



Μηχανή μηνυμάτων

Νικόλαος Καββαδίας
nkavn@physics.auth.gr

Rev. 1.0
June 7, 2010

1. Εισαγωγή

Αντικείμενο της πειραματικής άσκησης είναι η υλοποίηση μιας απλής μηχανής μηνυμάτων σε LCD 2x16 χαρακτήρων με τη βοήθεια του επεξεργαστή PicoBlaze, σε FPGA Spartan-3E. Η αναπτυξιακή πλακέτα που θα χρησιμοποιηθεί στην άσκηση είναι το board που συνοδεύει το “Spartan-3E Starter Kit” (S3ESK) [1]. Ο προγραμματισμός του FPGA γίνεται με περιγραφή κυκλώματος στην γλώσσα περιγραφής υλικού VHDL.

Η άσκηση χωρίζεται σε δύο μέρη. Στο μέρος 1 (part 1) δίνεται στους φοιτητές έτοιμο κύκλωμα απεικόνισης μηνύματος στο LCD του board [1, κεφ. 5] ώστε την εξοικείωσή τους με τα εξής:

- Την χρήση του επεξεργαστή PicoBlaze ως ελεγκτή γενικού σκοπού (όπως για περιφερειακές συσκευές), στην προκειμένη περίπτωση ως ελεγκτή του LCD 2x16 χαρακτήρων. Η άσκηση είναι οργανωμένη ώστε να μην απαιτούνται γνώσεις της αρχιτεκτονικής PicoBlaze, ενώ το πρόγραμμα οδήγησης του LCD δίνεται στους φοιτητές και είναι κοινό για τα 2 μέρη της άσκησης.
- Την κατανόηση της χρήσης μνήμης ROM για την αποθήκευση των χαρακτήρων του μηνύματος.

Στο μέρος 2 της άσκησης (part 2) ζητείται από τους φοιτητές να τροποποιήσουν το δοσμένο κύκλωμα ώστε την ολίσθηση του μηνύματος με 2 διαφορετικούς τρόπους: κατά μοτίβο “zig-zag” και μοτίβο “slider”.

Στα πλαίσια του δεύτερου μέρους της άσκησης δίνεται προτεινόμενο διάγραμμα βαθμίδων στο οποίο επισημαίνονται οι τροποποιήσεις ως προς το πρότυπο κύκλωμα του μέρους 1.

2. Μέρος 1: Μηχανή σταθερού μηνύματος

Στο πρώτο μέρος της άσκησης δίνεται σε πηγαίο κώδικα VHDL η περιγραφή του κυκλώματος για μηχανή σταθερού μηνύματος. Στο Σχήμα 1 δίνεται η ιεραρχία των αρχείων για το μέρος 1.

```
\fpgalab\msgmach\kcpsm3
control.psm : Πρόγραμμα οδήγησης του LCD για τον PicoBlaze
KCPSM3.EXE  : Συμβολομεταφραστής για τον PicoBlaze (εναλλακτικό
όνομα, KCPSM3)
ROM_form.coe : Xilinx COE πρότυπο αρχείο για την μνήμη
προγράμματος
ROM_form.v  : Verilog πρότυπο αρχείο για την μνήμη προγράμματος
ROM_form.vhd : VHDL πρότυπο αρχείο για την μνήμη προγράμματος

\fpgalab\msgmach\part1
control.vhd : Μνήμη προγράμματος για τον PicoBlaze που παρήχθη
με χρήση του “KCPSM3.EXE” από το πρόγραμμα “control.psm”.
kcpsm3.vhd  : Περιγραφή του PicoBlaze για Xilinx Spartan-3 FPGA.
lcd_cntrl_pkg.vhd : VHDL package με κωδικοποιήσεις για τους
απεικονίσημους ASCII χαρακτήρες από το LCD.
```

