Τμήμα Πληροφορικής
Μάθημα: Προηγμένες Εφαρμογές Ψηφιακής Σχεδίασης
Ονοματεπώνυμο Κωνσταντίνος Δαμιανός
ΑΕΜ: 5346
Εξάμηνο: 3ο

ΑΠΑΝΤΗΣΕΙΣ 1ης Σειράς Ασκήσεων

Άσκηση 1

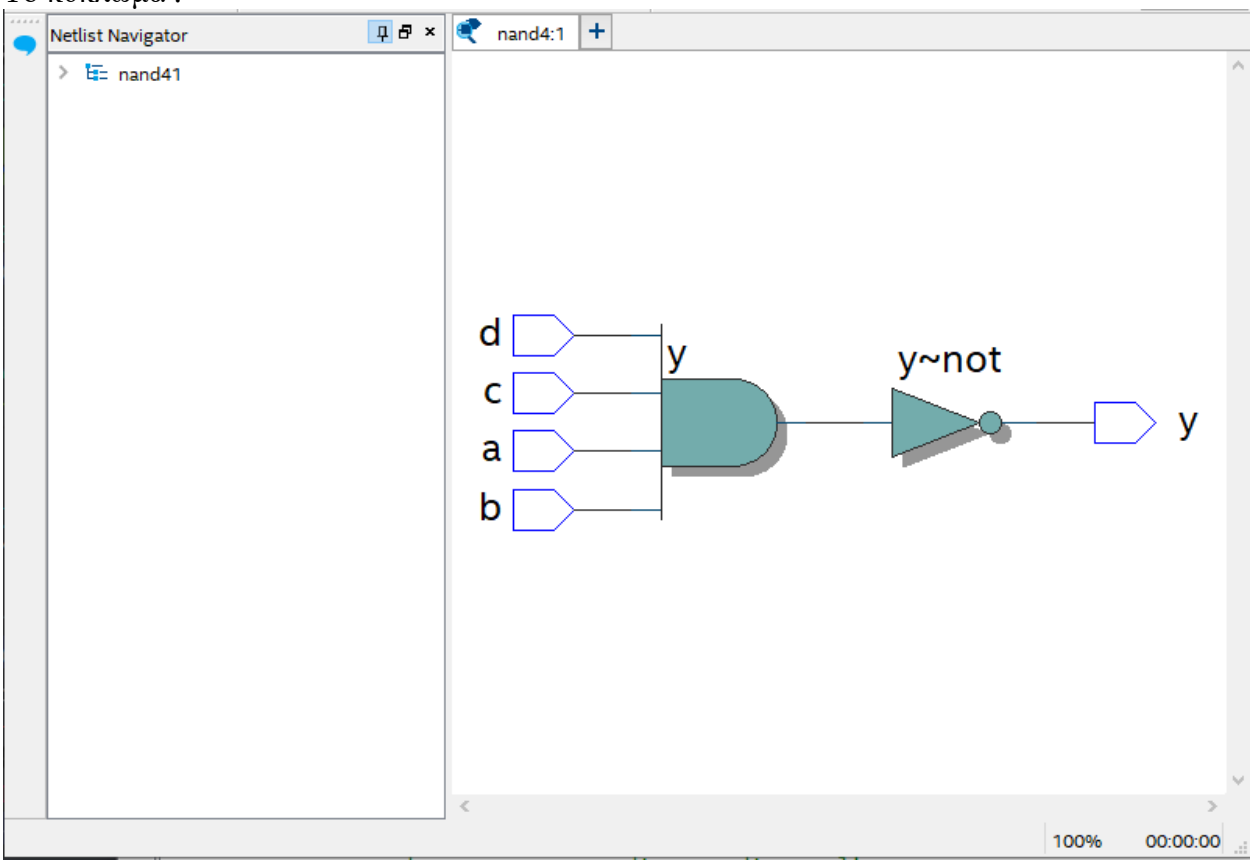
Ο κωδικας (sv) για την nand4 :

```
module nand41(input logic a,b,c,d , output logic y);
    assign y =~(a&b&c&d);
endmodule
```

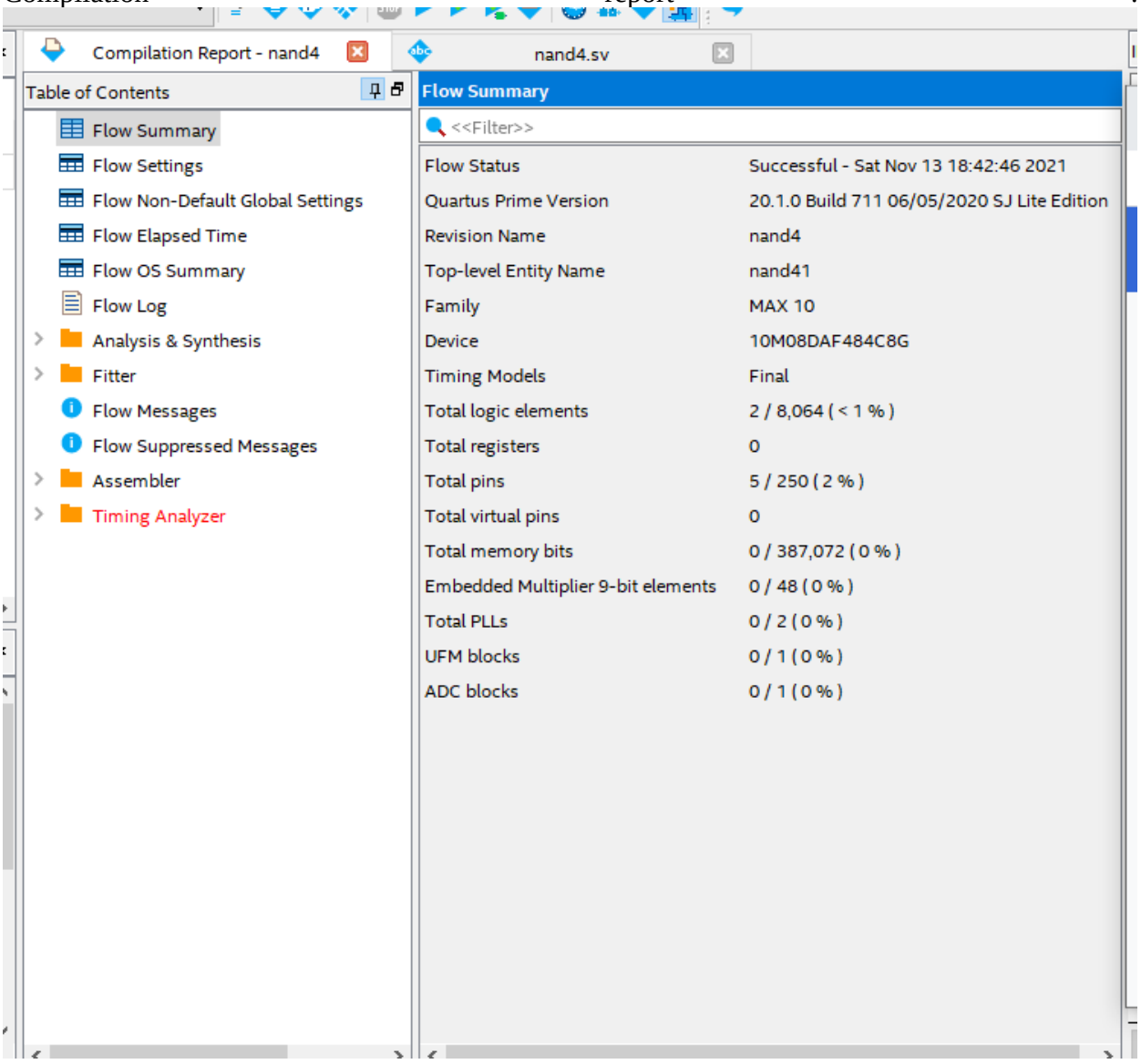
Ο κωδικας (do) για την nand4 :

vsim nand41
add wave *
force a 0 0,1 100 -repeat 200
force b 0 0,1 200 -repeat 400
force c 0 0,1 400 -repeat 800
force d 0 0,1 800 -repeat 1600
run 1600

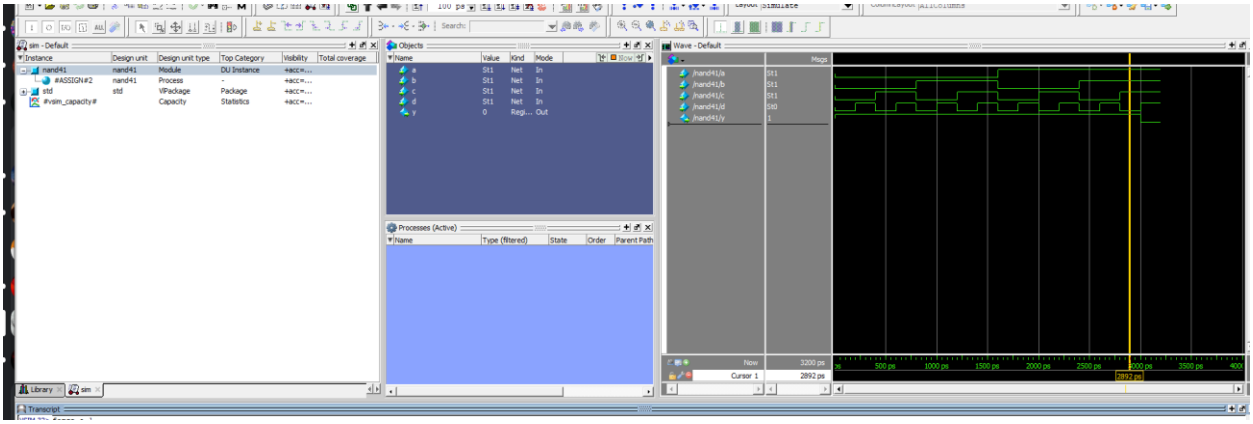
Το κυκλωμα :



Compilation report :



Simulation :



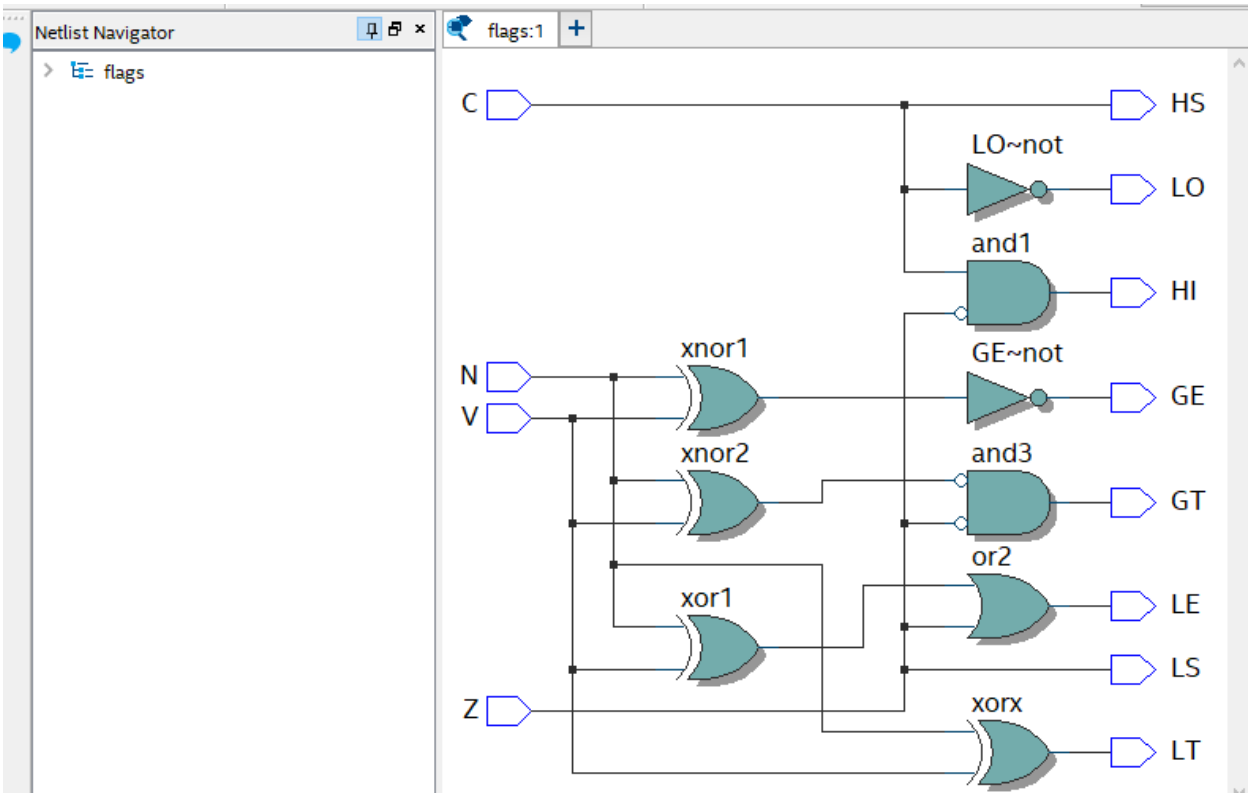
Ο κωδικας (sv) για τα flags:

```
module flags (input logic C,N,V,Z, output logic HS,LS,HI,LO,GE,LE,GT,LT);
        logic X1,X2,X3,X4;
        assign HS= C;
        not inv1(X1,C);
        or ogate1(LS,Z,X1);
        not inv2(X2,Z);
        and and1(HI,X2,C);
        not inv3(LO,C);
        xnor xnor1(GE,N,V);
        xor xor1(X3,N,V);
        or or2(LE,X3,Z);
        xnor xnor2(X4,N,V);
        and and3(GT,X2,X4);
        xor xorx(LT,N,V);
        endmodule
```

