



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 11

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

1.Σκοπός της άσκησης.....	4
2.Παραδοτέα.....	4
3.Ο επεξεργαστής Picoblaze.....	4
4.Παραδείγματα χρήσης του επεξεργαστή Picoblaze.....	5
5.Αρχιτεκτονικά χαρακτηριστικά του επεξεργαστή Picoblaze.....	6
6.Αρχιτεκτονική του Picoblaze.....	6
7.Το πλήρες σετ εντολών (ISA) του Picoblaze.....	8
8.Περιγραφή της άσκησης.....	8
9.Στοιχεία πλακέτας.....	9
10.Περιστροφή LED με υλοποίηση hardware.....	10
11.Περιστροφή LED με υλοποίηση hardware + software.....	11
12.Εμφάνιση μηνύματος hello world στην LCD οθόνη με hardware υλοποίηση.....	11
13.Εμφάνιση μηνύματος hello world στην LCD οθόνη με hardware και software υλοποίηση.....	13

1. Σκοπός της άσκησης

- Ανάπτυξη ενσωματωμένων συστημάτων με τον μικρο-επεξεργαστή Picoblaze¹.
- Δημιουργία System-on-Chip με τον picoblaze.

(A) 10 ερώτηση

(C) 6 ασκήσεις/προγράμματα

2 bonus

2. Παραδοτέα

- **Παραδοτέο C1: left_right_leds.vhd** [επίσης τα σχετιζόμενα παραδοτέα: **(a1)** left_right_leds.ucf, **(a2)** screenshot από το σχηματικό (*schematic*), και **(a3)** από το device utilization summary].
- **Παραδοτέο C2: left_right_leds.vhd** .
- **Παραδοτέο C3: left_right_leds_embedded_system.vhd** [καθώς και τα σχετιζόμενα παραδοτέα: **(a4)** screenshot από το σχηματικό (*schematic*), και **(a5)** από το “device utilization summary” “Resource Estimation”].
- **Παραδοτέο C4:** Κώδικας VHDL [επίσης τα αρχεία: **(a6)** ucf, **(a7)** screenshot από το σχηματικό (*schematic*), και **(a8)** από το “device utilization summary” “Resource Estimation”].
- **Παραδοτέο C5: control.vhd** .
- **Παραδοτέο C6: left_right_leds-lcd.ucf** [(**a9**) screenshot από το σχηματικό (*schematic*), **(a10)** από το “device utilization summary” ή το “Resource Estimation”].

3. Ο επεξεργαστής Picoblaze

Είναι ένας επεξεργαστής soft-core, δηλαδή επεξεργαστής που περιγράφεται σε γλώσσα HDL και μπορεί να ενσωματωθεί σε μια πλακέτα FPGA.

Συνοπτικά Χαρακτηριστικά:

- ISC 8-bit Microcontroller Reference Design.
- Supports Virtex/Virtex-E, Spartan™-II/E, Virtex-II/Pro devices.
- Very small size – only 84 Virtex-II slices, 33% of XC2V40.
- Up to 66 MIPS in Virtex™-II , 100 MIPS in Virtex™-IV.
- Everything in FPGA - no external components required.
- Highly integrated for implementing non-time critical state machine.

¹ Χρησιμοποιήθηκε υλικό από το εκπαιδευτικό σεμινάριο “Reconfigurable Architectures and Tools (K. Siozios & D. Diamantopoulos), NTUA”.

- Predictable fast interrupt response.
- Originally named KCPSM which stands for "Constant(K) Coded Programmable State Machine" (formerly "Ken Chapman's PSM").
- It is freely distributed under the BSD license.

4. Παραδείγματα χρήσης του επεξεργαστή Picoblaze

Παραδείγματα χρήσης του επεξεργαστή Picoblaze:

- Front panel switches and displays for Set Top Box.
- Dynamic Loop Bandwidth Multiplexer for frame to frame analysis.
- Link layer of IEEE 1394 Interface.
- Microcontroller for Compact Flash Programming engine.
- DECT Radio/Repeater.
- PCI board programming controller.
- Communications controller.
- Preprocessing for network processor.
- Motor controller.
- Programmable power supply controller.
- Part of Media Access Controller.
- Controller in broadcast video equipment.
- and many, many more...