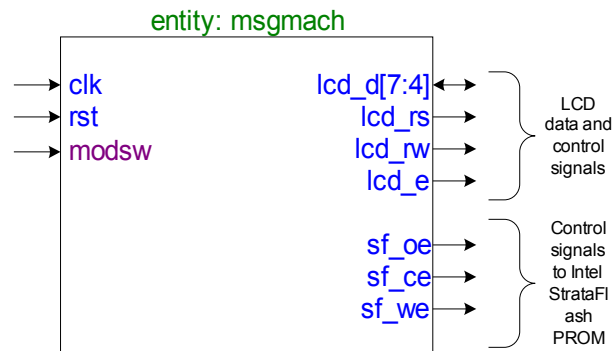
msg_rom.vhd : Περιγραφή ROM 32 θέσεων των 8-bit για την αποθήκευση του μηνύματος.
msgmach1.vhd : Top-level αρχείο του κυκλώματος.
msgmach1_s3esk.ucf : User-Constraint File (UCF αρχείο) για την αντιστοίχιση ακροδεκτών του FPGA με σήματα εισόδου/εξόδου του top-level αρχείου.

Σχήμα 1: Ιεραρχία αρχείων πηγαίου κώδικα για το μέρος 1 της άσκησης.

Η διεπαφή εισόδου/εξόδου της μηχανής μηνυμάτων δίνεται στο Σχ. 2 ενώ ο Πίνακας 1 καταγράφει λεπτομερώς τις θύρες εισόδου/εξόδου και τις ιδιότητές τους.

Πίνακας 1: Διεπαφή εισόδου/εξόδου για τη μηχανή μηνυμάτων.

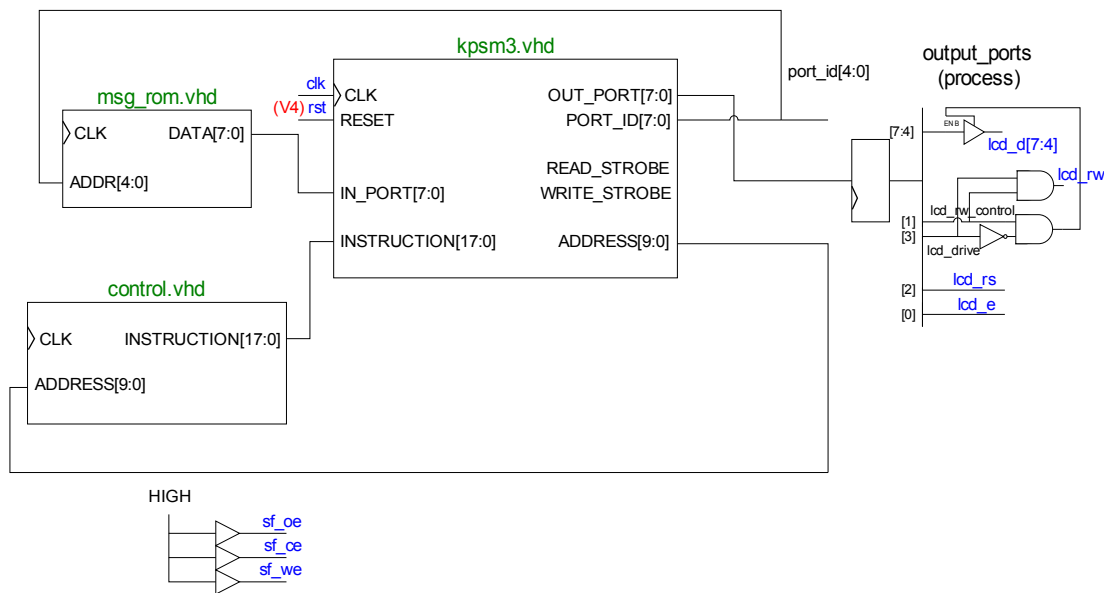
Θύρα	Εύρος	Κατεύθυνση	Περιγραφή
clk	1	Input	Master clock input
rst	1	Input	Asynchronous active high reset
modsw	1	Input	Toggle between shifting mode (part 2 only).
lcd_d[7:4]	4	Input/Output	LCD data bits (upper nibble for the 4-bit interface)
lcd_e	1	Output	Read/Write Enable Pulse 0: Disabled 1: Read/Write operation enabled
lcd_rs	1	Output	Register Select 0: Instruction register during write operations. Busy Flash during read operations 1: Data for read or write operations
lcd_rw	1	Output	Read/Write control 0: WRITE, LCD accepts data 1: READ, LCD presents data
sf_ce	1	Output	StrataFlash Chip Enable. Connects to FPGA pin LDC0 to support the BPI configuration.
sf_oe	1	Output	StrataFlash Chip Enable. Connects to FPGA pin LDC1 to support the BPI configuration.
sf_we	1	Output	StrataFlash Write Enable. Connects to FPGA pin HDC to support the BPI configuration.



