ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Πτυχιακή Εργασία

Σχεδίαση κυκλωμάτων επικοινωνίας με απλές οθόνες, με τη γλώσσα VHDL και υλοποίηση στις αναπτυξιακές πλακέτες LP-2900 και DE2.

Φοιτήτρια: Βουρδόγλου Γεωργίας (3431)

Επιβλέπων: Δρ. Καλόμοιρος Ιωάννης, Επίκ. Καθηγητής

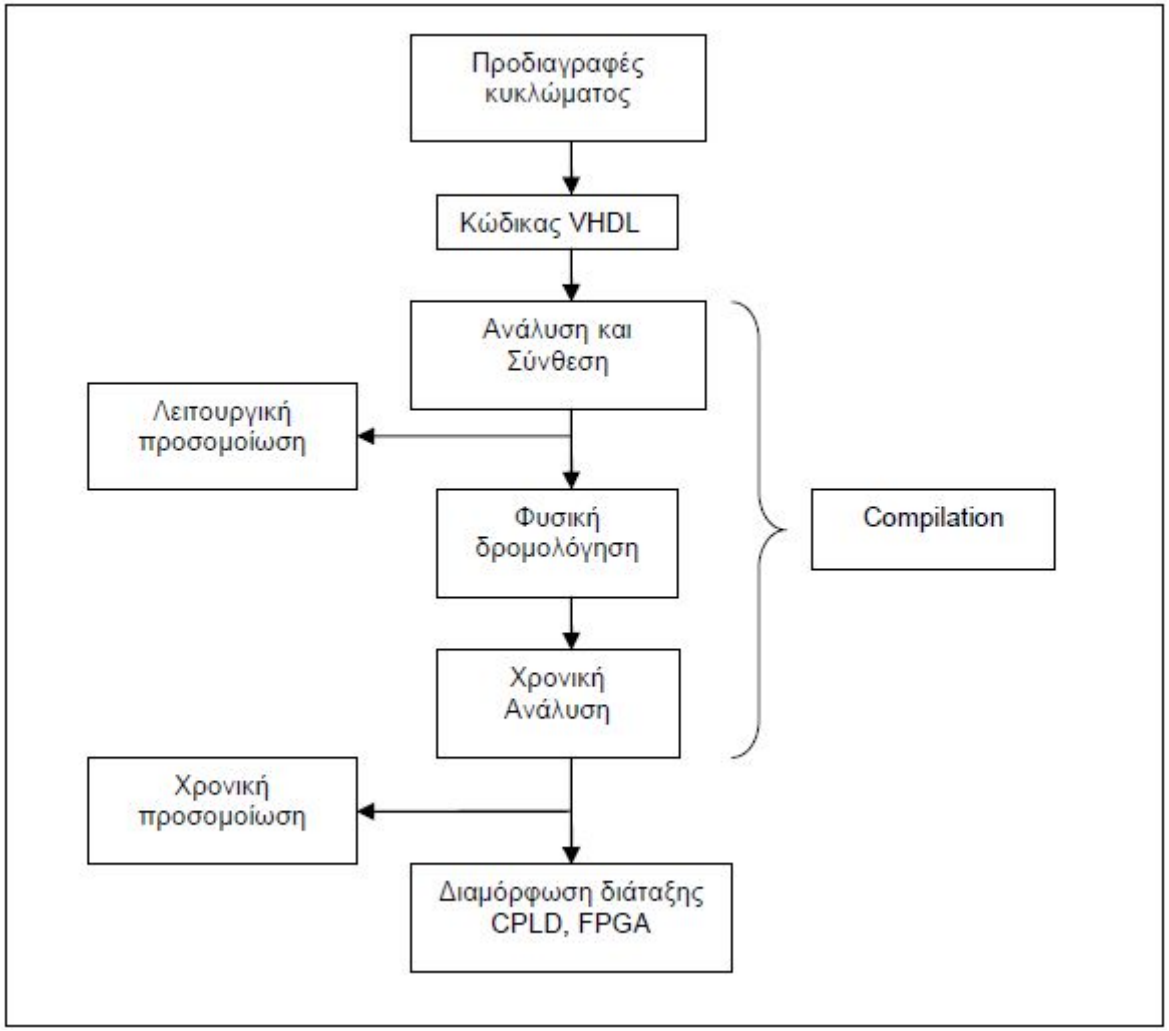
Περίληψη

Σκοπός της πτυχιακής εργασίας που ακολουθεί είναι η σχεδίαση σε γλώσσα VHDL (γλώσσα περιγραφής υλικού) κυκλωμάτων επικοινωνίας με οθόνες απεικόνισης λογικών καταστάσεων και η υλοποίησή τους σε FPGA (Διατάξεις Πυλών Προγραμματιζόμενες στο Πεδίο) μέσω των αναπτυξιακών πλακετών LP-2900 και DE2. Σε αυτές τις οθόνες ανήκουν οι απλές συστοιχίες LEDs, οι απεικονίσεις επτά τομέων (SSD) και οι οθόνες LCD.