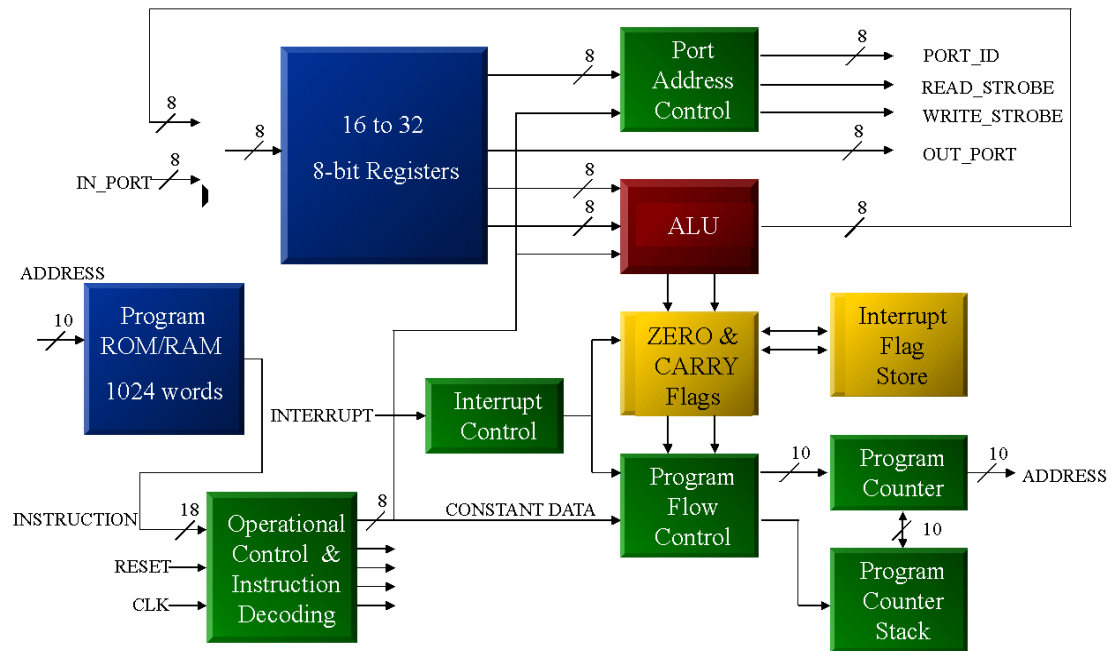
5. Αρχιτεκτονικά χαρακτηριστικά του επεξεργαστή Picoblaze

Χαρακτηριστικά:

- 32 General Purpose 8-Bit Registers.
- Arithmetic Logic Unit (ALU):
 - Performed with operands from any of the registers.
 - Operations include logical, arithmetic, shift and rotate operations.
- Flags (Program Flow Control):
 - ZERO and CARRY flags.
- Input/Output:
 - 256 input and 256 output ports (8 bits).
 - Shift data in and out of the unit.

- Interrupt:
 - Single interrupt that can be augmented by logic.
- Predictable execution rate:
 - Two clock cycles/instruction.

6. Αρχιτεκτονική του Picoblaze



7. Το πλήρες σετ εντολών (ISA) του Picoblaze

(Η ανάλυση των εντολών βρίσκεται στο εγχειρίδιο χρήσης στα έγγραφα του εργαστηρίου)

The following is a complete instruction set representing all op-codes

'X' and 'Y' refer to the definition of the storage registers 's' in the range 0 to 1F
 'kk' represents a constant value in the range 00 to FF
 'aa' represents an address in the range 00 to FF
 'pp' represents a port address in the range 00 to FF

SR0 sX
 SR1 sX
 SRX sX
 SRA sX
 RR sX

Program Control Group

JUMP aa
 JUMP Z,aa
 JUMP NZ,aa
 JUMP C,aa
 JUMP NC,aa

 CALL aa
 CALL Z,aa
 CALL NZ,aa
 CALL C,aa
 CALL NC,aa

Note that call and return supports
 up to a stack depth of 15.

Logical Group

LOAD sX,kk
 AND sX,kk
 OR sX,kk
 XOR sX,kk

 LOAD sX,sY
 AND sX,sY
 OR sX,sY
 XOR sX,sY

Arithmetic Group

ADD sX,kk
 ADDCY sX,kk
 SUB sX,kk
 SUBCY sX,kk

 ADD sX,sY
 ADDCY sX,sY
 SUB sX,sY
 SUBCY sX,sY

SL0 sX
 SL1 sX
 SLX sX
 SLA sX
 RL sX

Input/Output Group

INPUT sX,pp
 INPUT sX,(sY)

 OUTPUT sX,pp
 OUTPUT sX,(sY)

Interrupt Group

RETURNI ENABLE
 RETURNI DISABLE

 ENABLE INTERRUPT
 DISABLE INTERRUPT

8. Περιγραφή της άσκησης

Σε αυτό το εργαστήριο θα γνωρίσετε τον επεξεργαστή picoblaze. Το εργαστήριο αυτό αποτελείται από 4 ασκήσεις. Οι 2 ασκήσεις εκτελούν κάποια λειτουργία χρησιμοποιώντας μόνο hardware, ενώ οι άλλες 2 εκτελούν την ίδια λειτουργία χρησιμοποιώντας hardware + software.