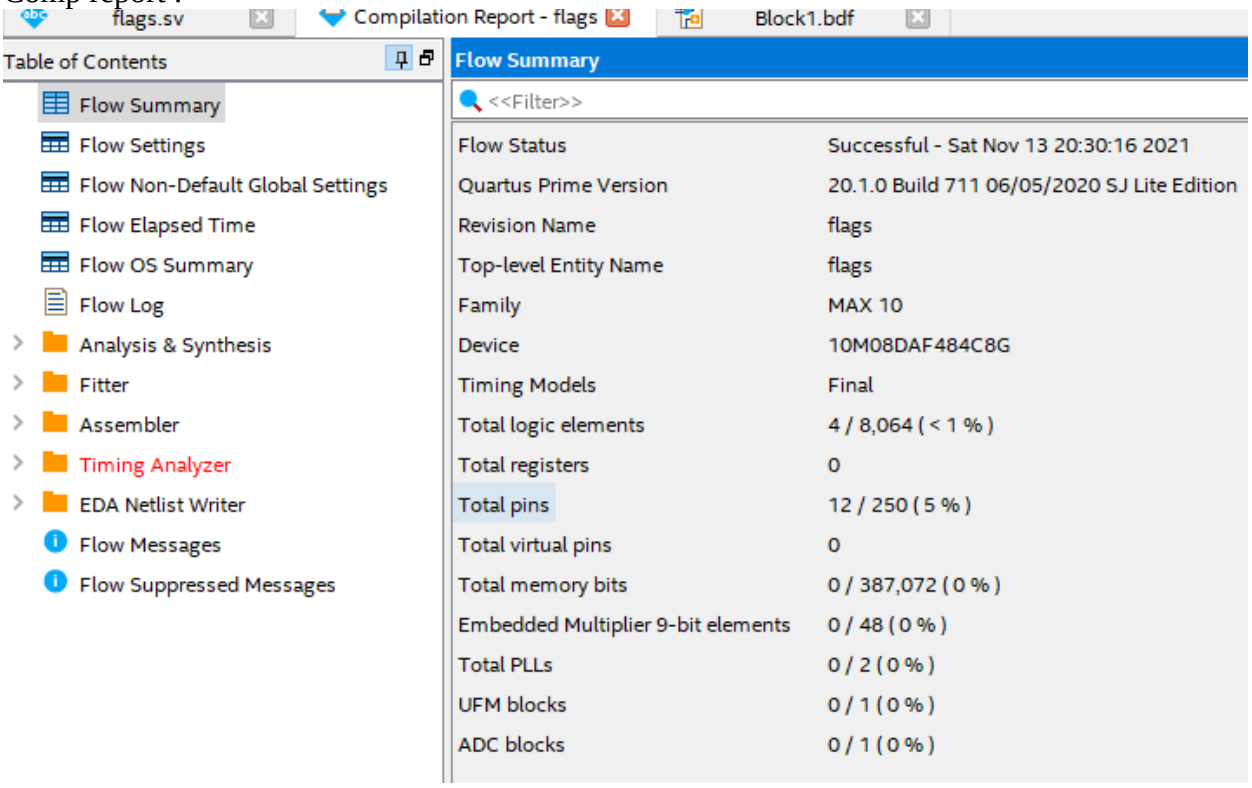
Ο κωδικας do :

```
vsim flags
add wave *
force Z 0 0,1 100 -repeat 200
force V 0 0,1 200 -repeat 400
force N 0 0,1 400 -repeat 800
force C 0 0,1 800 -repeat 1600
run 1600
```

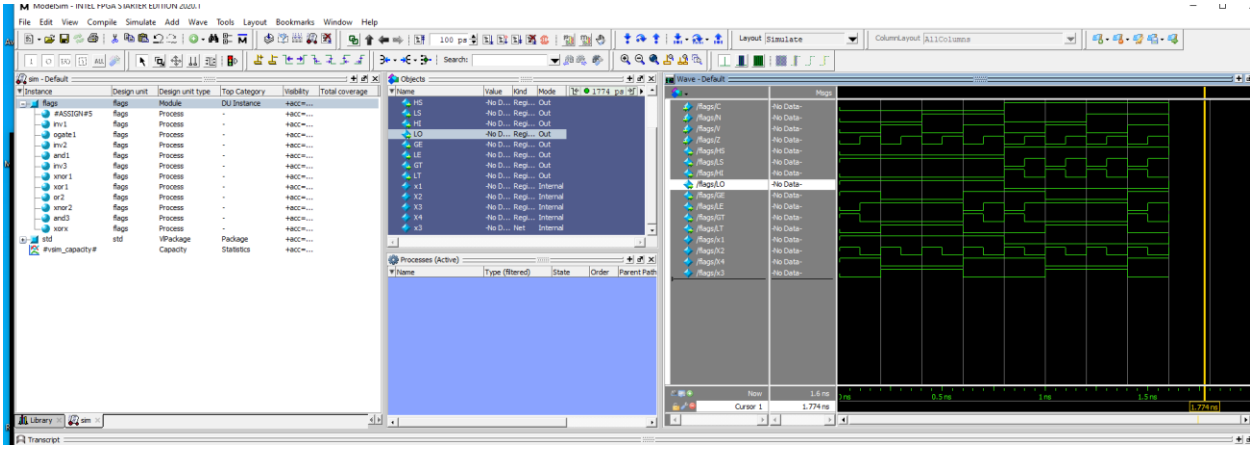
Το κυκλωμα :



Comp report :



Simulation :



<u>Άσκηση 2 :</u>

Ο κωδικας (sv) για την mux2 :

```
module mux2 (input logic x,y,s, output logic m);
        assign m = s ? y : x ;
endmodule
```

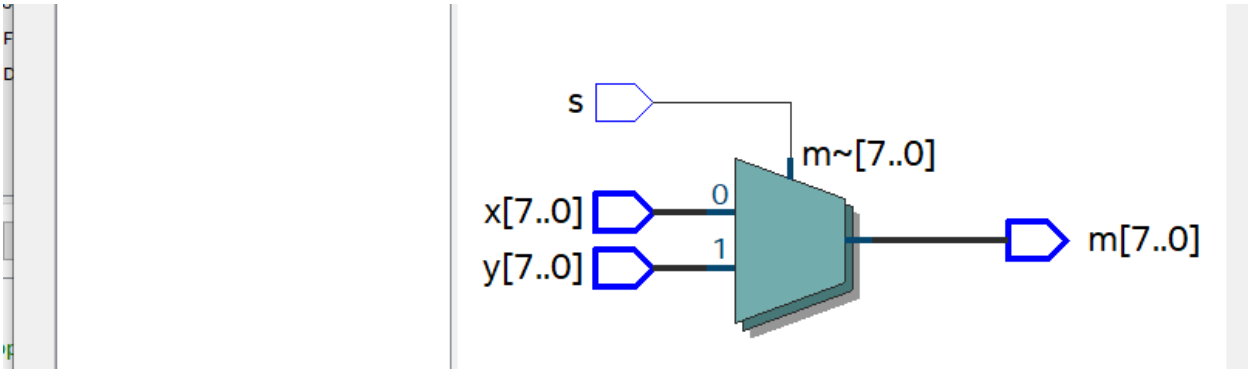
Ο κωδικας (sv) για την mux28 :

```
module mux28 ( input logic [7:0] x,y, input logic s , output logic [7:0] m);
        assign m = s ? y : x;
endmodule
```

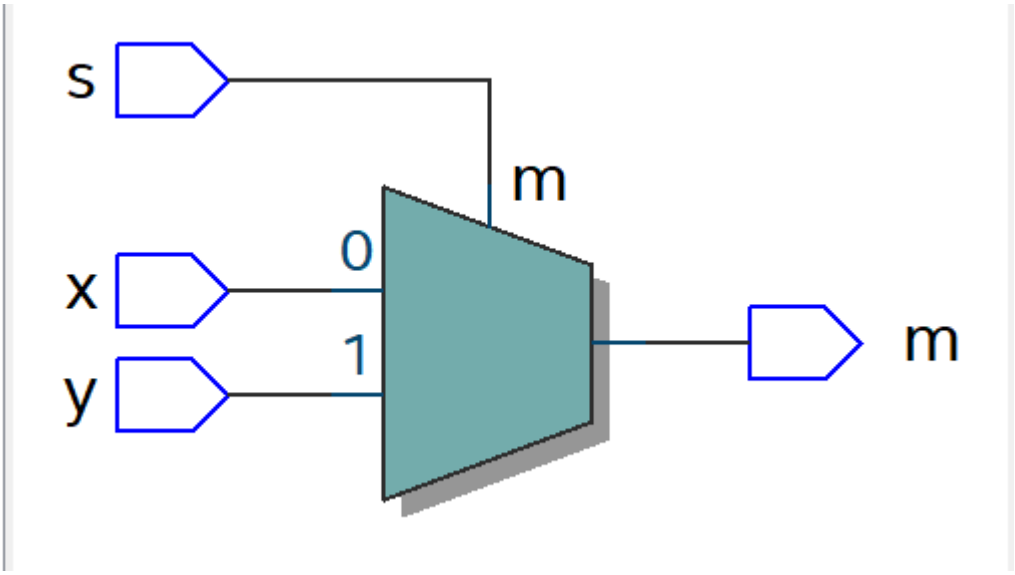
Ο κωδικας do για mux28 :

```
vsim mux28
add wave *
force s 1
force x 11010011
force y 01010101
run 200
force s 0
run 200
```

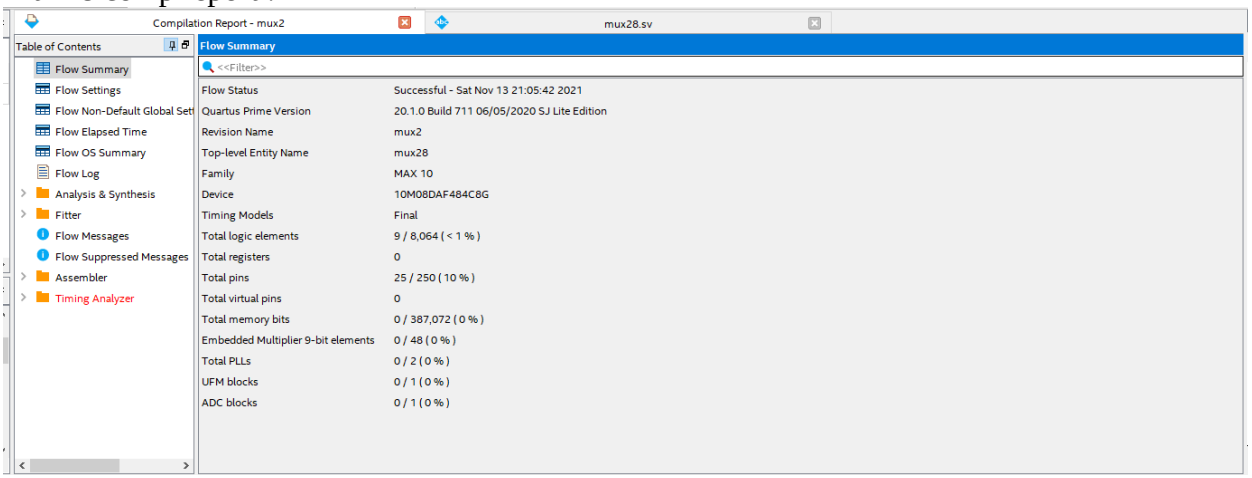
Mux28 κυκλωμα :



