

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΗΥ1901– Ειδικές Αρχιτεκτονικές Υπολογιστών

Εργαστήριο 3
Δημιουργία Πλατφόρμας στο SDSoC και ανάπτυξη
εφαρμογής επιταχυντή

Ν. ΤΑΜΠΟΥΡΑΤΖΗΣ - Π. ΜΟΥΣΟΥΛΙΩΤΗΣ

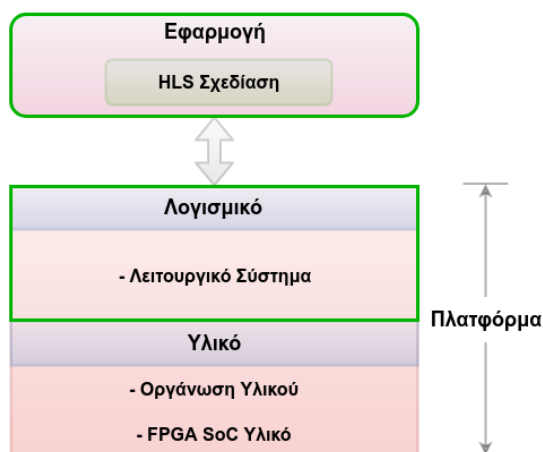
Διδάσκων: Ιωάννης Παπαευσταθίου

Version 0.1

Νοέμβριος 2019

1. Εισαγωγή

Στο εργαστήριο 1 ασχοληθήκαμε με τη σχεδίαση του επιταχυντή μας με τη χρήση του εργαλείου Vivado HLS. Στο εργαστήριο 2 οργανώσαμε το υλικό μέρος της πλατφόρμας που θα χρησιμοποιήσουμε για να διασυνδέσουμε τον επιταχυντή μας. Στο παρόν εργαστήριο θα ασχοληθούμε με την δημιουργία του λογισμικού μέρους της πλατφόρμας μας, τη δημιουργία της πλατφόρμας μας, καθώς και τη χρήση της πλατφόρμας στο SDSoc για την ανάπτυξη εφαρμογής που θα καλεί τον επιταχυντή που σχεδιάσαμε στο εργαστήριο 1. Μπορείτε να βρείτε περισσότερες λεπτομέρειες σχετικά με το SDSoc στο User Guide¹.



Η δημιουργία του λογισμικού μέρους αφορά τη ρύθμιση και την παραγωγή του λειτουργικού συστήματος που θα τρέχει πάνω στην πλατφόρμα. Όσον αφορά την εφαρμογή, αυτή θα είναι μια C++ εφαρμογή που θα χρησιμοποιεί τον επιταχυντή που σχεδιάσαμε στο εργαστήριο 1 με τη χρήση του Vivado HLS.

2. Δημιουργία Λειτουργικού PetaLinux

Το PetaLinux είναι μια διανομή Linux σχεδιασμένη για να τρέχει πάνω στα FPGA SoCs της Xilinx. Η Xilinx περιλαμβάνει εργαλεία για τη δημιουργία και ρύθμιση διανομής PetaLinux για ένα συγκεκριμένο FPGA SoC το οποίο θα έχει κάποια συγκεκριμένη οργάνωση υλικού. Κάνοντας χρήση των επόμενων εντολών, θα δημιουργήσουμε μια PetaLinux διανομή για το FPGA SoC του Ultra96 v1 (το `xczu3eg-sbva484-1-e`) και την οργάνωση υλικού με την οποία

¹ https://www.xilinx.com/html_docs/xilinx2018_2/sdsoc_doc/index.html

ασχοληθήκαμε στο εργαστήριο 2. Για τον σκοπό αυτό, ανοίγουμε ένα τερματικό (terminal emulator) και δίνουμε τις πιο κάτω εντολές:

