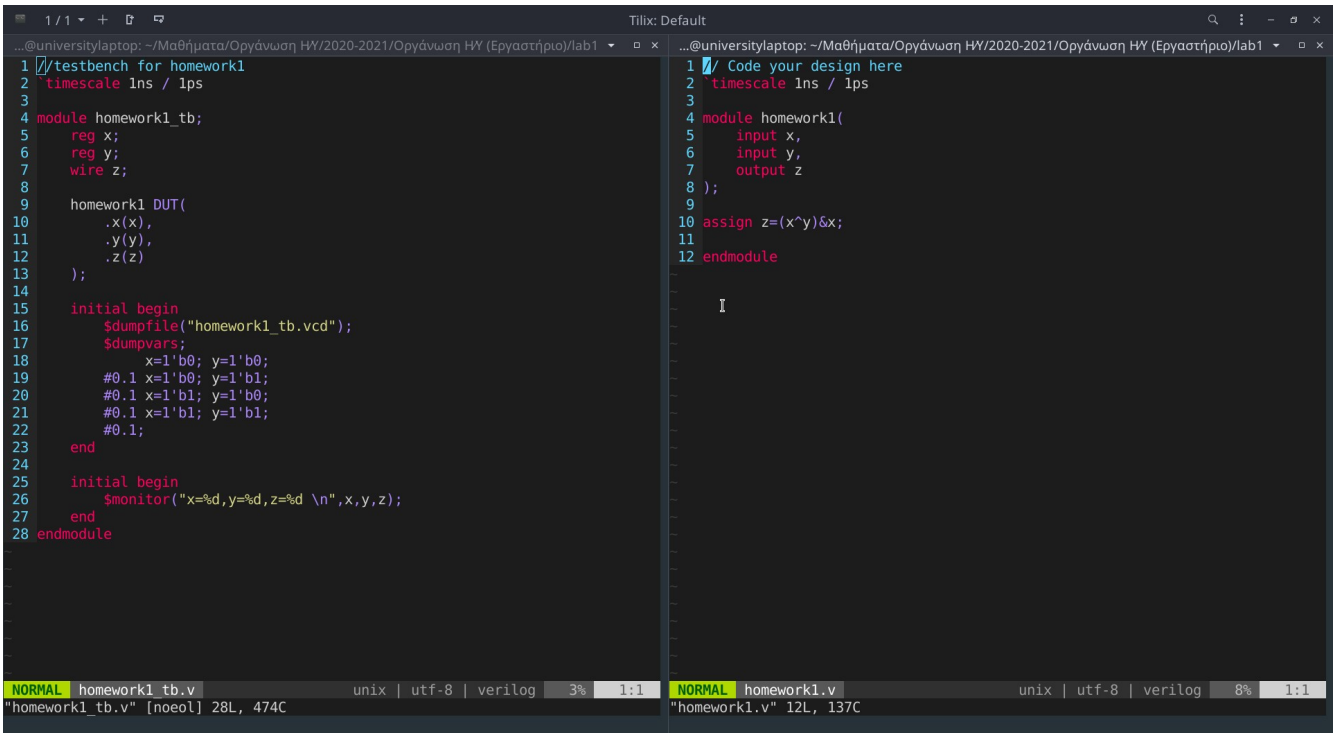
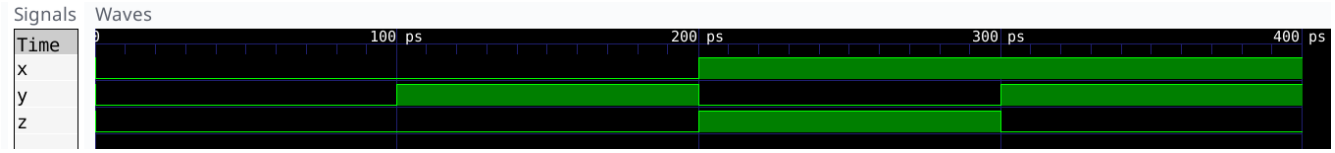
Μοσχογιάννης Πασχάλης ΑΜ 2114026
Άσκηση 1



