

CAD-Αναφορά Εργαστηρίου 2022

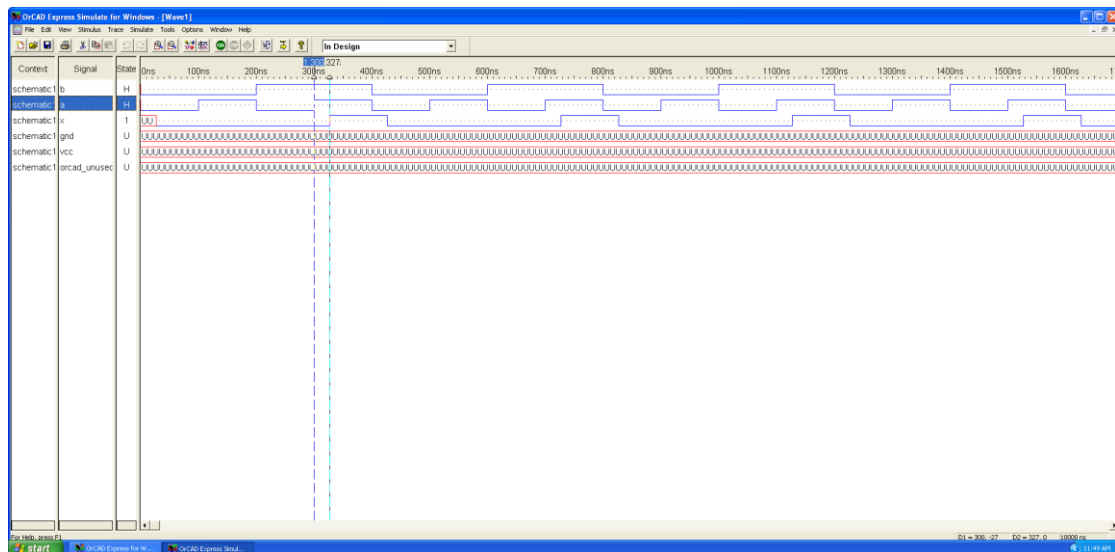
Γκίζας Μαρίνος

AM: 1054348

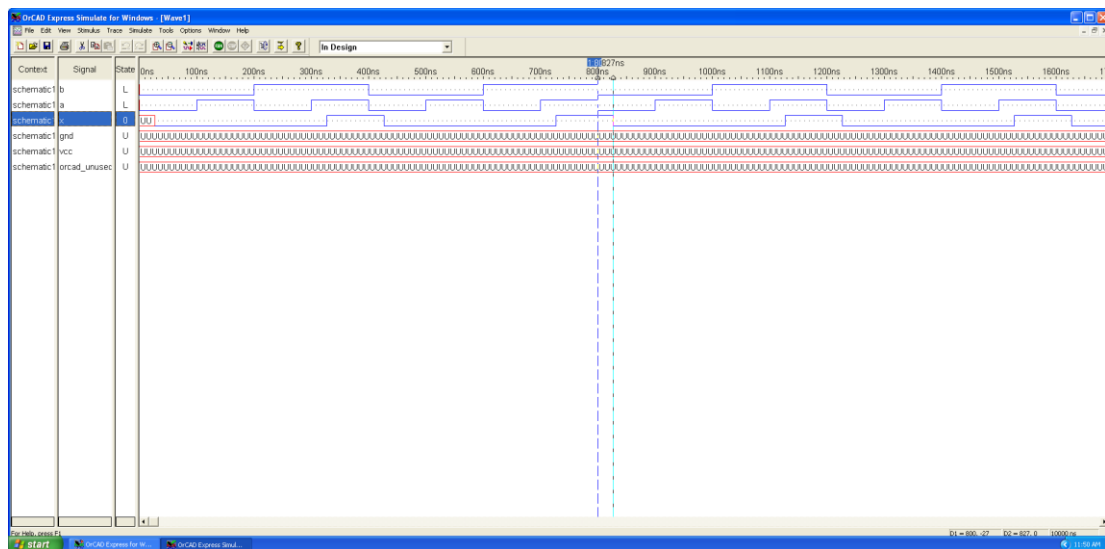
Άσκηση 1

	7408	74ALS08	74AS08	74S08	74LS08
Χρόνος καθυστέρησης μετάβασης στο λογικό 0	27ns	9ns	4ns	8ns	13ns
Χρόνος καθυστέρησης μετάβασης στο λογικό 1	27ns	9ns	4ns	8ns	13ns

1.Παράδειγμα μέτρησης καθυστέρησης για την 7408(0->1)



2. Παράδειγμα μέτρησης καθυστέρησης για την 7408(1->0)

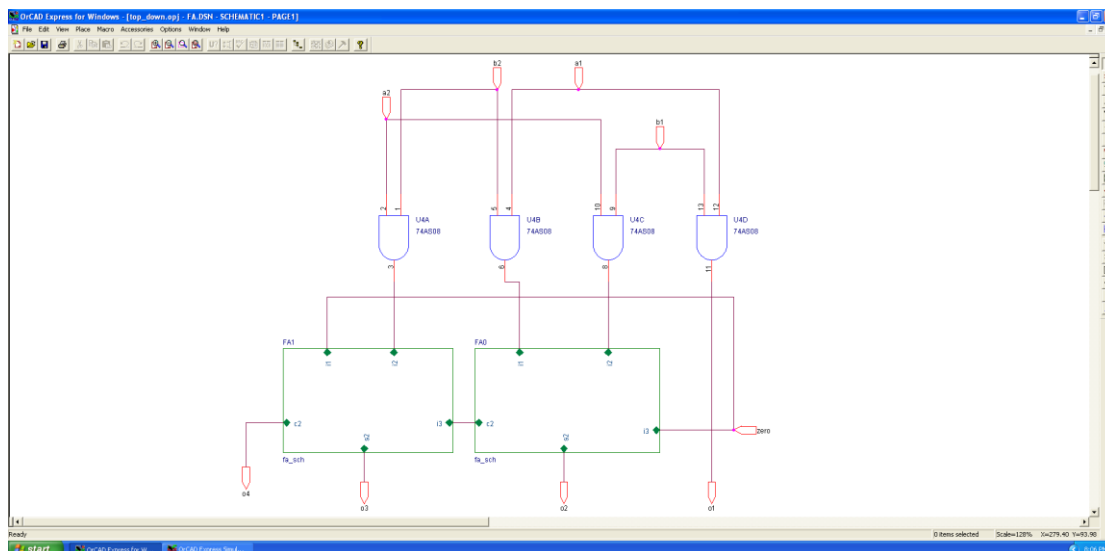


Ομοίως γίνεται η μέτρηση και για τις υπόλοιπες πύλες.

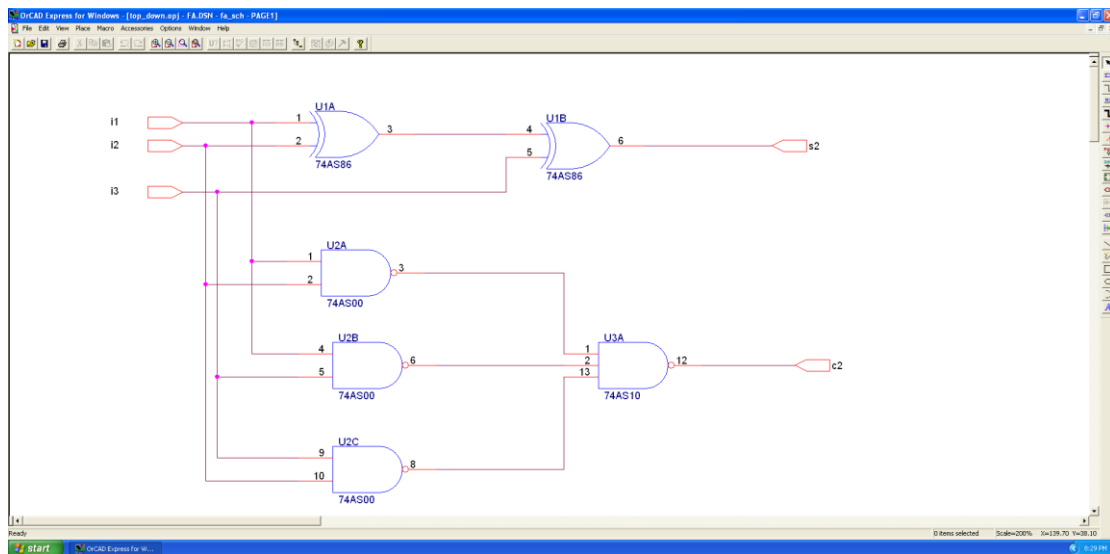
Άσκηση 2

Για την δεύτερη άσκηση θα σχεδιάσουμε, αρχικά, έναν carry-save πολλαπλασιαστή 2 bits με την top-down φιλοσοφία, ενώ έπειτα θα σχεδιάσουμε με την φιλοσοφία bottom-up έναν καταχωρητή 2 bits.