- `petalinux-create -t project -n <project_name> -s <path_to_base_BSP>`, όπου `<project_name>` το όνομα του PetaLinux project μας (π.χ. ultra96v1) και το `<path_to_base_BSP>` είναι η διαδρομή στο σύστημα αρχείων ενός αρχείου (Board Support Package) που περιλαμβάνει υποστηρικτικά αρχεία για την πλακέτα που χρησιμοποιούμε, όπως drivers και core applications. [e.g. `petalinux-create -t project -n ultra96v1 -s ultra96v1_full_2018_2.bsp`]
- `petalinux-config -p <project_name> --get-hw-description=<HDF_path>`, όπου `<HDF_path>` είναι η διαδρομή στο σύστημα αρχείων του αρχείου που περιγράφει τη Vivado σχεδίαση μας που υλοποιήσαμε στο εργαστήριο 2. Στην ncurses οθόνη που θα ανοίξει απλά επιλέξτε Exit και έπειτα επιλέξτε Yes για να αποθηκεύσετε το configuration. [e.g. `petalinux-config -p ultra96v1 --get-hw-description=.`]
- `petalinux-config -p <project_name> -c kernel`, για να ρυθμίσουμε το kernel της PetaLinux διανομής. Στην ncurses οθόνη που θα ανοίξει κάνουμε τις εξής επιλογές:
 - Device Drivers → Generic Driver Options → Size in MB(1024)
 - Device Drivers → Staging drivers (ON)
 - Device Drivers → Staging drivers → Xilinx APF Accelerator driver (ON)
 - Device Drivers → Staging drivers → Xilinx APF Accelerator driver → Xilinx APF DMA engines support (ON)
 - CPU Power Management → CPU idle → CPU idle PM support (OFF)
 - CPU Power Management → CPU Frequency scaling → CPU Frequency scaling (OFF)

Και έπειτα επιλέγουμε Exit και Yes για να αποθηκεύσουμε το configuration.

- `petalinux-config -p <project_name> -c rootfs`, για να ρυθμίσουμε το σύστημα αρχείων της PetaLinux διανομής. Στην ncurses οθόνη που θα ανοίξει κάνουμε την εξής επιλογή:
 - Filesystem Packages → misc → gcc-runtime → libstdc++ (ON)

Και έπειτα επιλέξτε Exit και Yes για να αποθηκεύσετε το configuration.

- `nano ultra96v1/project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi`, για να προσθέσουμε στο τέλος του αρχείου system-user.dts κώδικα σχετικό με τον Xilinx APF Accelerator driver:

```
/{
    xlnk {
        compatible = "xlnx,xlnk-1.0";
    };
};
```

- `petalinux-build -p <project_name>`, για να κάνουμε build την PetaLinux διανομή.

3. Δημιουργία Πλατφόρμας

Σε αυτό το βήμα θα χρησιμοποιήσουμε αρχεία που δημιουργήθηκαν στο προηγούμενο βήμα καθώς και αρχεία που δημιουργήθηκαν στο εργαστήριο 2 για να δημιουργήσουμε την πλατφόρμα μας μέσα από το SDSoC.

Αρχικά, θα πρέπει να ετοιμάσουμε τα αρχεία του PetaLinux. Για τον σκοπό αυτό, δίνουμε τις εξής εντολές σε ένα τερματικό:

```
cd <petalinux_project>/images/linux
mkdir boot
mkdir image
cp u-boot.elf boot/u-boot.elf
cp *fsbl.elf boot/fsbl.elf
cp bl31.elf boot/bl31.elf
cp pmufw.elf boot/pmufw.elf
cp image.ub image/image.ub
```

Ο κατάλογος boot περιλαμβάνει αρχεία σχετικά με την εκκίνηση του λειτουργικού και ο κατάλογος image περιλαμβάνει αρχεία σχετικά με το σύστημα αρχείων του λειτουργικού. Για να μετατρέψουμε τα αρχεία μέσα στον κατάλογο boot σε ένα αρχείο χρειαζόμαστε ένα αρχείο που περιγράφει αυτή τη διαδικασία μετατροπής και είναι τύπου BIF (Boot Image Format). Για το σκοπό αυτό δίνουμε την εξής εντολή για να δημιουργήσουμε ένα BIF αρχείο:

```
nano file.bif
```

 (είναι σημαντικό το .bif να είναι με μικρά γράμματα)