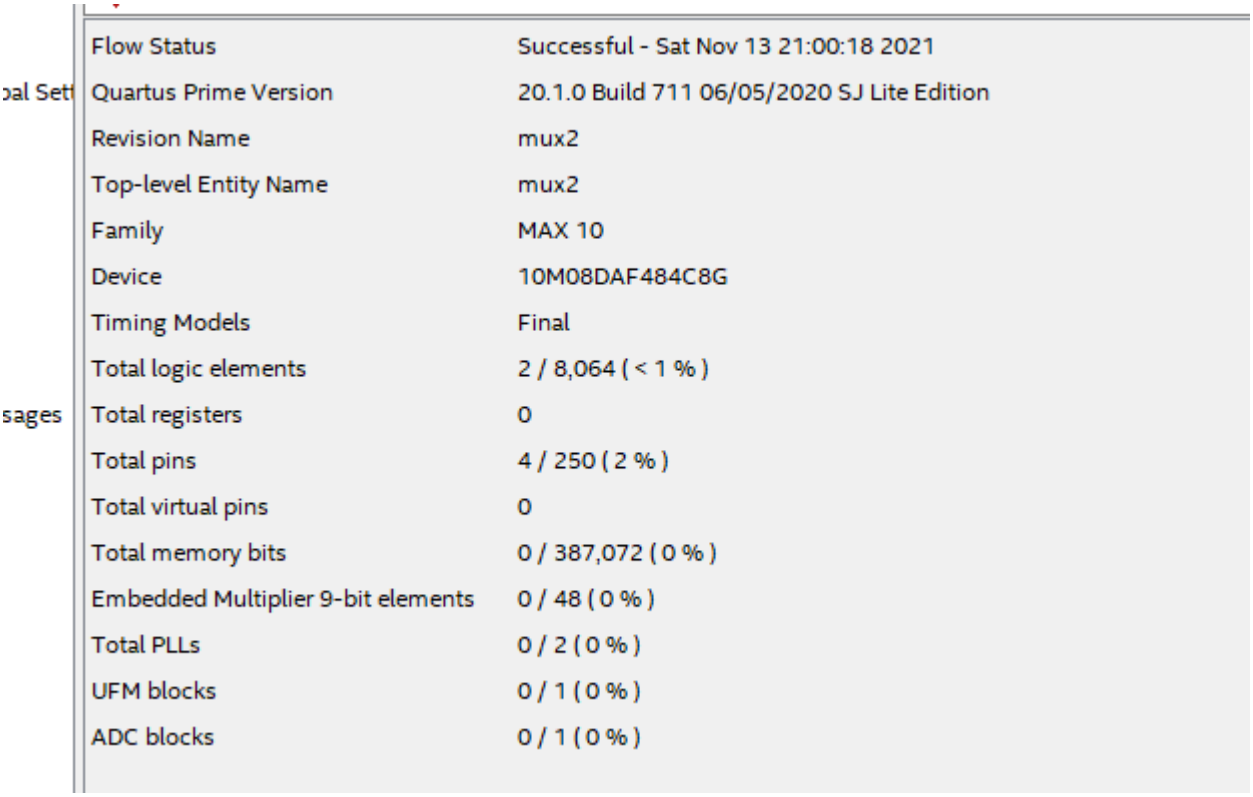
Mux2 κυκλωμα :



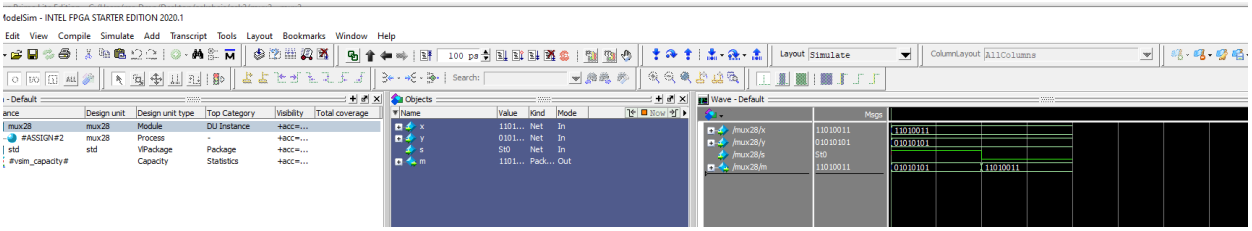
Mux28 comp report :



Mux2 comp report :



MUX 28 SIMULATION :



Άσκηση 3 :

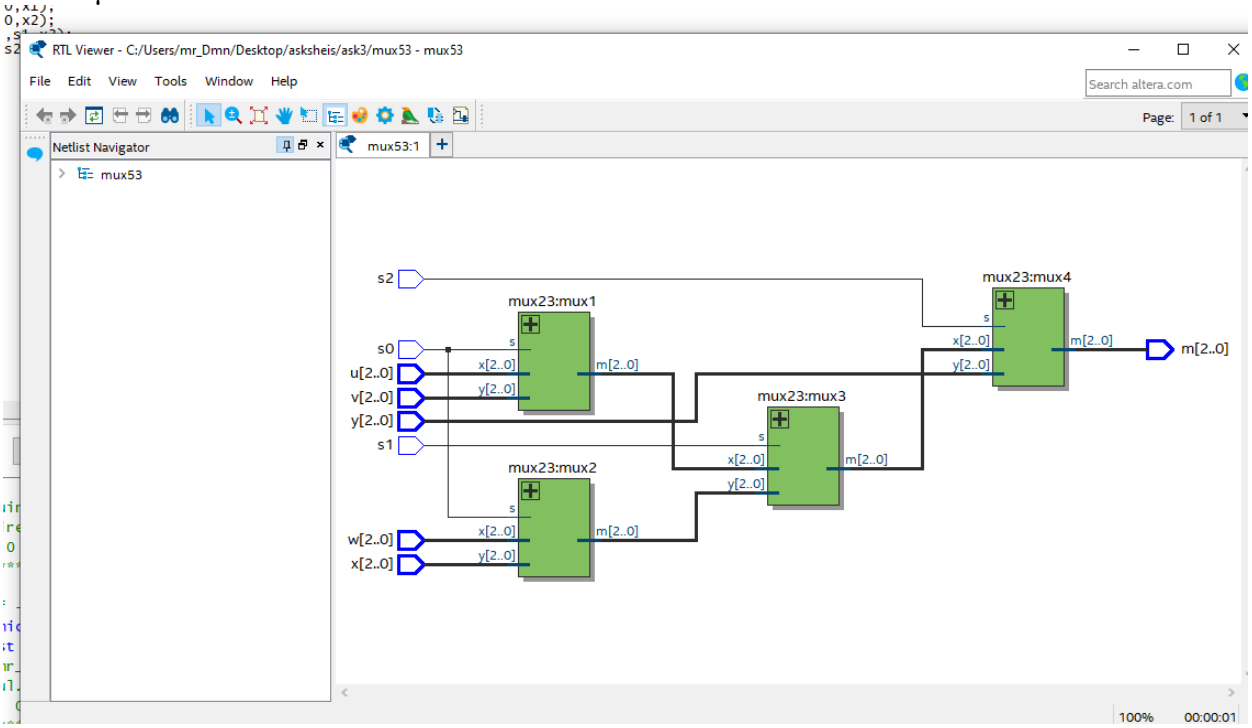
Ο κωδικας sv για mux53 :

```
module mux23(input logic [2:0] x,y,input logic s,output logic [2:0] m);
        assign m = s ? y:x;
endmodule

module mux53(input logic [2:0] u,v,w,x,y,input logic s0,s1,s2,output logic [2:0] m);
        logic [2:0] x1,x2,x3;
        mux23 mux1(u,v,s0,x1);
        mux23 mux2(w,x,s0,x2);
        mux23 mux3(x1,x2,s1,x3);
        mux23 mux4(x3,y,s2,m);
        endmodule
```

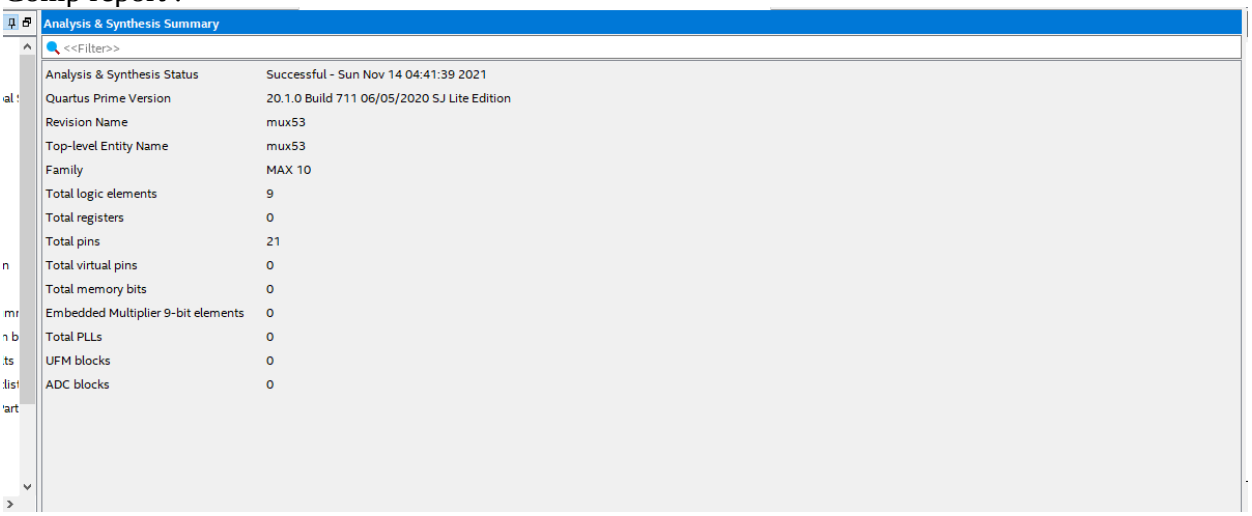
Ο κωδικας do για mux53 :

```
vsim mux53
add wave *
force u 101
force v 111
force w 110
force x 001
force y 010
force s0 0 0,1 100 -repeat 200
force s1 0 0,1 200 -repeat 400
force s2 0 0,1 400 -repeat 800
run 800
```

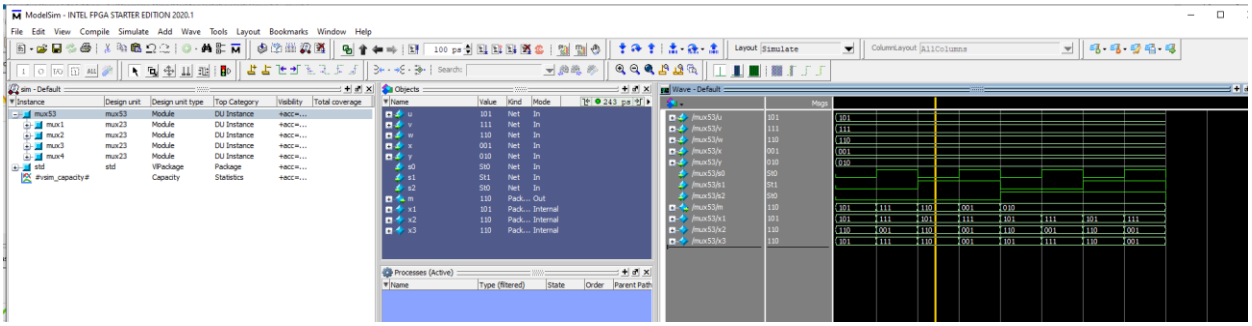
Κυκλωμα :



Comp report :



Simulation :



Άσκηση 4 :

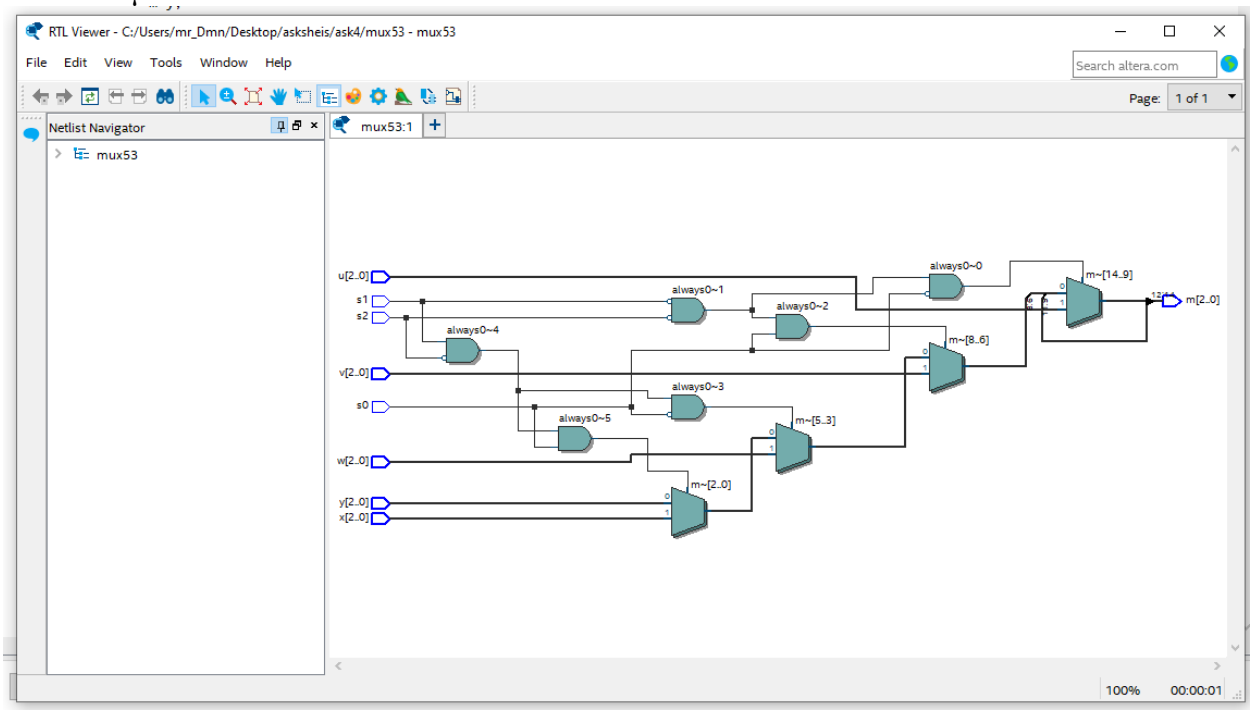
Ο κωδικας (sv) για τον mux53 :

```
module mux53(input logic [2:0] u,v,w,x,y,input logic s0,s1,s2,output logic [2:0] m);
        always_comb
        if (s2==0 & s1 ==0 & s0==0) m=u;
        else if (s2==0 & s1 ==0 & s0==0) m=u;
        else if (s2==0 & s1 ==0 & s0==1) m=v;
        else if (s2==0 & s1 ==1 & s0==0) m=w;
        else if (s2==0 & s1 ==1 & s0==1) m=x;
        else                                m=y;

        endmodule
```

