Μοσχογιάννης Πασχάλης ΑΜ 2114026

Άσκηση 1

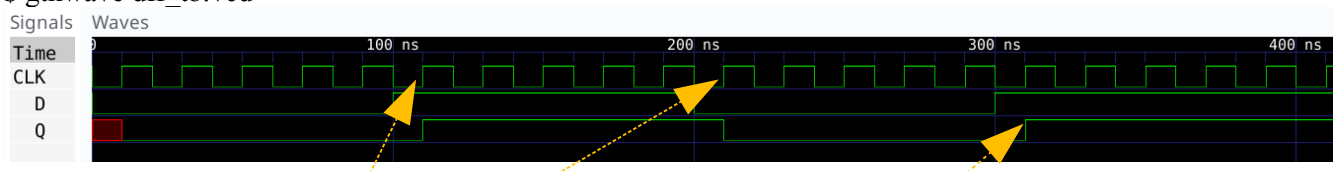
```
//testbench for DFF
`timescale 1ns / 1ps
module dff_tb;
    reg CLK;
    reg D;
    wire Q;

    dff DUT(
        .CLK(CLK),
        .D(D),
        .Q(Q));

    initial begin
        $dumpfile("dff_tb.vcd");
        $dumpvars;

        D=0;
        CLK=0;
        forever #10 CLK = ~CLK;
    end

    initial begin
            D <= 0;
        #100; D <= 1;
        #100; D <= 0;
        #100; D <= 1;
    end

endmodule
```

```
`timescale 1ns / 1ps
// Verilog code for D Flip FLop
module dff (CLK, D, Q);
    input CLK;
    input D;
    output reg Q;

always @ (posedge CLK)
    begin
        Q <= D;
    end

endmodule
```

iverilog - Icarus Verilog compiler Version 10.2 (stable)
vvp - Icarus Verilog vvp runtime engine Version 10.2 (stable)

```
$ iverilog -Wall -o dff_tb.vvp dff_tb.v dff.v
$ vvp dff_tb.vvp
$ ls
dff_tb.v
dff_tb.vcd
dff_tb.vvp
dff.v
```

$ gtkwave dff_tb.vcd



1. At rising edge of clock marked 1 - the Data D Input is 1, hence Q becomes 1 right after that.

2. At rising edge of clock marked 2, the Data is 0, hence Q becomes 0 right after the clock rising edge.

3. At 3, Data is 1 at the rising egdge of clock, hence Q becomes 1

Another testbench

Terminal Default

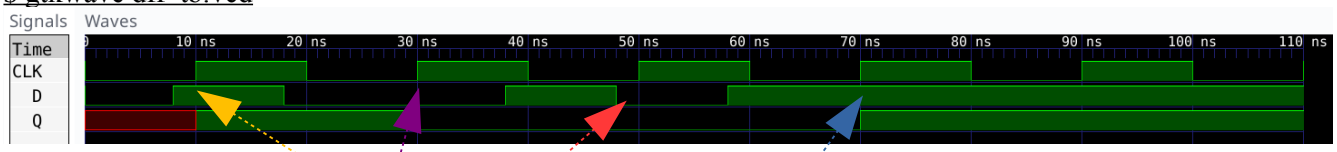
1: universitylaptop@universitylaptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab2/homework1

```
1 //testbench for DFF
2 `timescale 1ns / 1ps
3 module dff_tb;
4     reg CLK;
5     reg D;
6     wire Q;
7     dff DUT(
8         .CLK(CLK),
9         .D(D),
10        .Q(Q));
11    integer i;
12    initial begin
13        $dumpfile("dff_tb.vcd");
14        $dumpvars;
15        D=0;
16        CLK=0;
17        // forever #10 CLK = ~CLK;
18        //         OR
19        for (i=0; i <=10; i=i+1)
20        #10 CLK = ~CLK;
21    end
22    initial begin
23              D <= 0;
24         #8; D <= 1;
25        #10; D <= 0;
26        #10; D <= 0;
27        #10; D <= 1;
28        #10; D <= 0;
29        #10; D <= 1;
30        #10;
31    end
32    initial begin
33        $monitor("Clock=%d,D=%d,Q=%d \n",CLK,D,Q);
34    end
35 endmodule
36
```

