Εργασία Εξαμήνου για το μάθημα Λειτουργικά Συστήματα

Θέμα: Linux on Xilinx Spartan 3A – Starter Kit (F.P.G.A)

Όνομα: Παππάς Αλέξανδρος
Τμήμα: Τ.Μ.Π.Τ
Α.Μ: 443

Επιβλέπων Καθηγητής: Μηνάς Δασυγένης

Η εργασία αυτή αποτέλεσε εργασία εξαμήνου για το μάθημα Λειτουργικά Συστήματα με καθηγητή τον Δρ.Δασυγένη Μηνά το ακαδημαϊκό έτος 2011-2012 στο 4ο εξάμηνο σπουδών του φοιτητή Παππά Αλέξανδρου του τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών που έχει έδρα στην Κοζάνη.

Στόχοι της εργασίας:

1) Επαφή του φοιτητή με τα βασικά εργαλεία της Xilinx (Project Navigator, XPS, SDK, Impact, Plan Ahead), εξοικίωση με αυτά και χρήση τους στη συνέχεια για την επίτευξη των παρακάτω στόχων.
2) Εμφάνιση στην κονσόλα του SDK της φράσης “Hello World” ως ένδειξη μιας αρχικής εξοικίωσης με τα εργαλεία της Xilinx.
3) Σύνθεση του kernel των Embedded Linux (uClinux) σε περιβάλλον Linux και μεταφορά αυτού στην πλακέτα με απότερο σκοπό τη δυνατότητα χρήσης τερματικού των uClinux τα οποία βρίσκονται στην πλακέτα Xilinx Spartan 3A-Starter Kit.

Στη συνέχεια παρουσιάζονται συνοπτικά τα βήματα τα οποία ακολουθήθηκαν για να πραγματοποιηθούν αυτοί οι στόχοι. Είναι σημαντικό εδώ να σημειωθεί ότι έγινε χρήση των οδηγιών (οι οποίες όπως θα φανεί στη συνέχεια τροποποιήθηκαν μερικώς) από τα εξής αρχεία τα οποία και βρίσκονται στον ίδιο φάκελο με αυτό το έγγραφο:
1) edk_ctt12.3.pdf.
2) edk_ctt11.pdf.
3) ug334.pdf.

Παρακάτω ακολουθεί ο οδηγός που γράφτηκε από τους φοιτητές
Παππά Αλέξανδρο Α.Μ: 443
Κυπαρισσά Νικόλα Α.Μ: 414
όπως ζητήθηκε από τον Δρ.Δασυγένη Μηνά για χρήση του στο μάθημα Ενσωματωμένα Συστήματα. Ο οδηγός περιέχει συνοπτικά τα βήματα που ακολουθήθηκαν από τους 2 φοιτητές για εμφάνιση της φράσης “Hello World” στην κονσόλα του εργαλείου SDK της Xilinx. Προσέξτε ότι έγινε χρήση των αρχείων edk_ctt12.3.pdf και edk_ctt11.pdf που αναφέρθηκαν παραπάνω.
Αφού διαβάσετε την αρχή του αρχείου edk_ctt11.pdf και φτάσετε στο

Chapter 2 "Creating a new project",
ανοίξτε το "ISE Project Navigator"->New Project-><Εισάγετε όνομα>,<Εισάγετε μία διαδρομή καταλόγου (χωρίς κενά)>,<Αφήστε την προεπιλεγμένη γλώσσα σε HDL>,Πατήστε "Next" -> <Evaluation Development Board:"Spartan-3A Starter Kit">,
Σιγουρευτείτε ότι οι υπόλοιπες επιλογές είναι ως εξής:

- Synthesis Tool: "XST(VHDL/Verilog).
- Simulator: "Modelsim-PE VHDL".
- Preferred Language: "VHDL"
- P.S.P: "Store all values".
- Μην επιλέξετε το "Manual Compile Order"
- VHDL S.A.S: "VHDL-93".
- Μην ενεργοποιήσετε το "message filtering".

Πατήστε "Next" -> Πατήστε "Finish".

Ακολουθήστε τις οδηγίες του φυλλαδίου 11, σελίδες 14 (Μετά το "Select Source Type), 15, εκτός από την εξής επιλογή:

- Board Name: "Spartan-3A Starter Kit".

Chapter 3 "Using Xilinx Platform Studio".

page 23 "XPS Tools"
Ακολουθήστε τις οδηγίες της σελίδας 23 για να δημιουργήσετε το "Netlist".
Προσπεράστε τα 2 τελευταία βήματα:
- Hardware > Create or Import Peripheral.
- Software > Generare Libraries and BSPs.

Chapter 4 "Working with Your Embedded Platform".