Αρχικά θα υλοποιήσετε μια περιστροφή LED (led rotation) χρησιμοποιώντας **(a)** μόνο VHDL χωρίς επεξεργαστή και **(b)** τον PICOBLAZE και το κατάλληλο πρόγραμμα σε assembly. Στη συνέχεια θα επαναληφθεί η άσκηση με το να τυπωθεί το Hello World στην LCD οθόνη χρησιμοποιώντας **(c)** μόνο VHDL χωρίς επεξεργαστή και **(d)** τον PICOBLAZE και το κατάλληλο πρόγραμμα σε assembly.

Σημειώστε ότι και στα (a),(c), αλλά και στα (b),(d) χρησιμοποιείται η VHDL γλώσσα. Η διαφορά των (a),(c) με τα (b),(d) είναι ότι τα πρώτα είναι μια περιγραφή καθαρά υλικού (*hardware implementation*), ενώ τα υπόλοιπα είναι μια περιγραφή που χρησιμοποιεί υλικό (*τον επεξεργαστή*) και λογισμικό (*το πρόγραμμα σε assembly*) (*hardware & software implementation*).

Θα χρησιμοποιηθεί η πλακέτα SPARTAN 3A που υπάρχει στο εργαστήριο και λειτουργεί αρμονικά με τον επεξεργαστή picoblaze.

9. Στοιχεία πλακέτας

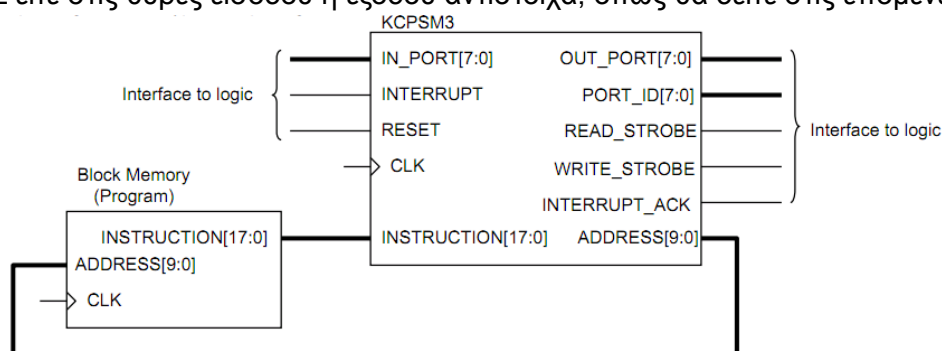
Το FPGA της πλακέτας Spartan που υπάρχει στο εργαστήριο, φέρει τον κωδικό XC3S700A και έχει τα παρακάτω τεχνικά στοιχεία:

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits ⁽¹⁾	Block RAM bits ⁽¹⁾	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	CLBs	Slices						
KC3S500A	50K	1,584	16	12	176	704	11K	54K	3	2	144	64
KC3S200A	200K	4,032	32	16	448	1,792	28K	288K	16	4	248	112
KC3S400A	400K	8,064	40	24	896	3,584	56K	576K	20	4	312	144
KC3S700A	700K	13,248	48	32	1,472	5,888	92K	360K	20	8	372	165

Η πλακέτα που έχει το συγκεκριμένο chip έχει πλήθος περιφερειακών, όπως:

- LCD 2x16 characters.
- 4 slide switches.
- 4 push buttons.
- 8 LEDs.
- 1 VGA output.
- 2 serial ports.
- 1 PS/2.
- 1 10/100 Ethernet PHY.
- A2D and D2A converters.
- Additional expansion ports.

Περισσότερες πληροφορίες μπορείτε να βρείτε στο **datasheet** που έχει ανέβει στα αρχεία του μαθήματος. Τέλος, παρουσιάζεται ο μικροεπεξεργαστής ricoblaze (KCPSM3), οι διασυνδέσεις εισόδου, εξόδου και η μνήμη εντολών (Program Block memory). Αν θέλετε να συνδέσετε ένα περιφερειακό εισόδου (π.χ. κουμπιά) ή εξόδου (π.χ. led) θα τα διασυνδέσετε με τον κατάλληλο κώδικα VHDL είτε στις θύρες εισόδου ή εξόδου αντίστοιχα, όπως θα δείτε στις επόμενες ασκήσεις.