3.Top-Down Design



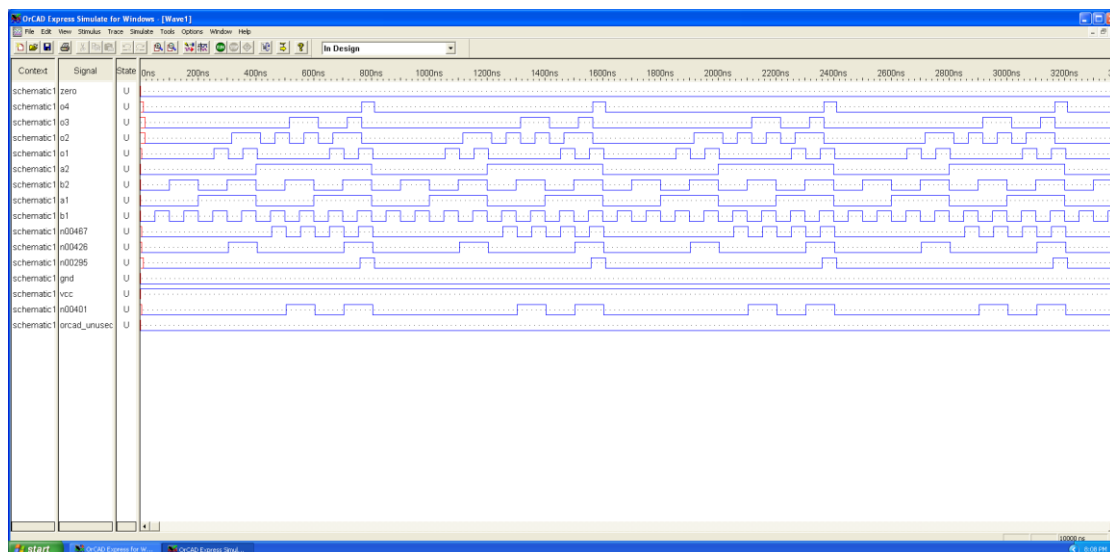
4.Full Adder



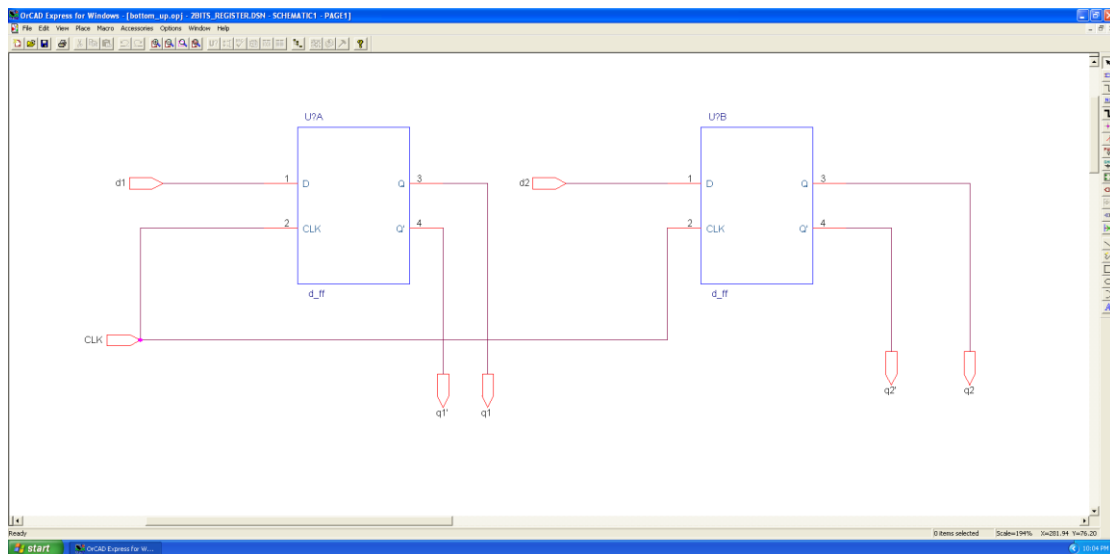
5.Stimulus

Absolute	Relative
schematic1.vcc	schematic1.a2
schematic1.gnd	schematic1.a1
	schematic1.b2
	schematic1.b1
	schematic1.zero
	schematic1.orcad_unused