σελίδα 29: Project > Export Hardware Design to SDK > Export Only.
σελίδες 30, 31: Ακολουθήστε τις οδηγίες ως έχουν.

- Στη σελίδα 31 πριν το βήμα 5, κάντε "Generate Top HDL Source" για να εμφανιστεί η επιλογή "Generate programming file". Στη συνέχεια κάντε "Generate programming file".

Chapter 5 "Introducing the Software Development Kit".

Ανοίξτε το φυλλάδιο 12.3 στη σελίδα 36.
Ανοίξτε το SDK και ακολουθήστε τις οδηγίες ως έχουν από τις σελίδες 36 έως 38.
Στη σελίδα 38 μην προχωρήσετε μετά την επιλογή "Finish".
Με την αναζήτηση των Windows βρείτε το impact και ανοίξτε το.

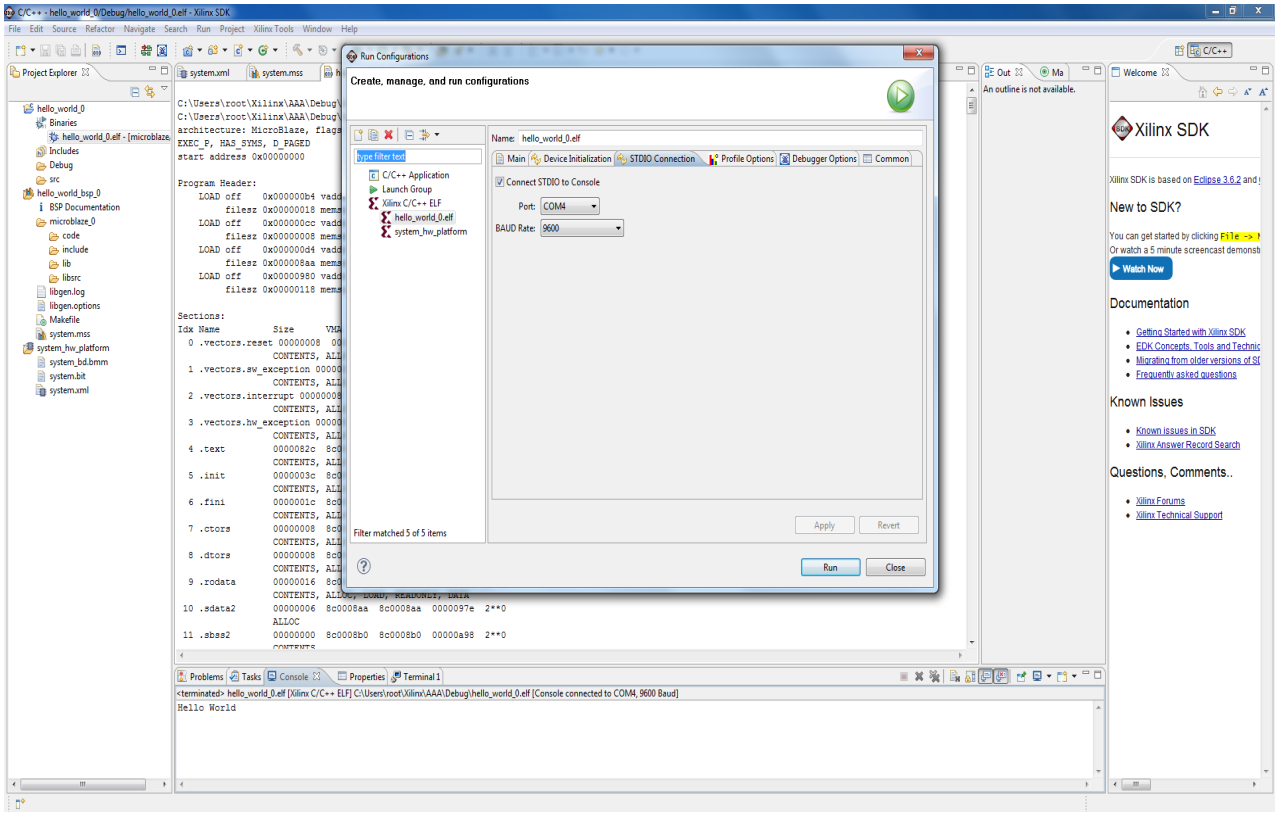
Στο παράθυρο "Automatically create and save project" επιλέξτε "No", στη συνέχεια "create a new project (.ipf)" και έπειτα πατήστε "Οκ".
Στο παράθυρο "Welcome to iMPACT" αφήστε τις επιλογές ως έχουν.
Στην κενή επιφάνεια "Boundary Scan" κάντε δεξί click και επιλέξτε "Add Xilinx Device". Στη συνέχεια εντοπίστε το αρχείο "system.bit".