NORMAL dff_tb.v unix | utf-8 | verilog 2% 1:1

"dff_tb.v" 37L, 652C written

$ gtkwave dff_tb.vcd



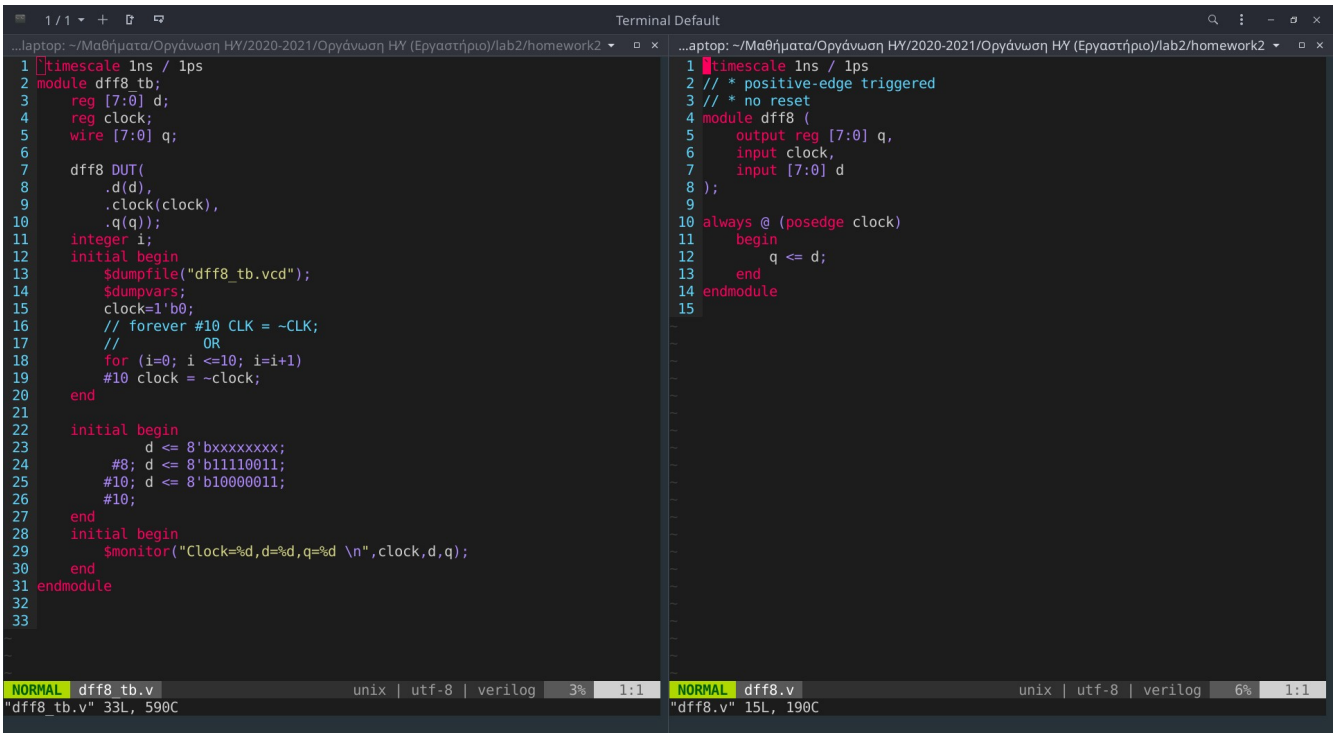
1. At rising edge of clock marked 1 - the Data D Input is 1, hence Q becomes 1 right after that. It is undefined x before that.

2. At rising edge of clock marked 2, the Data is 0, hence Q becomes 0 right after the clock rising edge.

3. At 3, the Data is again 0, hence Q stays 0.

4. At 4, Data is 1 at the rising egdge of clock, hence Q becomes 1

Άσκηση 2



$ iverilog -Wall -o dff8_tb.vvp dff8_tb.v dff8.v

$ vvp dff8_tb.vvp

VCD info: dumpfile dff8_tb.vcd opened for output.