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

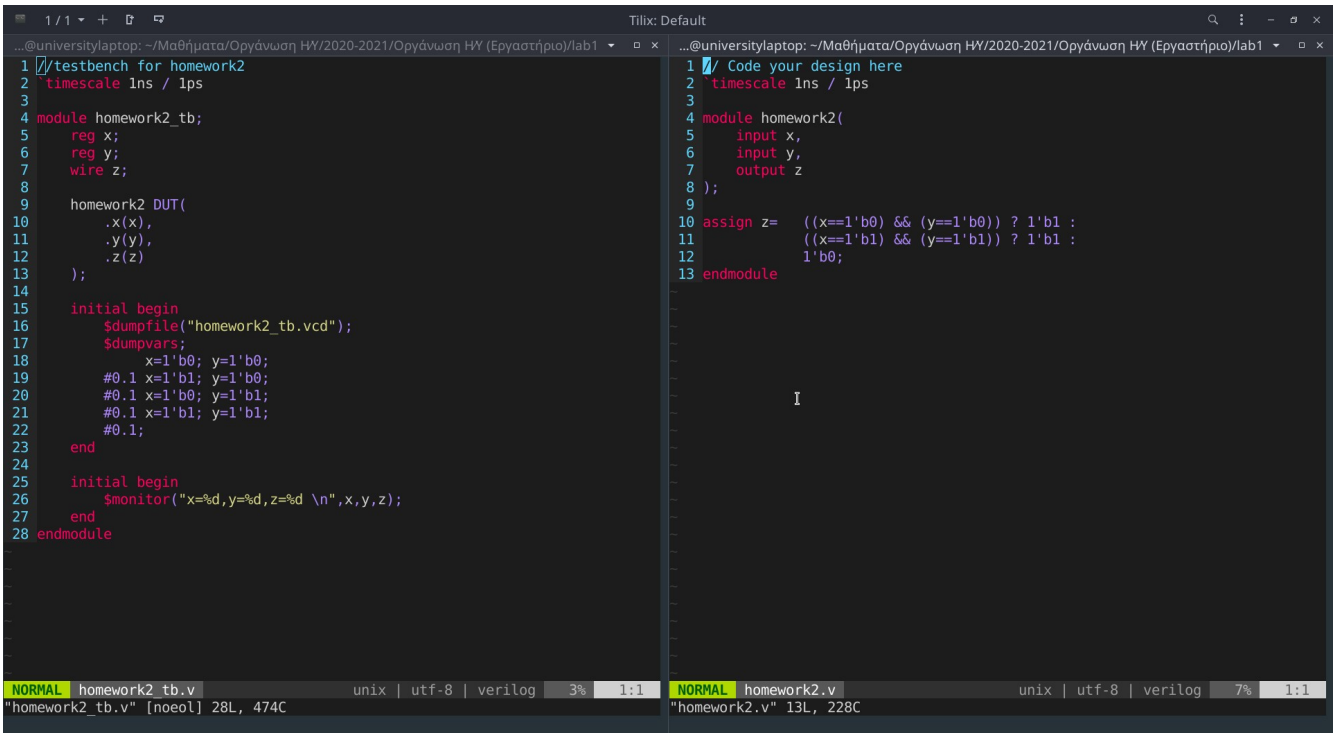
```
$ iverilog -Wall -o homework1_tb.vvp homework1_tb.v homework1.v
$ vvp homework1_tb.vvp
$ ls
 homework1_tb.v
 homework1_tb.vcd
 homework1_tb.vvp
 homework1.v
```

```
$ gtkwave homework1_tb.vcd
```

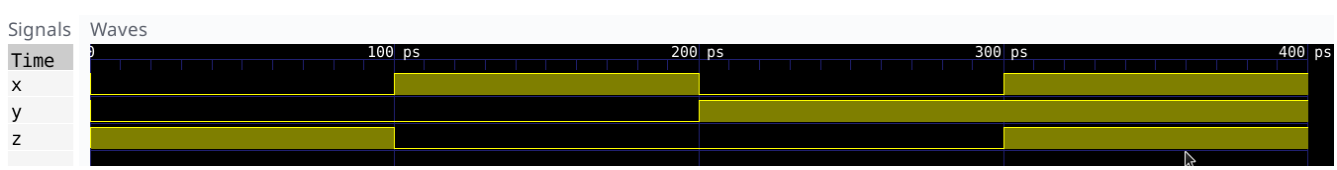


Άσκηση 2

X	Y	Z
0	0	1
1	0	0
0	1	0
1	1	1




```
$ iverilog -Wall -o homework2_tb.vvp homework2_tb.v homework2.v
$ vvp homework2_tb.vvp
$ ls
 homework2_tb.v
 homework2_tb.vcd
 homework2_tb.vvp
 homework2.v

$ gtkwave homework2_tb.vcd
```



Άσκηση 3

x(1)	x(0)	y(1)	y(0)	z
0	0	0	0	1
1	0	1	0	1
0	1	0	1	1
1	1	1	1	1

```
//testbench for homework3
`timescale 1ns / 1ps

module homework3_tb;
    reg [1:0] x, y;
    wire z;

    homework3 DUT(
        .x(x),
        .y(y),
        .z(z)
    );

    initial begin
        $dumpfile("homework3_tb.vcd");
        $dumpvars;
             x=2'b00; y=2'b00;
        #0.1 x=2'b00; y=2'b01;
        #0.1 x=2'b00; y=2'b10;
        #0.1 x=2'b00; y=2'b11;
        #0.1 x=2'b01; y=2'b00;
        #0.1 x=2'b01; y=2'b01;
        #0.1 x=2'b01; y=2'b10;
        #0.1 x=2'b01; y=2'b11;
        #0.1 x=2'b10; y=2'b00;
        #0.1 x=2'b10; y=2'b01;
        #0.1 x=2'b10; y=2'b10;
        #0.1 x=2'b10; y=2'b11;
        #0.1 x=2'b11; y=2'b00;
        #0.1 x=2'b11; y=2'b01;
        #0.1 x=2'b11; y=2'b10;
        #0.1 x=2'b11; y=2'b11;
        #0.1;
    end

    initial begin
```

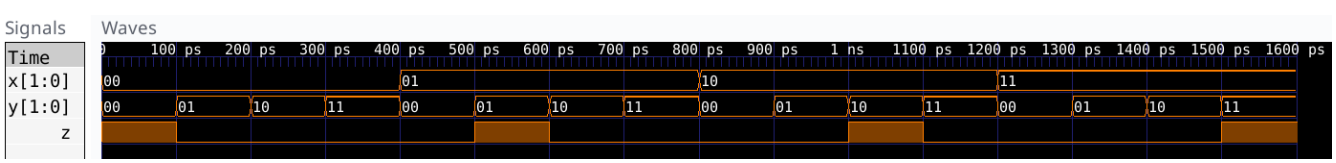
NORMAL homework3_tb.v unix | utf-8 | verilog 33% 13:0
"homework3_tb.v" [noeol] 39L, 852C

```
// Code your design here
`timescale 1ns / 1ps

module homework3(
    input [1:0]x,
    input [1:0]y,
    output z
);
assign z=   ((x==2'b00) && (y==2'b00)) ? 1'b1 :
            ((x==2'b10) && (y==2'b10)) ? 1'b1 :
            ((x==2'b01) && (y==2'b01)) ? 1'b1 :
            ((x==2'b11) && (y==2'b11)) ? 1'b1 :
            2'b0;
endmodule
```