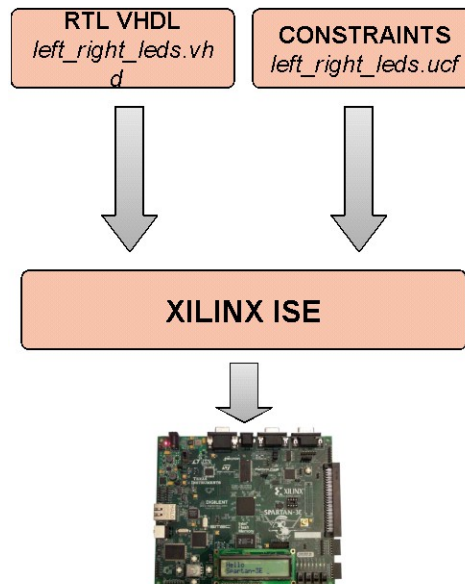
Στο εγχειρίδιο [KCPSM3_manual.pdf](#) στα έγγραφα του εργαστηρίου μπορείτε να διαβάσετε περισσότερες πληροφορίες.

(BONUS1: Αν υλοποιήσετε όλο το εργαστήριο στην πλακέτα ALTERA DE2, έχετε bonus +0.5 μονάδες)

10. Περιστροφή LED με υλοποίηση hardware

Θα κατασκευάσετε το αρχείο `left_right_leds.vhd` , το οποίο θα εκτελεί την περιστροφή των led, και θα το χρησιμοποιήσετε μαζί με το αρχείο των περιορισμών για την πλακέτα Spartan 3A.

➤ Using pure VHDL

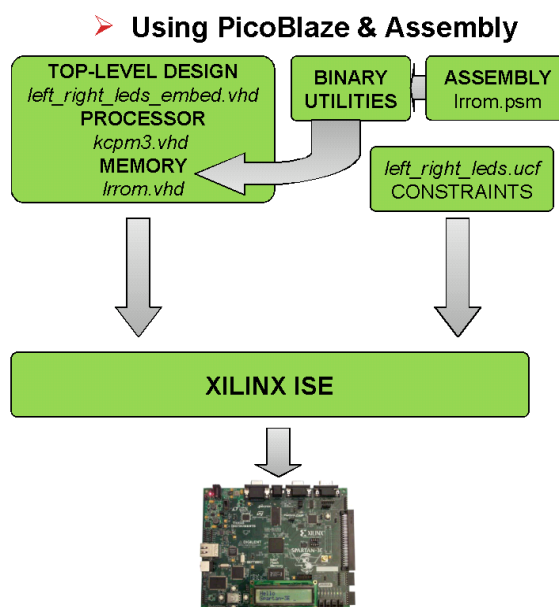


- Μεταφορτώστε το αρχείο `left_right_leds-Spartan3E.ucf` , το οποίο έχει τα constraints για την πλακέτα Spartan 3E και τροποποιήστε τα LOC, ώστε να ανταποκρίνονται στα constraints της πλακέτας Spartan 3A. Μπορείτε να χρησιμοποιήσετε το σχετικό αρχείο προηγούμενου εργαστηρίου. Αποθηκεύστε το αρχείο με το όνομα `left_right_leds.ucf` .
- Μεταφορτώστε το αρχείο `left_right_leds-incomplete.vhd` και τροποποιήστε τις τοποθεσίες που είναι σημειωμένες με το TODO, ώστε να λειτουργεί σωστά. Αποθηκεύστε το αρχείο με το όνομα `left_right_leds.vhd` .
- Δημιουργήστε ένα νέο project στο Xilinx IDE με τις κατάλληλες ρυθμίσεις για την πλακέτα του εργαστηρίου, κάντε σύνθεση και δημιουργία του bitstream, προγραμματίστε την πλακέτα και επιβεβαιώστε την ορθή λειτουργία.

Παραδοτέα: Τα αρχεία (a1) `left_right_leds.ucf`, (c1) `left_right_leds.vhd`, (a2) screenshot από το σχηματικό (*schematic*), και (a3) από το device utilization summary.

