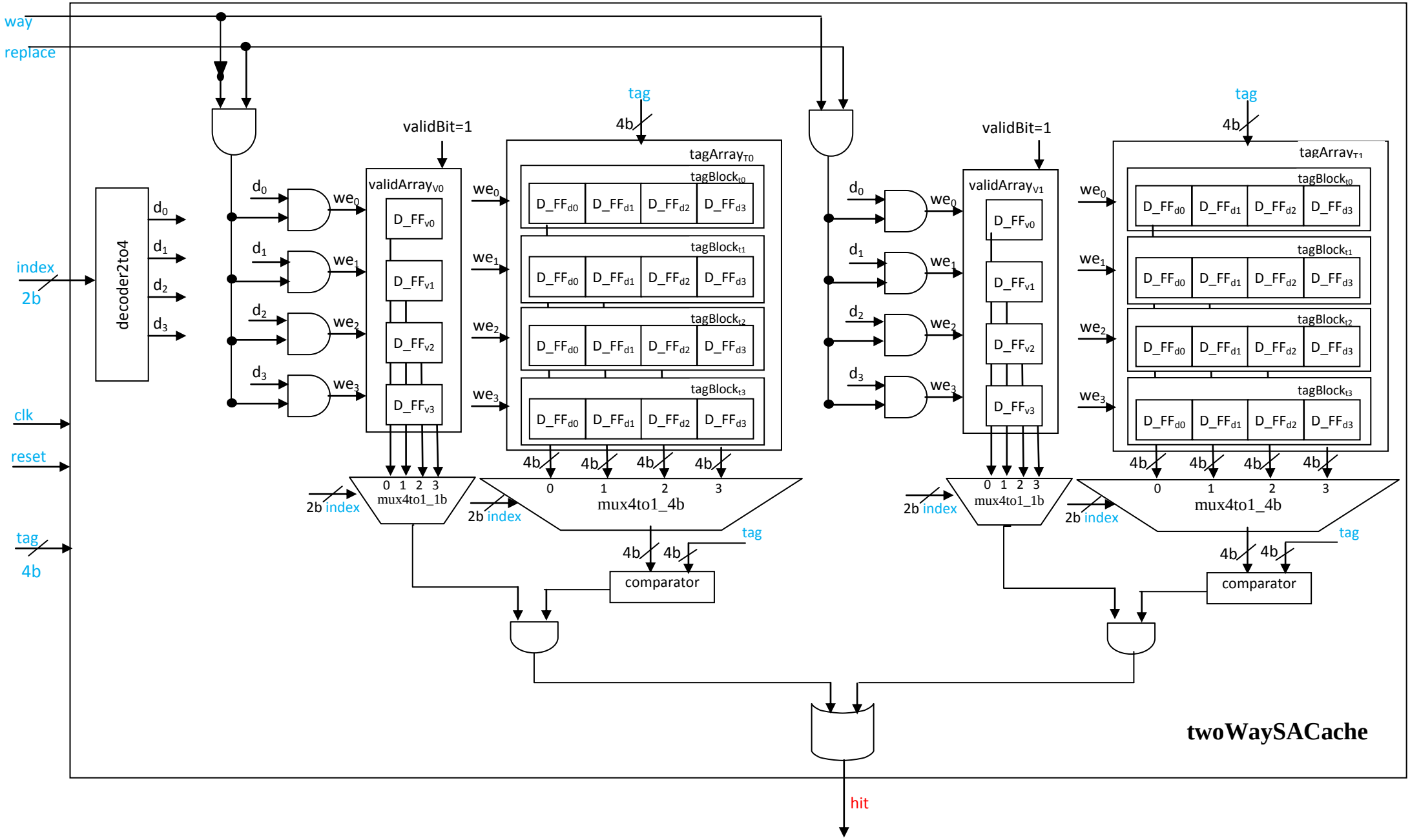
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI K. K. BIRLA GOA CAMPUS
FIRST SEMESTER 2014-2015
CS F342 COMPUTER ARCHITECTURE
ONLINE – 2