Εισαγωγή στη γλώσσα VHDL

- Η VHDL είναι μία γλώσσα περιγραφής υλικού που χρησιμοποιείται στον αυτόματο σχεδιασμό ηλεκτρονικών σχεδιάσεων για την περιγραφή και ανάπτυξη ολοκληρωμένων ψηφιακών κυκλωμάτων και συστημάτων.
- Ο όρος VHDL είναι συντόμευση των λέξεων VHSIC Hardware Description Language, όπου VHSIC σημαίνει Very High Speed Integrated Circuit.
- Η γλώσσα VHDL χρησιμοποιείται ευρύτατα για την περιγραφή και υλοποίηση ψηφιακών συστημάτων σε προγραμματιζόμενες λογικές διατάξεις, τύπου CPLDs (Complex Programmable Logic Devices-Σύνθετες προγραμματιζόμενες λογικές διατάξεις) και FPGAs (Field Programmable Gate Arrays-(Διατάξεις πυλών προγραμματιζόμενες στο πεδίο).

Διάγραμμα ροής της σχεδίασης με κώδικα VHDL



Εισαγωγή στη σχεδίαση

- Για τη σχεδίαση ενός λογικού κυκλώματος χρειάζεται ένας αριθμός εργαλείων CAD, που βρίσκονται συνήθως μαζί με τη μορφή ενός συστήματος σχεδίασης CAD. Αυτό περιλαμβάνει εργαλεία για την εκτέλεση των λειτουργιών της εισαγωγής σχεδίασης, της σύνθεσης και βελτιστοποίησης, της προσομοίωσης και της φυσικής σχεδίασης.

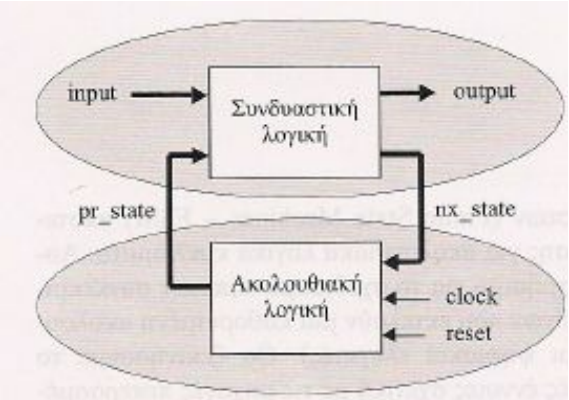
- Τέτοια εργαλεία λογισμικού για σχεδίαση με γλώσσα VHDL που χρησιμοποιούνται πιο συχνά είναι:
    1. Quartus II της Altera
    2. ISE της Xilinx
    3. Leonardo Spectrum της Mentor Graphics
    4. ModelSim της Mentor Graphics

Μηχανές πεπερασμένων καταστάσεων (Finite State Machines)

- Μία μηχανή πεπερασμένων καταστάσεων (FSM) είναι μία ειδική τεχνική μοντελοποίηση ακολουθιακών κυκλωμάτων που υλοποιείται με τη βοήθεια συνδιαστικής λογικής και ενός ή περισσότερων Flip-Flops.

- Στις περισσότερες περιπτώσεις υπάρχει ένα ωρολογιακό σήμα που ελέγχει τη λειτουργία ενός ακολουθιακού κυκλώματος και ένα τέτοιο κύκλωμα λέγεται σύγχρονο ακολουθιακό κύκλωμα (synchronous sequential circuits).

- Η άλλη περίπτωση, στην οποία δεν χρησιμοποιείται ωρολογιακό σήμα ονομάζεται ασύγχρονο ακολουθιακό κύκλωμα (asynchronous sequential circuits).



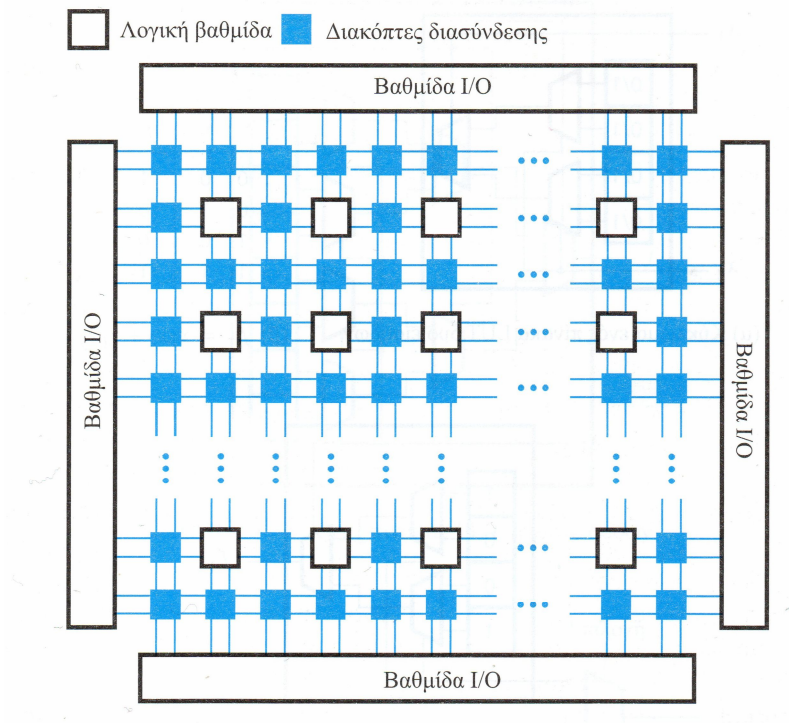
Αναπαράσταση δομικού διαγράμματος FSM

Διατάξεις Πυλών Προγραμματιζόμενων στο Πεδίο (FPGA)