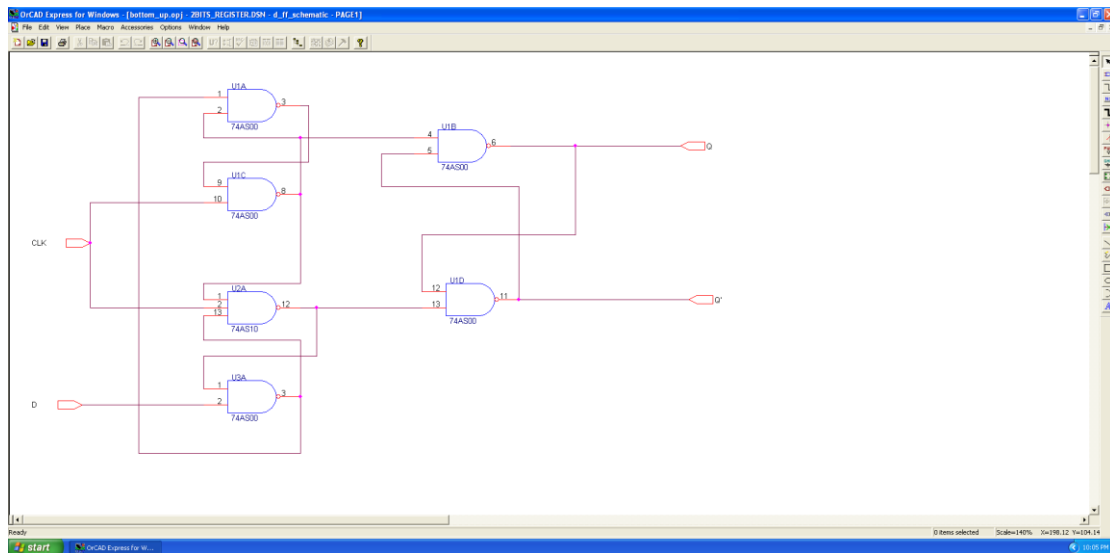
6.Waveforms



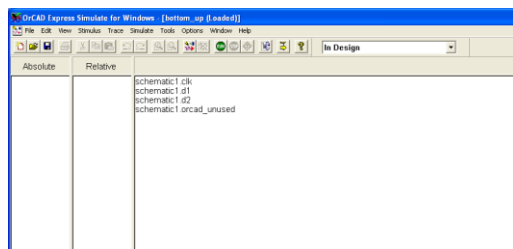
7. Bottom-Up Design



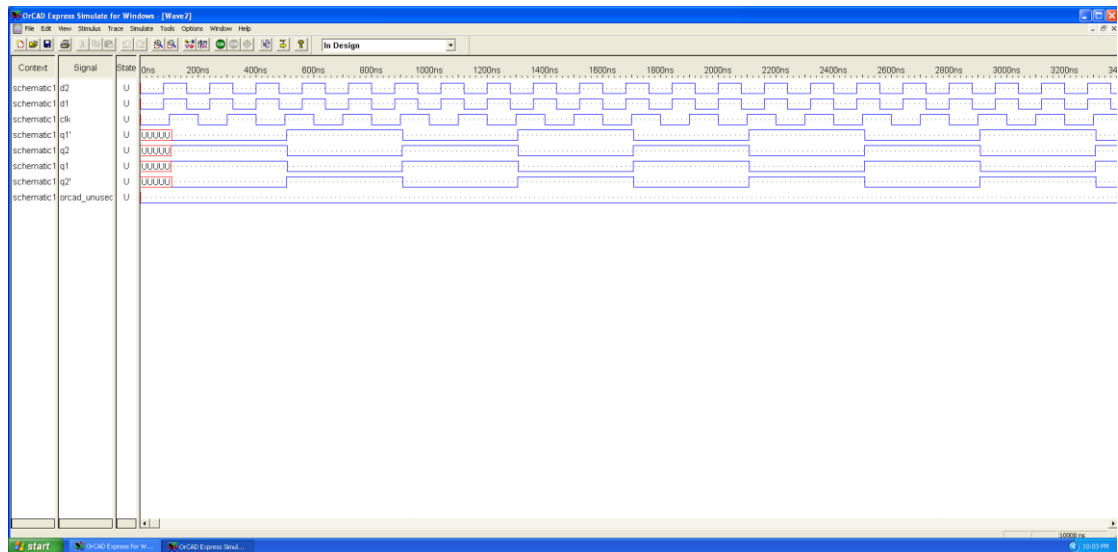
8. D FF



9. Stimulus

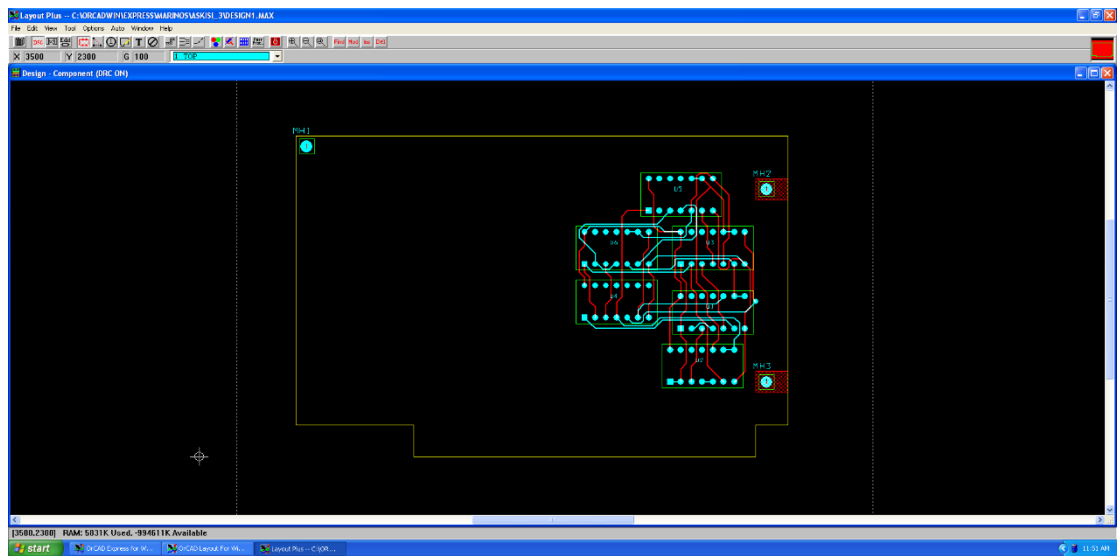


10.Waveforms

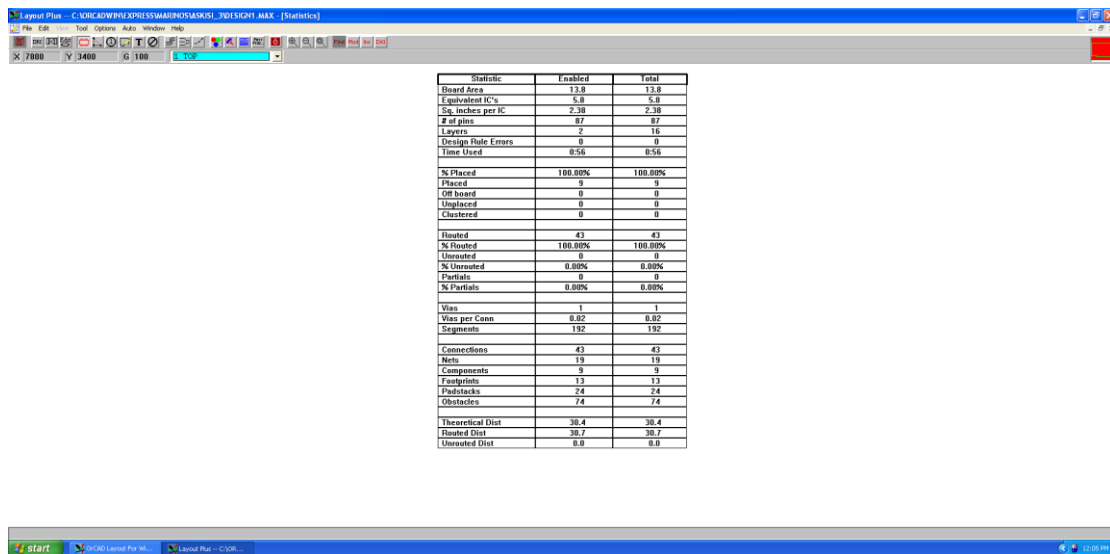


Άσκηση 3

11.Διαδρομηση δύο επιπέδων(DSIGN1.MAX)



12.Στατιστικά

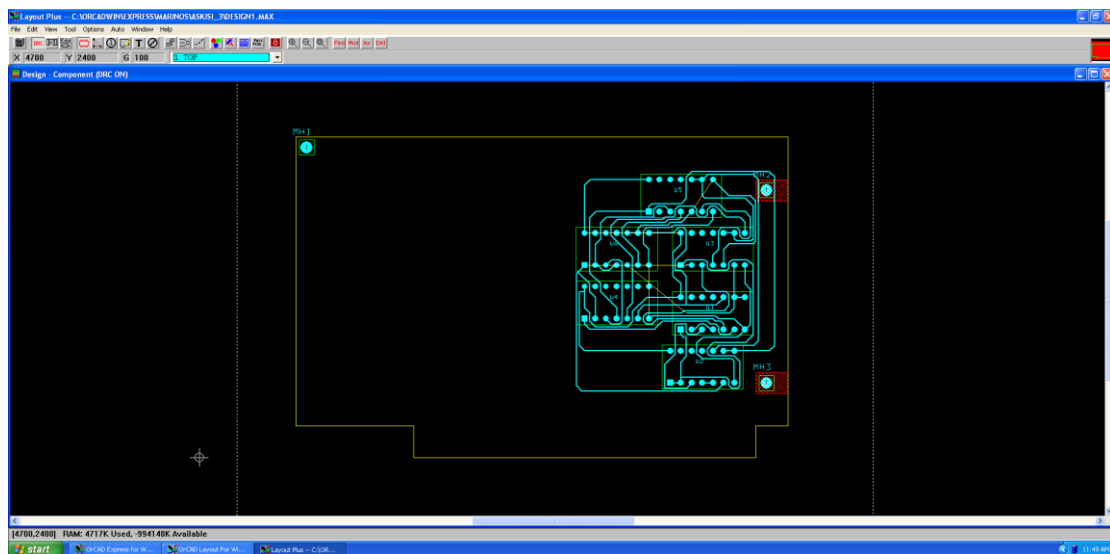


Statistic	Enabled	Total
Board Area	13.8	13.8
Equivalent IC's	5.8	5.8
Sig. Inches per IC	2.38	2.38
# of pins	87	87
Layers	2	16
Design Rule Errors	0	0
Time Used	0:56	0:56
% Placed	100.00%	100.00%
Placed	9	9
Off board	0	0
Unplaced	0	0
Channeled	0	0
Routed	43	43
% Routed	100.00%	100.00%
Unrouted	0	0
% Unrouted	0.00%	0.00%
Partials	0	0
% Partials	0.00%	0.00%
Vias	1	1
Vias per Conn	0.02	0.02
Segments	192	192
Connections	43	43
Net	19	19
Components	9	9
Footprints	13	13
Pads/Tracks	74	74
Obstacles	74	74
Theoretical Dist	38.4	38.4
Routed Dist	38.7	38.7
Unrouted Dist	0.0	0.0

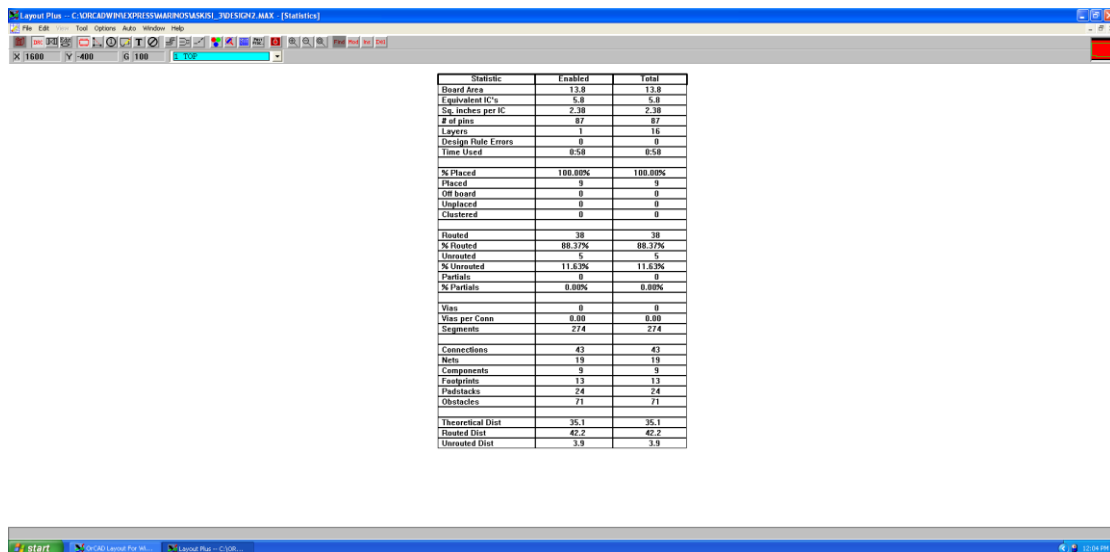
Παρατηρήσεις:

Παρατηρούμε ότι η διαδρομηση με τα δύο επίπεδα είναι πλήρης αφού το %Routed είναι 100%.

13.Διαδρομηση ενός επιπέδου(DSIGN2.MAX)



14.Στατιστικά



Statistic	Enabled	Total
Board Area	13.8	13.8
Equivalent IC's	5.8	5.8
Sq. Inches per IC	2.38	2.38
# of pins	87	87
Layers	1	16
Design Rule Errors	0	0
Time Used	0:58	0:58
% Placed	100.00%	100.00%
Placed	9	9
Off board	0	0
Unplaced	0	0
Challenged	0	0
Routed	38	38
% Routed	88.37%	88.37%
Unrouted	5	5
% Unrouted	11.63%	11.63%
Partials	0	0
% Partials	0.00%	0.00%
Vias	0	0
Vias per Conn	0.00	0.00
Segments	274	274
Connections	43	43
Net	19	19
Components	9	9
Footprints	13	13
Pads/stacks	24	24
Obstacles	71	71
Theoretical Dist	35.1	35.1
Routed Dist	42.2	42.2
Unrouted Dist	3.9	3.9

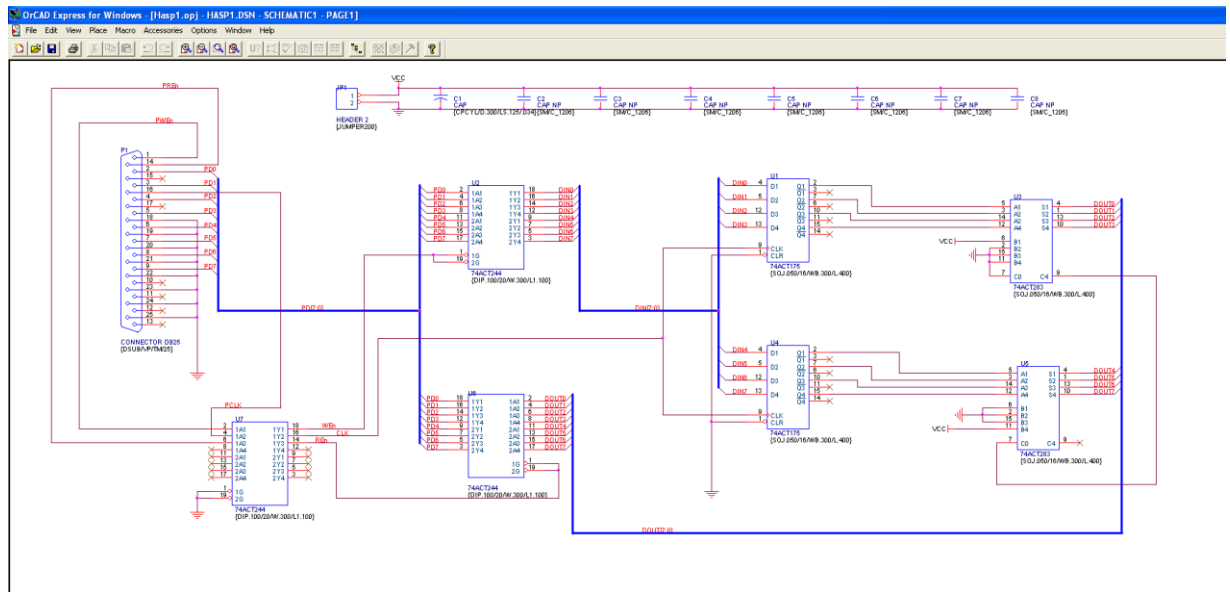
Παρατηρήσεις:

Παρατηρούμε ότι η διαδρομηση δεν είναι πλήρης αφού το %Routed είναι στο 88.37%.