Τώρα είστε έτοιμοι να περάσετε τον επεξεργαστή σας σε μορφή bitstream στην πλακέτα
FPGA. Για να γίνει αυτό βάλτε όλα τα jumpers όπως αυτά βρίσκονται στην πλακέτα νούμερο 5. Συνδέστε την πλακέτα στον υπολογιστή σας με το καλώδιο USB καθώς και με το καλώδιο Serial. Στις 2 πάνω άκρες της πλακέτας θα δείτε δεξιά τον διακόπτη
"Run-Suspend" και στα αριστερά τον διακόπτη "On-Off". Για να μπορέσετε να περάσετε
το bitstream στην πλακέτα πρέπει ο διακόπτης "Run-Suspend" να βρίσκεται στο Suspend. Στη συνέχεια γυρίστε τον διακόπτη στο "Run".
ΠΡΟΣΟΧΗ: Κάθε φορά που γυρνάτε τον διακόπτη "On-Off" στο "Off" το bitstream χάνεται και πρέπει όπως πριν να ξανακάνετε program.. το οποίο όμως διαρκεί μερικά δευτερόλεπτα. Το program πρέπει να γίνεται με την επιλογή "bootloop" το οποίο σημαίνει πως η πλακέτα είναι έτοιμη να δεχτεί αρχεία .elf, τα οποία παράγονται από

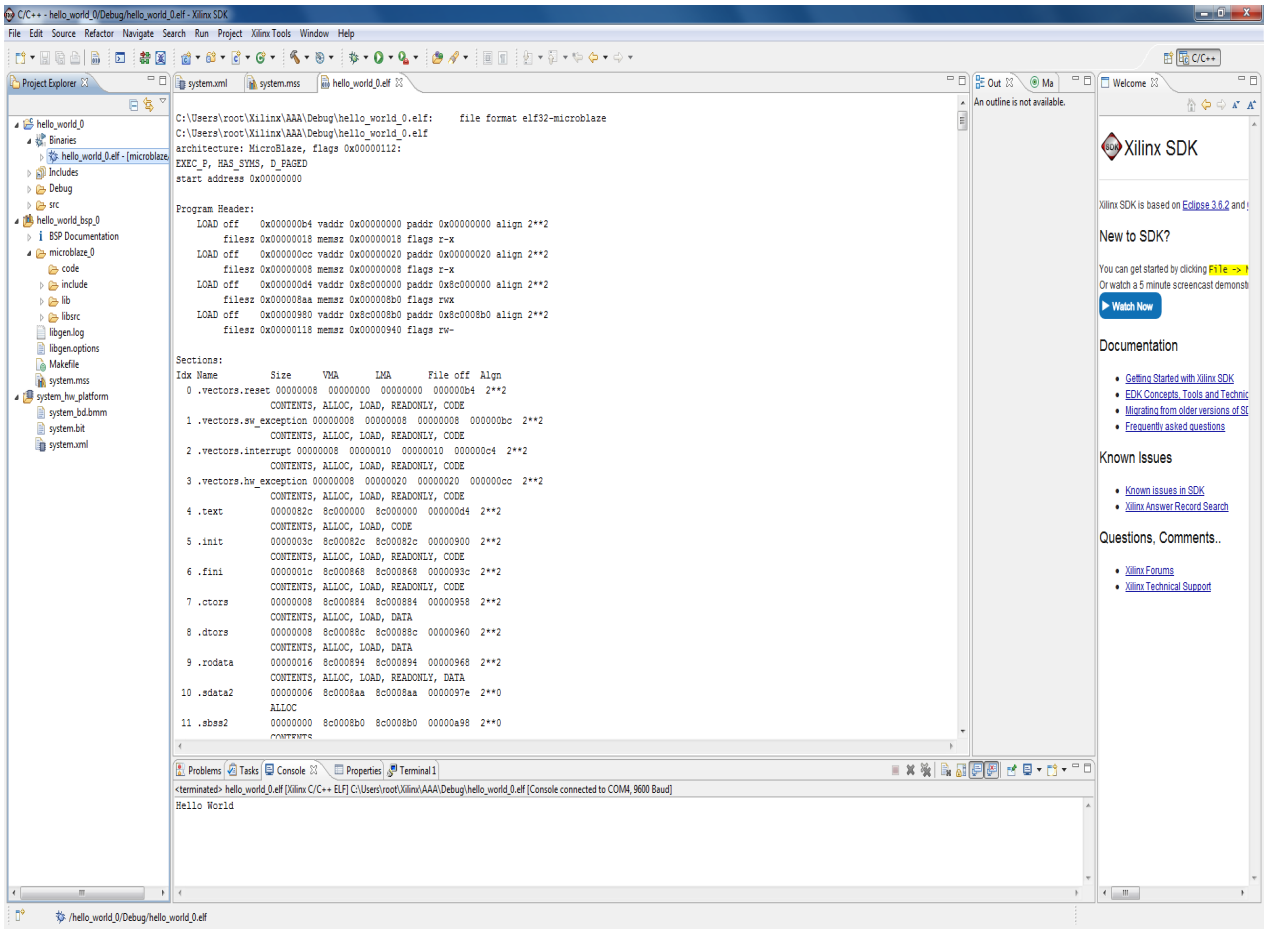
την διαδικασία compilation των αρχείων .c.