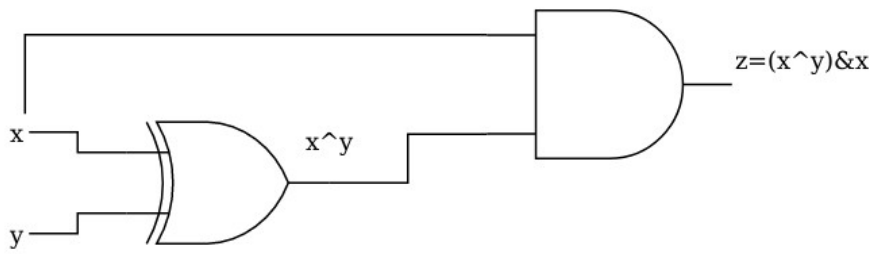
NORMAL homework3.v unix | utf-8 | verilog 7% 1:1
"homework3.v" [noeol] 14L, 336C

$ iverilog -Wall -o homework3_tb.vvp homework3_tb.v homework3.v
$ vvp homework3_tb.vvp
$ ls
homework3_tb.v
homework3_tb.vcd
homework3_tb.vvp
homework3.v

$ gtkwave homework3_tb.vcd



Άσκηση 4

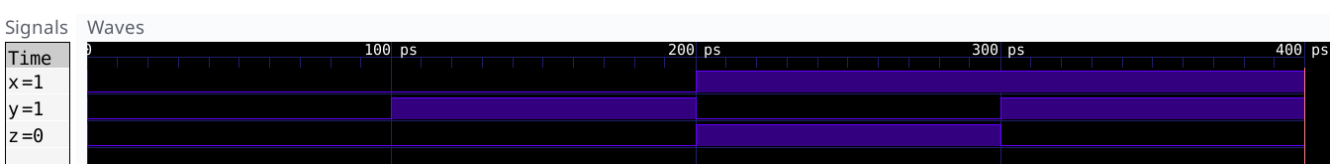



```
//testbench for homework1
`timescale 1ns / 1ps

module homework4_tb;
    reg x;
    reg y;
    wire z;

    homework4 DUT(
        .x(x),
        .y(y),
        .z(z)
    );

    initial begin
        $dumpfile("homework4_tb.vcd");
        $dumpvars;
            x=1'b0; y=1'b0;
        #0.1 x=1'b0; y=1'b1;
        #0.1 x=1'b1; y=1'b0;
        #0.1 x=1'b1; y=1'b1;
        #0.1;
    end

    initial begin
        $monitor("x=%d,y=%d,z=%d \n",x,y,z);
    end
endmodule
```

```
// Code your design here
`timescale 1ns/1ps

module homework4(
    output wire z,
    input wire x,y
);

wire xy_xor;
xor U0 (xy_xor, x, y);
and U1 (z, xy_xor, x);

endmodule
```

$ iverilog -Wall -o homework3_tb.vvp homework3_tb.v homework3.v
$ vvp homework3_tb.vvp
$ ls
homework3_tb.v
homework3_tb.vcd
homework3_tb.vvp
homework3.v

$ gtkwave homework3_tb.vcd



Άσκηση 5

```
...laptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework5
`timescale 1ns/1ps
`include "homework5.v"

module homework5_tb;
    reg [7:0] A, B;
    wire [7:0] Sum;
    wire Cout;

    ripple_carry DUT(
        .Sum(Sum),
        .Cout(Cout),
        .A(A),
        .B(B));

    initial begin
        $dumpfile("homework5_tb.vcd");
        $dumpvars;
             A=8'b00000000; B=8'b00001010;
        #0.1 A=8'b01110001; B=8'b11001110;
        #0.1 A=8'b01101100; B=8'b00100011;
        #0.1 A=8'b00011100; B=8'b01001011;
        #0.1;
    end

    initial begin
        $monitor ("Value of A is %d, B is %d, and the calculated sum is %d,
 as well as the Cout is %d at time %0t",A,B,Sum,Cout,$time);
    end

endmodule

INSERT  homework5_tb.v                    unix | utf-8 | verilog   3%   1:1
-- INSERT --
```