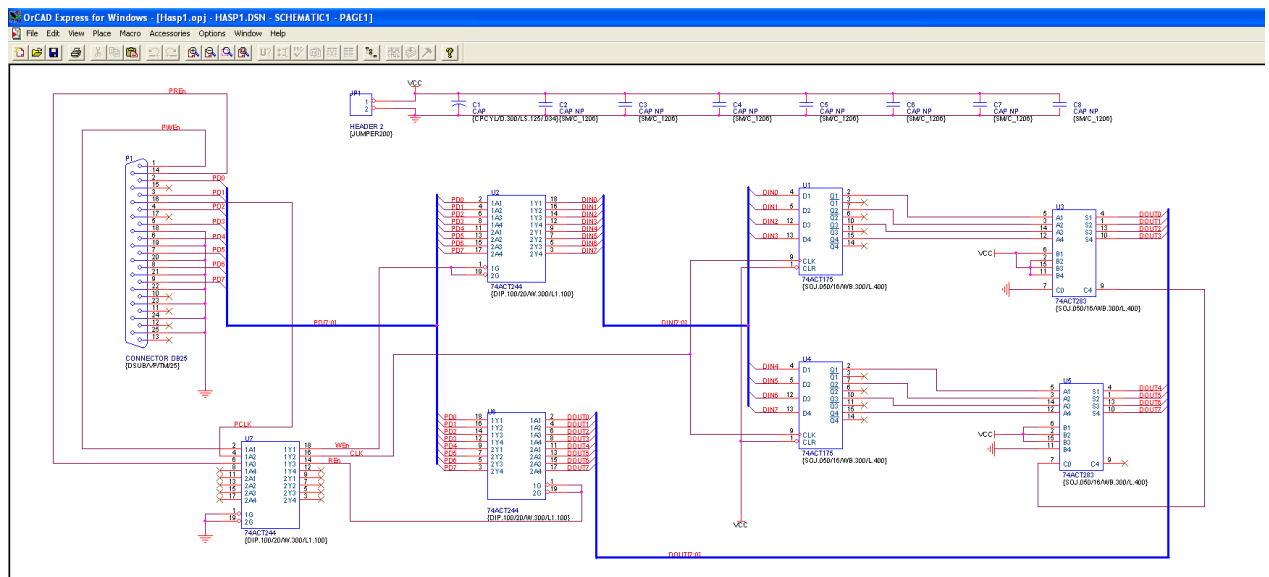
Άσκηση 4

Η καθυστέρηση που επιτυγχάνεται είναι **4ns** από το U7, **2ns** από το U2, **2ns** από το U1 και U4 (φροντίζουμε να έρθει το clk ακριβώς όποτε το χρειαζόμαστε), **16ns** για τον πρώτο adder U3 και άλλα **16ns** για τον δεύτερο adder U5 αφού πρέπει να περιμένουμε να μας έρθει το κρατούμενο από τον U3. Τέλος, έχουμε και άλλα **2ns** από το U6 για να πάρουμε στην έξοδο τα δεδομένα μας. Συνολικά έχουμε **42ns** καθυστέρηση. Συνεπώς η μέγιστη συχνότητα είναι $F_{max}=1/42$ Hz.

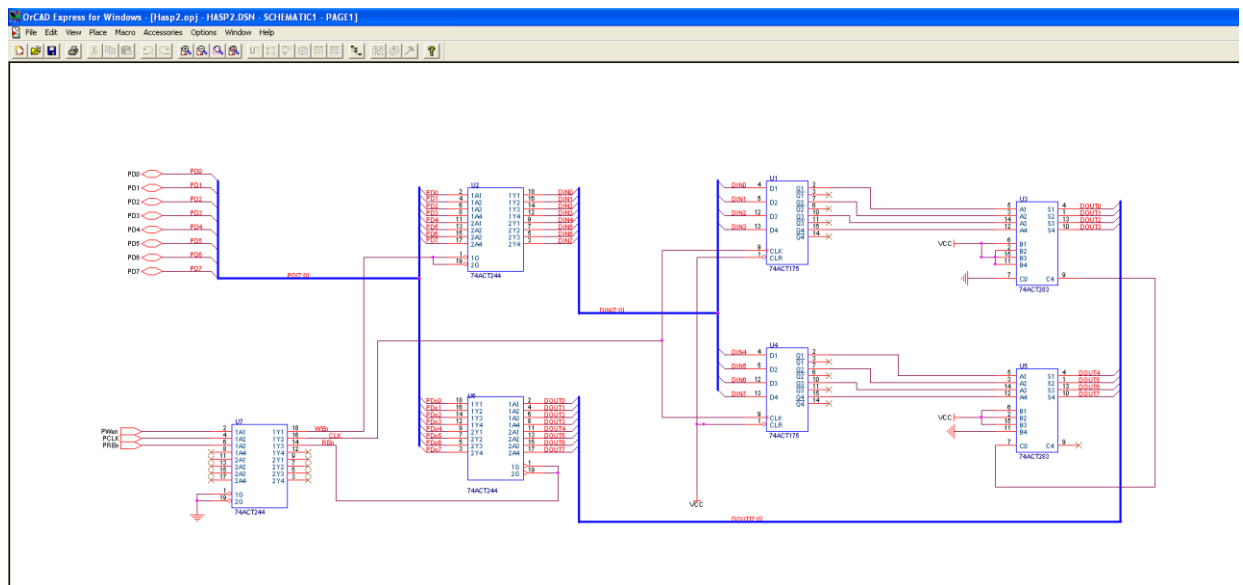
15. Hasp (Χωρίς τις αλλαγές)