Τα FPGAs (Field Programmable Gate Array- Διατάξεις Πυλών Προγραμματιζόμενων στο Πεδίο) είναι ψηφιακά ολοκληρωμένα κυκλώματα τα οποία περιέχουν προγραμματιζόμενα μπλοκ ψηφιακής λογικής. Αυτά τα μπλοκ συνδέονται μεταξύ τους με την βοήθεια προγραμματιζόμενων διασυνδέσεων. Τα FPGAs προγραμματίζονται είτε από τον καταναλωτή είτε από τον σχεδιαστή μετά την κατασκευή τους. Ο προγραμματισμός τους πραγματοποιείται κυρίως με τη χρήση μίας γλώσσας περιγραφής υλικού (HDL- Hardware Description Language), δηλαδή είτε VHDL, είτε AHDL, είτε Verilog.

Δομή της διάταξης FPGA

Η γενική δομή της διάταξης FPGA περιέχει τρία είδη πόρων: λογικές βαθμίδες, βαθμίδες εισόδου / εξόδου για τη σύνδεση με τους ακροδέκτες της συσκευασίας και τους διακόπτες και γραμμές εσωτερικής διασύνδεσης.



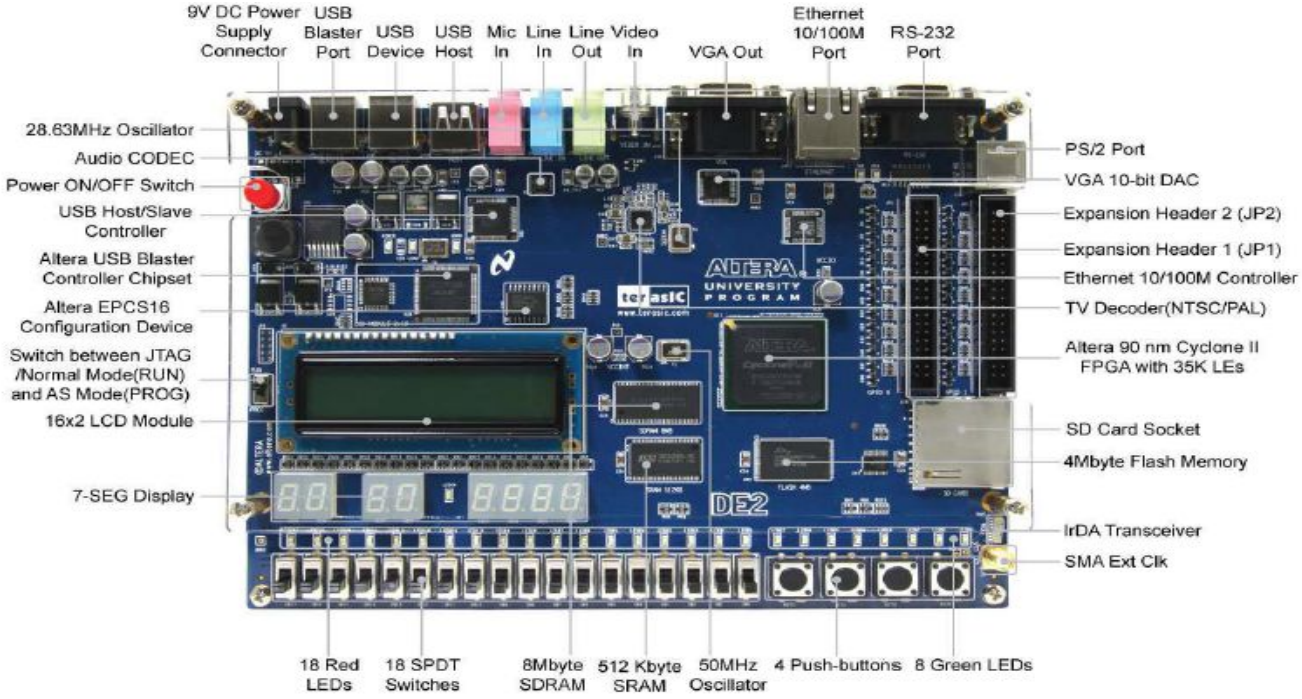
Κάτοψη της γενικής αρχιτεκτονικής FPGA

Αναπτυξιακή Πλακέτα LP-2900

Η πρώτη αναπτυξιακή πλακέτα που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής μας είναι η LP-2900 της εταιρείας Leap Electronic Co. Το αναπτυξιακό κύκλωμα LP-2900 βασίζεται στο ολοκληρωμένο FPGA EPF10K10TC144-4, το οποίο ανήκει στην οικογένεια FLEX10K της εταιρείας Altera. Διαθέτει 144 pins εισόδου/εξόδου από τα οποία τα 102 είναι ελεύθερα για χρήση και αποτελείται από 576 λογικά στοιχεία (logic elements) και 61444 bits μνήμης.