Chapter 5.1 "Load, Run, Debug C/C++ Projects on your FPGA".

Ανοίξτε το SDK και έπειτα επιλέξτε Run -> Run Configuration -> Double Click "Xilinx C/C++ELF -> STDIO Connection -> <Check "Connect STDIO to Console>, <Port: COM <Βρείτε σε ποιο COM αντιστοιχεί το Serial>>, <BAUD Rate "9600"> -> Apply -> Run.



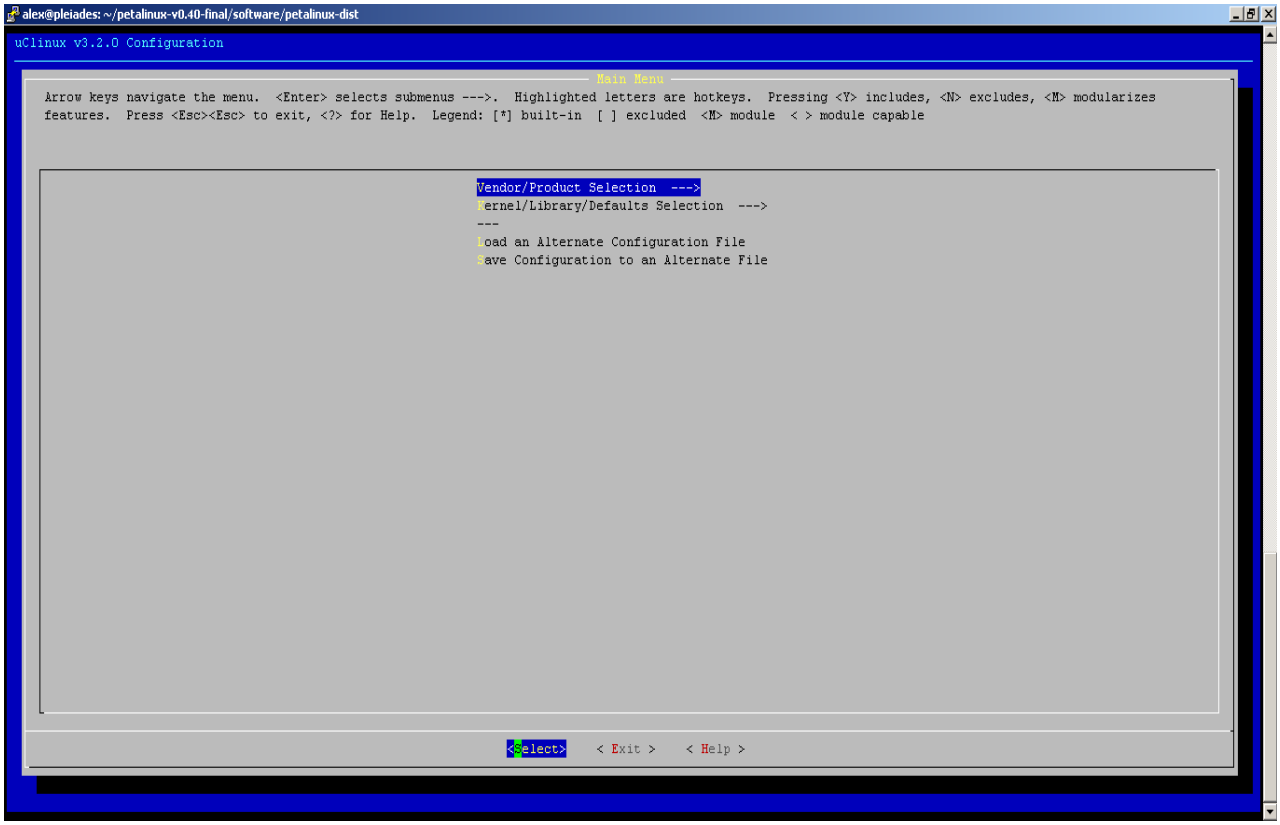
Θα πρέπει στην κονσόλα του SDK να εμφανιστεί το μήνυμα "Hello World".