16. Hasp (μετά τις αλλαγές)

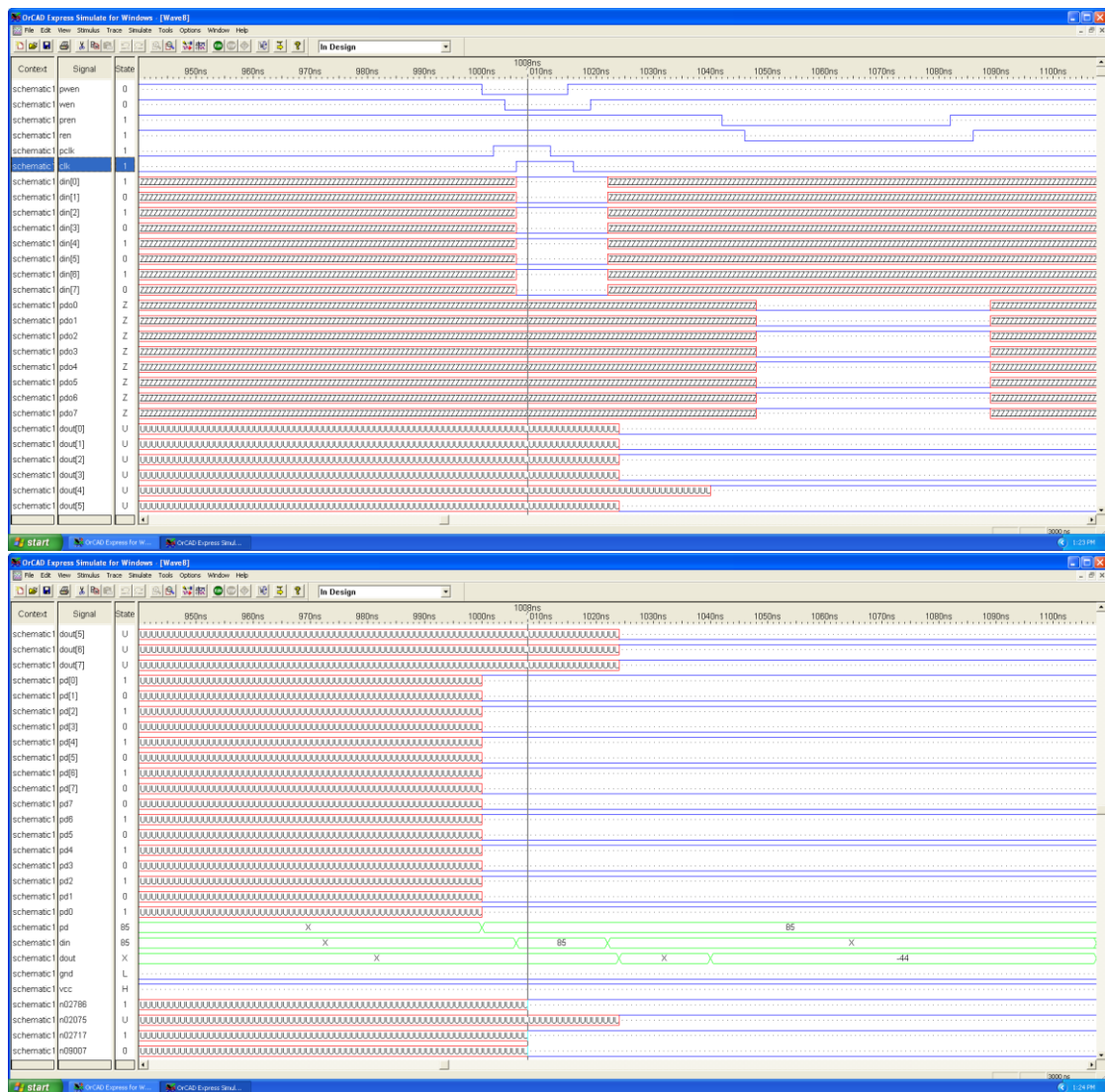


17. Hasp without connector, header and capacitor

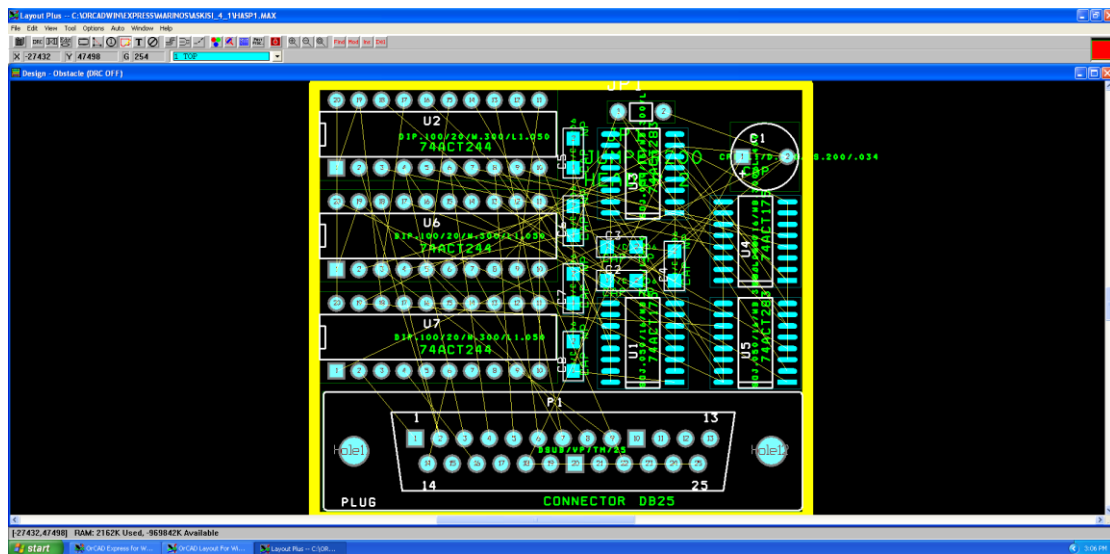


Οι αλλαγές που έγιναν στον δεύτερο σχεδιασμό (vcc αντι για gnd στο clear των FF και αλλαγή στις εισόδους των adder που παίρνουν gnd/vcc) για το simulation έχουν μεταφερθεί και στον πρώτο αφού έγινε η εξομοίωση.

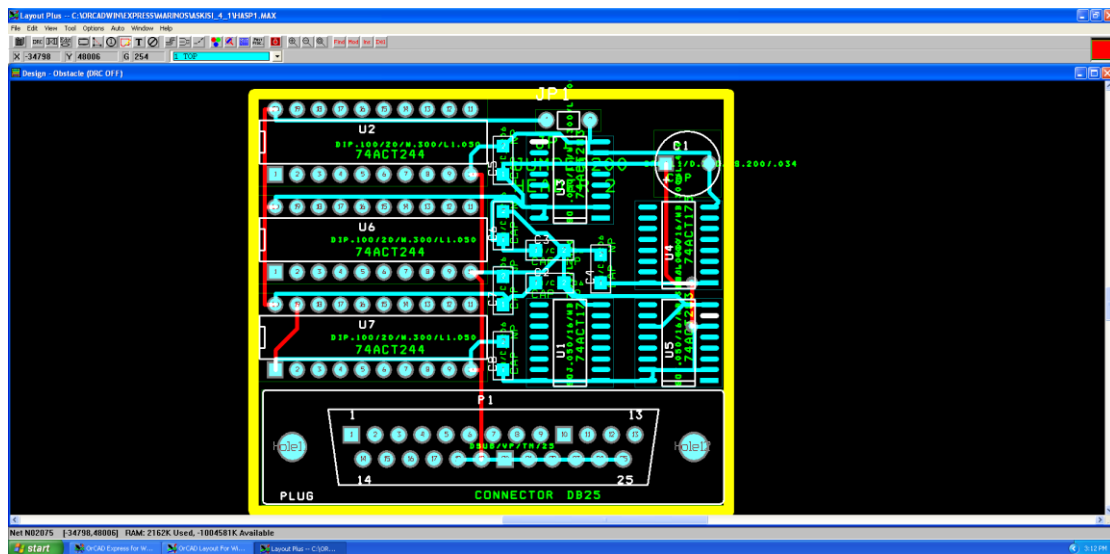
18.Simulation



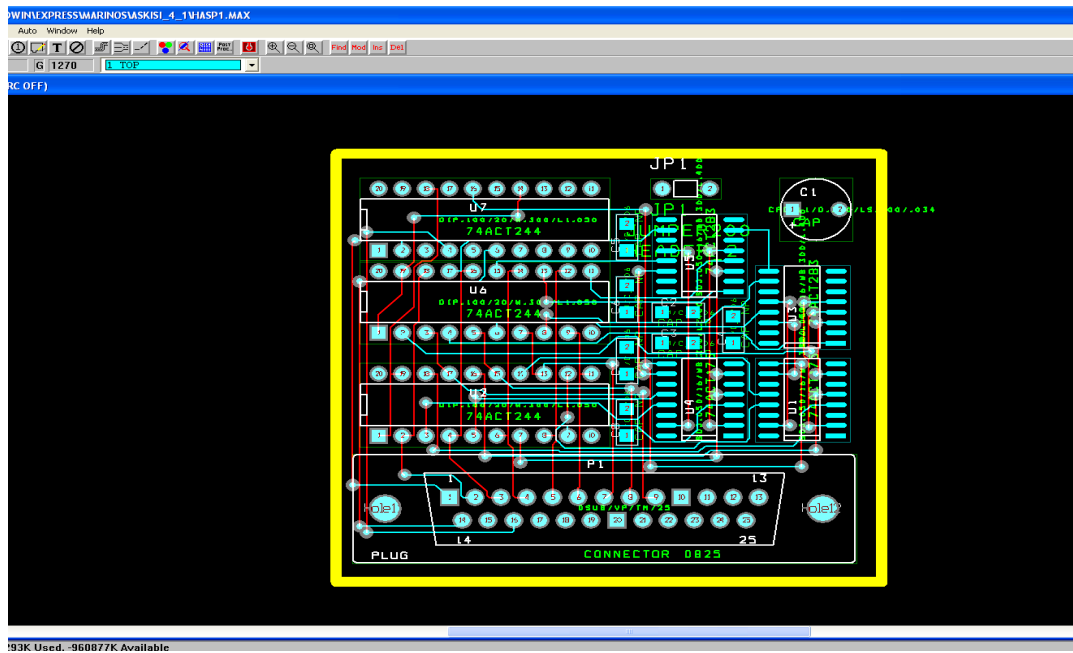
19.Layout



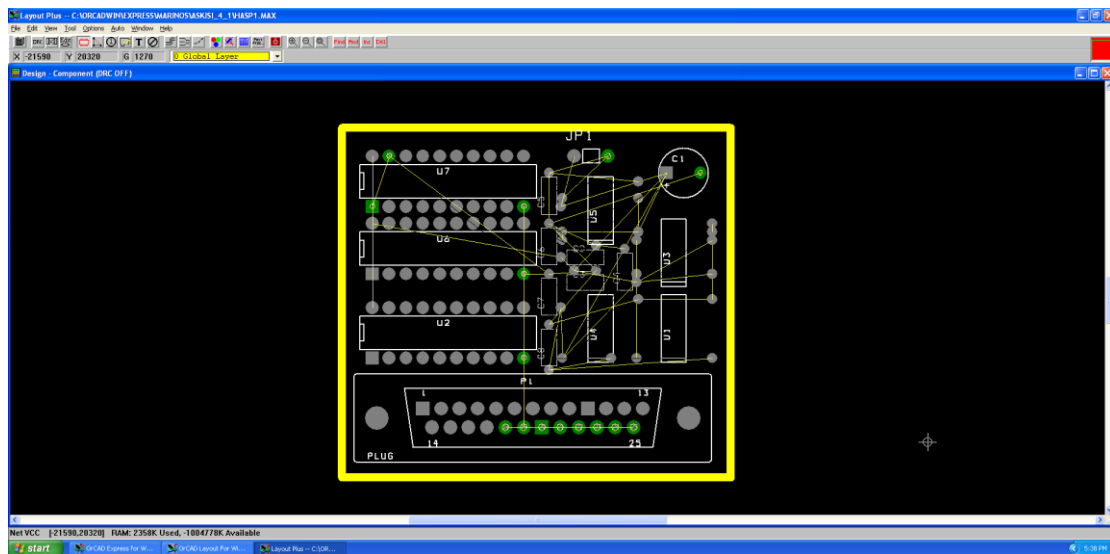
20..Διαδρόμηση μόνο με GND,VCC



21. Διαδρόμηση όλων των σημάτων εκτός GND,VCC



22. Fanout (GND-PWR)



Άσκηση 5

1.4 Bit Counter code

```
edit - counter.v
File Edit View Tools Window
C:\Modeltech_5.71\examples\work\counter.v
1 module counter4 (clear, clock, load, start_stop, count, data);
2   input [3:0] data;
3   output [3:0] count;
4   input start_stop;
5   input load;
6   input clock;
7   input clear;
8   reg [3:0] count;
9
10  always @(posedge clock or posedge clear)
11    if (clear) count <= 0;
12    else if (load) count <= data;
13    else if (start_stop) count <= count + 1;
14 endmodule
15
```