Αναπτυξιακή Πλακέτα DE2 της ALTERA

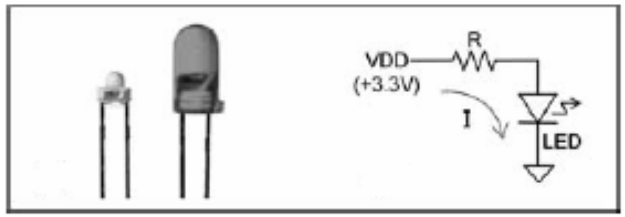
Η αναπτυξιακή και εκπαιδευτική πλακέτα DE2 της Altera είναι η δεύτερη πλακέτα που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής μας, έχει αναπτυχθεί για την εκμάθηση της ψηφιακής λογικής και την υπολογιστική οργάνωση σε εργαστηριακό περιβάλλον. Η πλακέτα αυτή, βασίζεται στο ολοκληρωμένο FPGA EP2C35F672C6, το οποίο ανήκει στην οικογένεια Cyclone II της εταιρείας Altera.



Οθόνες απεικόνισης λογικών καταστάσεων

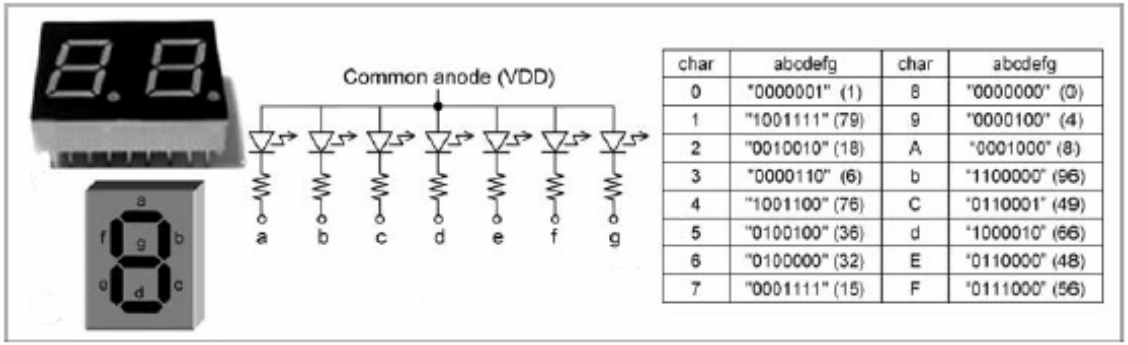
- **Απλές συστοιχίες LEDs**

Η Δίοδος Εκπομπής Φωτός, (LED, Light Emitting Diode) είναι ένας ημιαγωγός, ο οποίος εκπέμπει φωτεινή ακτινοβολία στενού φάσματος όταν του παρέχεται μία ηλεκτρική τάση κατά τη φορά ορθής πόλωσης.



- **Ενδείκτης επτά τομέων (SSD)**

Ο ενδείκτης επτά τομέων (SSD, seven-segment-display) μετατρέπει ένα δεκαδικό ψηφίο σε σήματα που οδηγούν τις διόδους φωτοεκπομπής (LEDs) του ενδείκτη.



char	abcdefg	char	abcdefg
0	"0000001" (1)	8	"0000000" (0)
1	"1001111" (79)	9	"0000100" (4)
2	"0010010" (18)	A	"0001000" (8)
3	"0000110" (6)	b	"1100000" (96)
4	"1001100" (76)	C	"0110001" (49)
5	"0100100" (36)	d	"1000010" (66)
6	"0100000" (32)	E	"0110000" (48)
7	"0001111" (15)	F	"0111000" (56)

Οθόνες απεικόνισης λογικών καταστάσεων (συνέχεια)

- **Οθόνη υγρών κρυστάλλων (LCD)**

  Η οθόνη υγρών κρυστάλλων (LCD, Liquid Crystal Display) περιέχει δύο γραμμές των 16 χαρακτήρων η καθεμία. Στο πίσω μέρος της οθόνης υπάρχει ένας ελεγκτής (HD44780U, της Hitachi) που ενεργεί ως διεπαφή μεταξύ της οθόνης LCD και του εξωτερικού κόσμου. Αυτός ο ελεγκτής μπορεί να προσεγγιστεί μέσω 16 ακίδων, οι οποίες περιλαμβάνουν τροφοδοσία, γραμμές αντίθεσης, ελέγχου και δεδομένων.