Σχήμα 2: Η μηχανή σταθερού μηνύματος ως μονάδα.

Το διάγραμμα βαθμίδων της μηχανής σταθερού μηνύματος φαίνεται στο Σχ. 3. Το κύκλωμα αποτελείται από τρεις επιμέρους μονάδες.

- `kpsm3.vhd` συμβολίζει τον PicoBlaze core [2]. Ο PicoBlaze είναι ένας 8-bit RISC μικροελεγκτής, του οποίου όλες οι εντολές εκτελούνται σε 2 κύκλους. Ένα ιδιαίτερο χαρακτηριστικό του είναι η χρήση ως 256 8-bit θυρών εισόδου (που συνδέονται με την είσοδο `in_port[7:0]`) και 256 8-bit θυρών εξόδου (που συνδέονται με την έξοδο `out_port[7:0]`) οι οποίες διευθυνσιοδοτούνται από το σήμα `port_id[7:0]`. Οι εντολές που πραγματοποιούν είσοδο και έξοδο είναι οι INPUT και OUTPUT, αντίστοιχα. Τα σήματα `READ_STROBE` και `WRITE_STROBE` γίνονται '1' (logic HIGH) κατά το δεύτερο κύκλο μιας εντολής INPUT και OUTPUT, αντίστοιχα.
- `control.vhd` είναι η μνήμη προγράμματος του PicoBlaze. Οι εντολές του PicoBlaze είναι εύρους 18-bit και αποθηκεύονται σε Block RAM. Χωρίς καμία τροποποίηση, το μέγιστο μέγεθος προγράμματος του PicoBlaze είναι 1024 λέξεις.
- `msg_rom.vhd` είναι μια ROM σύγχρονης ανάγνωσης που υλοποιείται σε Block RAM. Το `signal mem` υλοποιεί τα 32 στοιχεία αποθήκευσης των 8-bit (όσοι είναι και οι ταυτόχρονα απεικονιζόμενες διευθύνσεις του LCD).



Σχήμα 3: Διάγραμμα βαθμίδων της μηχανής σταθερού μηνύματος.

Στο κύκλωμα θέλουμε να απενεργοποιήσουμε την Intel StrataFlash PROM για αυτό και τα σήματα `sf_oe`, `sf_ce`, `sf_we` είναι σταθερά σε logic HIGH. Τα σήματα ελέγχου του LCD λαμβάνονται είτε άμεσα είτε με επεξεργασία από το `out_port` κατά τις πρότυπες ρουτίνες οδήγησης του LCD (από τον Ken Chapman, author του PicoBlaze).

Βασικό ρόλο στην απεικόνιση του αποθηκευμένου μηνύματος στην `msg_rom` έχει η χρήση των θυρών εισόδου και το πώς αυτές διευθυνσιοδοτούνται (σήματα `port_id` και `in_port`). Το `port_id` διευθυνσιοδοτεί τους χαρακτήρες της ROM (διευθύνσεις 0 ως 31 δηλ. 00 ως 1F σε δεκαεξαδικό, για αυτό και γίνεται χρήση των 5 bit χαμηλότερης σημαντικότητας του `port_id`).