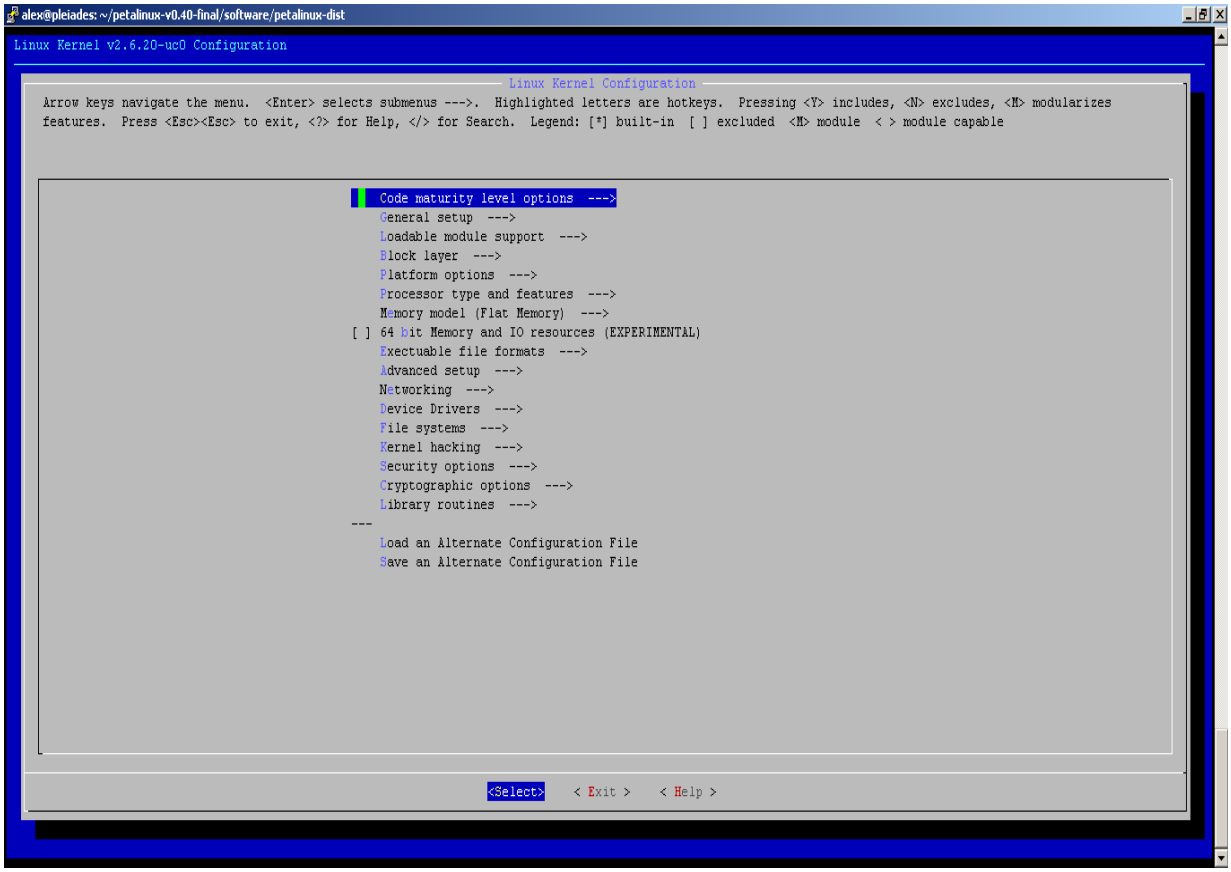
Στη συνέχεια ακολουθούν τα απαραίτητα βήματα για την σύνθεση του πυρήνα (kernel) των uClinux. Είναι σημαντικό ε΄δω να σημειωθεί ότι η σύνθεση του πυρήνα αυτού έγινε στον server του τμήματος Τ.Μ.Π.Τ ονόματι “Pleiades” με σκοπό τη μείωση του χρόνου σύνθεσης χάρις στις πολύ υψηλές επιδόσεις του συγκεκριμένου συστήματος.

Αφού τα αρχεία βρίσκονται στον Βασικό Κατάλογο (Home Directory) προχωρούμε ως εξής:

1) Αποσυμπίεση του αρχείου petalinux-v0.40-final.tar.gz με την εντολή “tar –zxvf petalinux-v0.40-final.tar.gz”.
2) Μετακίνηση στον κατάλογο “petalinux-v0.40-final” με την εντολή “cd petalinux-v0.40-final”.
3) Θέσιμο των περιβαλλοντικών μεταβλητών (βήμα που πρέπει να επαναλαμβάνεται σε κάθε νέα συνεδρία (session)) με την εντολή “source settings.sh”.
4) Μετακίνηση στον κατάλογο “software” (όπως και παραπάνω) και έπειτα στον κατάλογο “petalinux-dist”.
5) Πληκτρολόγηση της εντολής “make menuconfig” και δεδομένου ότι στο σύστημα είναι εγκατεστημένες οι βιβλιοθήκες Ncurses θα εμφανιστεί το παράθυρο όπως φαίνεται παρακάτω.