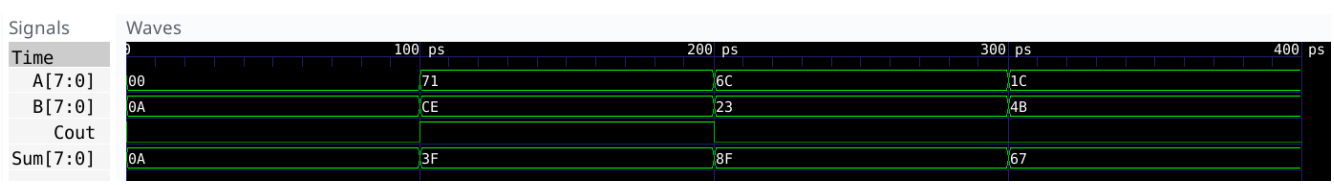
```
...aptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework5
`include "full_adder.v"
module ripple_carry(
    output wire [7:0] Sum,
    output wire Cout,
    input wire [7:0] A, B);

    wire c1, c2, c3, c4, c5, c6, c7;

    full_adder U0(.Sum(Sum[0]), .Cout(c1), .A(A[0]), .B(B[0]), .Cin(1'b0));
    full_adder U1(.Sum(Sum[1]), .Cout(c2), .A(A[1]), .B(B[1]), .Cin(c1));
    full_adder U2(.Sum(Sum[2]), .Cout(c3), .A(A[2]), .B(B[2]), .Cin(c2));
    full_adder U3(.Sum(Sum[3]), .Cout(c4), .A(A[3]), .B(B[3]), .Cin(c3));
    full_adder U4(.Sum(Sum[4]), .Cout(c5), .A(A[4]), .B(B[4]), .Cin(c4));
    full_adder U5(.Sum(Sum[5]), .Cout(c6), .A(A[5]), .B(B[5]), .Cin(c5));
    full_adder U6(.Sum(Sum[6]), .Cout(c7), .A(A[6]), .B(B[6]), .Cin(c6));
    full_adder U7(.Sum(Sum[7]), .Cout(Cout), .A(A[7]), .B(B[7]), .Cin(c7));
endmodule
NORMAL  homework5.v                       unix | utf-8 | verilog   5%   1:1
"homework5.v" 17L, 770C written
```

```
...aptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework5
`include "half_adder.v"
module full_adder(output wire Sum, Cout, input wire A, B, Cin);
    wire ha_1_sum, ha_1_cout, ha_2_cout;

    half_adder U0(.Sum(ha_1_sum), .Cout(ha_1_cout), .A(A), .B(B));
    half_adder U1(.Sum(Sum), .Cout(ha_2_cout), .A(ha_1_sum), .B(Cin));
    or U2(Cout, ha_2_cout, ha_1_cout);
endmodule
NORMAL  full_adder.v                      unix | utf-8 | verilog  50%   4:0
"full_adder.v" 8L, 321C written
```

```
...aptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework5
module half_adder(output wire Sum, Cout, input wire A,B);
xor U0 (Sum, A, B);
and U1 (Cout, A, B);
endmodule
NORMAL  half_adder.v                      unix | utf-8 | verilog  25%   1:1
```

$ iverilog -Wall -o homework5_tb.vvp homework5_tb.v
$ vvp homework5_tb.vvp
$ ls
full_adder.v
half_adder.v
homework5_tb.v
homework5_tb.vcd
homework5_tb.vvp
homework5.v

$ gtkwave homework5_tb.vcd



Άσκηση 6

Tilix: Default

...laptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework6

```verilog
`timescale 1ns / 1ps

module BCD_tb;
    reg [3:0] binary;
    wire [3:0] tens;
    wire [3:0] ones;
    BCD DUT (
        .binary(binary),
        .ones(ones)
    );

    initial begin
    $dumpfile("BCD_tb.vcd");
    $dumpvars;
        binary=4'b1101;
    #10 binary=4'b1000;
    #10 binary=4'b1001;
    #10 binary=4'b1010;
    #10;
    end
endmodule
```

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

"BCD_tb.v" [noeol] 21L, 354C

...aptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework6

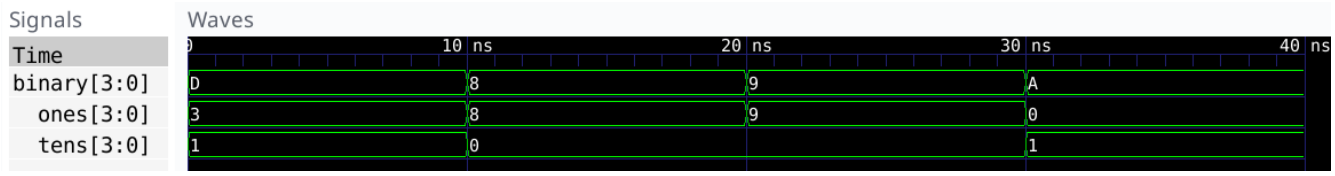
```verilog
module BCD(
    input [3:0] binary,
    output reg [3:0] tens,
    output reg [3:0] ones
);
    integer i;
    always @(binary)
    begin
        tens = 4'd0;
        ones = 4'd0;
        for(i=3; i>=0; i=i-1)
        begin
            if(tens >= 5)
            tens = tens + 3;
            if(ones >= 5)
            ones = ones + 3;
            tens = tens << 1;
            tens[0] = ones[3];
            ones = ones << 1;
            ones[0] = binary[i];
        end
    end
endmodule
```

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

"BCD.v" [noeol] 23L, 366C

$ iverilog -Wall -o BCD_tb.vvp BCD_tb.v BCD.v
$ vvp BCD_tb.vvp
$ ls
BCD_tb.v BCD_tb.vcd BCD_tb.vvp BCD.v