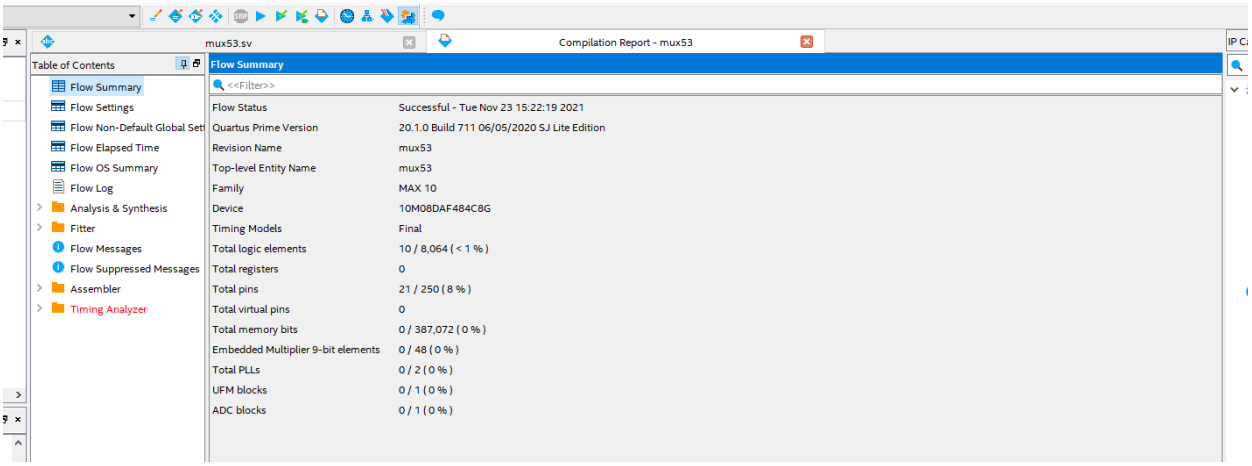
Ο κωδικας do :

```
vsim mux53
add wave *
force u 101
force v 111
force w 110
force x 001
force y 010
force s0 0 0,1 100 -repeat 200
force s1 0 0,1 200 -repeat 400
force s2 0 0,1 400 -repeat 800
run 800
```

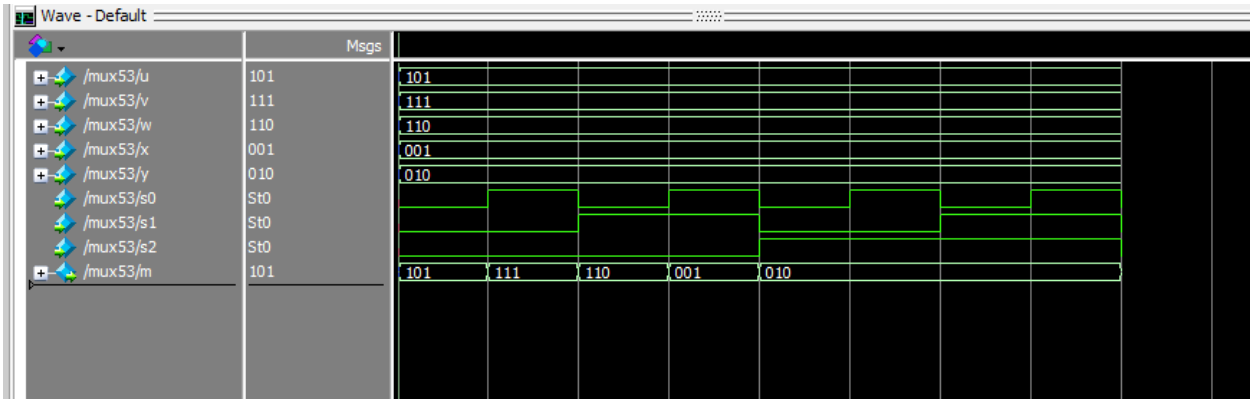
Το κυκλωμα :



Comp report :



Simulation :



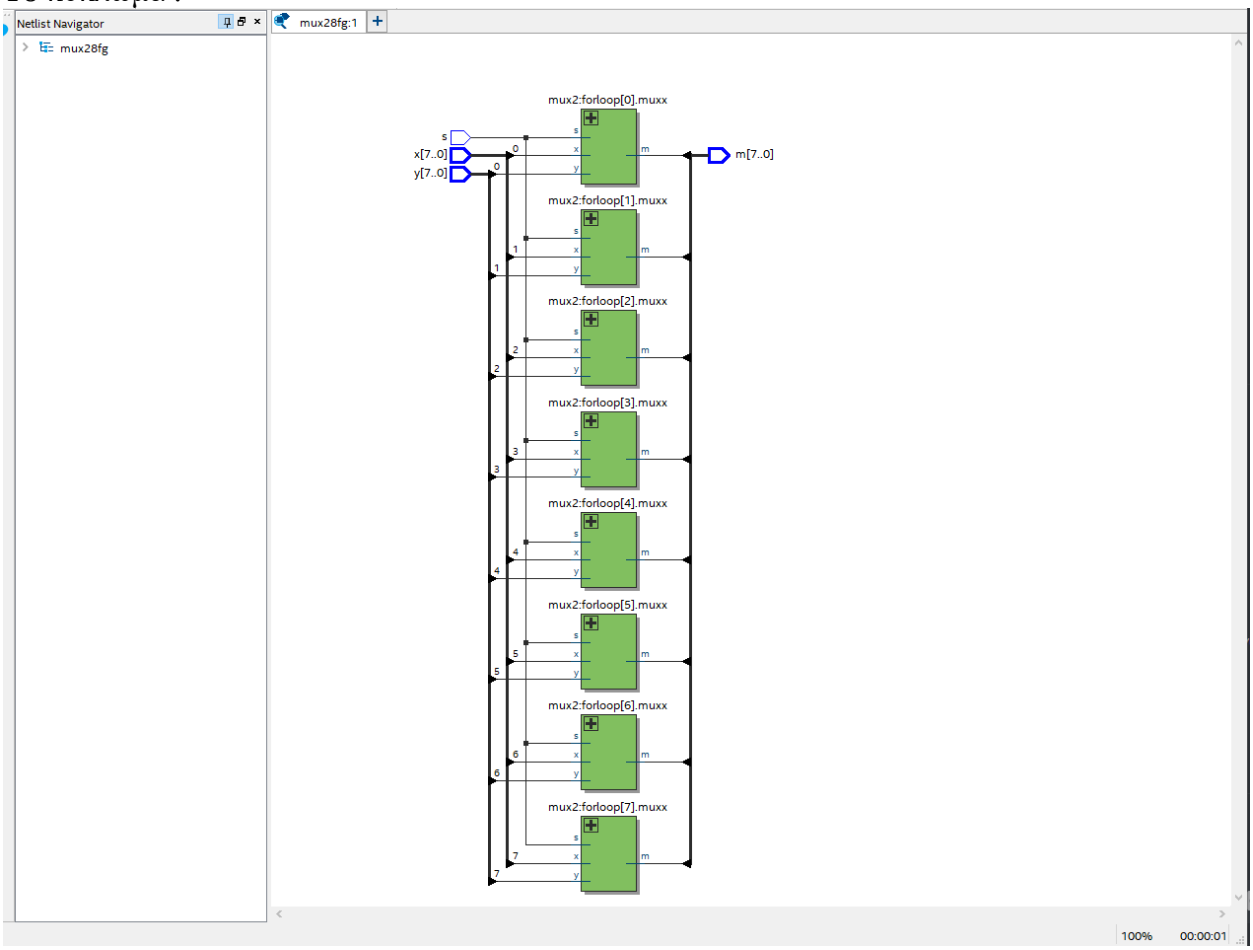
Ο κωδικας sv για το mux28fg :

```
module mux2 (input logic x,y,s, output logic m);
        assign m = s ? y : x ;
endmodule


module mux28fg (input logic [7:0] x,y, input logic s,output logic [7:0]m);
        genvar i;
        generate
        for (i=0; i<=7;i=i+1) begin: forloop
                mux2 muxx(x[i],y[i],s,m[i]);
        end
        endgenerate
endmodule
```

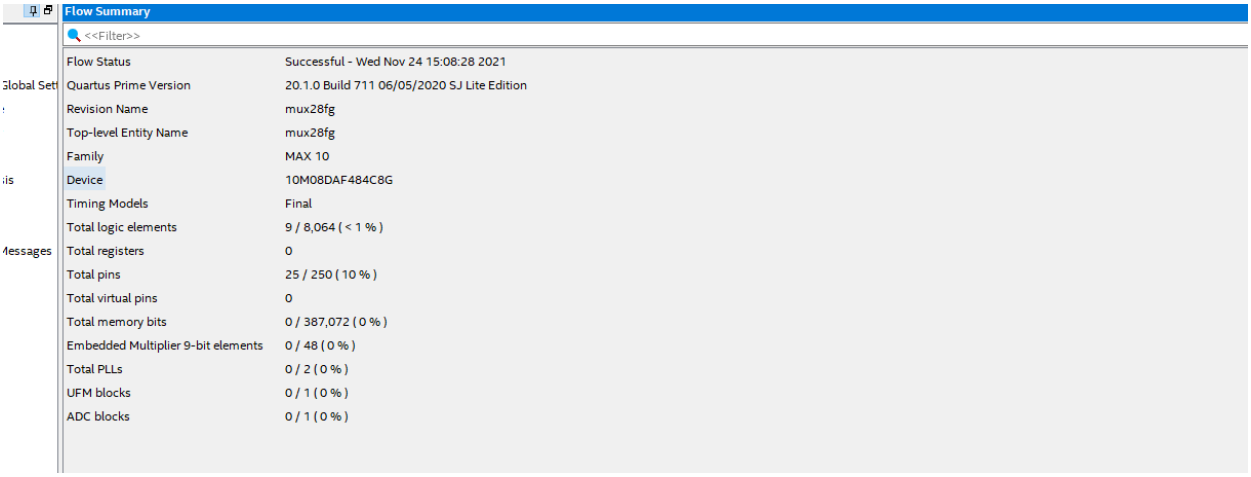
Ο κωδικας do :

```
vsim mux28fg
add wave *
force x 10010011
force y 11111111
force s 1

run 200
force s 0
run 200
```

Το κυκλωμα :



Comp report :

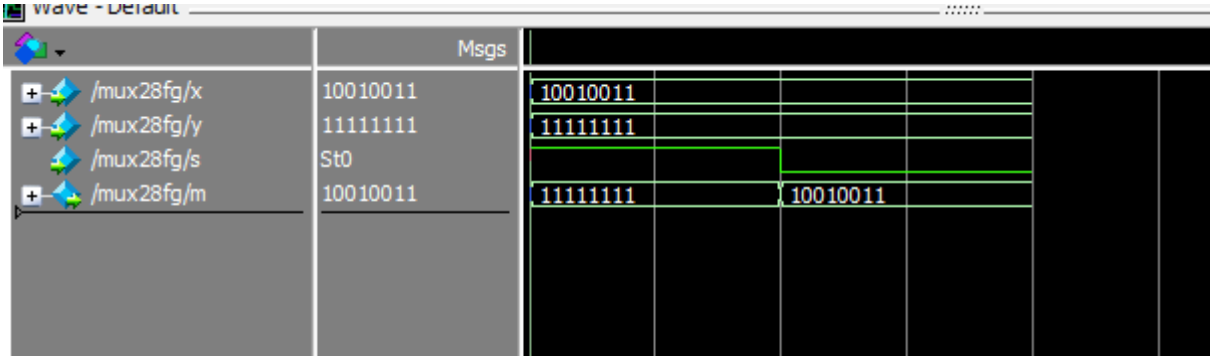


Flow Summary

<<Filter>>

Flow Status	Successful - Wed Nov 24 15:08:28 2021
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	mux28fg
Top-level Entity Name	mux28fg
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	9 / 8,064 (< 1 %)
Total registers	0
Total pins	25 / 250 (10 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 2 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

Simulation :



Άσκηση 5 :