Pin#	Name		In/Out/Pwr
1	GND	Ground	Power
2	VCC	LCD Controller Power (+3 to +5V)	Power
3	VLCD	LCD Display Bias (+5 to -5V *see text)	Analog
4	RS	Register Select: H: Data L: Command	Input
5	R/W	H: Read L: Write	Input
6	E	Enable (Data strobe, active high)	Input
7	DB0	Data LSB	I/O
8	DB1	Data	I/O
9	DB2	Data	I/O
10	DB3	Data	I/O
11	DB4	Data	I/O
12	DB5	Data	I/O
13	DB6	Data	I/O
14	DB7	Data MSB	I/O
15	A	LED Backlight Anode (optional)	Power
16	K	LED Backlight Cathode (optional)	Power

Εφαρμογή οδήγησης οθονών και προσομοίωση

```
1.  LIBRARY ieee;
2.  USE ieee.std_logic_1164.all;
3.  ENTITY led_ssd_lcd_driver IS
4.      GENERIC (clk_divider: POSITIVE :=50000);
5.      PORT(clk: IN std_logic;
6.           switches: IN std_logic_vector(3 DOWNTO 0);
7.           leds: OUT std_logic_vector(3 DOWNTO 0);
8.           ssd: OUT std_logic_vector(6 DOWNTO 0);
9.           RS, RW, LCD_ON, BKL_ON: OUT std_logic;
10.          E: BUFFER std_logic;
11.          DB: OUT std_logic_vector(7 DOWNTO 0));
12. END led_ssd_lcd_driver;
13. --------------------------------------------------------------------------------
14. ARCHITECTURE Behaviour OF led_ssd_lcd_driver IS
15.     TYPE STATE IS( FunctionSet1, FunctionSet2, FunctionSet3,
16.                    FunctionSet4, ClearDisplay, DisplayControl,
17.                    EntryMode, WriteData,  ReturnHome);
18.     SIGNAL pr_state, nx_state: state;
19.     SIGNAL input_lcd: std_logic_vector(7 DOWNTO 0);
20. BEGIN
21.     ---------Part1: LED driver-----------------------------------
22.     leds<= switches;
23.     ---------Part2: SSD driver-----------------
24.     WITH switches SELECT
25.         ssd<= "0000001" WHEN "0000",
26.               "1001111" WHEN "0001",
27.               "0010010" WHEN "0010",
28.               "0000110" WHEN "0011",
29.               "1001100" WHEN "0100",
30.               "0100100" WHEN "0101",
31.               "0100000" WHEN "0110",
32.               "0001111" WHEN "0111",
33.               "0000000" WHEN "1000",
34.               "0000100" WHEN "1001",
35.               "0001000" WHEN "1010",
36.               "1100000" WHEN "1011",
37.               "0110001" WHEN "1100",
38.               "1000010" WHEN "1101",
39.               "0110000" WHEN "1110",
40.               "0111000" WHEN OTHERS;
```

Εφαρμογή οδήγησης οθονών και προσομοίωση (συνέχεια)

```
41.  ---------Part3: LCD driver (FSM-based)----------------------------
42.  WITH switches SELECT
43.      input_lcd<= "00110000" WHEN "0000",
44.                  "00110001" WHEN "0001",
45.                  "00110010" WHEN "0010",
46.                  "00110011" WHEN "0011",
47.                  "00110100" WHEN "0100",
48.                  "00110101" WHEN "0101",
49.                  "00110110" WHEN "0110",
50.                  "00110111" WHEN "0111",
51.                  "00111000" WHEN "1000",
52.                  "00111001" WHEN "1001",
53.                  "01000001" WHEN "1010",
54.                  "01000010" WHEN "1011",
55.                  "01000011" WHEN "1100",
56.                 "01000100" WHEN "1101",
57.                 "01000101" WHEN "1110",
58.                 "01000110" WHEN OTHERS;
59.  -----Ενεργοποίηση οθόνης LCD και του οπίσθιου φωτισμού της-
60.  LCD_ON<='1'; BKL_ON<='1';
```

```
61.  -----Ρύθμιση γεννήτριας ρολογιού για την LCD (E=500Hz)----------
62.  PROCESS(clk)
63.     VARIABLE count: INTEGER RANGE 0 TO clk_divider;
64.  BEGIN
65.     IF (clk'EVENT AND clk='1') THEN
66.        count:=count+1;
67.        IF (count = clk_divider ) THEN
68.           E<=NOT E;
69.           count:=0;
70.        END IF;
71.     END IF;
72.  END PROCESS;
73.  -----Κατώτερο τμήμα μηχανής πεπερασμένων καταστάσεων(FSM)---
74.  PROCESS(E)
75.  BEGIN
76.     IF (E'EVENT AND E='1') THEN
77.        pr_state<= nx_state;
78.     END IF;
79.  END PROCESS;
80.  -----Ανώτερο τμήμα μηχανής πεπερασμένων καταστάσεων(FSM)----
81.  PROCESS( pr_state, input_lcd)
```