Clock=0,d= x,q= x

Clock=0,d=243,q= x

Clock=1,d=243,q=243

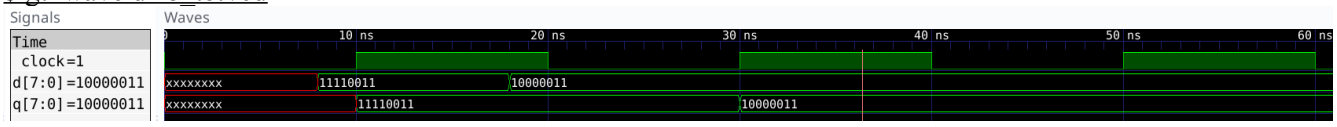
Clock=1,d=131,q=243

Clock=0,d=131,q=243

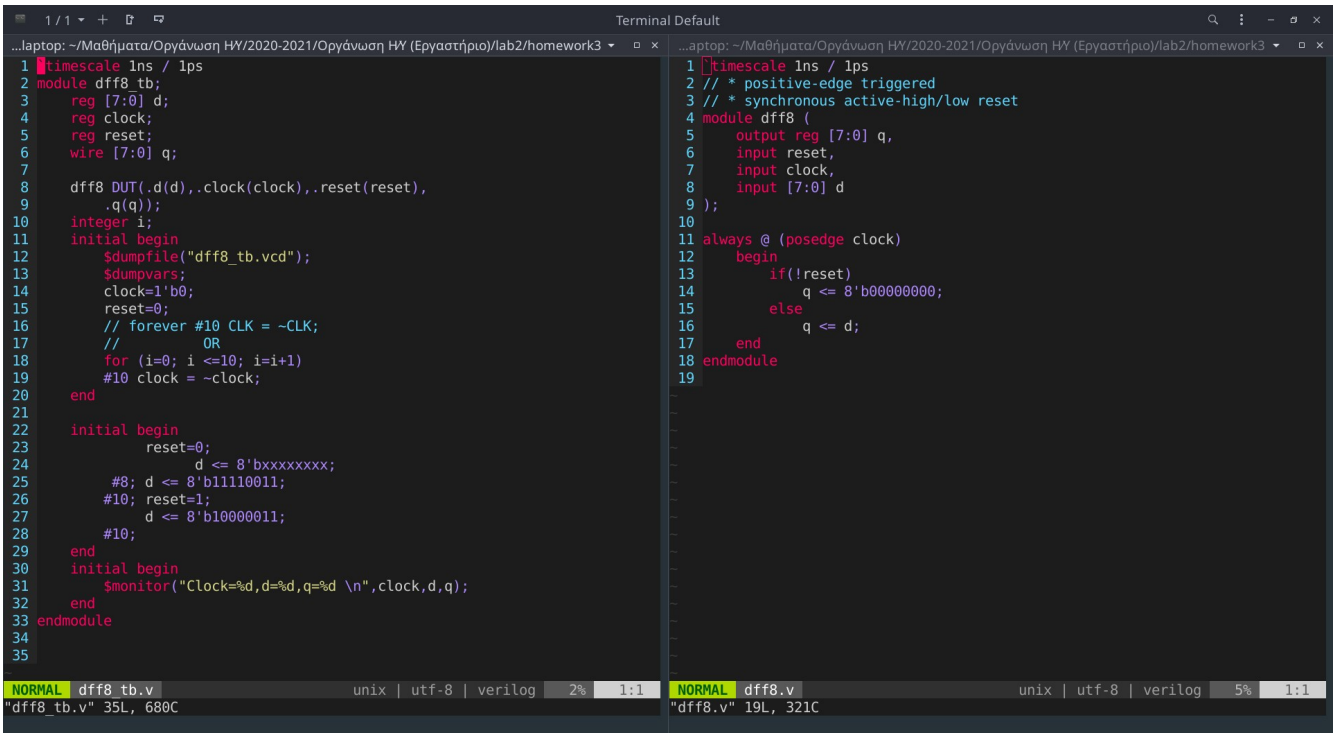
Clock=1,d=131,q=131

...