Ο κωδικας sv για το prm :

```
module prm (input logic D,A,E,F, output logic [3:0] y);
        always_comb
        if     (D==1)    y=4'b1000;
        else if (A==1) y=4'b0100;
        else if (E==1) y=4'b0010;
        else if (F==1)  y=4'b0001;
        else                y=4'b0000;

endmodule
```

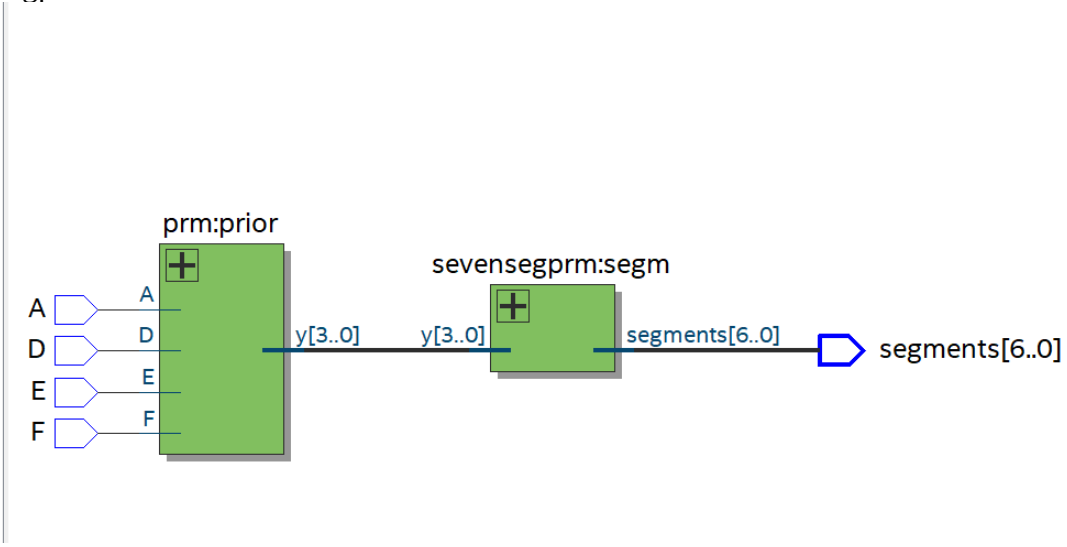
Ο κωδικας sv για το sevensegprm:

```
module sevensegprm(input logic[3:0] y,output logic [6:0] segments);
	always_comb
		case (y)

		8:	segments=	7'b0111101;
		4:	segments=	7'b1110111;
		2:	segments=	7'b1001111;
		1:	segments=	7'b1000111;
		default :segments = 7'b0000000;
	endcase
endmodule
```

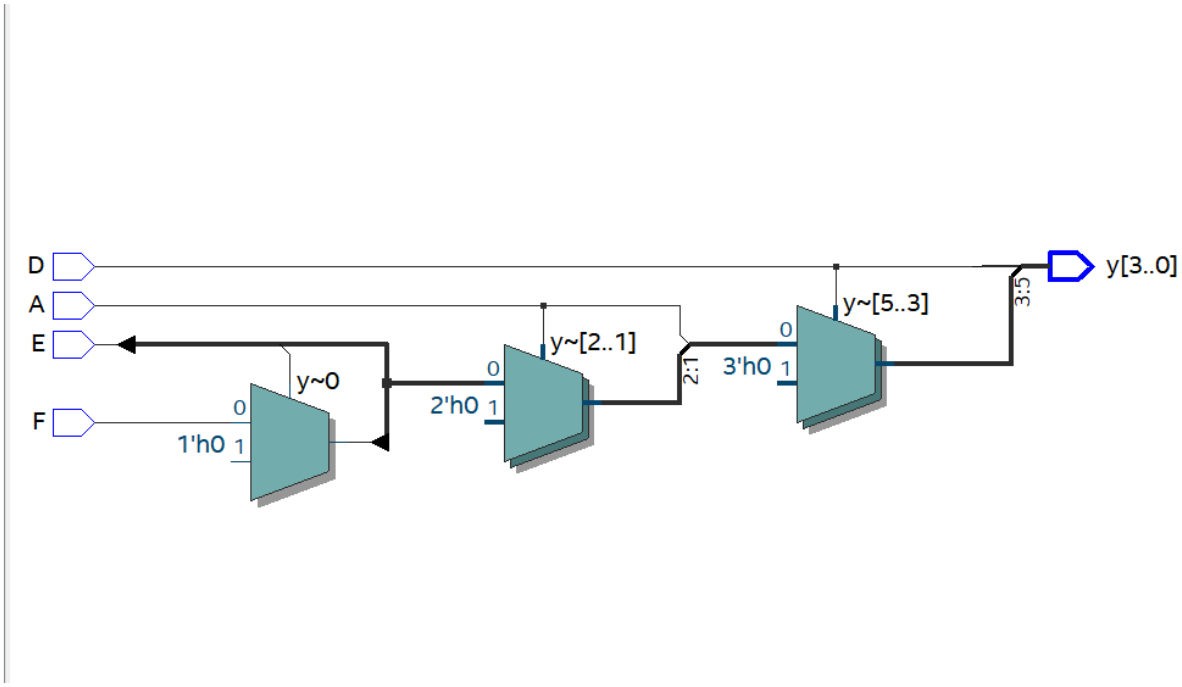
Ο κωδικας sv για το segprm:

```
module segprm(input logic D,A,E,F,output logic [6:0] segments);
	logic [3:0]y;
	prm prior(D,A,E,F,y);
	sevensegprm segm(y,segments);
endmodule
```

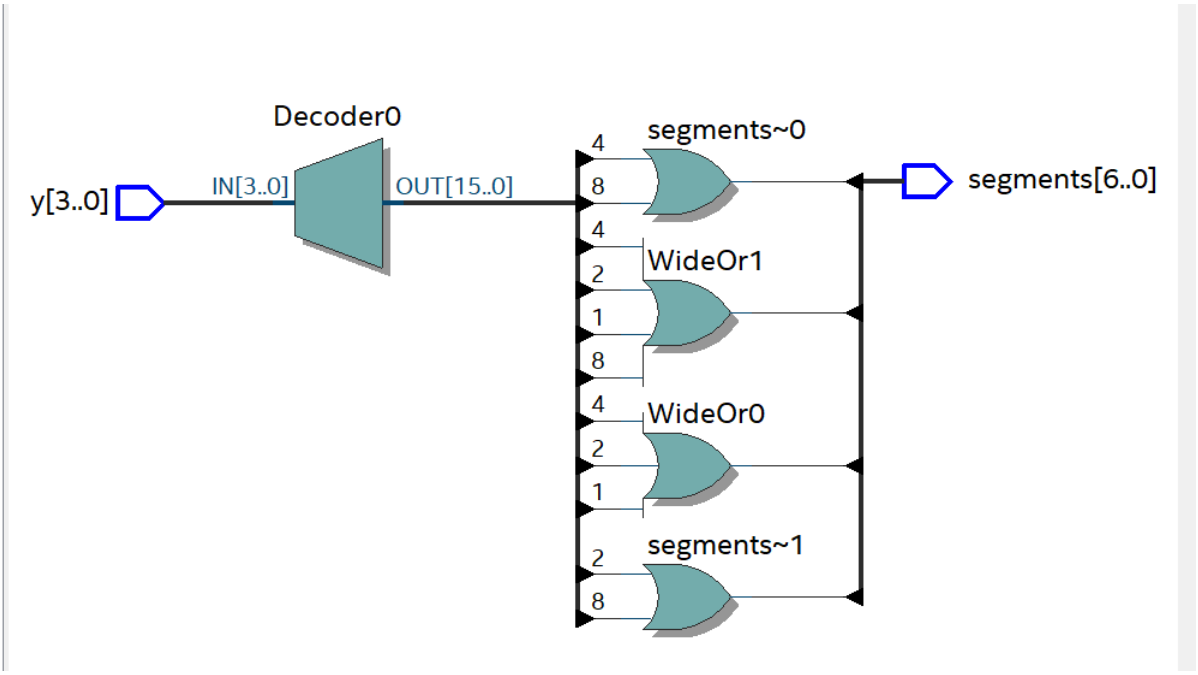
Το κυκλωμα segprm :



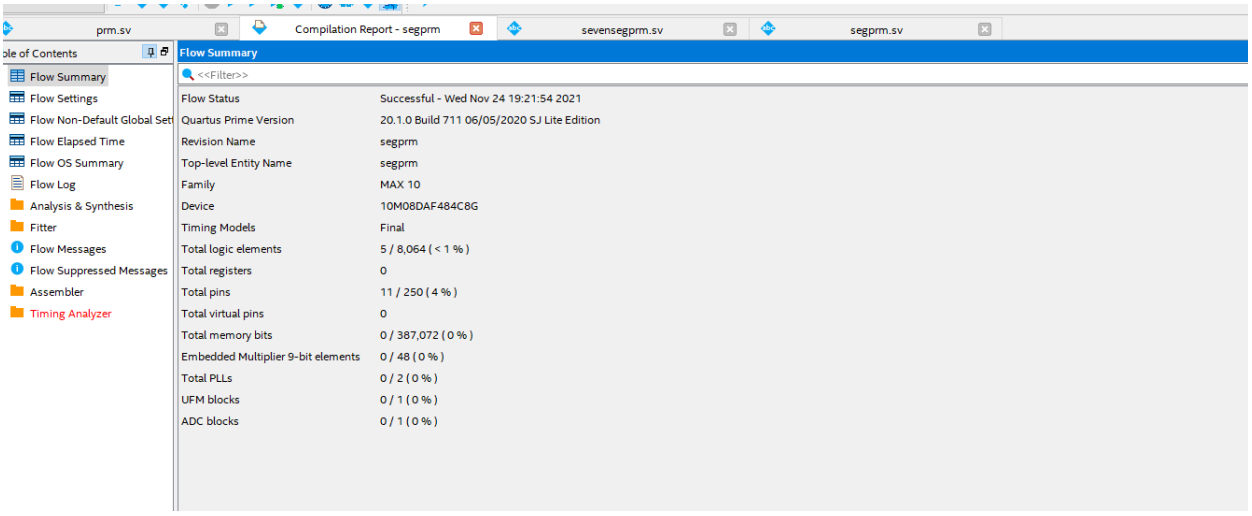
Το κυκλωμα prm :



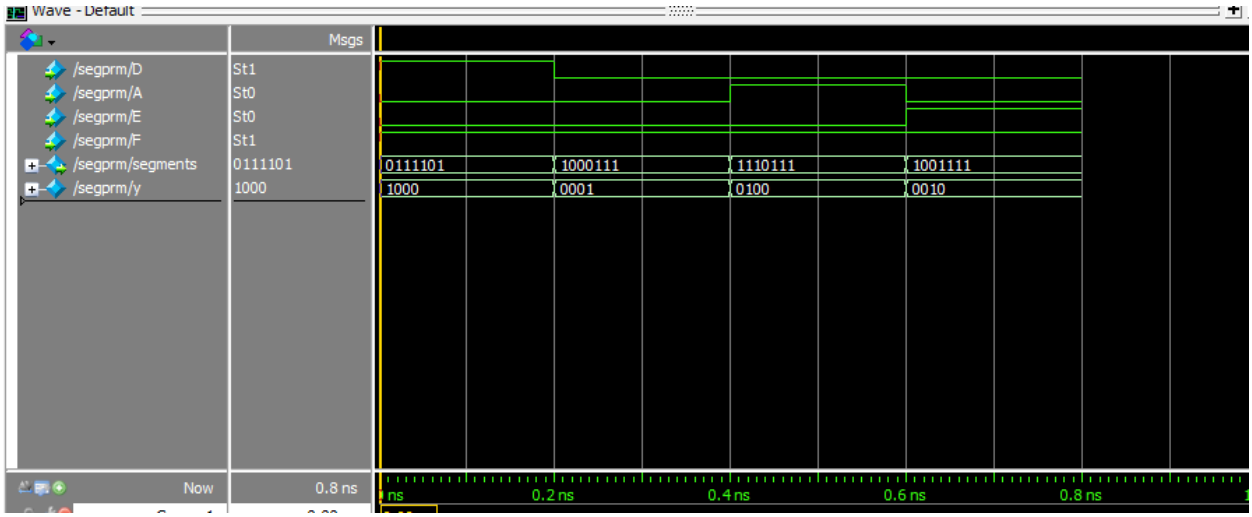
Το κυκλωμα sevensegprm:



Comp report :



Rtl simulation :



Ο κωδικας do :

```
vsim segprm
add wave *
force D 1
force F 1
force A 0
force E 0
run 200
force D 0
run 200
force A 1
run 200
force A 0
force E 1
run 200
```

Άσκηση 6 :

Ο κωδικας για το fa1bit :

```
module fa1bit(input logic ci,a,b , output logic s,co);
	logic x;
	xor(x,a,b);

	xor(s,ci,x);

	assign co= x ? ci : b;
endmodule
```