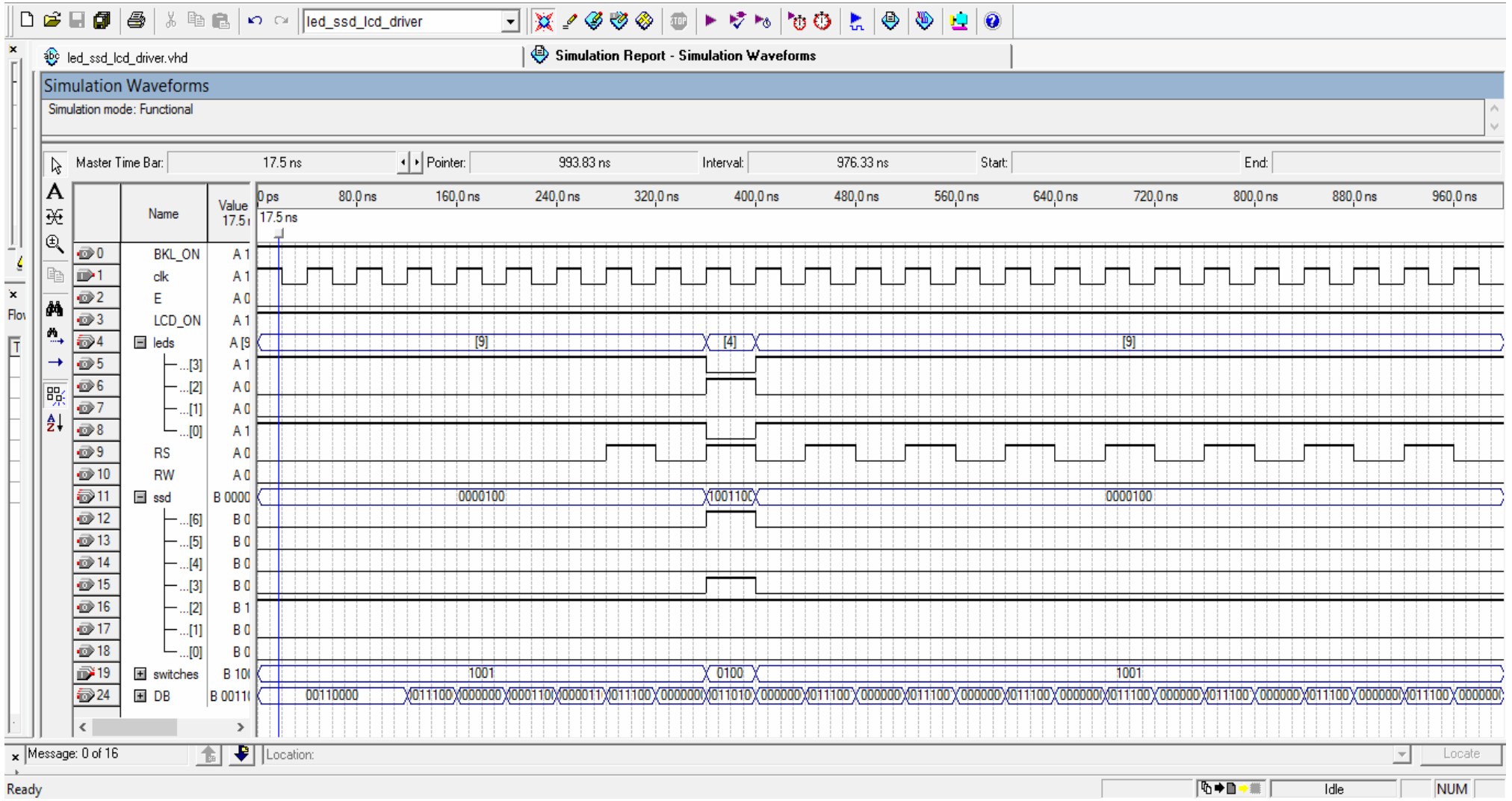
Εφαρμογή οδήγησης οθονών και προσομοίωση (συνέχεια)

```
82.     BEGIN
83.       CASE pr_state IS
84.       ---------Αρχικοποίηση της LCD---------
85.           WHEN FunctionSet1=>
86.                RS<='0'; RW<='0';
87.                DB<= "0011XX00";
88.                 nx_state<= FunctionSet2;
89.           WHEN FunctionSet2=>
90.                RS<='0'; RW<='0';
91.                DB<= "0011XX00";
92.                nx_state<= FunctionSet3;
93.           WHEN FunctionSet3=>
94.                RS<='0'; RW<='0';
95.                DB<= "0011XX00";
96.                nx_state<= FunctionSet4;
97.           WHEN FunctionSet4=>
98.                RS<='0'; RW<='0';
99.                DB<= "00111000";
100.               nx_state<= ClearDisplay;
101.          WHEN ClearDisplay =>
102.               RS<='0'; RW<='0';
103.               DB<= "00000001";
104.                nx_state<= DisplayControl;
105.          WHEN DisplayControl =>
106.               RS<='0'; RW<='0';
107.               DB<= "00001100";
108.                nx_state<= EntryMode;
109.          WHEN EntryMode =>
110.               RS<='0'; RW<='0';
111.               DB<= "00000110";
112.                nx_state<= WriteData;
113.          -------Εγγραφή δεδομένων στην LCD---
114.          WHEN WriteData =>
115.               RS<='1'; RW<='0';
116.               DB<= input_lcd;
117.               nx_state<= ReturnHome;
118.          WHEN ReturnHome =>
119.               RS<='0'; RW<='0';
120.               DB<= "10000000";
121.                nx_state<= WriteData;
122.       END CASE;
123.    END PROCESS;
124. END Behaviour;
```