$ gtkwave dff8_tb.vcd

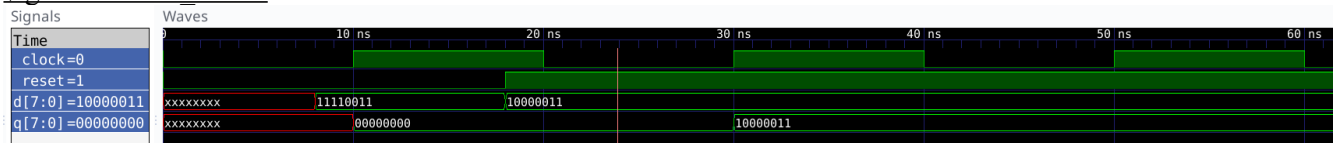


Άσκηση 3



```
$ iverilog -Wall -o dff8_tb.vvp dff8_tb.v dff8.v
$ vvp dff8_tb.vvp
VCD info: dumpfile dff8_tb.vcd opened for output.
Clock=0,d=  x,q=  x
Clock=0,d=243,q=  x
Clock=1,d=243,q=  0
Clock=1,d=131,q=  0
Clock=0,d=131,q=  0
Clock=1,d=131,q=131
Clock=0,d=131,q=131
Clock=1,d=131,q=131
...
```

$ gtkwave dff8_tb.vcd



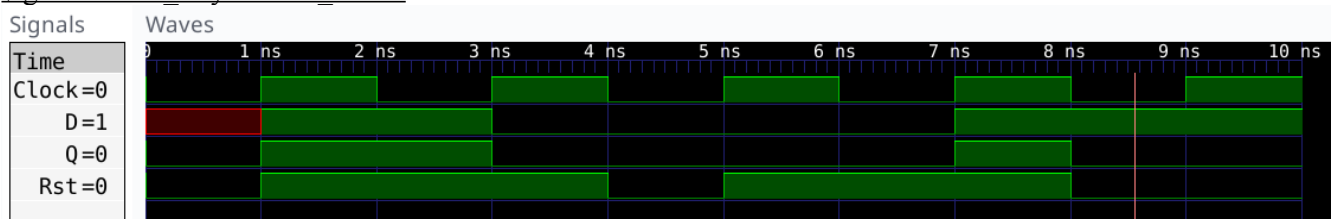
Άσκηση 4

```
1 `timescale 1ns / 1ps
2 module dff_AsyncReset_tb;
3     reg Clock, Reset, D;
4     wire Q;
5     dff_AsyncReset DUT(.D(D),.Clock(Clock),.Reset(Reset),
6         .Q(Q));
7     initial begin
8         $dumpfile("dff_AsyncReset_tb.vcd");
9         $dumpvars;
10                Clock=0;
11                Reset = 0;
12                D = 1'bx;
13        #1;     Clock=1;
14                D = 1;
15                Reset = 1;
16        #1;     Clock=0;
17        #1;     Clock=1;
18                D = 0;
19        #1;     Clock=0;
20                Reset=0;
21        #1;     Clock=1;
22                Reset = 1;
23        #1;     Clock=0;
24        #1;     Clock=1;
25                D = 1;
26        #1;     Clock=0;
27                Reset=0;
28        #1;     Clock=1;
29        #1;
30    end
31    initial begin
32        $monitor("Clock=%d,D=%d,Q=%d \n",Clock,D,Q);
33    end
34 endmodule
```

```
1 `timescale 1ns / 1ps
2 // D Flip Flop with Asyncronous Reset
3
4 module dff_AsyncReset
5    (
6     input wire Clock,
7     input wire D,
8     input Reset,
9     output reg Q
10   );
11
12   always @(negedge Reset or  posedge Clock  )
13      begin
14        if (!Reset)
15            Q = 0;  // Q <= 1'b0;
16                    // Qn <= 1'b1; στο σχήμα της
17                    // άσκησης δεν έχει ανάστροφη έξοδο
18        else
19            Q = D;
20                    // Qn <= -D; στο σχήμα της
21                    // άσκησης δεν έχει ανάστροφη έξοδο
22      end
23 endmodule
```

```
$ iverilog -Wall -o dff_AsyncReset_tb.vvp dff_AsyncReset_tb.v dff_AsyncReset.v
$ vvp dff_AsyncReset_tb.vvp
VCD info: dumpfile dff_AsyncReset_tb.vcd opened for output.
Clock=0,D=x,Q=0
Clock=1,D=1,Q=1
Clock=0,D=1,Q=1
Clock=1,D=0,Q=0
Clock=0,D=0,Q=0
Clock=1,D=0,Q=0
Clock=0,D=0,Q=0
Clock=1,D=1,Q=1
Clock=0,D=1,Q=0
Clock=1,D=1,Q=0
$ ls
dff_AsyncReset_tb.v
dff_AsyncReset_tb.vcd
dff_AsyncReset_tb.vvp
dff_AsyncReset.v
```