$ gtkwave BCD_tb.vcd



Άσκηση 7

a

```
//testbench for homework7
`timescale 1ns / 1ps

module homework7_tb;
    reg A;
    reg B;
    reg C;
    reg D;
    wire Y;
    reg [1:0] sel;

    homework7 DUT(
        .A(A),
        .B(B),
        .C(C),
        .D(D),
        .sel(sel),
        .Y(Y)
    );

    initial begin
        $dumpfile("homework7_tb.vcd");
        $dumpvars;
             A=1'b1; B=1'b0; C=1'b1; D=1'b0; sel=2'b00;
        #0.1 A=1'b1; B=1'b0; C=1'b1; D=1'b0; sel=2'b10;
        #0.1 A=1'b1; B=1'b0; C=1'b1; D=1'b0; sel=2'b01;
        #0.1 A=1'b1; B=1'b0; C=1'b1; D=1'b0; sel=2'b11;
        #0.1;
    end

    initial begin
        $monitor("SELECT=%d,A=%d,B=%d,C=%d,D=%d,Y=%d \n",sel,A,B,C,D,Y);
    end
endmodule
```

```
// Code your design here
`timescale 1ns / 1ps

module homework7(
    output Y,
    input A,B,C,D,
    input [1:0] sel
);

assign Y = (sel == 2'b00) ? A :
           (sel == 2'b10) ? B :
           (sel == 2'b01) ? C :
           (sel == 2'b11) ? D :
           1'bX;
endmodule
```

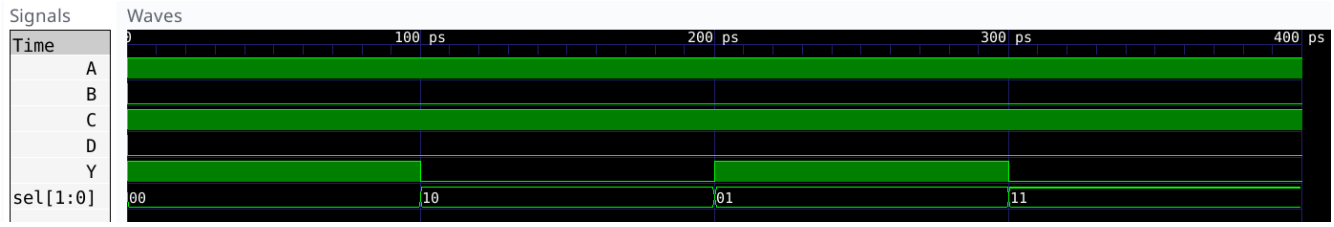
$ iverilog -Wall -o homework7_tb.vvp homework7_tb.v homework7.v
$ ls
homework7_tb.v homework7_tb.vcd homework7_tb.vvp homework7.v
$ vvp homework7_tb.vvp
VCD info: dumpfile homework7_tb.vcd opened for output.
SELECT=0,A=1,B=0,C=1,D=0,Y=1

SELECT=2,A=1,B=0,C=1,D=0,Y=0

SELECT=1,A=1,B=0,C=1,D=0,Y=1

SELECT=3,A=1,B=0,C=1,D=0,Y=0

$ gtkwave homework7_tb.vcd



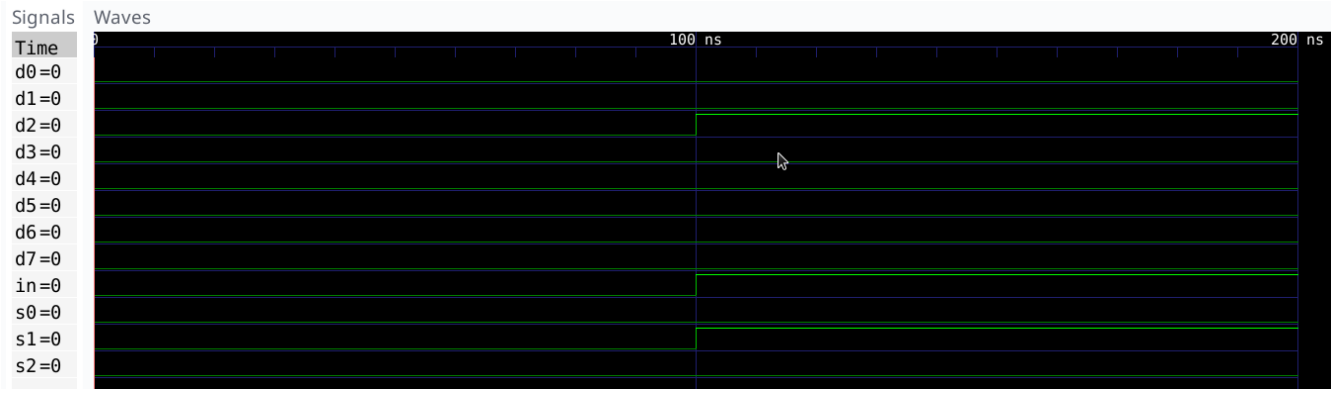
b

```
...laptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework7
`timescale 1ns / 1ps

module demultiplexer_tb;
// Inputs
reg in;
reg s0,s1,s2;

// Outputs
wire d0,d1,d2,d3,d4,d5,d6, d7;

// Instantiate the Unit Under Test (UUT)
demultiplexer DUT(
    .in(in),
    .s0(s0),.s1(s1),.s2(s2),
    .d0(d0),.d1(d1),.d2(d2),.d3(d3),
    .d4(d4),.d5(d5),.d6(d6),.d7(d7)
);
initial begin
    $dumpfile("demultiplexer_tb.vcd");
    $dumpvars;
    // Initialize Inputs
          in = 0;
          s0 = 0;
          s1 = 0;
          s2 = 0;
    // Wait 100 ns for global reset to finish
    #100; in = 1;
          s0 = 0;
          s1 = 1;
          s2 = 0;
    // Wait 100 ns for global reset to finish
    #100;
    // Add stimulus here
    end
endmodule
NORMAL  demultiplexer_tb.v   unix | utf-8 | verilog   2%   1:1
"demultiplexer_tb.v" 35L, 687C written
```

```
...aptop: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework7
`timescale 1ns / 1ps

module demultiplexer(
    in,
    s0,s1,s2,
    d0,d1,d2,d3,d4,d5,d6,d7);
input in,s0,s1,s2;
output d0,d1,d2,d3,d4,d5,d6,d7;
assign  d0=(in & ~s2 & ~s1 &~s0),
        d1=(in & ~s2 & ~s1 &s0),
        d2=(in & ~s2 & s1 &~s0),
        d3=(in & ~s2 & s1 &s0),
        d4=(in & s2 & ~s1 &~s0),
        d5=(in & s2 & ~s1 &s0),
        d6=(in & s2 & s1 &~s0),
        d7=(in & s2 & s1 &s0);
endmodule
INSERT  demultiplexer.v   unix | utf-8 | verilog   5%   1:2
-- INSERT --
```