Προσομοίωση με το Quartus II της εταιρείας ALTERA

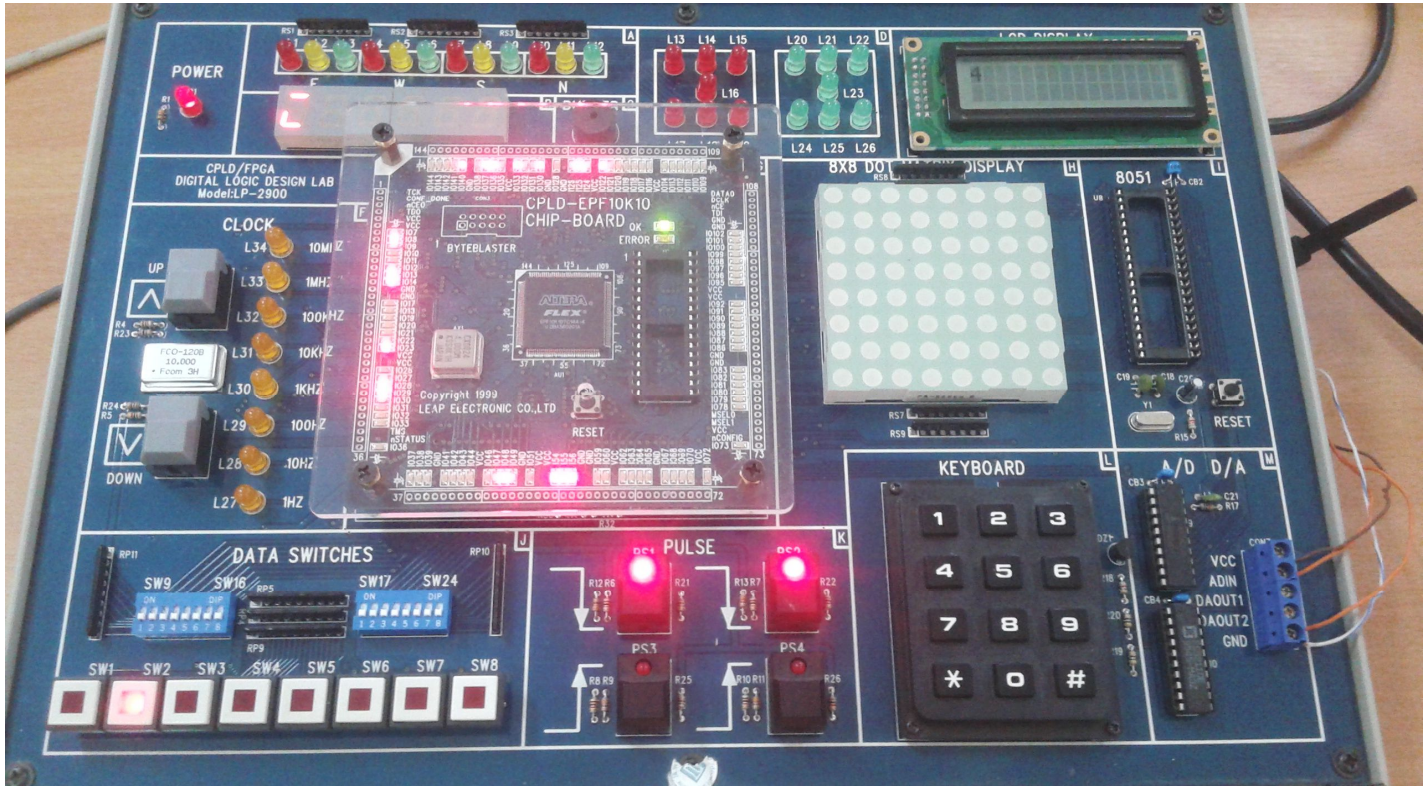
- Το λογισμικό Quartus II είναι ένα από τα γνωστότερα προγράμματα σχεδίασης CAD (Computer-Aided Design) και είναι πνευματική ιδιοκτησία της ALTERA. Το Quartus II χρησιμοποιείται για την ανάπτυξη και τον προγραμματισμό όλων των αναπτυξιακών κυκλωμάτων της εταιρείας ALTERA, δηλαδή των διατάξεων CPLDs και FPGAs που κατασκευάζει η εταιρεία.