Για λόγους πληρότητας, στο Σχ. 4 δίνονται οι υπορουτίνες σε κώδικα συμβολομεταφραστή του PicoBlaze για την απεικόνιση των δύο σειρών του μηνύματος. Στην υπορουτίνα disp_line1, ο καταχωρητής s6 έχει την τρέχουσα διεύθυνση στην mem της msg_rom. Με την εντολή INPUT s5, (s6) το περιεχόμενο της θέσης αυτής στην mem, αντιγράφεται στον s5. Αυτός συγκρίνεται με την τιμή του ASCII χαρακτήρα του CARRIAGE RETURN (πλήκτρο ENTER). Αν έχει την ίδια τιμή, σημαίνει ότι έχουμε τέλος της 1^{ης} σειράς και γίνεται άλμα στη διεύθυνση προγράμματος disp_line1_end (αντίστοιχο label). Όταν δεν συμβαίνει αυτό, η τρέχουσα διεύθυνση αυξάνεται κατά 1 και συγκρίνεται με το 16 (επόμενη θέση μετά την τελευταία στην 1^η σειρά). Όταν η διεύθυνση είναι μικρότερη του 16 (δεκαεξαδικό 10) τότε έχουμε νέα επανάληψη του βρόχου disp_line1_loop. Παρομοίως εξηγείται και η λειτουργία της disp_line2.

```

;*****
*
;LCD text messages
;*****
*
;
;Display content for 1st line starting from the current cursor
position
;
disp_line1: LOAD s6, 00
disp_line1_loop: INPUT s5, (s6)
                COMPARE s5, character_CR
                JUMP Z, disp_line1_end
                CALL LCD_write_data
                ADD s6, 01
                COMPARE s6, 10
                JUMP NZ, disp_line1_loop
disp_line1_end: RETURN
;
;
;Display content for 2nd line starting from the current cursor
position
;
disp_line2: LOAD s6, 10
disp_line2_loop: INPUT s5, (s6)
                COMPARE s5, character_CR
                JUMP Z, disp_line2_end
                CALL LCD_write_data
                ADD s6, 01
                COMPARE s6, 20
                JUMP NZ, disp_line2_loop
disp_line2_end: RETURN

```

Σχήμα 4: Το πρόγραμμα control.psm για τον PicoBlaze. Όλες οι τιμές γράφονται στο δεκαεξαδικό.

Κατά την εκτέλεση του μέρους 1 της άσκησης, οι φοιτητές θα πρέπει να βλέπουν το μήνυμα της επιλογής τους στο LCD του board (τροποποιήστε την msg_rom για σύνταξη νέου μηνύματος, οι διαθέσιμοι χαρακτήρες δίνονται στο αρχείο lcd_cntr_pkg.vhd).