$ gtkwave dff_AsyncReset_tb.vcd

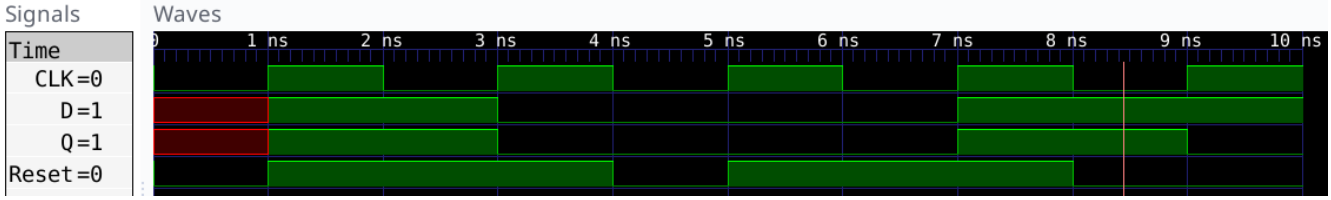


Άσκηση 5

```
`timescale 1ns / 1ps
module dff_SyncReset_tb;
    reg Clock, Reset, D;
    wire Q;

    dff_SyncReset DUT(.Clock(Clock),.D(D),.Reset(Reset),
        .Q(Q));
    initial begin
        $dumpfile("dff_SyncReset_tb.vcd");
        $dumpvars;
                Clock=0;
                Reset = 0;
                D = 1'bx;
        #1;     Clock=1;
                D = 1;
                Reset = 1;
        #1;     Clock=0;
        #1;     Clock=1;
                D = 0;
        #1;     Clock=0;
                Reset=0;
        #1;     Clock=1;
                Reset = 1;
        #1;     Clock=0;
        #1;     Clock=1;
                D = 1;
        #1;     Clock=0;
                Reset=0;
        #1;     Clock=1;
        #1;
    end
    initial begin
        $monitor("Clock=%d,D=%d,Q=%d \n",Clock,D,Q);
    end
endmodule
```

```
`timescale 1ns / 1ps
// D Flip FLop with Synchronous Reset

module dff_SyncReset (Clock, D, Q, Reset);
    input Clock;
    input D;
    input Reset;
    output reg Q;

always @ (posedge Clock)
    begin
        // Q <= (!Reset)? 0 : D;
        if(!Reset)
            Q <= 0; // Q <= 1'b0;
                    // Qn <= 1'b1; στο σχήμα της
                    // άσκησης δεν έχει ανάστροφη έξοδο
        else
            Q <= D;
                    // Qn <= -D; στο σχήμα της
                    // άσκησης δεν έχει ανάστροφη έξοδο
    end

endmodule
```

```
$ iverilog -Wall -o dff_SyncReset_tb.vvp dff_SyncReset_tb.v dff_SyncReset.v
$ vvp dff_SyncReset_tb.vvp
VCD info: dumpfile dff_SyncReset_tb.vcd opened for output.
Clock=0,D=x,Q=x
Clock=1,D=1,Q=1
Clock=0,D=1,Q=1
Clock=1,D=0,Q=0
Clock=0,D=0,Q=0
Clock=1,D=0,Q=0
Clock=0,D=0,Q=0
Clock=1,D=1,Q=1
Clock=0,D=1,Q=1
Clock=1,D=1,Q=0
$ ls
dff_SyncReset_tb.v
dff_SyncReset_tb.vcd
dff_SyncReset_tb.vvp
dff_SyncReset.v

$ gtkwave dff_SyncReset_tb.vcd
```