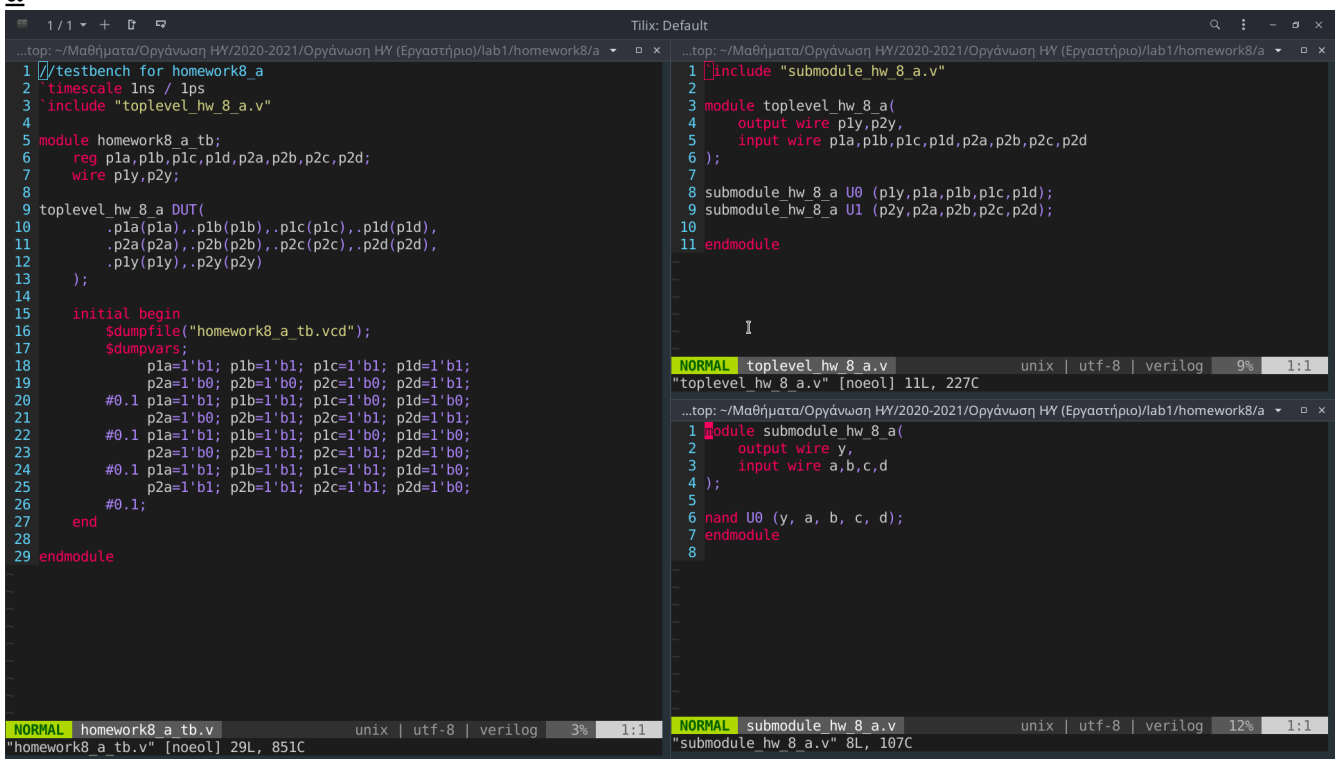
$ iverilog -Wall -o demultiplexer_tb.vvp demultiplexer_tb.v demultiplexer.v
$ vvp demultiplexer_tb.vvp
$ ls
demultiplexer_tb.v demultiplexer_tb.vcd demultiplexer_tb.vvp demultiplexer.v