Προσομοίωση με το Quartus II της εταιρείας ALTERA (συνέχεια)



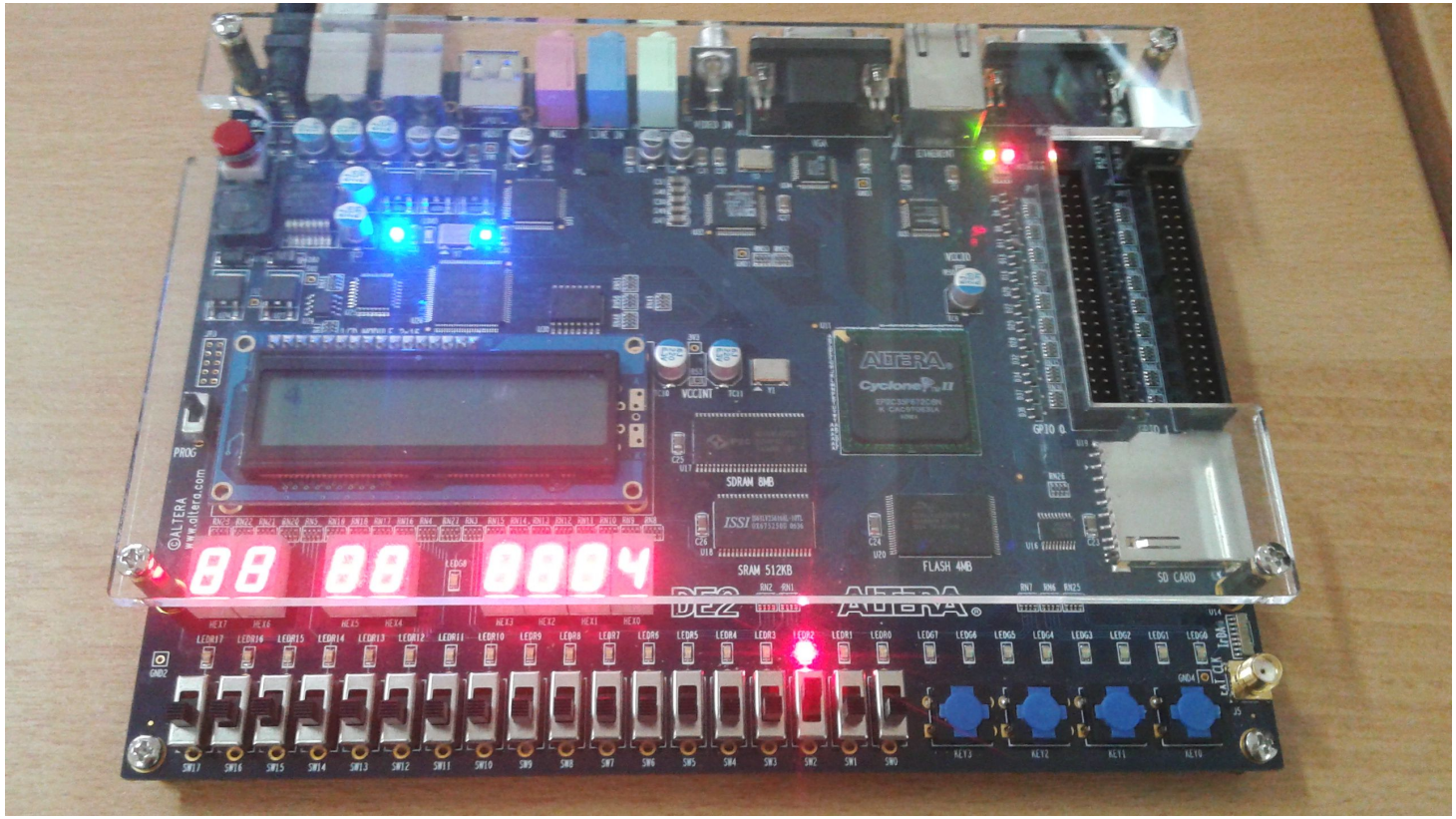
Υλοποίηση του κυκλώματος με αναπτυξιακή πλακέτα LP-2900

- Έχουμε ορίσει τις εισόδους $Switches_{0,1,3}$ = 0 ($SW_{4,3,1}$) και $Switch_2$=1 (SW_2). Άρα σύμφωνα με τη λειτουργία των leds, της οθόνης επτά τομέων και της lcd οθόνης διαπιστώνουμε ότι θα πρέπει να ανάβει μόνο το SMD led_8, στην οθόνη επτά τομέων τα $LEDs_{23,28,29}$ και στην οθόνη LCD να απεικονίζεται ο αριθμός τέσσερα.



Υλοποίηση του κυκλώματος με αναπτυξιακή πλακέτα DE2

- Έχουμε ορίσει τις εισόδους $Switches_{0,1,3}$ = 0 ($SW_{0,1,3}$) και $Switch_2$=1 (SW_2). Έτσι έχουμε ως αποτέλεσμα να ανάβει μόνο το $LEDR_2$ και στην οθόνη επτά τομέων, όπως και στην οθόνη lcd να απεικονίζεται ο αριθμός τέσσερα.



Τέλος παρουσίασης

Σας ευχαριστώ για το χρόνο σας