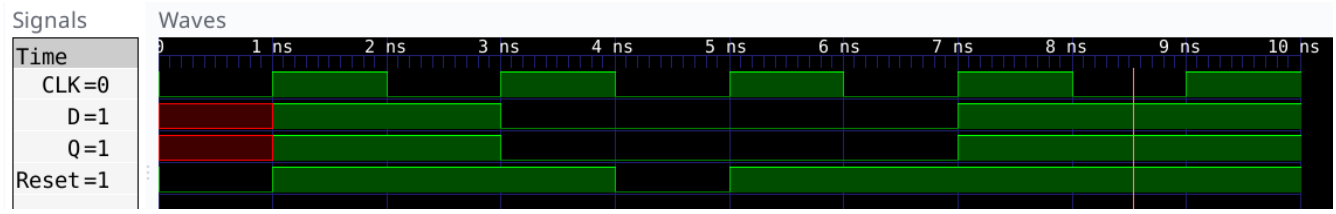
Another testbench

Terminal Default

1: universitylaptop@universitylaptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab2/homework5

```
1  timescale 1ns / 1ps
2  module dff_SyncReset_tb;
3      reg Clock, Reset, D;
4      wire Q;
5
6      dff_SyncReset DUT(.Clock(Clock),.D(D),.Reset(Reset),
7          .Q(Q));
8      initial begin
9          $dumpfile("dff_SyncReset_tb.vcd");
10         $dumpvars;
11                 Clock=0;
12                 Reset = 0;
13                 D = 1'bx;
14         #1;     Clock=1;
15                 D = 1;
16                 Reset = 1;
17         #1;     Clock=0;
18         #1;     Clock=1;
19                 D = 0;
20         #1;     Clock=0;
21                 Reset=0;
22         #1;     Clock=1;
23                 Reset = 1;
24         #1;     Clock=0;
25         #1;     Clock=1;
26                 D = 1;
27         #1;     Clock=0;
28         #1;     Clock=1;
29         #1;
30     end
31     initial begin
32         $monitor("Clock=%d,D=%d,Q=%d \n",Clock,D,Q);
33     end
34 endmodule
35
36
```

NORMAL dff_SyncReset_tb.v unix | utf-8 | verilog 2% 1:1



Άσκηση 6

Terminal Default

...laptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework6

```
1  `timescale 1ns / 1ps
2  module dff16_SyncReset_tb;
3      reg [15:0] d;
4      reg clock;
5      reg reset;
6      reg CE;
7      wire [15:0] q;
8
9      dff16_SyncReset DUT(.d(d),.clock(clock),.reset(reset),.CE(CE),
10         .q(q));
11     integer i;
12     initial begin
13         $dumpfile("dff16_SyncReset_tb.vcd");
14         $dumpvars;
15         clock=1'b0;
16         reset=0;
17         // forever #10 CLK = ~CLK;
18         //          OR
19         for (i=0; i <=10; i=i+1)
20         #10 clock = ~clock;
21     end
22
23     initial begin
24             reset=0;
25             CE=1'b0;
26             d <= 16'bxxxxxxxxxxxxxxxx;
27          #8; d <= 16'b1110111110010011;
28         #10; reset=1;
29             CE=1'b1;
30             d <= 16'b1000001000000001;
31         #10;
32     end
33     initial begin
34         $monitor("Clock=%d,d=%d,q=%d \n",clock,d,q);
35     end
36 endmodule
```

NORMAL dff16_SyncReset_tb.v unix | utf-8 | verilog 2% 1:1

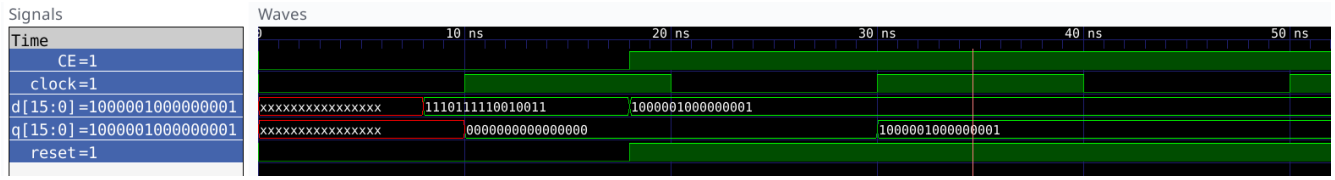
"dff16_SyncReset_tb.v" [noeol] 36L, 797C

...aptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework6

```
1  `timescale 1ns / 1ps
2
3  module dff16_SyncReset (clock, CE, d, q, reset);
4      input clock;
5      input CE;
6      input [15:0] d;
7      input reset;
8      output reg [15:0] q;
9
10
11 always @ (posedge clock)
12     begin
13
14         if(!reset)
15             q <= 16'b0000000000000000;
16         else
17             if ( CE == 1)
18                 q <= d;
19     end
20
21 endmodule
```