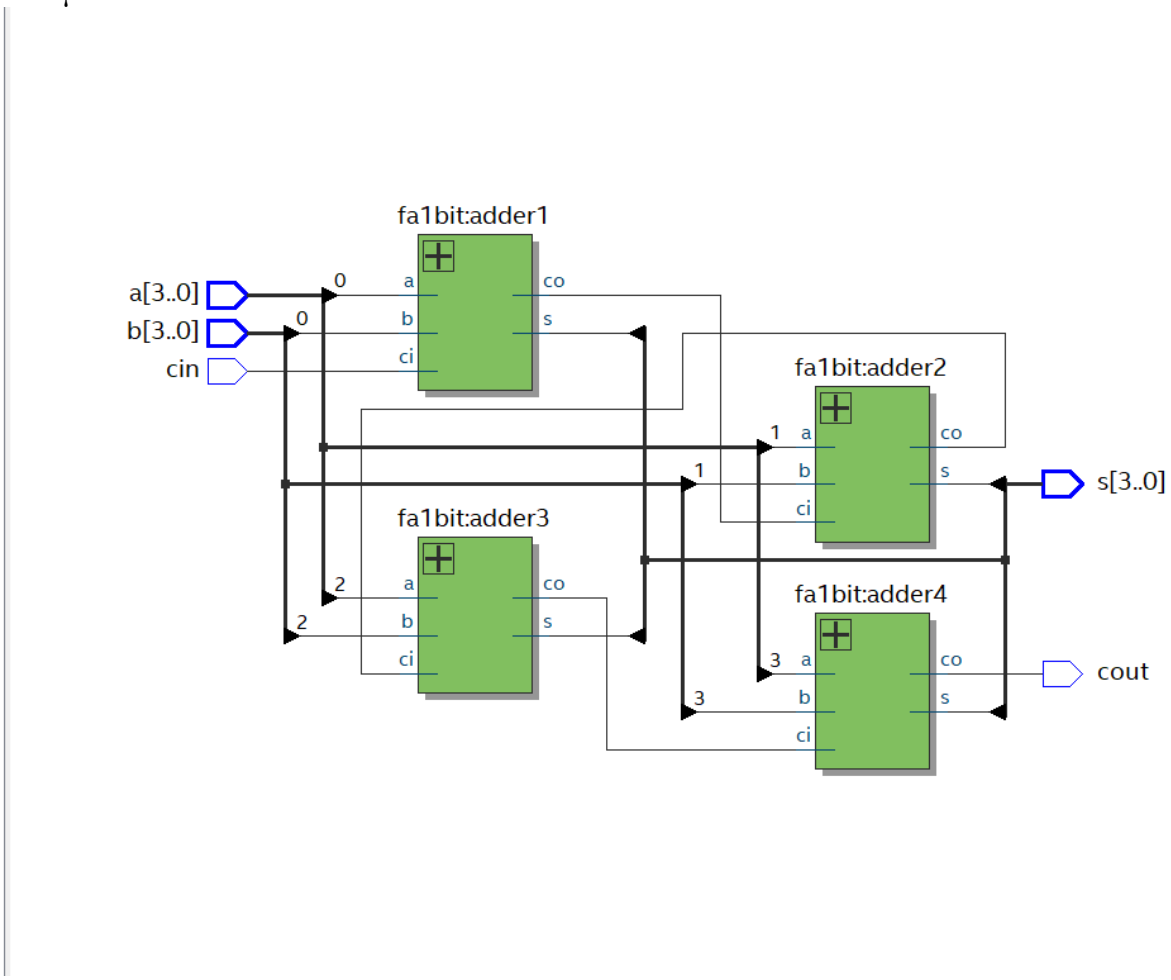
Ο κωδικας για το farc4bit :

```
module farc4bit (input logic [3:0] a , b,input logic cin , output logic [3:0] s , output logic cout);

	logic c1,c2,c3;
	fa1bit adder1 (cin,a[0],b[0],s[0],c1);
	fa1bit adder2 (c1,a[1],b[1],s[1],c2);
	fa1bit adder3 (c2,a[2],b[2],s[2],c3);
	fa1bit adder4 (c3,a[3],b[3],s[3],cout);
endmodule
```

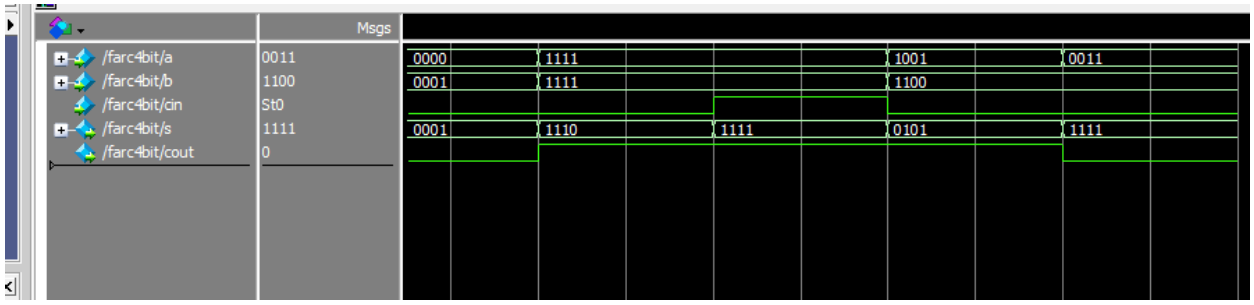
Ο κωδικας do :

```
vsim farc4bit
add wave *
force a 0000
force b 0001
force cin 0
run 100
force a 1111
force b 1111
run 100
force cin 1
run 100
force cin 0
force a 1001
force b 1100
run 100
force a 0011
run 100
```

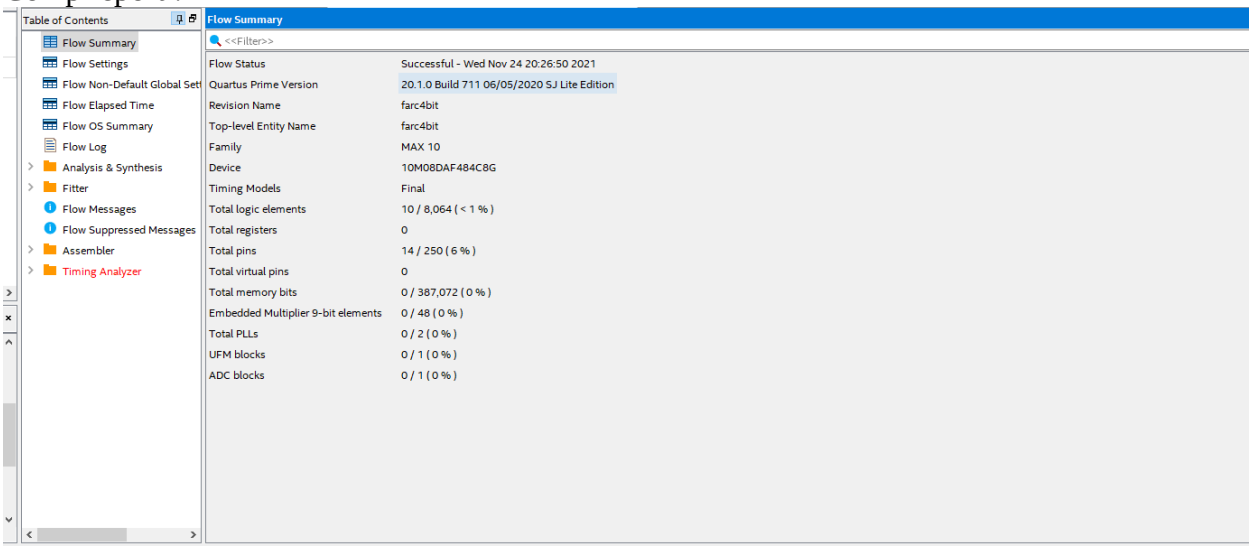
Κυκλωμα :



Simulation :



Comp report :



Άσκηση 7 :

Ο κωδικας sv για τον alunbit :

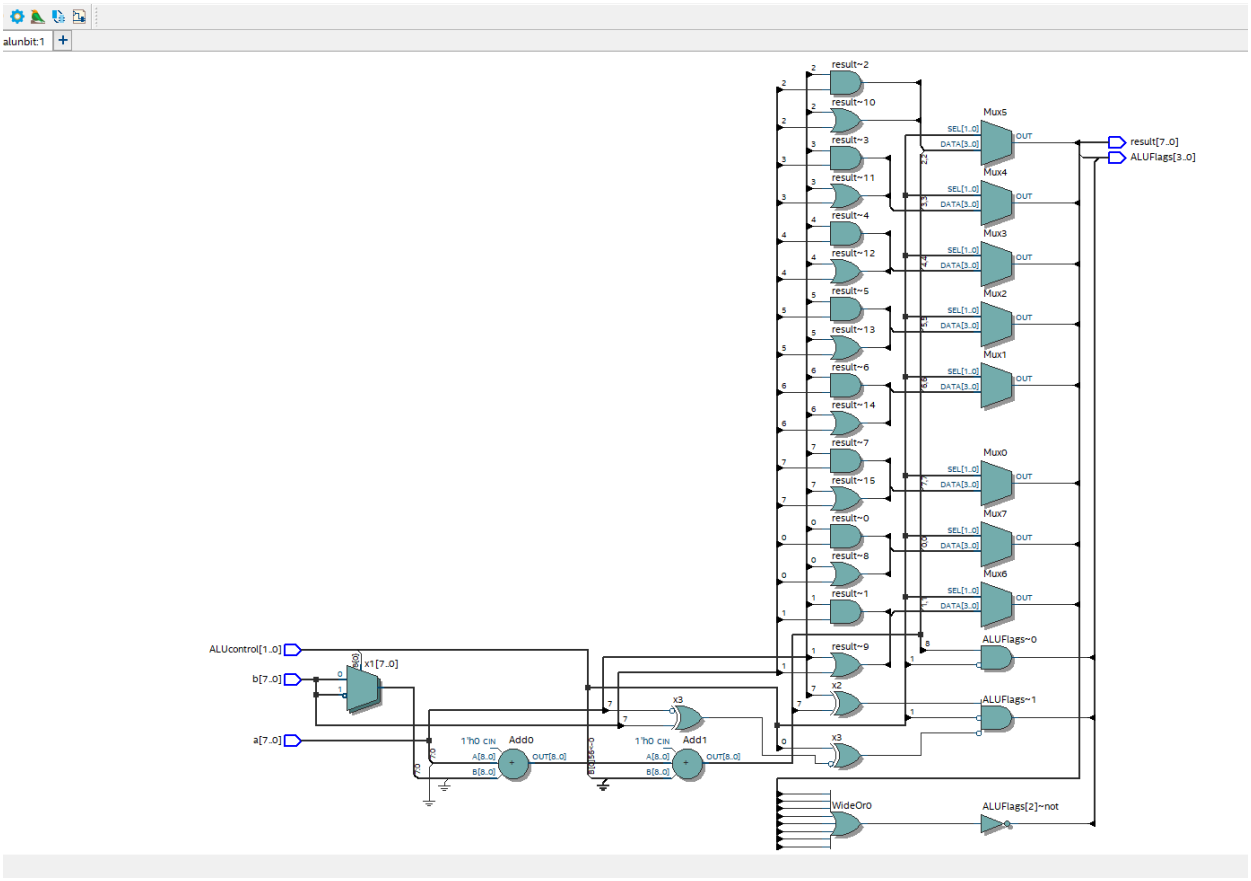
```
module alunbit
        #(parameter n = 8)
        (input logic [n-1:0] a,b,input logic [1:0]ALUcontrol,output logic [n-1:0] result ,output logic [3:0] ALUFlags);
        logic [n-1:0] x1,sum;
        logic cout,x2,x3;
        assign x1 = ALUcontrol[0] ? ~b : b;
        assign {cout,sum} = a + x1 + ALUcontrol[0];
        always_comb
        casex   (ALUcontrol[1:0])
                2'b0?: result = sum;
                2'b10: result = a&b;
                2'b11: result = a|b;
        endcase
        assign ALUFlags[3] = result[n-1];

        assign ALUFlags[2]= ~(|result);

        assign ALUFlags[1] = (~ALUcontrol[1] &  cout);
        assign x2 = sum[n-1]^a[n-1];
        assign x3 = ~a[n-1]~^b[n-1]~^ALUcontrol[0];
        assign ALUFlags[0]= x2&x3& ~ALUcontrol[1];


endmodule
```

Ο κωδικας do :

```
vsim alunbit
add wave *
force a 00000000
force b 00000000
force ALUcontrol 00
run 100
force a 00000000
force b 11111111
run 100
force a 00000001
run 100
force a 11111111
force b 00000001
run 100
force ALUcontrol 01
force a 00000000
force b 00000000
run 100
force b 11111111
run 100
force a 00000001
force b 00000001
run 100
force a 10000000
force b 00000001
run 100
force ALUcontrol 10
force a 11111111
force b 11111111
run 100
force b 01111000
run 100
force a 01111000
force b 00100001
run 100
force a 00000000
force b 11111111
run 100
force ALUcontrol 11
force a 11111111
force b 11111111
run 100
force a 01111000
force b 00100001
run 100
force a 00000000
force b 11111111
run 100
force b 00000000
```

run 100

Κυκλωμα :