11. Περιστροφή LED με υλοποίηση hardware + software

Θα χρησιμοποιήσετε την περιγραφή του picoblaze που βρίσκεται στο αρχείο **kcpsm3.vhd** . Θα γράψετε την assembly για την εκτέλεση της συγκεκριμένης λειτουργίας (*led rotation*) στο αρχείο **lrrom.psm**. Όπως σε ένα πραγματικό ενσωματωμένο σύστημα, το πρόγραμμα θα πρέπει να τοποθετηθεί σε μια μνήμη ROM που θα συνδέεται με τον μικρο-επεξεργαστή για να τον τροφοδοτεί με εντολές. Θα πρέπει λοιπόν κάθε εντολή assembly που έχετε στο αρχείο psm να τη μετατρέψετε στον αντίστοιχο κώδικα μηχανής, ο οποίος θα πρέπει να τοποθετηθεί σε ένα VHDL module που θα λειτουργεί ως μνήμη ROM. Αν και μπορείτε να κάνετε εσείς αυτή τη διαδικασία, μπορείτε να γλυτώσετε χρόνο με το να χρησιμοποιήσετε ένα ειδικό πρόγραμμα για το picoblaze που δέχεται ως είσοδο ένα αρχείο psm και εξάγει ένα αρχείο VHDL που περιγράφει μια μνήμη ROM με τον κώδικά μας. Το αρχείο αυτό που θα εξαχθεί θα το ονομάσετε **lrrom.vhd**. Χρησιμοποιώντας (**α**) την περιγραφή του επεξεργαστή (*hardware*) και (**β**) την περιγραφή της μνήμης ROM (*software*) θα κατασκευάσετε το **left_right_leds_embed.vhd** το οποίο θα εκτελεί την περιοδική ενεργοποίηση των LED.



- Μεταφορτώστε το αρχείο **kcpsm3.vhd** το οποίο περιγράφει τον επεξεργαστή picoblaze. Αυτό το αρχείο δε θα το τροποποιήσετε.
- Χρησιμοποιήστε το αρχείο των περιορισμών **left_right_leds.vhd** από την προηγούμενη άσκηση.
- Μεταφορτώστε το αρχείο με το όνομα **left_right_leds_embed-incomplete.vhd** το οποίο θα αποτελεί την περιγραφή του SoC, στο οποίο περιέχονται ως components, ο επεξεργαστής, η μνήμη ROM προγράμματος, η διασύνδεση με τις θύρες εισόδου, και η διασύνδεση με τις θύρες εξόδου. Τροποποιήστε το αρχείο στα σημεία που επισημαίνονται ως TODO σύμφωνα με τα σχόλια και αποθηκεύστε το αρχείο με το όνομα **left_right_leds_embedded_system.vhd** .
- Μεταφορτώστε το αρχείο **lrrom-incomplete.psm** το οποίο περιέχει την assembly που θα χρησιμοποιήσετε για τη δημιουργία του αρχείου της ROM. Τροποποιήστε τις κατάλληλες γραμμές που έχουν σημειωθεί με το TODO. Αποθηκεύστε το νέο αρχείο με το όνομα **lrrom.psm** .
- Μεταφορτώστε τα αρχεία **KCPSM3.EXE** και **ROM_form.vhd** τα οποία θα βοηθήσουν στη δημιουργία του αρχείου **lrrom.vhd**. Τοποθετήστε αυτά τα 2 αρχεία, όπως και το αρχείο που **lrrom.psm**, σε έναν κατάλογο που μπορείτε να μεταβείτε εύκολα από τη γραμμή εντολών. Από τη γραμμή εντολών (**cmd.exe**) μεταβείτε στον κατάλογο που έχετε αποθηκεύσει τα αρχεία και δώστε την εντολή **KCPSM3.EXE lrrom.psm**. Αν εκτελεστεί χωρίς σφάλμα

(παρατηρήστε προσεκτικά την έξοδο του προγράμματος) θα δείτε στο τέλος ότι έχει δημιουργηθεί με επιτυχία το αρχείο **lrom.vhd** (όπως και άλλα αρχεία).

- Δημιουργήστε ένα νέο project στο Xilinx IDE με τις κατάλληλες ρυθμίσεις για την πλακέτα του εργαστηρίου, κάντε σύνθεση και δημιουργία του bitstream, προγραμματίστε την πλακέτα και επιβεβαιώστε την ορθή λειτουργία.