2.8 Bit counter code

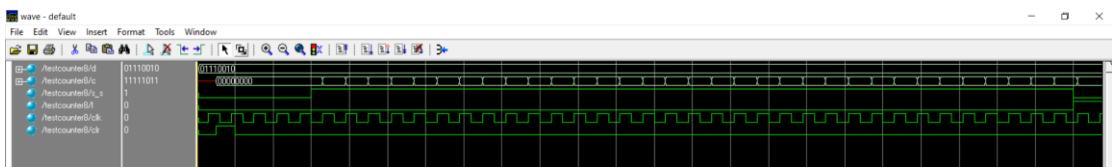
```
edit - counter8.v
File Edit View Tools Window
C:\Modeltech_5.71\examples\work\counter8.v
1 module counter8 (clear, clock, load, start_stop, count, data);
2   input [7:0] data;
3   output [7:0] count;
4   input start_stop, load, clock, clear;
5   assign start_stop1 = (count[0] && count[1] && count[2] && count[3]);
6   //Είναι το σήμα επίτρεψης για τον counter4 c2.
7
8   counter4 c1 (clear, clock, load, start_stop, count[3:0], data[3:0]);
9   counter4 c2 (clear, clock, load, start_stop1, count[7:4], data[7:4]);
10 endmodule
```

3. Testbench for 8 Bit counter

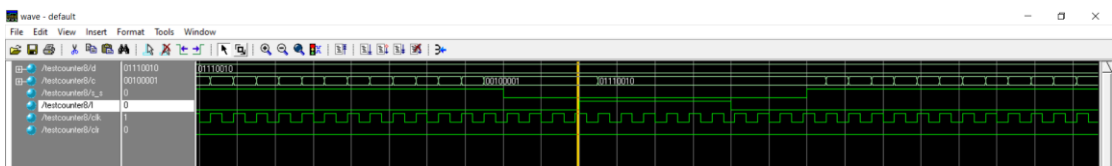
```
1 module testcounter8();
2 reg [7:0] d;
3 wire [7:0] c;
4 reg s_s, l, clk, clr;
5
6 counter8 c1(clr,clk,l,s_s,c,d);
7
8 initial
9 begin
10     #5 clk <= 0; clr <= 0; l <= 0; s_s <= 0; d <= 8'b01110010;
11 end
12 always #30 clk <= !clk;
13 initial
14 begin
15     #50 clr <= 1;
16     #50 clr <= 0;
17 end
18 initial
19 begin
20     #300 s_s <= 1;
21     #2000 s_s <= 0;
22 end
23 initial
24 begin
25     #2500 l <= 1;
26     #400 l <= 0;
27     #200 s_s <= 1;
28 end
29 endmodule
30
```

- Simulation

4. Clear and start_stop



5. Load<=1



- Synchronous clear

6.4 bit counter-syn clr

```
edit - syn_clr_counter4.v
File Edit View Tools Window
C:\Modeltech_5.7\examples\work\syn_clr_counter4.v
1 module syn_clr_counter4 (clear, clock, load, start_stop, count, data);
2   input [3:0] data;
3   output [3:0] count;
4   input start_stop;
5   input load;
6   input clock;
7   input clear;
8   reg [3:0] count;
9
10  always @(posedge clock)
11    if (clear) count <= 0;
12    else if (load) count <= data;
13    else if (start_stop) count <= count + 1;
14    else count <= count; //προαιρετικό
15 endmodule
16
```

7.8 bit counter-syn clr

```
edit - syn_clr_counter8.v
File Edit View Tools Window
C:\Modeltech_5.7\examples\work\syn_clr_counter8.v
1 module syn_clr_counter8 (clear, clock, load, start_stop, count, data);
2   input [7:0] data;
3   output [7:0] count;
4   input start_stop, load, clock, clear;
5   wire start_stop1;
6   assign start_stop1 = (count[0] & count[1] & count[2] & count[3]);
7
8
9   syn_clr_counter4 c1 (clear, clock, load, start_stop, count[3:0], data[3:0]);
10  syn_clr_counter4 c2 (clear, clock, load, start_stop1, count[7:4], data[7:4]);
11 endmodule
```

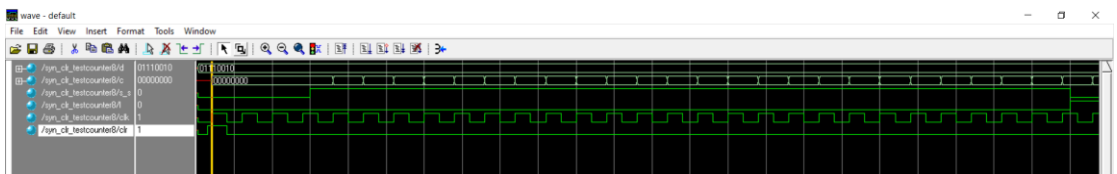
Testbench for 8 bit counter: (έγινε αλλαγή στις καθυστερήσεις των clk ,clr για να φανεί στις κυματομορφές ότι είναι σύγχρονο το clr).

```

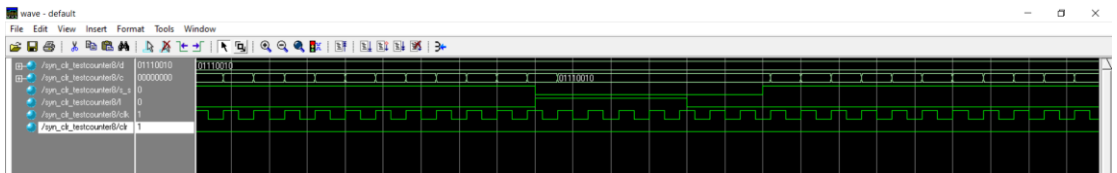
1 module syn_clr_testcounter8();
2 reg [7:0] d;
3 wire [7:0] c;
4 reg s_s, l, clk, clr;
5
6 syn_clr_counter8 c1(clr,clk,l,s_s,c,d);
7
8 initial
9 begin
10     #5 clk <= 0; clr <= 0; l <= 0; s_s <= 0; d <= 8'b01110010;
11 end
12 always #40 clk <= !clk;
13 initial
14 begin
15     #30 clr <= 1;
16     #50 clr <= 0;
17 end
18 initial
19 begin
20     #300 s_s <= 1;
21     #2000 s_s <= 0;
22 end
23 initial
24 begin
25     #2300 l <= 1;
26     #400 l <= 0;
27     #200 s_s <= 1;
28 end
29 endmodule
30