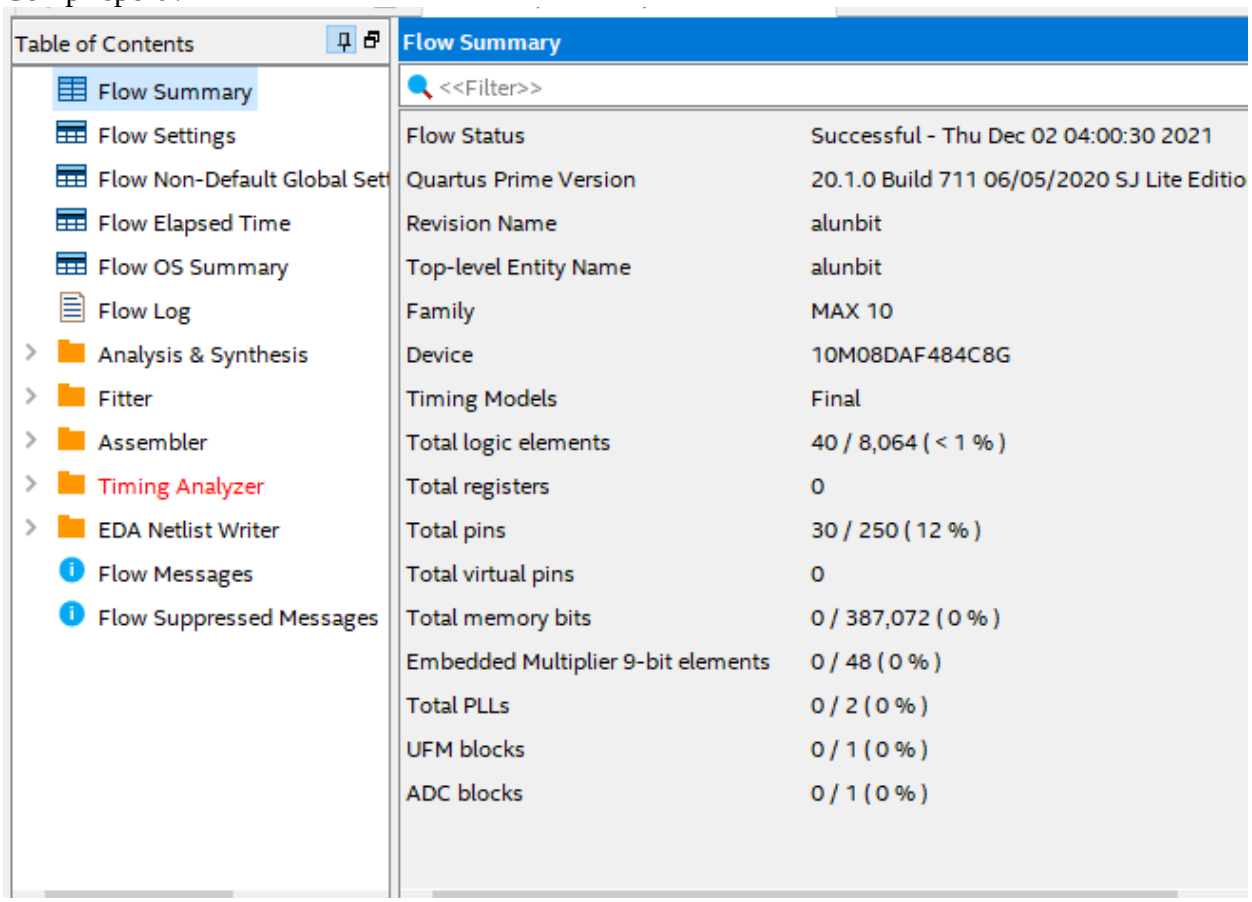


Πινακας :

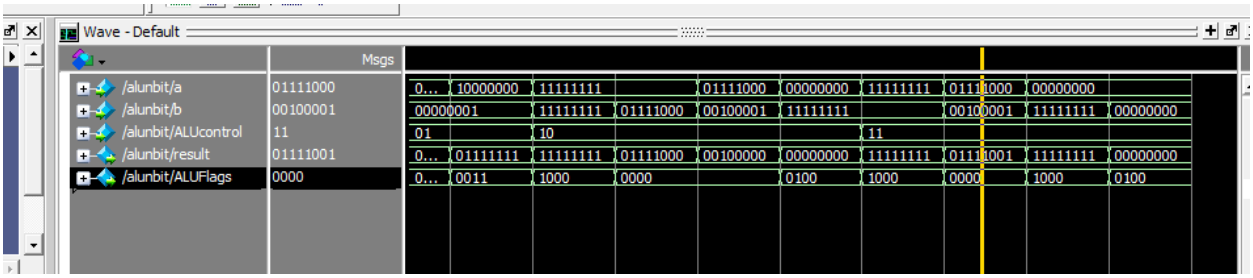
Πίνακας 1. Πράξεις ALU για N=8 (οι αριθμοί είναι hex).

Test	ALUControl[1:0]	A	B	Result	ALUFlags (NZCV)
ADD 0+0	0	00	00	00	4
ADD 0+(-1)	0	00	FF	FF	8
ADD 1+(-1)	0	01	FF	00	6
ADD FF+1	0	FF	01	00	6
SUB 0-0	1	00	00	00	6
SUB 0-(-1)	1	00	FF	01	0
SUB 1-1	1	01	01	00	6
SUB 80-1	1	80	01	7F	3
AND FF, FF	2	FF	FF	FF	8
AND FF, 78	2	FF	78	78	0
AND 78, 21	2	78	21	20	0
AND 00, FF	2	00	FF	00	4
OR FF, FF	3	FF	FF	FF	8
OR 78, 21	3	78	21	79	0
OR 00, FF	3	00	FF	FF	8
OR 00, 00	3	00	00	00	4

Comp report :



Simulation :



Άσκηση 8 :

Κωδικας :

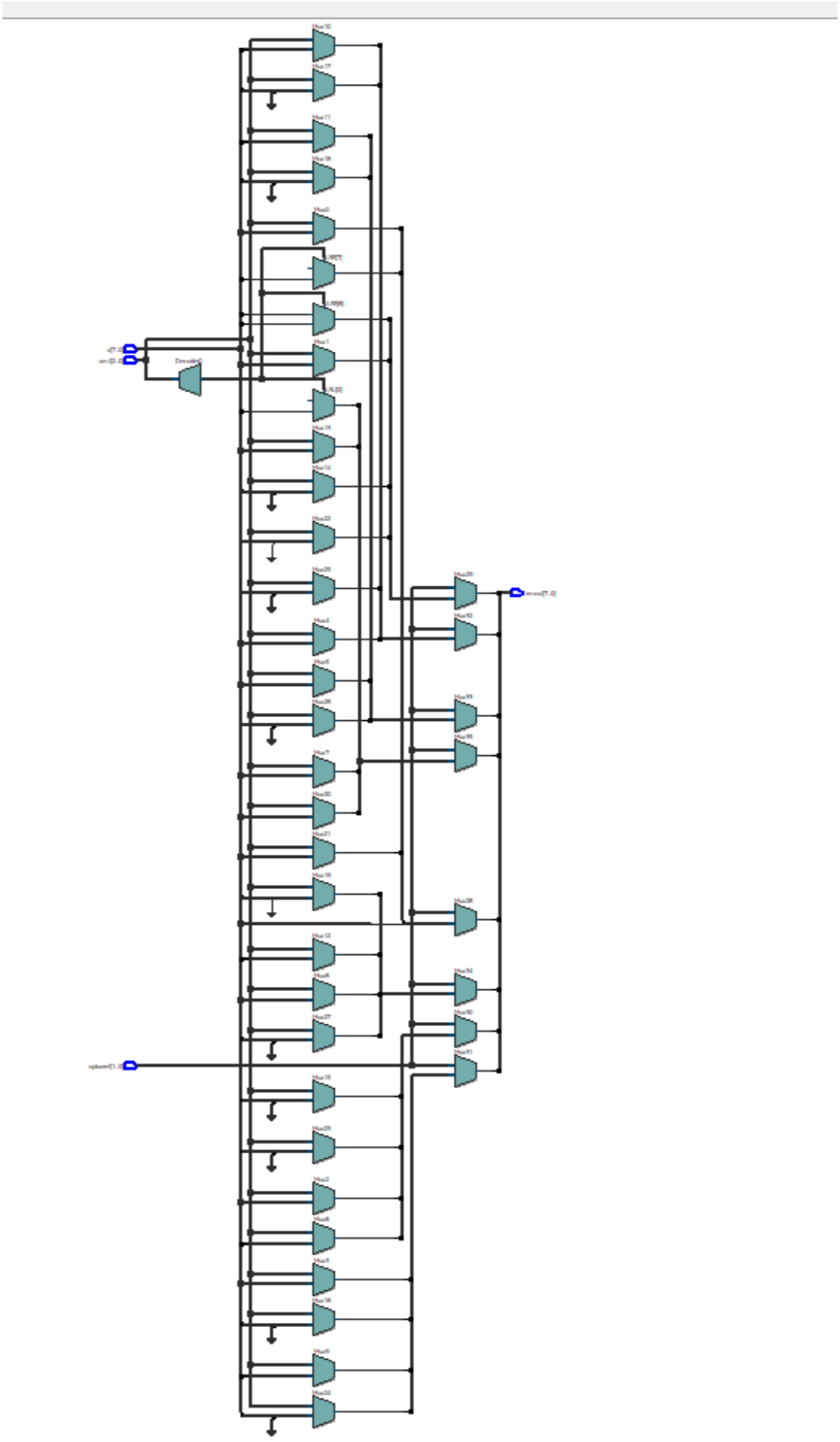
```
module Barrel8bit ( input logic [2:0] amt, input logic [7:0] a ,input logic [1:0] opbarrel ,output logic [7:0] rmout);
	logic [7:0] LSL,LSR,ASR,ROR;
	// ROR
	always_comb
		case(amt)
			3'b000:ROR=a;
			3'b001:ROR={a[0],a[7],a[6],a[5],a[4],a[3],a[2],a[1]};
			3'b010:ROR={a[1],a[0],a[7],a[6],a[5],a[4],a[3],a[2]};
			3'b011:ROR={a[2],a[1],a[0],a[7],a[6],a[5],a[4],a[3]};
			3'b100:ROR={a[3],a[2],a[1],a[0],a[7],a[6],a[5],a[4]};
			3'b101:ROR={a[4],a[3],a[2],a[1],a[0],a[7],a[6],a[5]};
			3'b110:ROR={a[5],a[4],a[3],a[2],a[1],a[0],a[7],a[6]};
			3'b111:ROR={a[6],a[5],a[4],a[3],a[2],a[1],a[0],a[7]};
			default:ROR=8'bxxxxxxxx;
			endcase
	//ASR
	always_comb
		case(amt)
			3'b000:ASR=a;
			3'b001:ASR={a[7],a[7],a[6],a[5],a[4],a[3],a[2],a[1]};
			3'b010:ASR={a[7],a[7],a[7],a[6],a[5],a[4],a[3],a[2]};
			3'b011:ASR={a[7],a[7],a[7],a[7],a[6],a[5],a[4],a[3]};
			3'b100:ASR={a[7],a[7],a[7],a[7],a[7],a[6],a[5],a[4]};
			3'b101:ASR={a[7],a[7],a[7],a[7],a[7],a[7],a[6],a[5]};
			3'b110:ASR={a[7],a[7],a[7],a[7],a[7],a[7],a[7],a[6]};
			3'b111:ASR={a[7],a[7],a[7],a[7],a[7],a[7],a[7],a[7]};
			default:ASR=8'bxxxxxxxx;
			endcase
	//LSR
	always_comb
		case(amt)
			3'b000:LSR=a;
			3'b001:LSR={1'b0,a[7],a[6],a[5],a[4],a[3],a[2],a[1]};
			3'b010:LSR={1'b0,1'b0,a[7],a[6],a[5],a[4],a[3],a[2]};
			3'b011:LSR={1'b0,1'b0,1'b0,a[7],a[6],a[5],a[4],a[3]};
			3'b100:LSR={1'b0,1'b0,1'b0,1'b0,a[7],a[6],a[5],a[4]};
			3'b101:LSR={1'b0,1'b0,1'b0,1'b0,1'b0,a[7],a[6],a[5]};
			3'b110:LSR={1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,a[7],a[6]};
			3'b111:LSR={1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,a[7]};
			default:LSR=8'bxxxxxxxx;
			endcase

		//LSL
	always_comb
		case(amt)
			3'b000:LSL=a;
			3'b001:LSL={a[6],a[5],a[4],a[3],a[2],a[1],a[0],1'b0};
```

```
                3'b010:LSL={a[5],a[4],a[3],a[2],a[1],a[0],1'b0,1'b0};
                3'b011:LSL={a[4],a[3],a[2],a[1],a[0],1'b0,1'b0,1'b0};
                3'b100:LSL={a[3],a[2],a[1],a[0],1'b0,1'b0,1'b0,1'b0};
                3'b101:LSL={a[2],a[1],a[0],1'b0,1'b0,1'b0,1'b0,1'b0};
                3'b110:LSL={a[1],a[0],1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
                3'b111:LSL={a[0],1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
                default:LSL=8'bxxxxxxxx;
                endcase

    //barrel
    always_comb
        case(opbarrel)
            2'b00:rmout=LSR;
            2'b01:rmout=LSL;
            2'b11:rmout=ASR;
            2'b10:rmout=ROR;
        endcase


    endmodule
```