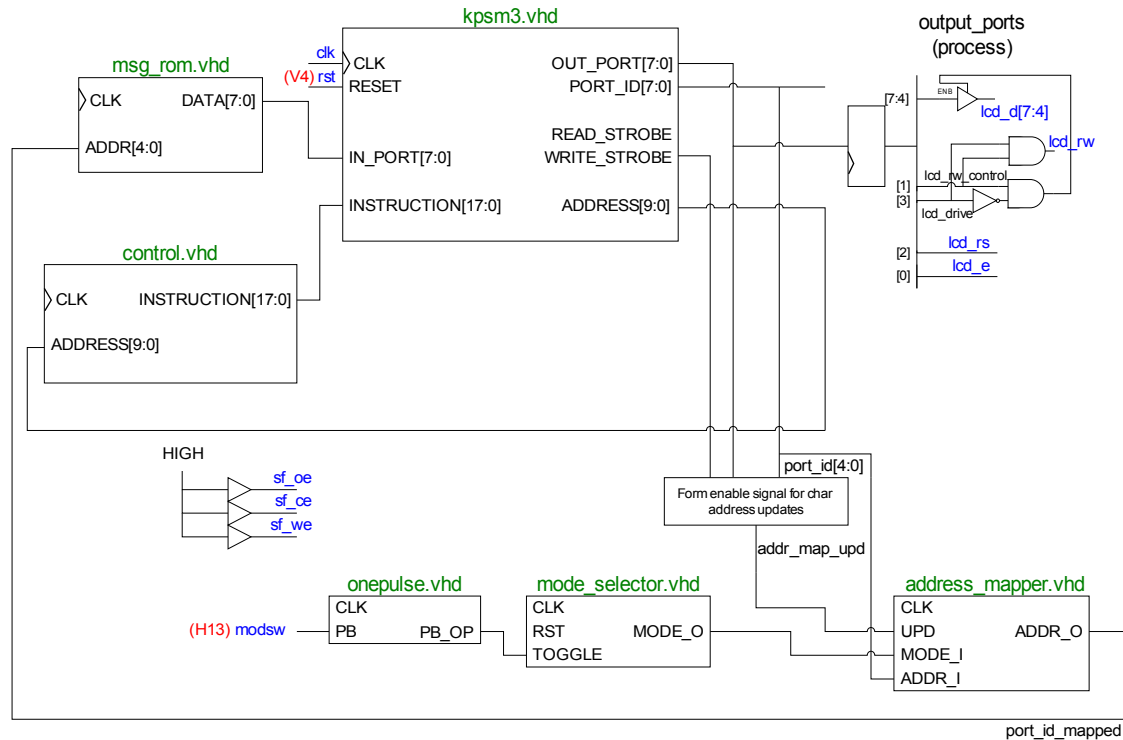
3. Μέρος 2: Μηχανή κινούμενου μηνύματος

Στο μέρος 2 της άσκησης (part 2) ζητείται από τους φοιτητές να τροποποιήσουν το δοσμένο κύκλωμα ώστε την ολίσθηση του μηνύματος με 2 διαφορετικούς τρόπους:

- Κατά μοτίβο “zig-zag” (ο τελευταίος χαρακτήρας της 1^{ης} σειράς στο LCD γίνεται πρώτος χαρακτήρας της 2^{ης} σειράς, όμοια με την ανάγνωση κειμένου από αριστερά προς τα δεξιά)

- Κατά μοτίβο “slider” το οποίο επιτυγχάνεται με ταυτόχρονη ολίσθηση της 1^{ης} και 2^{ης} σειράς από αριστερά προς τα δεξιά.

Το κύκλωμα του μέρους 2 αποτελεί τροποποίηση του κυκλώματος σταθερού μηνύματος του Σχήματος 3. Μια προτεινόμενη υλοποίηση ακολουθεί το διάγραμμα βαθμίδων του Σχήματος 5.



Σχήμα 5: Διάγραμμα βαθμίδων της μηχανής κινούμενου μηνύματος.

Η βασική διαφορά ως προς το Σχήμα 3 είναι ότι η msg_rom τώρα διευθυνσιοδοτείται από την νέα μονάδα address_mapper (έξοδος addr_o, αντίστοιχο σήμα port_id_mapped) και όχι άμεσα από την έξοδο port_id[4:0] του kpsm3 (PicoBlaze). Επίσης, η νέα είσοδος modsw (push-button) χρησιμοποιείται για την εναλλαγή από ακίνητο σε κινούμενο μήνυμα. Το ακίνητο μήνυμα πρέπει να απεικονίζεται στις ίδιες θέσεις χαρακτήρων στο LCD όπως στο μέρος 1 της άσκησης.