```

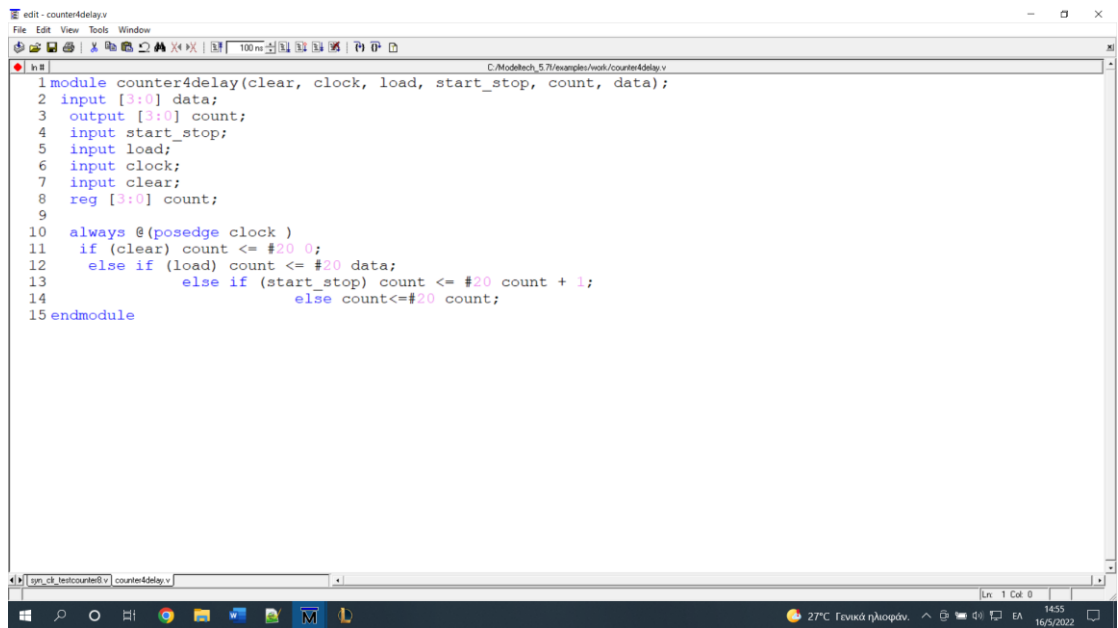
8. Clear and start_stop



9. Load <= 1



10.4 bit counter with 20ns delay



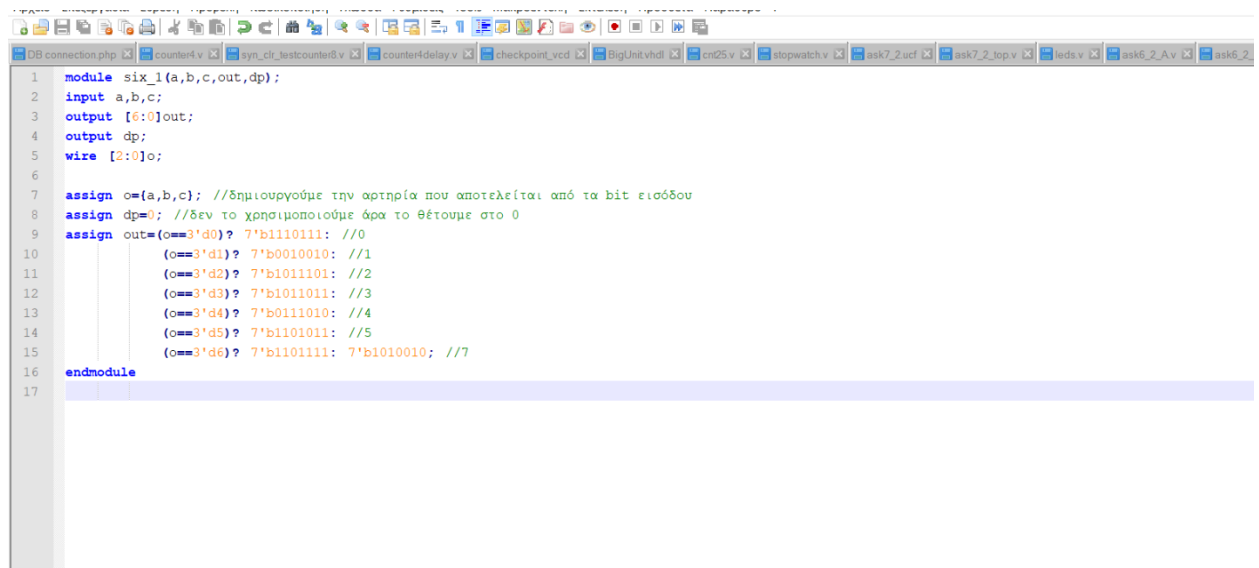
```
1 module counter4delay(clear, clock, load, start_stop, count, data);
2   input [3:0] data;
3   output [3:0] count;
4   input start_stop;
5   input load;
6   input clock;
7   input clear;
8   reg [3:0] count;
9
10  always @(posedge clock)
11    if (clear) count <= #20 0;
12    else if (load) count <= #20 data;
13    else if (start_stop) count <= #20 count + 1;
14    else count <= #20 count;
15 endmodule
```

Αφού έχουμε 20ns delay, το clock έχει ως ελάχιστη τιμή το 20ns για να επιτελούνται οι λειτουργίες του μετρητή. Συνεπώς $F_{max} = 1/20ns = 5 \cdot 10^{-11}$.

Άσκηση 6

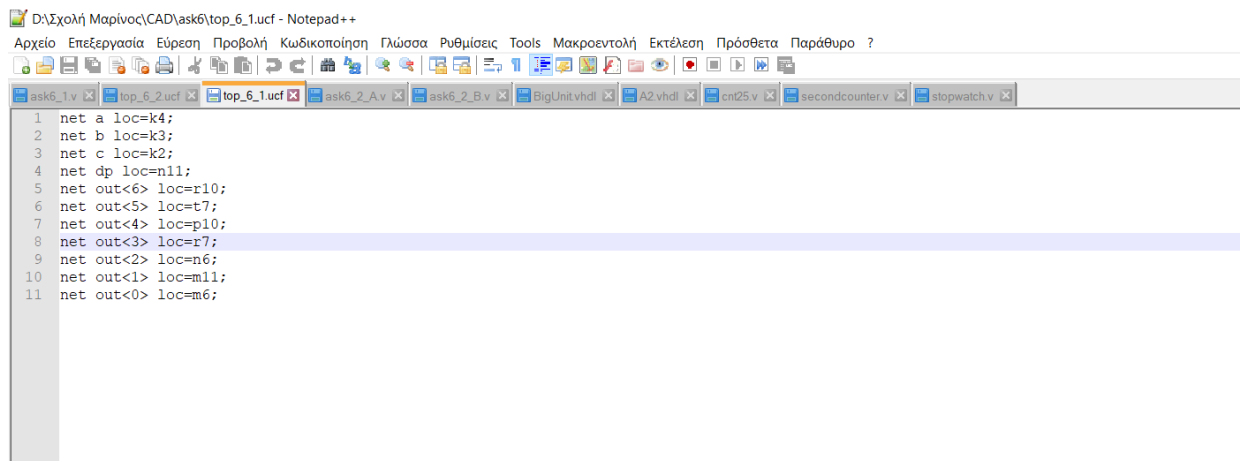
Ερώτημα 1

11.Κώδικας



```
1 module six_1(a,b,c,out,dp);
2   input a,b,c;
3   output [6:0]out;
4   output dp;
5   wire [2:0]o;
6
7   assign o={a,b,c}; //δημιουργούμε την αρτηρία που αποτελείται από τα bit εισόδου
8   assign dp=0; //δεν το χρησιμοποιούμε άρα το θέτουμε στο 0
9   assign out=(o==3'd0)? 7'b1110111: //0
10      (o==3'd1)? 7'b0010010: //1
11      (o==3'd2)? 7'b1011101: //2
12      (o==3'd3)? 7'b1011011: //3
13      (o==3'd4)? 7'b0111010: //4
14      (o==3'd5)? 7'b1101011: //5
15      (o==3'd6)? 7'b1101111: 7'b1010010: //7
16 endmodule
17
```

12.Top.ucf



```
D:\Σχολή Μαρίνου\CAD\ask6\top_6_1.ucf - Notepad++
Αρχείο Επεξεργασία Εύρεση Προβολή Κωδικοποίηση Γλώσσα Ρυθμίσεις Tools Μακροεντολή Εκτέλεση Πρόσθετα Παράθυρο ?

ask6_1.v top_6_2.ucf top_6_1.ucf ask6_2_A.v ask6_2_B.v BigUnit.vhdl A2.vhdl cn25.v secondcounter.v stopwatch.v

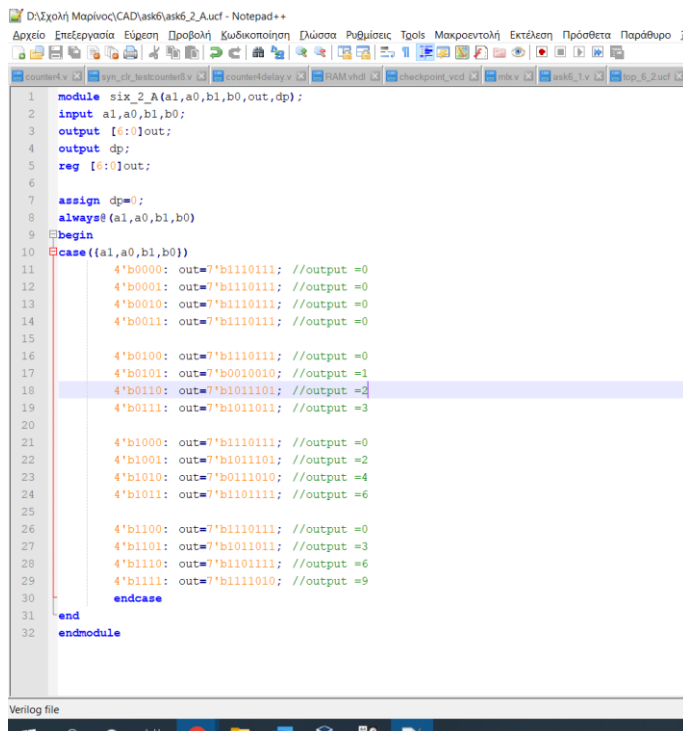
1 net a loc=k4;
2 net b loc=k3;
3 net c loc=k2;
4 net dp loc=n11;
5 net out<6> loc=r10;
6 net out<5> loc=t7;
7 net out<4> loc=p10;
8 net out<3> loc=r7;
9 net out<2> loc=n6;
10 net out<1> loc=m11;
11 net out<0> loc=m6;
```

Μέγιστη συχνότητα λειτουργίας: Θα βρούμε το path με την μεγαλύτερη καθυστέρηση και αυτό αν το αντιστρέψουμε θα βρούμε την μέγιστη συχνότητα λειτουργίας.

Το path με την μεγαλύτερη καθυστέρηση έχει 11.778ns. Άρα $F_{max}=1/11.778ns=84.9MHz$.