6) Επιλογή του “Vendor/Product Selection” όπου επιλέγεται ως Vendor η Xilinx και ως “Targeted Product” (προϊον που θέλουμε να χρησιμοποιήσουμε), η πλακέτα “Spartan3E1600-RevA-lite-edk101” και έπειτα “exit”.
7) Επιλογή “Kernel/Library/Defaults Selection” και στη συνέχεια επιλογή του “Customize Kernel Settings”, τέλος επιλέγουμε 2 φορές “exit” και στην προτροπή αποθήκευση των ρυθμίσεων ή όχι επιλέγουμε ναι.
Στη συνέχεια θα εμφανιστεί το παρακάτω παράθυρο όπου και θα γίνουν οι ρυθμίσεις για τον πυρήνα των uClinux.



Χάριν ευκολίας, στη συνέχεια θα αναφερθούν μόνο το ποιες επιλογές έμειναν επιλεγμένες. Προσέξτε ότι κάθε γραμμή ξεκινά με επιλογή του βασικού καταλόγου. Κατά συνέπεια όποτε η αρχή της γραμμής αλλάζει σημαίνει ότι έχουμε επιστρέψει στον αρχικό κατάλογο που φαίνεται στην παραπάνω φωτογραφία.

1) "Code maturity level options" → "Prompt for development and/or incomplete code/drivers".
2) "General Setup" → "Automatically append version info to the version string"
3) "General Setup" → "Kernel .config support".
4) "General Setup" → "Create deprecated sysfs files".
5) "General Setup" → "Optimize for size (Look out for broken compilers!)".
6) "General Setup" → "Configure standard kernel features (for small systems)".
7) "General Setup" → "Configure standard kernel features (for small systems)" → "Sysctl syscall support".
8) "General Setup" → "Configure standard kernel features (for small systems)" → "Enable ELF core dumbs".
9) "General Setup" → "Configure standard kernel features (for small systems)" → "Use full SLAB allocator".
10) "General Setup" → "Configure standard kernel features (for small systems)" → "Enable VM event counters for /proc/vmstat".
11) "Block layer" → "Enable the block layer".
12) "Processor type and features" → "Allow allocating large blocks (>1 MB) of memory".