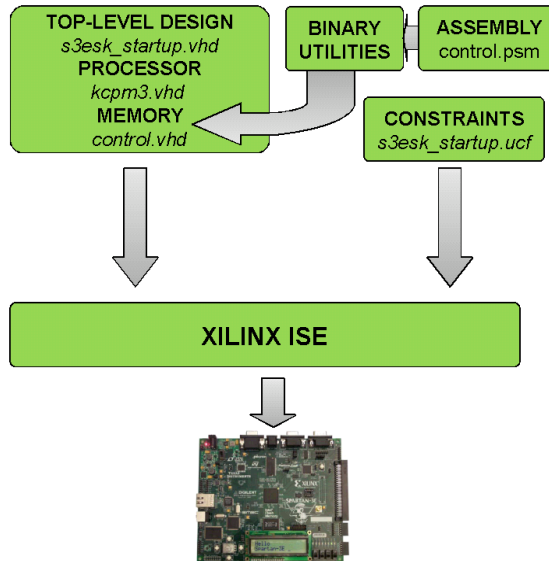
Παραδοτέα: Τα αρχεία (c2) **left_right_leds.vhd**, (c3) **left_right_leds_embedded_system.vhd**, (a4) screenshot από το σχηματικό (schematic), και (a5) από το “device utilization summary” “Resource Estimation”.

12. Εμφάνιση μηνύματος hello world στην LCD οθόνη με hardware υλοποίηση

- Να δημιουργήσετε μια VHDL υλοποίηση (μπορείτε να χρησιμοποιήσετε το υλικό που έχετε αναπτύξει σε προηγούμενα εργαστήρια), η οποία θα κάνει τα εξής:
 - Θα εμφανίζει και θα κάνει SCROLL στην οθόνη LCD το μήνυμα SPARTAN-3A STARTER KIT " και " www.xilinx.com .
 - SW0 turns on LD0.
 - SW1 turns on LD1.
 - SW2 turns on LD2.
 - SW3 turns on LD3.
 - BTN East turns on LD4.
 - BTN South turns on LD5.
 - BTN North turns on LD6.
 - BTN West turns on LD7.
 - Single LED is moved left or right by rotation of control OR by pressing centre button of rotary encoder toggle mode.

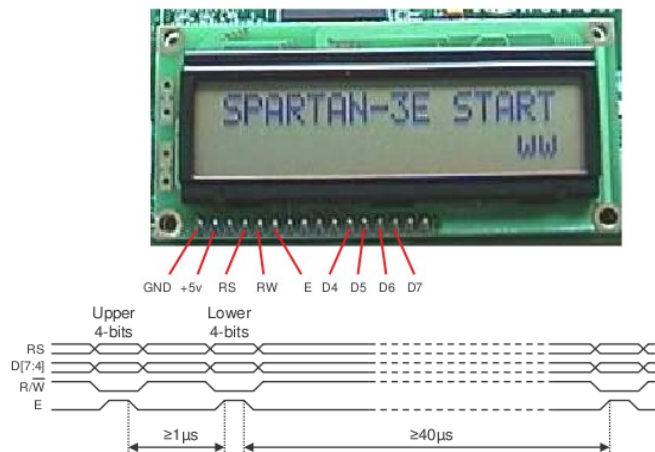
Παραδοτέα: Τα αρχεία (a6) **ucf**, (c4) **κώδικας vhdl**, (a7) screenshot από το σχηματικό (schematic), και (a8) από το “device utilization summary” “Resource Estimation”.

13. Εμφάνιση μηνύματος hello world στην LCD οθόνη με hardware και software υλοποίηση

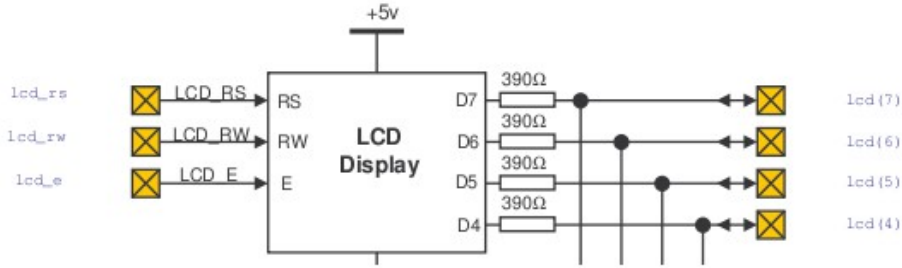


Η LCD της πλακέτας χρησιμοποιείται ως εξής:

- The board is set up to use the 4-wire/8-wire (both supported) data interface to the LCD character module.
- After initial display communication is established, all data transfers are 8-bit ASCII character codes, data bytes or 8-bit addresses.
- Each 8-bit transfer obviously has to be decomposed into two 4-bit or one 8-bit transfer which must be spaced by at least 1μs.
- Following an 8-bit write operation, there must be an interval of at least 40μs before the next communication.



Υλοποίηση με μεταφορές των 4bit/8bit:



Caution! When using four-bit mode, the FPGA must drive the LCD_DB<3:0> signals High.