Ερώτημα 2Α

13.Κώδικας



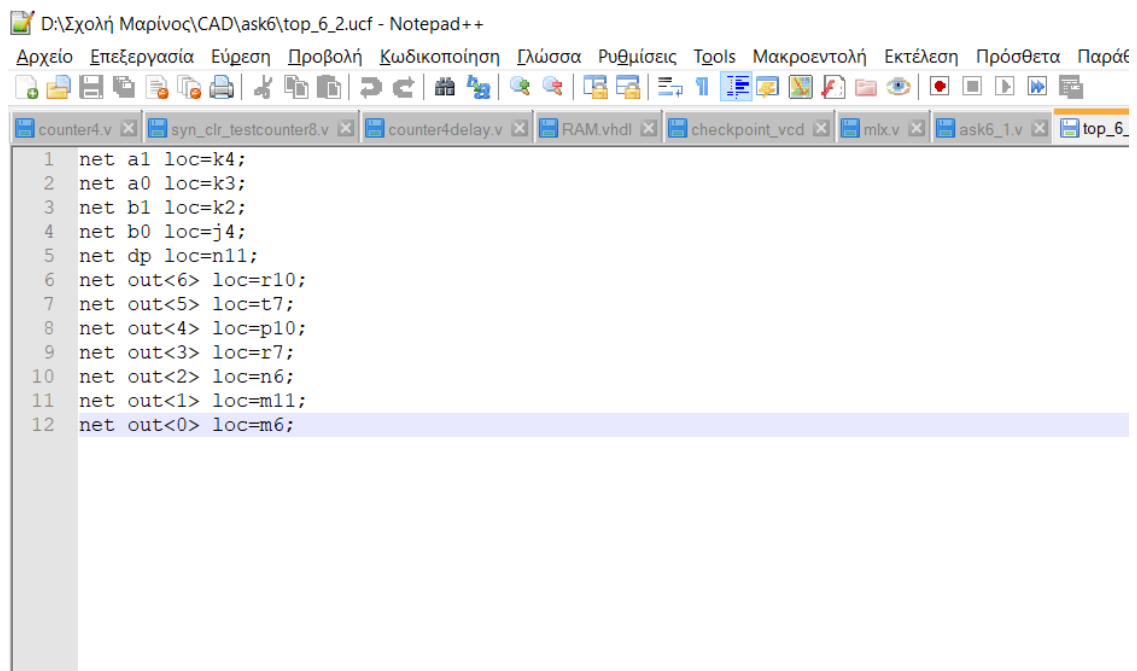
```
D:\Σχολή Μαρίνου\CAD\ask6\ask6_2_A.ucf - Notepad++
Αρχείο Επεξεργασία Εύρεση Προβολή Κωδικοποίηση Γλώσσα Ρυθμίσεις Tools Μακροεντολή Εκτέλεση Πρόσθετα Παράθυρο 2

counter.v syn_00_testcounter.v counterdelay.v RAM.vhdl checkpoint_vcd mix.v ask6_1.v top_6_2.ucf

1 module six_2_A(a1,a0,b1,b0,out,dp);
2 input a1,a0,b1,b0;
3 output [6:0]out;
4 output dp;
5 reg [6:0]out;
6
7 assign dp=0;
8 always@ (a1,a0,b1,b0)
9 begin
10 case({a1,a0,b1,b0})
11 4'b0000: out=7'b1110111; //output =0
12 4'b0001: out=7'b1110111; //output =0
13 4'b0010: out=7'b1110111; //output =0
14 4'b0011: out=7'b1110111; //output =0
15
16 4'b0100: out=7'b1110111; //output =0
17 4'b0101: out=7'b0010010; //output =1
18 4'b0110: out=7'b1011101; //output =2
19 4'b0111: out=7'b1011011; //output =3
20
21 4'b1000: out=7'b1110111; //output =0
22 4'b1001: out=7'b1011101; //output =2
23 4'b1010: out=7'b0111010; //output =4
24 4'b1011: out=7'b1101111; //output =6
25
26 4'b1100: out=7'b1110111; //output =0
27 4'b1101: out=7'b1011011; //output =3
28 4'b1110: out=7'b1101111; //output =6
29 4'b1111: out=7'b1111010; //output =9
30 endcase
31 end
32 endmodule

Verilog file
```

14.Top.ucf

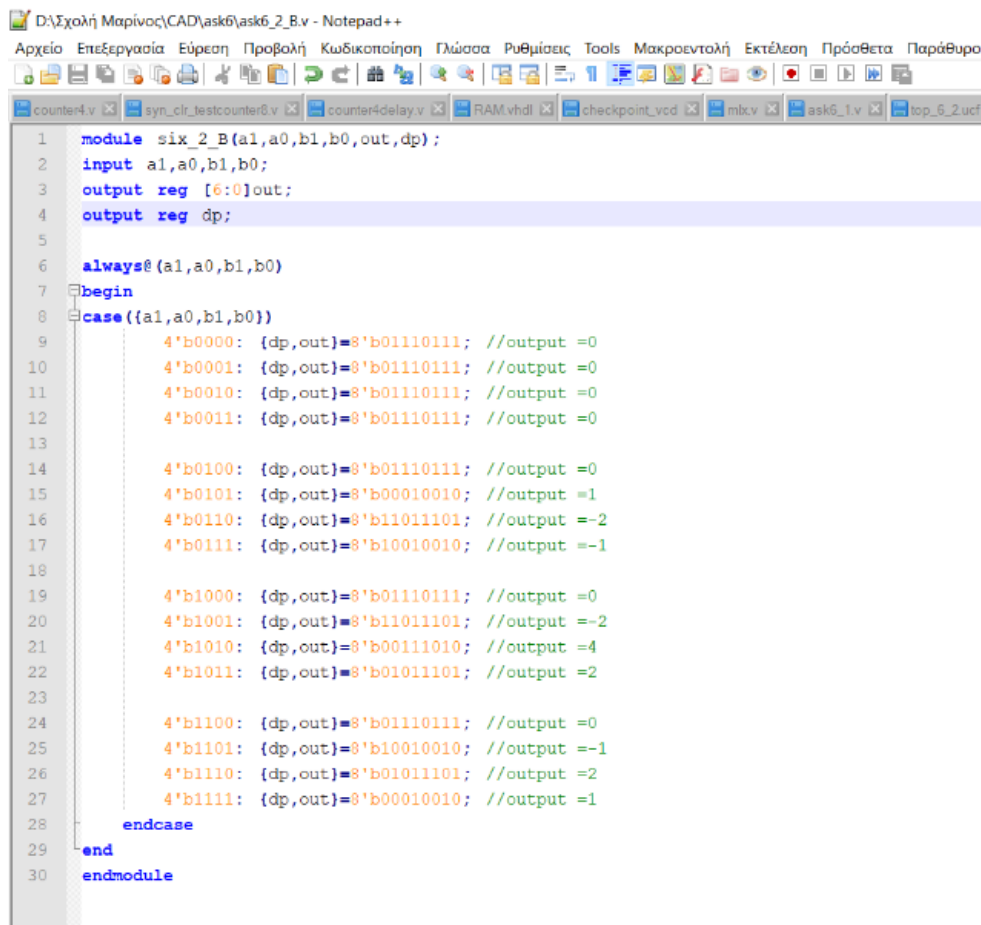


```
1 net a1 loc=k4;
2 net a0 loc=k3;
3 net b1 loc=k2;
4 net b0 loc=j4;
5 net dp loc=n11;
6 net out<6> loc=r10;
7 net out<5> loc=t7;
8 net out<4> loc=p10;
9 net out<3> loc=r7;
10 net out<2> loc=n6;
11 net out<1> loc=m11;
12 net out<0> loc=m6;
```

Μέγιστη καθυστέρηση=12.652ns. Συνεπώς η μέγιστη συχνότητα λειτουργίας είναι $F_{\max}=1/12.652\text{ns}=79\text{ MHz}$.

Ερώτημα 2B

15.Κώδικας



```
1 module six_2_B(a1,a0,b1,b0,out,dp);
2 input a1,a0,b1,b0;
3 output reg [6:0]out;
4 output reg dp;
5
6 always@ (a1,a0,b1,b0)
7 begin
8 case ({a1,a0,b1,b0})
9 4'b0000: {dp,out}={3'b01110111; //output =0
10 4'b0001: {dp,out}={3'b01110111; //output =0
11 4'b0010: {dp,out}={3'b01110111; //output =0
12 4'b0011: {dp,out}={3'b01110111; //output =0
13
14 4'b0100: {dp,out}={3'b01110111; //output =0
15 4'b0101: {dp,out}={3'b00010010; //output =1
16 4'b0110: {dp,out}={3'b11011101; //output =-2
17 4'b0111: {dp,out}={3'b10010010; //output =-1
18
19 4'b1000: {dp,out}={3'b01110111; //output =0
20 4'b1001: {dp,out}={3'b11011101; //output =-2
21 4'b1010: {dp,out}={3'b00111010; //output =4
22 4'b1011: {dp,out}={3'b01011101; //output =2
23
24 4'b1100: {dp,out}={3'b01110111; //output =0
25 4'b1101: {dp,out}={3'b10010010; //output =-1
26 4'b1110: {dp,out}={3'b01011101; //output =2
27 4'b1111: {dp,out}={3'b00010010; //output =1
28 endcase
29 end
30 endmodule
```

Top.ucf: είναι το ίδιο με το A

Μέγιστη καθυστέρηση=13.274ns. Συνεπώς η μέγιστη συχνότητα λειτουργίας είναι $F_{max}=1/13.274ns = 75.3 \text{ MHz}$.

Άσκηση 7

Οι κώδικες βρίσκονται στα αρχεία
ask7_2.ucf,ask7_2_top.v,leds.v,stopwatch.v,cnt5_128.v

Για το stopwatch φτιάξαμε έναν καταχωρητή που θα κρατάει την τιμή του πραγματικού χρόνου την δεδομένη χρονική στιγμή (now_time) και αν κάνουμε το stop==1 τότε θα περνάει στην έξοδο μόνο αυτή η τιμή που είναι αποθηκευμένη έναν κύκλο πριν. Αυτή η τιμή θα είναι στην έξοδο μέχρι να αλλάξουμε και πάλι το stop.

Για να δημιουργήσουμε τα δέκατα θέλουμε ο χρόνος να είναι 10 φορές μικρότερος από ότι στα δευτερόλεπτα. Συνεπώς, θέλουμε συχνότητα 10πλάσια δηλαδή 10 Hz. Αν την διαιρέσουμε με τον χρονιστή μας δίνει $10^7 = 25^3 * 2^7 * 5$. Θα χρειαστούμε λοιπόν για την μέτρηση 3 counter mod25 που έχουμε ήδη, έναν 7 bits και έναν 5 bits. Η σειρά που τους τοποθετήσαμε παίζει ρόλο στον χρονισμό του κυκλώματος.