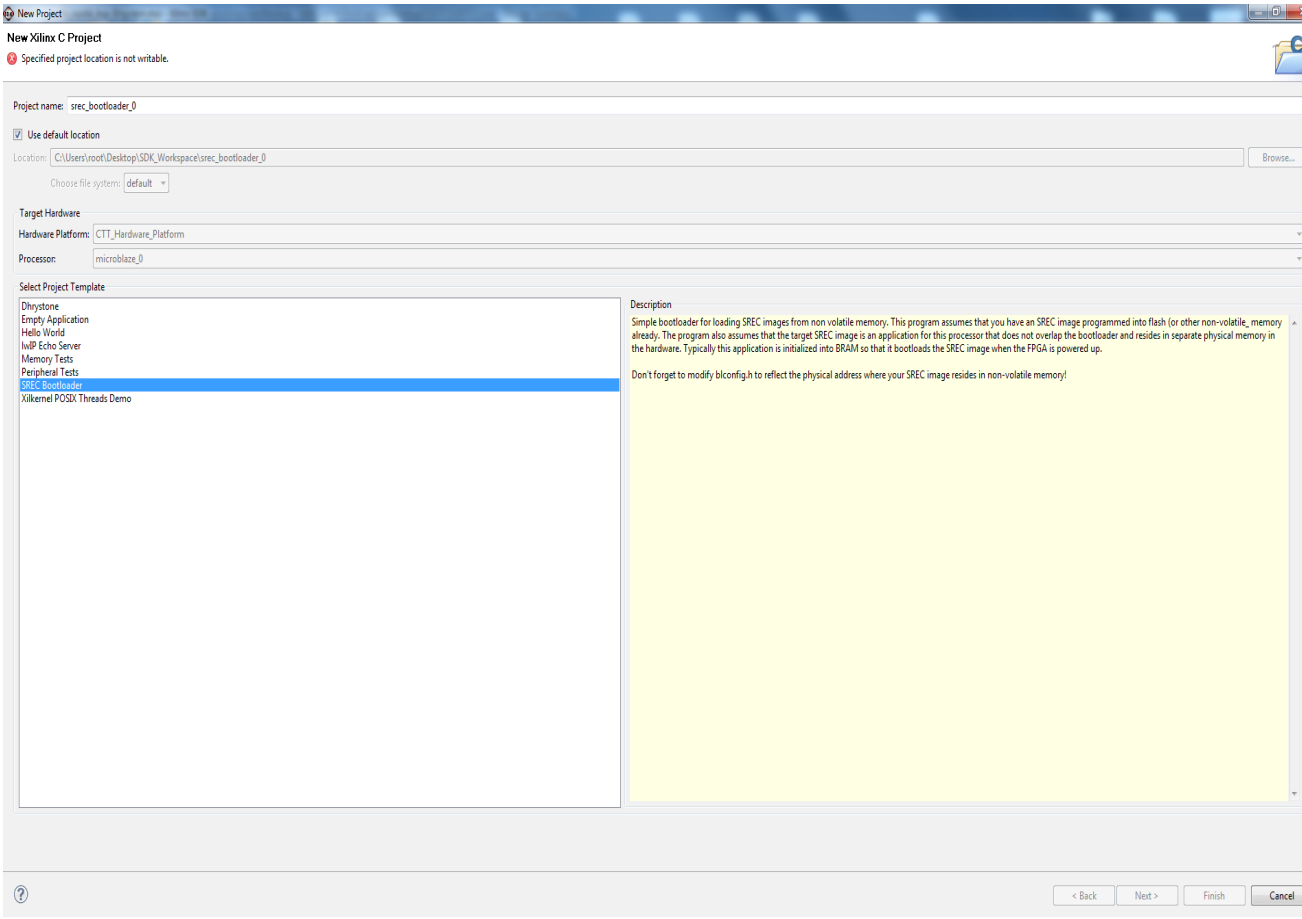
13) “Memory model” → “Flat memory”.
14) “Executable file formats” → “Kernel support for flat binaries”.
15) “Device Drivers” → “Generic Driver Options” → “Select only drivers that don’t need compile-time external firmware”.
16) “Device Drivers” → “Generic Driver Options” → “Prevent firmware from being built”.
17) “Device Drivers” → “Memory Technology Devices (MTD)” → “MTD concatenating support”.
18) “Device Drivers” → “Memory Technology Devices (MTD)” → “Command line partition table parsing”.
19) “Device Drivers” → “Memory Technology Devices (MTD)” → “Direct char device access to MTD devices”.
20) “Device Drivers” → “Memory Technology Devices (MTD)” → “Caching block device access to MTD devices”.
21) “Device Drivers” → “Memory Technology Devices (MTD)” → “RAM/ROM/Flash chip drivers → “Detect flash chips by Common Flash Interface (CFI) probe”.
22) “Device Drivers” → “Memory Technology Devices (MTD)” → “RAM/ROM/Flash chip drivers → “Flash chip driver advanced configuration options”.
23) “Device Drivers” → “Memory Technology Devices (MTD)” → “RAM/ROM/Flash chip drivers → “Support for RAM chips in bus mapping”.
24) “Device Drivers” → “Memory Technology Devices (MTD)” → “Mapping drivers for chip access” → “Support non-linear mappings of flash chips”.
25) “Device Drivers” → “Memory Technology Devices (MTD)” → “Mapping drivers for chip access” → “CFI Flash device in physical memory map”.
26) “Device Drivers” → “Memory Technology Devices (MTD)” → “Mapping drivers for chip access” → “Generic uClinux RAM/ROM filesystem support”.
27) “Device Drivers” → “Memory Technology Devices (MTD)” → “Mapping drivers for chip access” → “uClinux RAM/ROM filesystem is located at ebss”.
28) “Device Drivers” → “Block devices” → “RAM disk support”.
29) “Device Drivers” → “Serial drivers” → “Xilinx uartlite serial port support”.
30) “Device Drivers” → “Serial drivers” → “Support for console on Xilinx uartlite serial port”.
31) “Device Drivers” → “Character devices” → “Unix98 PTY support”.
32) “Device Drivers” → “Character devices” → “Legacy (BSD) PTY support”.
33) “Device Drivers” → “Character devices” → “Hardware Random Number Generator Core support”.
34) “Device Drivers” → “Character devices” → “Xilinx GPIO support”.
35) “File systems” → “Second extended fs support”.
36) “File systems” → “ROM file system support”.
37) “File systems” → “Direct I/O support”.
38) “File systems” → “Pseudo filesystems” → “/proc file system support”.
39) “File systems” → “Pseudo filesystems” → “Sysctl support (/proc/sys).
40) “File systems” → “Pseudo filesystems” → “sysfs file system support”.
41) “File systems” → “Pseudo filesystems” → “Virtual memory file system support (former shm fs).