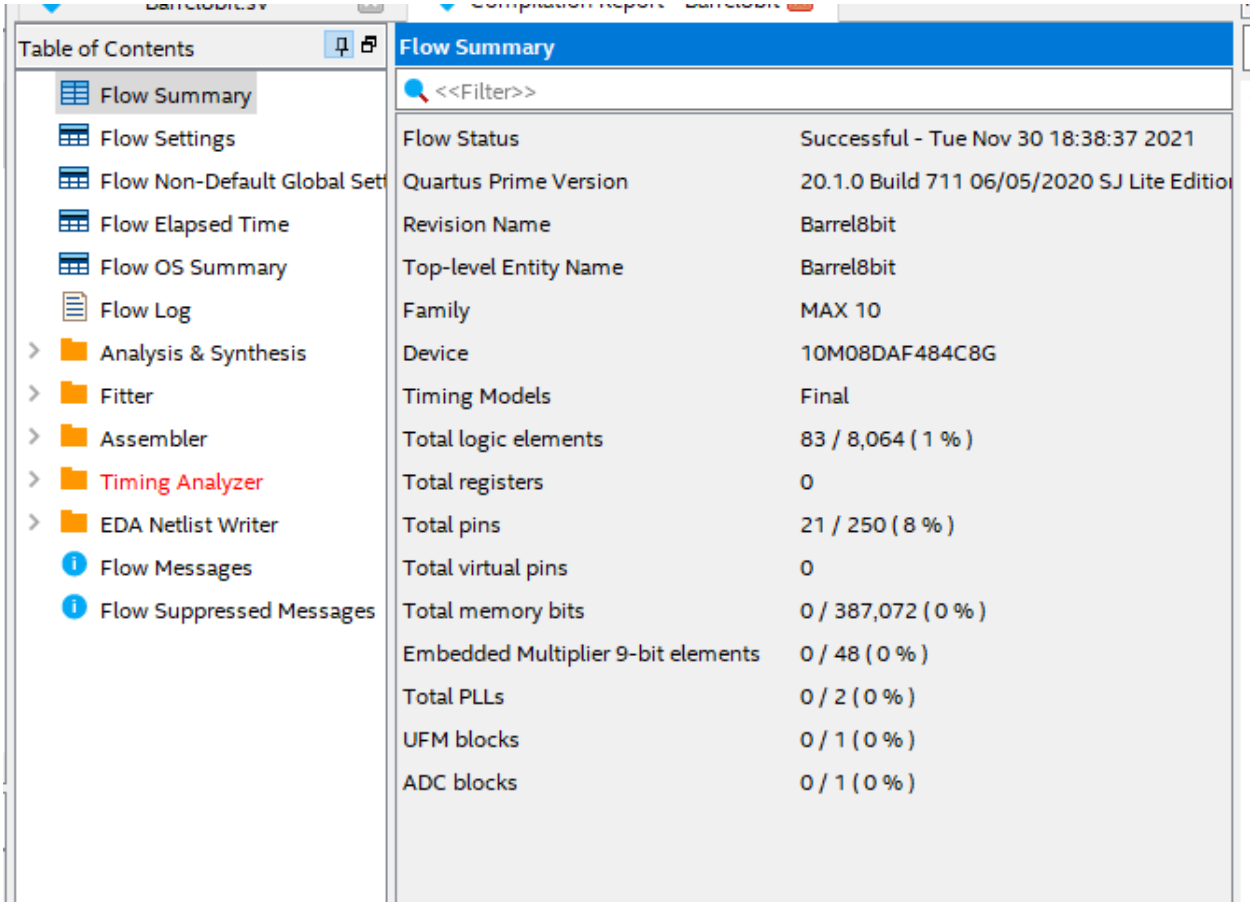
Ο κωδικας do :

```
vsim Barrel8bit
add wave *
force amt 010
force a 11001100
force opbarrel 00
run 100
force opbarrel 01
run 100
force opbarrel 10
run 100
force opbarrel 11
run 100
```

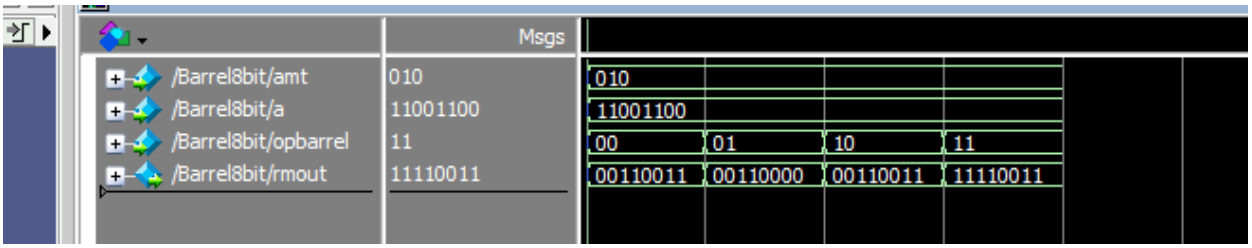
Κυκλωμα :



Comp report :



Simulation :



Άσκηση 9 :

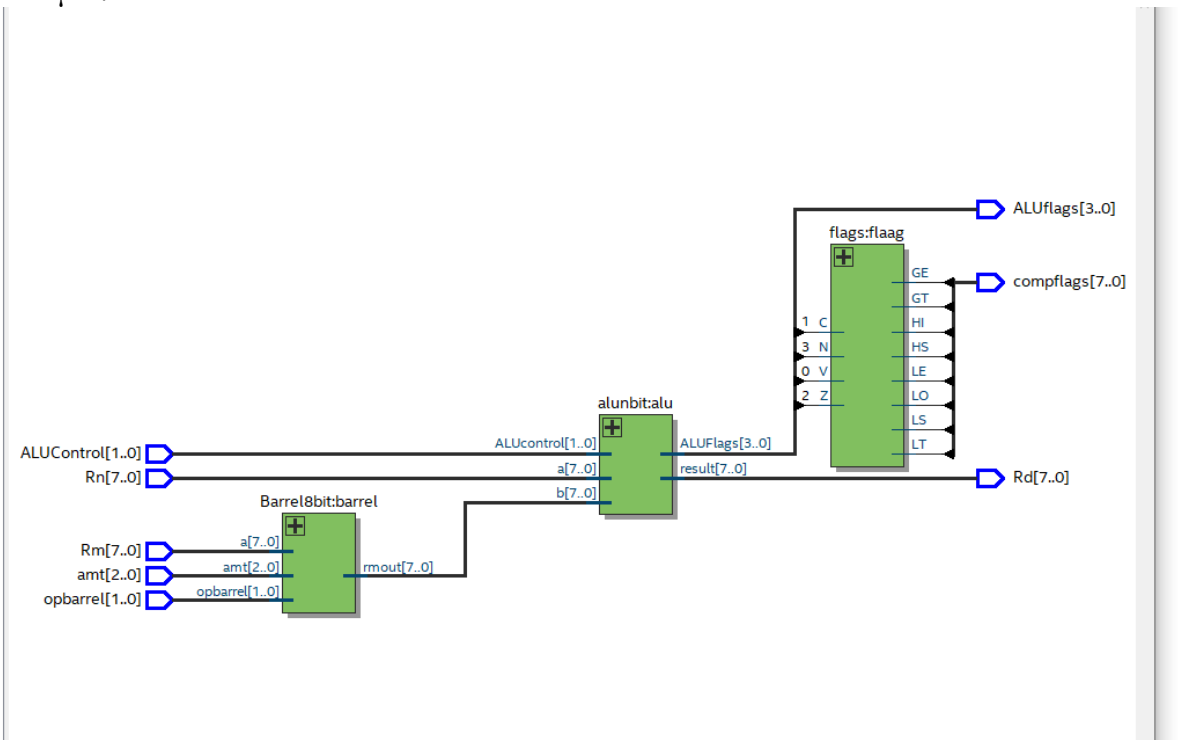
Ο κωδικας :

```
module ALUBarrel
#(parameter n = 8)

(input logic [n-1:0] Rn,Rm,input logic [2:0] amt,input logic [1:0] opbarrel,ALUControl,output logic [n-1:0] Rd,output logic [3:0] ALUflags,output logic [7:0] compflags);
logic [n-1:0] barrelx;
logic  [3:0] flaggs;
logic HS,LS,HI,LO,GE,LE,GT,LT;
Barrel8bit barrel(amt,Rm,opbarrel,barrelx);
alunbit alu(Rn,barrelx,ALUControl,Rd,flaggs);
assign ALUflags= flaggs;
flags flaag (flaggs[1],flaggs[3],flaggs[0],flaggs[2],HS,LS,HI,LO,GE,LE,GT,LT);
assign compflags[7:0]={HS,LS,HI,LO,GE,LE,GT,LT};
endmodule
```

Κωδικας do :

```
vsim ALUBarrel
add wave *
force Rn 00000001
force Rm 00000001
force amt 000
force opbarrel 00
force ALUControl 00
run 100
force Rm 10010011
force Rn 00110011
force ALUControl 01
run 100
force Rn 00011101
force Rm 00001001
force amt 100
force opbarrel 01
run 100
force amt 101
run 100
force ALUControl 00
run 100
force amt 100
run 100
```

Κυκλωμα :



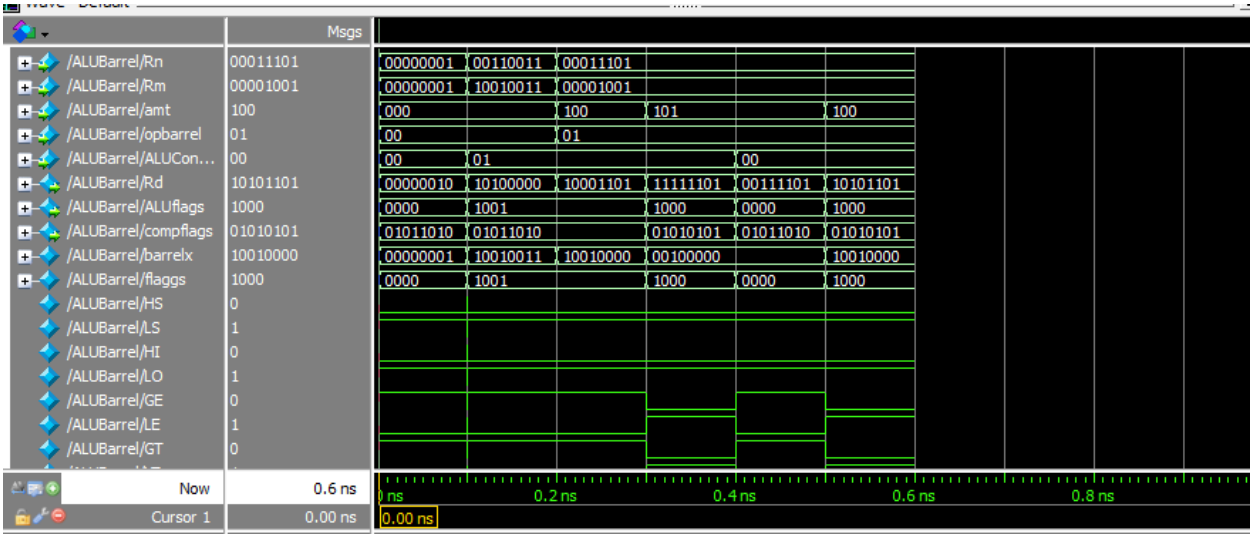
Comp report :

Flow Summary

<<Filter>>

Flow Status	Successful - Thu Dec 02 18:03:05 2021
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Editio
Revision Name	ALUBarrel
Top-level Entity Name	ALUBarrel
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	125 / 8,064 (2 %)
Total registers	0
Total pins	43 / 250 (17 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 2 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

Simulation :


Msgs
/ALUBarrel/Rn
/ALUBarrel/Rm
/ALUBarrel/amt
/ALUBarrel/opbarrel
/ALUBarrel/ALUCon...
/ALUBarrel/Rd
/ALUBarrel/ALUflags
/ALUBarrel/compflags
/ALUBarrel/barrelx
/ALUBarrel/flaggs
/ALUBarrel/HS
/ALUBarrel/LS
/ALUBarrel/HI
/ALUBarrel/LO
/ALUBarrel/GE
/ALUBarrel/LE
/ALUBarrel/GT
00011101
00001001
100
01
00
10101101
1000
01010101
10010000
1000
0
1
0
1
0
1
0
Now
0.6 ns
Cursor 1
0.00 ns
0.2 ns
0.4 ns
0.6 ns
0.8 ns