Assignment 3
Submitted By :
Snehal Gupta
2016201

Question 1

Verilog codes :

```
//testb.v
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 10.11.2017 13:15:30
// Design Name:
// Module Name: testb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module testb;
    reg inp_clk;
    wire out_clk1;
    wire out_clk2;
    wire out_clk3;
    wire out_clk4;
    wire out_clk5;
    wire out_clk6;
    //divideby3 d1(inp_clk,out_clk1,out_clk4,out_clk5);
    //second d2(inp_clk,out_clk3,out_clk2,out_clk6);
    divideby3 g1(
    .clk(inp_clk),
    .out_clk1(out_clk1),
```

```
    .out_clk4(out_clk4),
    .out_clk5(out_clk5),
    .out_clk3(out_clk3),
    .out_clk2(out_clk2),
    .out_clk6(out_clk6));
   initial
       begin
       inp_clk = 0;
       end
   always #5 inp_clk = ~inp_clk;

endmodule

//divideby3.v
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 10.11.2017 00:59:55
// Design Name:
// Module Name: divideby3
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module divideby3(
    input clk,
    output out_clk1,
    output out_clk2,
    output out_clk3,
    output out_clk4,
    output out_clk5,
```

```
    output out_clk6
    );
    reg [1:0] pos,pos1;
    initial
    begin
    pos <= 2'b00;
    pos1 <= 2'b00;
    end
    always @(posedge clk)
    if(pos == 2'b10)
    pos <= 2'b00;
    else
    pos <= pos+1;
    always @(negedge clk)
    begin
    pos1 = pos1 == 2'b10 ? 2'b00 : pos1 + 1;
    end

    assign out_clk1 = ~pos[1] ;
    assign out_clk2 = ~pos1[0] & ~pos1[1];
    assign out_clk3 = ~pos1[1] ;
    assign out_clk4 = pos[0];
    assign out_clk5 = pos[0] | pos[1];
    assign out_clk6 = pos1[0];

endmodule
```

Simulated waveform :

Explanation :
I have created a mod 3 counter . So, it has three states : 00,01 and 10 . The MSBs of these states are pos[1] and LSBs are pos[0] . Two separate registers have been used for creating two counters at posedge and negedge of the input clock.
Clock outputs :
1 - We observe that the values of this clock output in one time period at positive edges of the input clock is 1,1,0 respectively . So, we output the complement of pos[1] as output clock.
2 - We observe that the values of this clock output in one time period at negative edges of the input clock is 1,0,0 respectively . So, we output the and of not of pos1[1] and not of pos1[0] as output clock.
3 - We observe that the values of this clock output in one time period at negative edges of the input clock is 1,1,0 respectively . So, we output the not of pos1[1] as output clock.
4 - We observe that the values of this clock output in one time period at positive edges of the input clock is 0,1,0 respectively . So,we output the value of pos[0] as output clock.
5 - We observe that the values of this clock output in one time period at positive edges of the input clock is 0,1,1 respectively . So,we output the value of OR of pos[0] and pos[1] as output clock.
6 - We observe that the values of this clock output in one time period at negative edges of the input clock is 0,1,0 respectively . So,we output the value of pos1[0] as output clock.

Question 2 :
Number of transactions in a burst i.e. the burst length is specified by the following signals **which are sent by master** :