Προσοχή: Στον πυρήνα των uClinux περιέχονται οι επιλογές 38-41, αν όμως δεν επιλεχτούν η διαφορά στο μέγεθος του πυρήνα είναι πολύ μικρή και ανώφελη. Κατά συνέπεια κράτησα αυτές τις επιλογές. Δεν υποστηρίζω πως αυτές είναι οι καλύτερες δυνατές επιλογές καθώς η εμπειρία μου στα λειτουργικά συστήματα είναι περιορισμένη και για την επιλογή ή όχι των περισσότερων κατέφυγα στο Google για πληροφορίες σχετικά με τις επιλογές. Σίγουρα όμως ο πυρήνας των uClinux δε μπορεί να περιοριστεί πολύ περισσότερο και να καταλήξει σε μία λειτουργική μορφή που να «χωράει» στην πλακέτα Xilinx Spartan 3A Starter-Kit. Έτσι ολοκληρώνεται ένα μεγάλο μέρος μέρος της συγκεκριμένης εργασίας. Το επόμενο βήμα ήταν το λειτουργικό αυτό σύστημα αφού συντέθηκε, να περαστεί στην πλακέτα Xilinx Spartan 3A Starter-Kit όπου και θα έπρεπε να αποδειχτεί λειτουργικό. Δυστυχώς όμως η ολοκλήρωση της συγκεκριμένης εργασίας στην πλακέτα αυτή αποδείχτηκε αδύνατη. Στη συνέχεια παρουσιάζονται οι εναλλακτικές που υπήρχαν και γιατί καμία από αυτές δεν μπόρεσε να υπηρετήσει τον σκοπό της εργασίας.

Εναλλακτικές:
1) Netboot. Μετά τη σύνθεση του πυρήνα των uClinux δημιουργείται ο φάκελος “images” ο οποίος περιέχει τα αρχεία: image.bin, image.elf, image.srec, image.ub, Linux.bin, romfs.img, rootfs.cpio, ub.config.img, u-boot.bin, u-boot.elf, u-boot-s.bin, u-boot-s.elf, u-boot.srec, u-boot-s.srec, όπως φαίνεται στην παρακάτω εικόνα. (screenshot).

Τα αρχεία αυτά αντιγράφονται και στον φάκελο “tftpboot”. Σύμφωνα λοιπόν με τις οδηγίες του φυλλαδίου “embedded_linux_xupv5_lab_manual.pdf αφού ήδη βρισκόμαστε στον φάκελο αυτό πληκτρολογούμε την εντολή /sbin/ifconfig για να σιγουρευτούμε ότι η I.P του host είναι 192.168.0.1. Έπειτα ενεργοποιούμε την πλακέτα και παρακολουθούμε την διαδικασία boot στο “Kermit window”. Τέλος για να ξεκινήσει το boot δίνουμε στο τερματικό του u-boot (Kermit) “run netboot”. Το βασικό όμως πρόβλημα με αυτήν την εναλλακτική είναι ότι η μνήμη της πλακέτας Xilinx Spartan 3A Starter-Kit δεν επαρκεί για να φιλοξενήσει τις εικόνες του πυρήνα των uClinux. Κατά συνέπεια είναι αδύνατο το πέρασμα του πυρήνα στην πλακέτα.

2) SREC Bootloader. Από το εργαλείο της Xilinx SDK όπως βλέπετε στην παρακάτω εικόνα.



Από το κείμενο δεξιά της εν λόγω επιλογής

```
"Simple bootloader for loading SREC images from non
volatile memory. This program assumes that you have an
SREC image programmed into flash (or other non-volatile_
memory already. The program also assumes that the target
SREC image is an application for this processor that does
not overlap the bootloader and resides in separate
physical memory in the hardware. Typically this
application is initialized into BRAM so that it bootloads
the SREC image when the FPGA is powered up. Don't forget
to modify blconfig.h to reflect the physical address
where your SREC image resides in non-volatile memory!"
```

καταλαβαίνουμε ότι η εφαρμογή SREC Bootloader υπάρχει ήδη σε εσωτερική μνήμη της πλακέτας όπου υπάρχουν και άλλες εφαρμογές της Xilinx και μέσω του SDK περνάει από την εσωτερική αυτή μνήμη στην SDRAM. Η ειδοποίηση λοιπόν αναφέρεται όχι στο αρχείο image.srec. Πρέπει λοιπόν να περαστεί σε μία serial flash μνήμη το αρχείο image.srec το οποίο περιέχει τον kernel μαζί με το filesystem. Δυστυχώς όμως όσο και αν μικρύνει ο kernel το αρχείο image.srec είναι ακόμη πολύ μεγάλο (~ 5M) για να χωρέσει στην serial Flash μνήμη της πλακέτας Xilinx Spartan 3A-Starter Kit. Κατά συνέπεια και αύτή η εναλλακτική δεν μπορεί να υπηρετήσει τον σκοπό μας.
Σε αυτό το σημείο η εργασία σταματά καθώς οι γνώσεις μου και η εμπειρία μου δεν μπόρεσαν να μου προσφέρουν κάποια άλλη εναλλακτική.