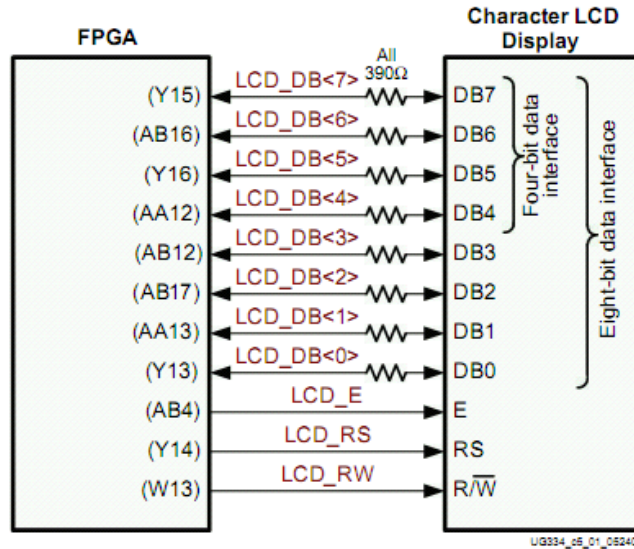


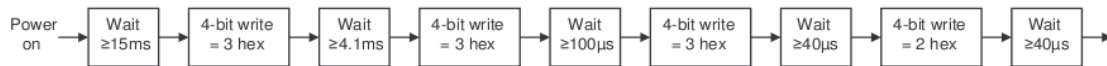
Figure 5-1: Character LCD Interface

Περισσότερες πληροφορίες μπορείτε να δείτε στο datasheet [ug334.pdf](#) για το Spartan 3A στα έγγραφα του εργαστηρίου, στο κεφάλαιο “Character LCD Screen”.

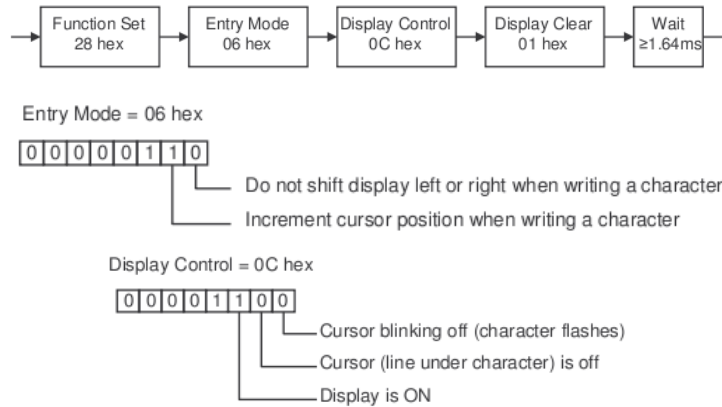
Προκειμένου να υπάρξει συμβατότητα με άλλες πλακέτες της Xilinx που διαθέτουν μόνο 4bit δεδομένων, σε αυτό το εργαστήριο θα χρησιμοποιήσουμε αρχικά μόνο τα 4bit.

(BONUS2: Αν υλοποιήσετε το εργαστήριο-κώδικας VHDL+assembly -- με μεταφορές 8bit, τότε έχετε +0.5)

Πριν χρησιμοποιηθεί το LCD display από τον επεξεργαστή, θα πρέπει να αρχικοποιηθεί με μια συγκεκριμένη ακολουθία. Θα χρησιμοποιηθεί η ρουτίνα LCD_reset η οποία εκτελεί την παρακάτω ακολουθία αρχικοποίησης:

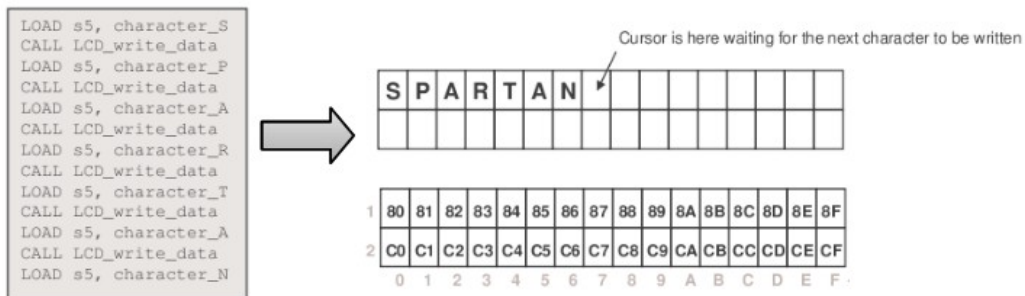


Το επόμενο τμήμα της ακολουθίας χρησιμοποιείται για να εγκαθιδρύσει τη λειτουργία της οθόνης:



Μόλις ολοκληρωθεί η αρχικοποίηση, η εγγραφή χαρακτήρων στο LCD έχει ως συνέπεια να τοποθετηθούν στην άνω αριστερή γωνία και να τοποθετούνται σειριακά προς τα δεξιά και στη συνέχεια, μόλις γεμίσει η γραμμή, στην κάτω γραμμή.