1. ARLEN [3:0] : Specifies the read burst length.
2. AWLEN [3:0] : Specifies the write burst length.

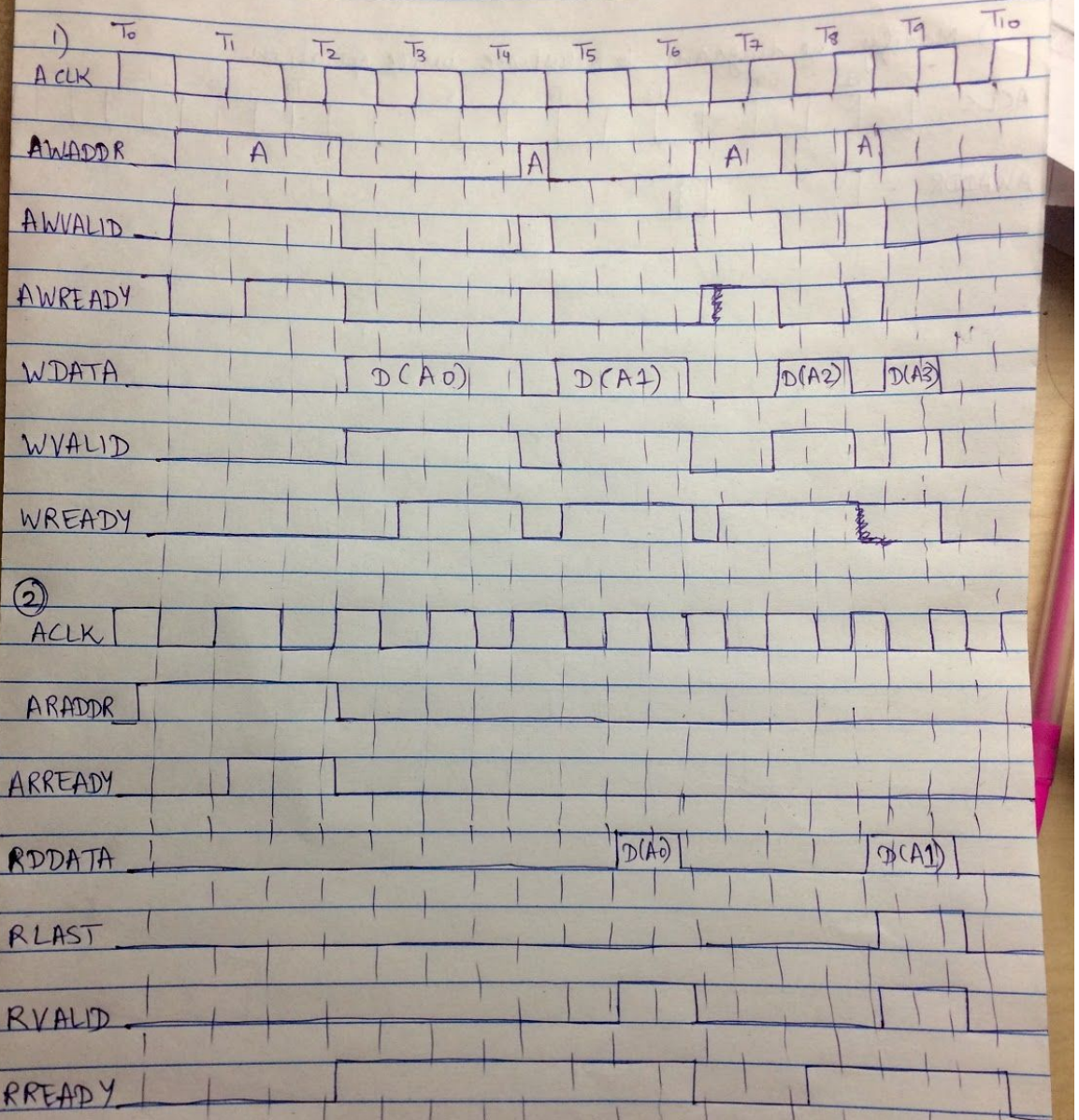
Size of each transaction in a burst i.e. the burst size is specified by the following signals **which are sent by master :**

1. ARSIZE [2:0] : Specifies the read burst size.
2. AWSIZE [2:0] : Specifies the write burst size.

Question 3 :

Q3

1)


T0
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
ACLK
AWADDR
A
A
A
A
AWVALID
AWREADY
WDATA
D(A0)
D(A1)
D(A2)
D(A3)
WVALID
WREADY

2)

ACLK
ARADDR
ARREADY
RDDATA
D(A0)
D(A1)
RLAST
RVALID
RREADY