Το νέο top-level αρχείο ονομάζεται msgmach2.vhd. Στο top-level υπάρχουν συνολικά οι εξής αλλαγές:

- Προστίθεται μια ανάθεση για το σχηματισμό του σήματος επίτρεψης για την ανανέωση του χάρτη διευθύνσεων στην msg_rom. Η ανάθεση δίνει τιμή στο signal addr_map_upd του Σχ. 5 και για το δεύτερο κύκλο OUTPUT εντολής όταν συμβαίνει να είναι out_port=X"01" και port_id=X"20".
- Προστίθεται το αρχείο onepulse.vhd. Το αντίστοιχο υποκύκλωμα αυτό δέχεται ως είσοδο σήμα από push-button (η είσοδος modsw) και ως έξοδο PB_OP παράγει HIGH για μία περίοδο ρολογιού ανεξάρτητα από την χρονική διάρκεια του HIGH στην είσοδο PB.
- Προστίθεται το αρχείο mode_selector.vhd. Όταν η είσοδος TOGGLE γίνεται ένα, η έξοδος του κυκλώματος πρέπει να αλλάζει κατάσταση, δηλ. να αντιστρέφεται (από '0' γίνεται '1' και το αντίστροφο). Το σήμα current_mode

που αντιστοιχεί στην έξοδο MODE_O καθορίζει την κατάσταση του μηνύματος ('0' για ακίνητο μήνυμα, '1' για κινούμενο).

- Προστίθεται το αρχείο address_mapper.vhd. Όταν η είσοδος UPD είναι '1' δίνεται εντολή νέου υπολογισμού της τρέχουσας διεύθυνσης στην msg_rom. Αυτό σημαίνει ότι θα πρέπει για ακίνητο μήνυμα να είναι port_id_mapped = port_id (όταν είναι MODE_I = '0') αλλά η σχέση να τροποποιείται κατάλληλα για κινούμενο μήνυμα (MODE_I = '1') στον χάρτη των διευθύνσεων.

Να πραγματοποιηθούν τα εξής:

1. Να γραφεί η ανάθεση (concurrent assignment ή process) για τη δημιουργία του σήματος addr_map_upd.
2. Να γραφεί ο κώδικας του αρχείου onepulse.vhd.
3. Να γραφεί ο κώδικας του αρχείου mode_selector.vhd.
ΣΗΜΕΙΩΣΗ: Η λειτουργία που επιτελείται από τον mode_selector θα μπορούσε να ενσωματωθεί στο αρχείο address_mapper.
4. Να γραφεί ο κώδικας του address_mapper για τους δύο διαφορετικούς τρόπους ολίσθησης του μηνύματος. Τα αντίστοιχα αρχεία να ονομαστούν address_mapper1.vhd (zig-zag) και address_mapper2.vhd (slider).
ΣΗΜΕΙΩΣΗ: Για διευκόλυνση των φοιτητών αναφέρεται ότι η διαδοχή των διευθύνσεων για τις δύο περιπτώσεις είναι η εξής:
 - Μοτίβο zig-zag: 00 → 01 ... → 0F → 10 → 11 ... → 1F → 00
 - Μοτίβο slider: 00 → 01 ... → 0F → 00 και ταυτόχρονα: 10 → 11 → ... → 1F → 10
5. Να τροποποιηθεί κατάλληλα το top-level αρχείο (msgmach2.vhd).
6. Να τροποποιηθεί κατάλληλα το UCF αρχείο (msgmach2_s3esk.ucf).
7. Να γίνει synthesis, implementation, FPGA programming για τις δύο εναλλακτικές υλοποιήσεις (πρώτα για address_mapper1.vhd και έπειτα για address_mapper2.vhd).

Κατά την εκτέλεση του μέρους 2 της άσκησης, οι φοιτητές θα πρέπει να βλέπουν το μήνυμα της επιλογής τους στο LCD του board με τους δύο τρόπους ολίσθησης.

4. Αναφορές

[1] Xilinx Inc., UG230: Spartan-3E Starter Kit Board User Guide, v1.0, March 9, 2006.

[2] Xilinx Inc., UG 129: PicoBlaze 8-bit Embedded Microcontroller User Guide for Spartan-3, Virtex-II, and Virtex-II Pro FPGAs, v1.1.1, November 21, 2005.

[3] http://www.xilinx.com/ipcenter/processor_central/picoblaze/index.htm