INSTRUCTIONS

1. The ONLINE EXAM will begin at 1:45 P.M. Duration of ONLINE EXAM is 120 minutes.
2. Make sure you are occupying the computer allotted to you for the ONLINE EXAM at least 5 minutes before the ONLINE EXAM.
3. The ONLINE EXAM is an OPEN BOOK EXAM. You are allowed to use only the e-BOOKS provided to you.
4. You are NOT allowed to carry TEXT BOOKS, NOTE BOOKS and photo copied materials to Computer Centre for ONLINE EXAM
5. We will provide the softcopies of Digital Design by Morris Mano and Verilog HDL: A Guide to Digital Design and Synthesis by Samir Palnitkar.
6. You are NOT allowed to bring Mobile phone(s), Storage devices and other electronic gadget(s) to Computer Centre during ONLINE EXAMINATION.
7. Make sure ModelSim software is working correctly on the computer allocated to you. If not, report immediately.
8. All modules INCLUDING testBench should be saved as a SINGLE file.
9. File name should be CA_O2_<Your_Id>.v [Eg. CA_O2_2012A7PS670G.v]
10. Use of additional modules other than the given modules is strictly NOT allowed.
11. Change of module prototype is strictly NOT allowed.
12. Upload your file on backup server (We will provide the IP address during the exam) before evaluation.
13. The weightage of Online2 is 13 Marks

Design and implement the following using verilog HDL in **ModelSim software** with the following specifications.



Modules

1. **module D_FF(input clk, input reset, input write, input d, output reg q);**

```
always @(negedge clk)
    if(reset)
        q=0;
    else
        if(write)
            q=d;
```

endmodule

2. **module tagBlock(input clk, input reset, input write, input [3:0] tag,output [3:0] tagData);**
3. **module tagArray(input clk, input reset, input [3:0] we, input [3:0] tag, output [3:0] tagOut0, output [3:0] tagOut1, output [3:0] tagOut2, output [3:0] tagOut3);**
4. **module vaildArray(input clk, input reset, input [3:0] we, input validBit,output validOut0, output validOut1, output validOut2, output validOut3);**
5. **module decoder2to4(input [1:0] in,output reg [3:0] decOut);**
6. **module mux4to1_1b(input in1, input in2, input in3, input in4, input[1:0] sel,output regvalidOut);**
7. **module mux4to1_4b(input [3:0] in1,input [3:0] in2, input [3:0] in3, input [3:0] in4, input[1:0] sel,output reg[3:0] tagOut);**
8. **module comparator(input [3:0] in1, input [3:0] in2,output reg compOut);**
9. **module twoWaySACache(input clk, input reset, input way, input replace, input [3:0] tag, input [1:0] index,output hit);**
10. **module testBench;**

```
regclk, reset, way, replace;
reg [3:0] tag;
reg [1:0] index;
wire hit;
twoWaySACache uut (clk, reset, way, replace, tag, index, hit);

always  #5 clk=~clk;
initialbegin
    way=0; replace=0; index=2'd0; tag=4'd0; reset=1; clk=0;
    #5  way=0; replace=1; index=2'd0; tag=4'd4; reset=0;
    #10 way=1; replace=1; index=2'd1; tag=4'd5;
    #10 way=0; replace=1; index=2'd2; tag=4'd8;
    #10 way=1; replace=1; index=2'd3; tag=4'd9;
    #10 way=1; replace=0; index=2'd3; tag=4'd10;
    #10 way=0; replace=0; index=2'd2; tag=4'd8;
    #10 way=0; replace=0; index=2'd1; tag=4'd6;
```

```
        #10 way=0; replace=0; index=2'd0; tag=4'd4;
        #10 way=1; replace=1; index=2'd1; tag=4'd6;
        #10 way=1; replace=0; index=2'd1; tag=4'd6;
        #10 $finish;
    end
```

endmodule

```
/* ----------------------------------------------------------------------------------------------------------------------- */
```