$ gtkwave demultiplexer_tb.vcd



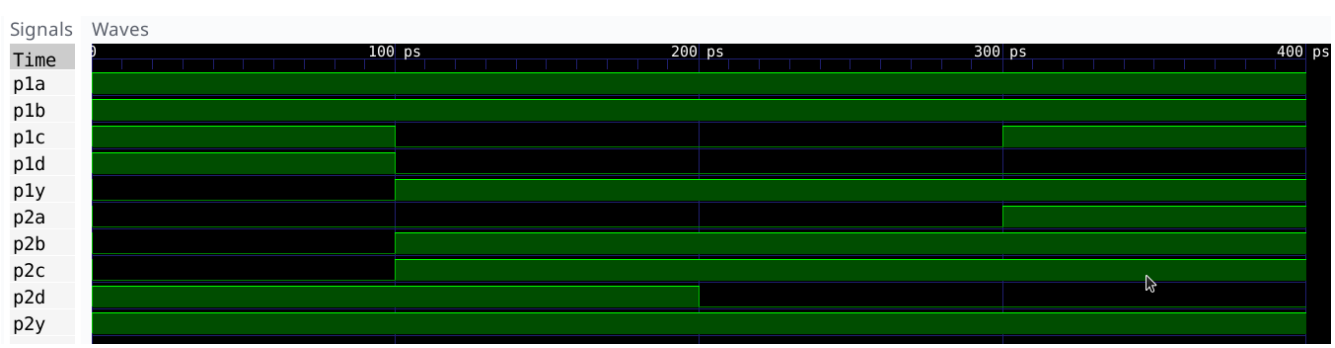
Άσκηση 8

a



$ iverilog -Wall -o homework8_a_tb.vvp homework8_a_tb.v
$ vvp homework8_a_tb.vvp
$ ls
homework8_a_tb.v
homework8_a_tb.vcd
homework8_a_tb.vvp
submodule_hw_8_a.v
toplevel_hw_8_a.v

$ gtkwave homework8_a_tb.vcd



b

Tilix: Default

...top: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework8/b

```verilog
`timescale 1ns / 1ns
`include "toplevel_hw_8_b.v"

module homework_b_tb;
    reg S0, S1, S2, D0, D1, D2, D3, D4, D5, D6, D7;
    wire O;

    toplevel_hw_8_b DUT(
        .S0(S0), .S1(S1), .S2(S2),
        .D0(D0), .D1(D1), .D2(D2), .D3(D3),
        .D4(D4), .D5(D5), .D6(D6), .D7(D7),
        .O(O)
    );

    initial begin
        $dumpfile("homework_b_tb.vcd");
        $dumpvars;
            S0=1'b0; S1=1'b0; S2=1'b0;
            D0=1'b0; D1=1'b0; D2=1'b0; D3=1'b0;
            D4=1'b0; D5=1'b0; D6=1'b0; D7=1'b0;
        #10 S0=1'b0; S1=1'b0; S2=1'b0;
            D0=1'b1; D1=1'b1; D2=1'b1; D3=1'b1;
            D4=1'b1; D5=1'b1; D6=1'b1; D7=1'b1;
        #10;
    end
endmodule
```

NORMAL homework8_b_tb.v unix | utf-8 | verilog 3% 1:1

"homework8_b_tb.v" [noeol] 26L, 694C

...top: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework8/b

```verilog
`include "submodule_hw_8_b_or.v"
`include "submodule_hw_8_b_not.v"
`include "submodule_hw_8_b_and.v"

module toplevel_hw_8_b(
    output wire O,
    input wire S0, S1, S2, D0, D1, D2, D3, D4, D5, D6, D7
);
wire NOTS0,NOTS1,NOTS2,AND0,AND1,AND2,AND3,AND4,AND5,AND6,AND7;

submodule_hw_8_b_not U0 (NOTS0, S0);
submodule_hw_8_b_not U1 (NOTS1, S1);
submodule_hw_8_b_not U2 (NOTS2, S2);
submodule_hw_8_b_and U3 (AND0, NOTS0, NOTS1, NOTS2, D0);
submodule_hw_8_b_and U4 (AND1, S0, NOTS1, NOTS2, D1);
submodule_hw_8_b_and U5 (AND2, NOTS0, S1, NOTS2, D2);
submodule_hw_8_b_and U6 (AND3, S0, S1, NOTS2, D3);
submodule_hw_8_b_and U7 (AND4, NOTS0, NOTS1, S2, D4);
submodule_hw_8_b_and U8 (AND5, S0, NOTS1, S2, D5);
submodule_hw_8_b_and U9 (AND6, NOTS0, S1, S2, D6);
submodule_hw_8_b_and U10 (AND7, S0, S1, S2, D7);
submodule_hw_8_b_or U11 (O, AND0,AND1,AND2,AND3,AND4,AND5,AND6,AND7);
endmodule
```

NORMAL toplevel_hw_8_b.v unix | utf-8 | verilog 4% 1:1

...top: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework8/b

```verilog
module submodule_hw_8_b_or(
    output wire F1,
    input wire A, B, C, D, E, F, G, H
);
or U0 (F1, A, B, C, D, E, F, G, H);
endmodule
```

NORMAL submodule_hw_8_b_or.v unix | utf-8 | verilog 16% 1:1

"submodule_hw_8_b_or.v" [noeol] 6L, 139C

Tilix: Default

...op: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework8/b

```verilog
module submodule_hw_8_b_not(
    output wire F1,
    input wire A
);
not U0 (F1, A);
endmodule
```