NORMAL dff16_SyncReset.v unix | utf-8 | verilog 4% 1:1

"dff16_SyncReset.v" 21L, 285C



Άσκηση 7

Terminal Default

...laptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework7

```
1  //`timescale 1ns / 1ps
2  `include "toplevel_hw_7.v"
3
4  module homework7_tb;
5      reg in;
6      reg Clock;
7      wire out;
8
9      toplevel_hw_7 DUT(.in(in),.Clock(Clock),
10         .out(out));
11     integer i;
12     initial begin
13         $dumpfile("homework7_tb.vcd");
14         $dumpvars;
15         in=1;
16         Clock=0;
17         // forever #10 CLK = ~CLK;
18         //          OR
19         for (i=0; i <=10; i=i+1)
20         #10 Clock = ~Clock;
21     end
22     initial begin
23             in <= 0;
24          #8; in <= 0;
25         #10; in <= 0;
26         #10; in <= 1;
27         #10; in <= 1;
28         #10; in <= 1;
29         #10; in <= 1;
30         #10;
31     end
32     initial begin
33         $monitor("Clock=%d,D=%d,out=%d \n",Clock,in,out);
34     end
35 endmodule
```

NORMAL homework7_tb.v unix | utf-8 | verilog 2% 1:1

"homework7_tb.v" 35L, 702C

...aptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework7

```
1  `include "submodule_dff.v"
2  `include "submodule_xor.v"
3  module toplevel_hw_7(
4      output wire out,
5      input wire in,Clock
6  );
7  wire XOR;
8  submodule_xor U0 (XOR,in,out); // XOR is data in to D Flip Flop
9  submodule_dff U1 (out,XOR,Clock);
10 endmodule
```

NORMAL toplevel_hw_7.v unix | utf-8 | verilog 10% 1:1

"toplevel_hw_7.v" 10L, 242C written

...aptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework7

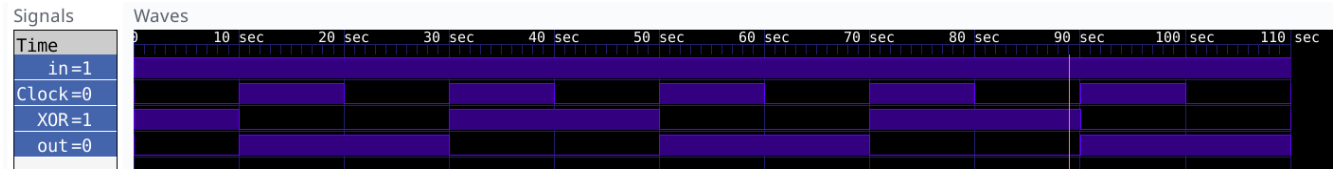
```
1  module submodule_dff (Q, D, Clock);
2      output reg Q;
3      input Clock;
4      input D;
5      initial Q<=0;
6  always @ (posedge Clock)
7      begin
8          Q <= D;
9      end
10 endmodule
```

NORMAL submodule_dff.v unix | utf-8 | verilog 10% 1:1

...aptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework7

```
1  module submodule_xor(
2      output wire XOR,
3      input wire A,B
4  );
5  xor U0 (XOR, A, B);
6  endmodule
```

NORMAL submodule_xor.v unix | utf-8 | verilog 16% 1:1



Άσκηση 10

Terminal Default

...ptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab2/homework10

```
1  `timescale 1ns/1ps
2
3  module shiftReg4_tb;
4  reg [3:0]din;
5  reg clock, reset;
6  wire [3:0]D00, D01, D02, D03;
7
8  shiftReg4 DUT( .din(din), .clock(clock), .reset(reset),
9  .D00(D00), .D01(D01), .D02(D02), .D03(D03));
10
11 initial begin
12     $dumpfile("shiftReg4_tb.vcd");
13     $dumpvars;
14
15              reset=1'b1;
16              clock=1'b0;
17              din=4'b1011;
18     #0.1     clock=1'b1;
19     #0.1     clock=1'b0;
20     #0.1     clock=1'b1;
21              reset=1'b0;
22     #0.1     clock=1'b0;
23     #0.1     clock=1'b1;
24              reset=1'b1;
25     #0.1     clock=1'b0;
26     #0.1     clock=1'b1;
27     #1
28     end
29 endmodule
```

