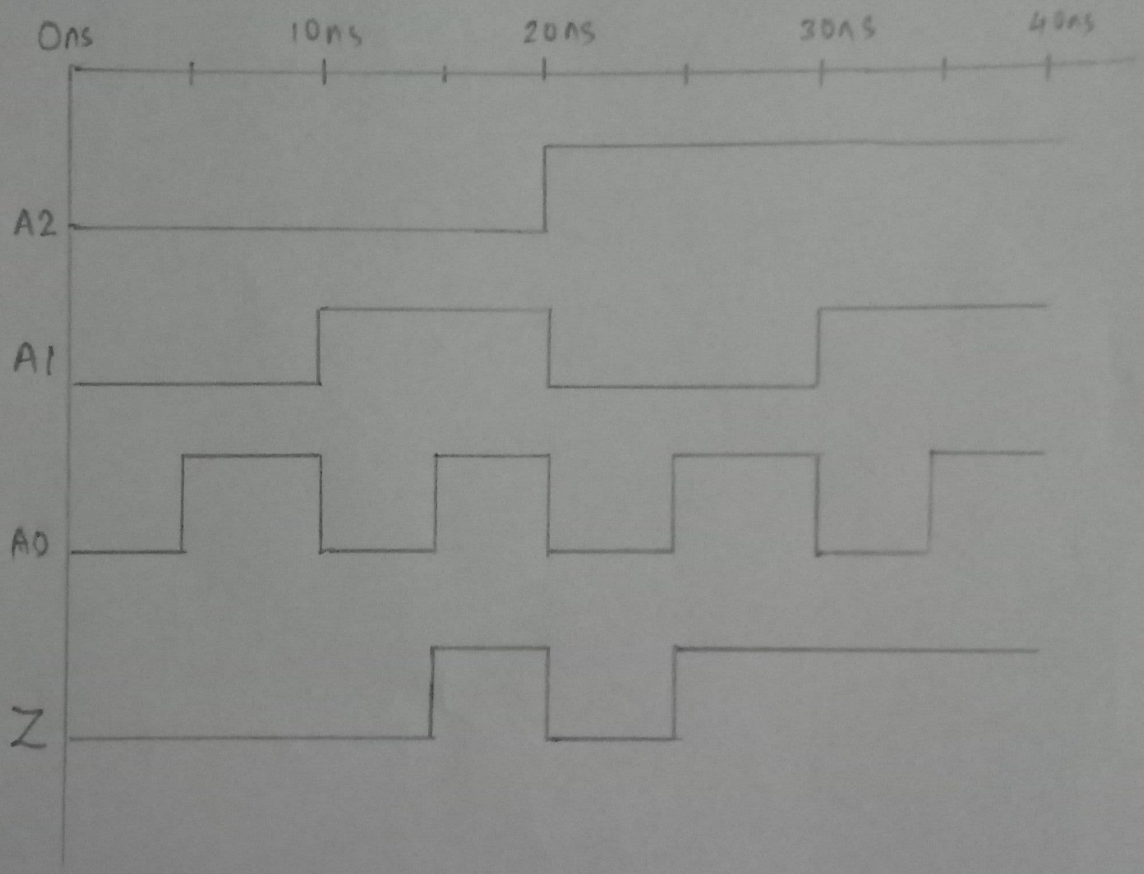
```
module majority (A2, A1, A0, Z)
    input A2, A1, A0;
    output Z;
    assign z = ((A2 & A0) | (A2 & A1) | (A1 & A0));
endmodule
```

TESTBENCH CODE :

```
module majority_tb();
    reg a2, a1, a0;
    wire z1;
  majority uut (.A2(a2), .A1(a1), .A0(a0), .z(z1));
    initial begin;
    $ dumpfile ("dump.vcd");
    $ dumpvars;
    a2=0; a1=0; a0=0; #5;
    a2=0; a1=0; a0=1; #5;
    a2=0; a1=1; a0=0; #5;
    a2=0; a1=1; a0=1; #5;
```

```
a2=1 ; a1=0 ; a0=0 ; #5;
a2=1 ; a1=0 ; a0=1 ; #5;
a2=1 ; a1=1 ; a0=0 ; #5;
a2=1 ; a1=1 ; a0=1 ; #5;
end
endmodule
```

TIMING DIAGRAM:



DISCUSSION:

In this experiment, we have implemented a majority circuit using verilog code. We have learned various operators and operations in verilog.

QUESTIONNAIRES :

(3.1) Explain difference between $ display and $ monitor.

Ans>

$ display

$ display statement is used to display the immediate values of variables or signals. It gets executed in active region.

$ monitor

$ monitor statement displays the value of a variable or signal whenever its value changes. It gets executed in the postponed region. We need to call it only one time and it will print a value of a variable or a signal every time when its value is getting changed.

(3.2) Given the following verilog code, what is the value of 'a' displayed ?

```
always @(clk) begin
    a = 0;
    a <= 1;
    $ display (a);
end
```

Ans> Since, 'a=0' is an active event, it is scheduled into the first Queue. The 'a<=1' is a non-blocking event. So, it is placed in 3rd Queue. Finally the display statement is placed into 4th Queue.

Only events in the active queue are completed in this sim cycle, so the 'a=0' happens and then display shows a=0. If we were to look at value of a in the next sim cycle, it would show '1'.

(33) For the following verilog code segment, if the initial value of IR is ABCD3456 (in hexadecimal) the value of 'data' in decimal will be _____

```
wire [31:0] IR;
wire [3:0] data; → wire [15:0] d1;
wire [31:16] d2;
assign d1 = IR[31:16];
assign d2 = IR[15:0];
assign data = d1[11:8] + d2[19:16] + d2[31:28];
```

Ans) d1 = 1010 1011 1100 1101

d2 = 0011 0100 0101 0110

∴ data = 1011 + 0110 + 0011

= 0100 = 4 (carry out = 1)

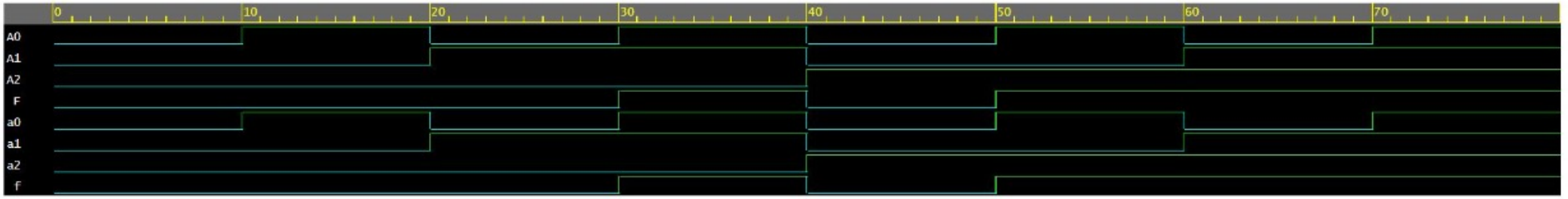
□ CO JUSTIFICATION:

In this experiment we have done a majority circuit by using verilog code. So, CO6 is justified.

EPWave

From: 0s To: 80s

Get Signals Radix 100%



Note: To revert to EPWave opening in a new browser window, set that option on your user page.