NORMAL submodule_hw_8_b_not.v unix | utf-8 | verilog 16% 1:1

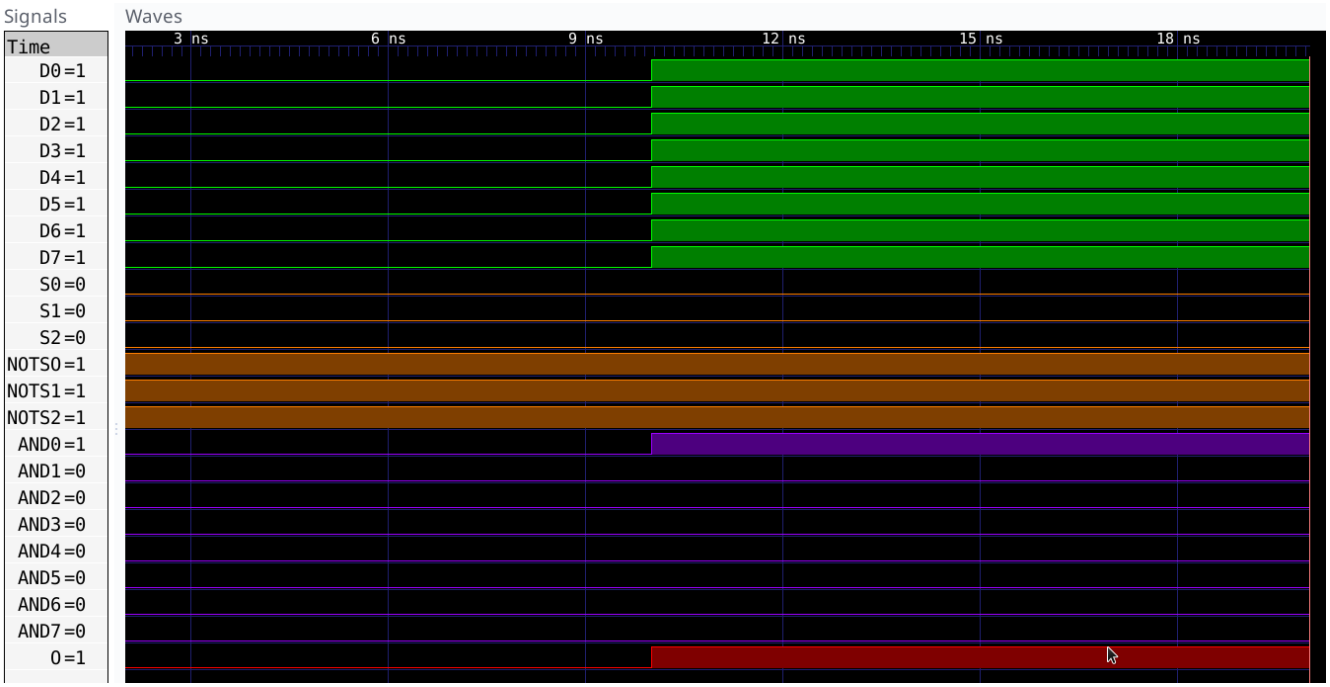
...op: ~/Μαθήματα/Οργάνωση ΗΥ/2020-2021/Οργάνωση ΗΥ (Εργαστήριο)/lab1/homework8/b

```verilog
module submodule_hw_8_b_and(
    output wire F1,
    input wire A, B, C, D
);
and U0 (F1, A, B, C, D);
endmodule
```

NORMAL submodule_hw_8_b_and.v unix | utf-8 | verilog 16% 1:1

```
$ iverilog -Wall -o homework8_b_tb.vvp homework8_b_tb.v
$ vvp homework8_b_tb.vvp
$ ls
homework8_b_tb.v
homework8_b_tb.vvp
homework_b_tb.vcd
submodule_hw_8_b_and.v
submodule_hw_8_b_not.v
submodule_hw_8_b_or.v
toplevel_hw_8_b.v
```

$ gtkwave homework8_b_tb.vcd



c

```verilog
//testbench for homework1
`timescale 1ns / 1ps
`include "toplevel_hw_8_c.v"

module homework8_c_tb;
    reg X;
    reg Y;
    reg Z;
    wire F;

    toplevel_hw_8_c DUT(
        .X(X),
        .Y(Y),
        .Z(Z),
        .F(F)
    );

    initial begin
        $dumpfile("homework8_c_tb.vcd");
        $dumpvars;
            X=1'b0; Y=1'b0; Z=1'b0;
        #10 X=1'b0; Y=1'b0; Z=1'b1;
        #10 X=1'b0; Y=1'b1; Z=1'b0;
        #10 X=1'b0; Y=1'b1; Z=1'b1;
        #10 X=1'b1; Y=1'b0; Z=1'b0;
        #10 X=1'b1; Y=1'b0; Z=1'b1;
        #10 X=1'b1; Y=1'b1; Z=1'b0;
        #10 X=1'b1; Y=1'b1; Z=1'b1;
        #10;
    end

    initial begin
        $monitor("X=%d,Y=%d,Z=%d,F=%d \n",X,Y,Z,F);
    end
endmodule
```

```verilog
`include "submodule_hw_8_c_or.v"
`include "submodule_hw_8_c_not.v"
`include "submodule_hw_8_c_xor.v"

module toplevel_hw_8_c(
    output wire F,
    input wire X,Y,Z
);

wire OR1,NOTX,OR2,XOR1;
submodule_hw_8_c_or U0 (OR1,X,Y);
submodule_hw_8_c_not U1 (NOTX,X);
submodule_hw_8_c_xor U2 (XOR1,Y,Z);
submodule_hw_8_c_or U3 (OR2,NOTX,XOR1);
submodule_hw_8_c_xor U4 (F,XOR1,OR2);
endmodule
```

```verilog
module submodule_hw_8_c_not(
    output wire F1,
    input wire A
);
not U0 (F1, A);
endmodule
```

```verilog
module submodule_hw_8_c_or(
    output wire F1,
    input wire A,B
);
or U0 (F1, A, B);
endmodule
```

```verilog
module submodule_hw_8_c_xor(
    output wire F1,
    input wire A,B
);
xor U0 (F1, A, B);
endmodule
```

$ iverilog -Wall -o homework8_c_tb.vvp homework8_c_tb.v
$ vvp homework8_c_tb.vvp
$ ls
homework8_c_tb.v
homework8_c_tb.vcd
homework8_c_tb.vvp
submodule_hw_8_c_not.v
submodule_hw_8_c_or.v
submodule_hw_8_c_xor.v
toplevel_hw_8_c.v

$ gtkwave homework8_c_tb.vcd