INSERT shiftReg4_tb.v | + unix | utf-8 | verilog 3% 1:1

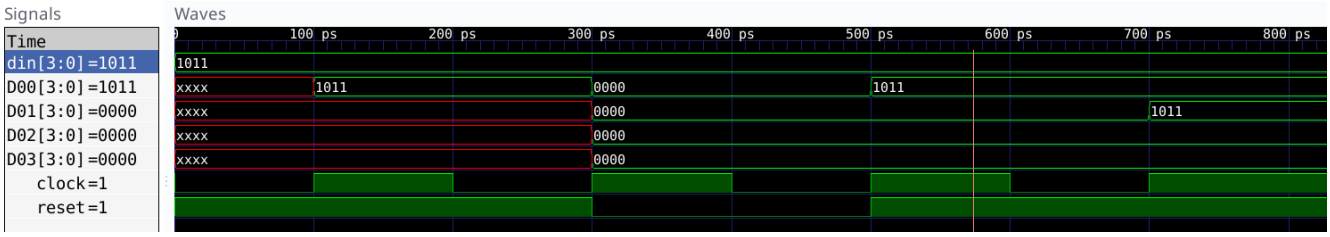
-- INSERT --

...ptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab2/homework10

```
1  `timescale 1ns/1ps
2
3  module shiftReg4(
4      input [3:0] din,
5      input clock, reset,
6      output reg [3:0] D00, D01, D02, D03);
7
8  always @(posedge clock)
9      begin
10         if(!reset)
11             begin
12                 D00 <= 4'b0;
13                 D01 <= 4'b0;
14                 D02 <= 4'b0;
15                 D03 <= 4'b0;
16             end
17         else
18             begin
19                 D00 <= din;
20                 D01 <= D00;
21                 D02 <= D01;
22                 D03 <= D02;
23             end
24
25     end
26
27 endmodule
```

NORMAL shiftReg4.v unix | utf-8 | verilog 3% 1:1

"shiftReg4.v" 27L, 517C



Άσκηση 12

Terminal Default

...ptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework12

```
1  `timescale 1ns/1ps
2  module stimulus;
3  reg clk, load,enable;
4  reg reset;
5  reg s_in;
6  wire s_out;
7  free_run_shift_reg #(2) s1 (
8  .clk(clk),
9  .reset(reset),
10 .s_in(s_in),
11 .s_out(s_out),
12 .enable(enable),
13 .load(load)
14 );
15 integer i, j;
16 initial
17 begin
18 clk = 0;
19 for(i =0; i<=40; i=i+1)
20 begin
21 #10 clk = ~clk;
22 end
23 end
24 initial
25 begin
26 $dumpfile("test.vcd");
27 $dumpvars(0,stimulus);
28 s_in = 0; reset =1;
29 enable=1; load=1;
30 #2 s_in = 0 ; reset = 0;
31 #2 reset =1;
32 for(i =0; i<=10; i=i+1)
33 begin
34 #20 s_in = ~s_in;
35 end
36 #20 s_in =1;
```

INSERT shiftReg4_tb.v | + unix | utf-8 | verilog 2% 1:1

-- INSERT --

...ptop: ~/Μαθήματα/Οργάνωση Η/Υ/2020-2021/Οργάνωση Η/Υ (Εργαστήριο)/lab2/homework12

```
1  module free_run_shift_reg
2  #(parameter N=4)
3  (
4  input wire clk, reset,
5  input wire s_in,
6  input enable,load,
7  output wire s_out
8  );
9  reg [N-1:0] r_reg;
10 wire [N-1:0] r_next;
11
12 always @(posedge clk, negedge reset)
13 begin
14 if (!reset)
15 r_reg <= 0;
16 else if(!load)
17 r_reg=3;
18 else
19 if(enable)
20 r_reg <= r_next;
21 end
22 assign r_next = {s_in, r_reg[N-1:1]};
23 assign s_out = r_reg[0];
24
25 endmodule
```

NORMAL shiftReg4.v unix | utf-8 | verilog 4% 1:1

"shiftReg4.v" 25L, 368C