Θα γράφετε στην οθόνη με εντολές assembly, όπως παρακάτω:



Στον παραπάνω κώδικα, κάθε φορά τοποθετούμε στον καταχωρητή **s5** τη δεκαεξαδική τιμή του χαρακτήρα σύμφωνα με τον κώδικα ASCII και καλούμε τη συνάρτηση `LCD_write_data` που θα κατασκευάσουμε, η οποία γράφει αυτόν τον χαρακτήρα στο LCD και δέχεται μια παράμετρο ως είσοδο, η οποία παράμετρος είναι η τιμή του καταχωρητή `s5`.

Επίσης, θα χρησιμοποιήσετε και άλλες συναρτήσεις σε assembly για το χειρισμό του LCD, όπως τοποθέτηση του δρομέα σε συγκεκριμένο σημείο, καθαρισμό του display κ.α.

Τα βήματα που θα ακολουθήσετε είναι ίδια με τα βήματα της άσκησης (b) για την hardware + software υλοποίηση του LED Rotation.

- Μεταφορτώστε το αρχείο **kcpsm3.vhd** , το οποίο περιγράφει τον επεξεργαστή picoblaze.
- Μεταφορτώστε το αρχείο **left_right_leds-lcd-Spartan3E.ucf** , το οποίο έχει τα constraints για την πλακέτα Spartan 3E και τροποποιήστε τα LOC, ώστε να ανταποκρίνονται στα constraints της πλακέτας Spartan 3A. Μπορείτε να χρησιμοποιήσετε σχετικό αρχείο προηγούμενου εργαστηρίου. Αποθηκεύστε το αρχείο με το όνομα **left_right_leds-lcd.ucf** .
- Μεταφορτώστε το αρχείο **s3A-led-lcd-interrupts.vhd** και παρατηρήστε τις συνδέσεις εισόδου (*κουμπιά και rotary knob*) και εξόδου (*LCD*). Επίσης, παρατηρήστε τον κώδικα χειρισμού interrupt, που ενεργοποιείται μόλις συμβεί κάποια αλλαγή στο rotary knob.
- Μεταφορτώστε το αρχείο **control-incomplete.psm** , το οποίο περιέχει την assembly που θα χρησιμοποιήσετε για τη δημιουργία του αρχείου της ROM. Τροποποιήστε τις κατάλληλες γραμμές που έχουν σημειωθεί με το TODO. Τις τιμές μπορείτε να τις βρείτε με τη μελέτη του προηγούμενου αρχείου μαζί με τη γνώση των εντολών assembly από το εγχειρίδιο χρήσης του picoblaze. Αποθηκεύστε το νέο αρχείο με το όνομα **control.psm** .
- Μεταφορτώστε τα αρχεία KCPSM3.EXE και ROM_form.vhd , τα οποία θα βοηθήσουν στη δημιουργία του αρχείου lrrom.vhd. Τοποθετήστε αυτά τα 2 αρχεία, όπως και το αρχείο που control.psm σε έναν κατάλογο που μπορείτε να μεταβείτε εύκολα από τη γραμμή εντολών. Από τη γραμμή εντολών μεταβείτε στον κατάλογο που έχετε αποθηκεύσει τα αρχεία και δώστε την εντολή **KCPSM3.EXE control.psm**. Αν εκτελεστεί χωρίς σφάλμα (*παρατηρήστε προσεκτικά την έξοδο του προγράμματος*), θα δείτε στο τέλος ότι έχει δημιουργηθεί με επιτυχία το αρχείο **control.vhd** (*όπως και άλλα αρχεία*).
- Δημιουργήστε ένα νέο project στο Xilinx IDE με τις κατάλληλες ρυθμίσεις για την πλακέτα του εργαστηρίου, κάντε σύνθεση και δημιουργία του bitstream, προγραμματίστε την πλακέτα και επιβεβαιώστε την ορθή λειτουργία.

Παραδοτέα: Τα αρχεία (c5) **control.vhd**, (c6) **left_right_leds-lcd.ucf**, (a9) screenshot από το σχηματικό (*schematic*), και (a10) από το “device utilization summary” ή το “Resource Estimation”.