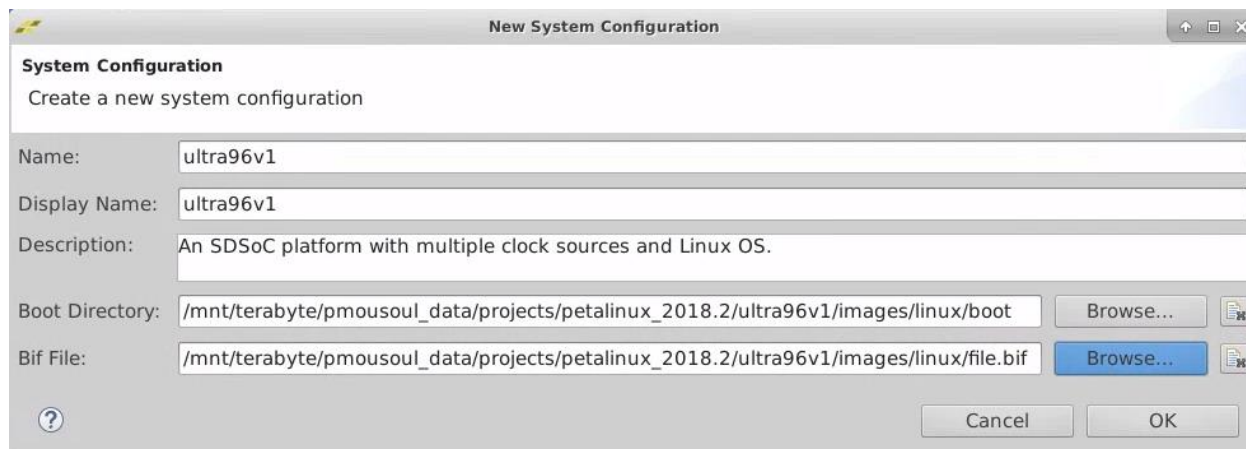
και μεταβάλλουμε τα περιεχόμενά του όπως παρακάτω:

```
the_ROM_image:
{
    [fsbl_config] a53_x64
    [bootloader]<fsbl.elf>
    [pmufw_image]<pmufw.elf>
    [destination_device=pl] <bitstream>
    [destination_cpu=a53-0, exception_level=el-3, trustzone] <bl31.elf>
    [destination_cpu=a53-0, exception_level=el-2] <u-boot.elf>
}
```

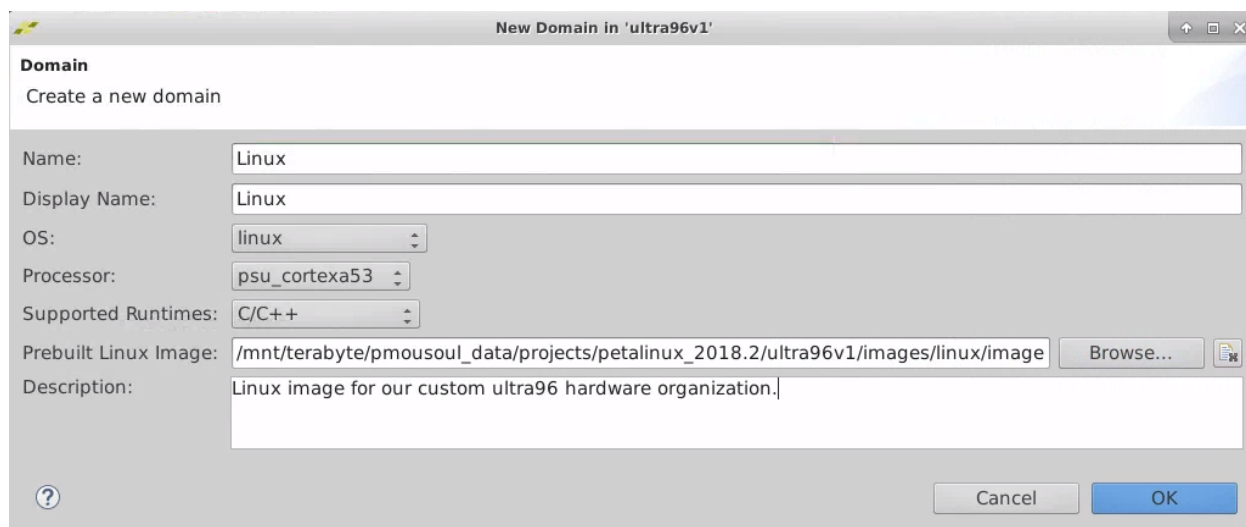
Για να δημιουργήσουμε την πλατφόρμα, ανοίγουμε το SDSoC και επιλέγουμε File -> New SDx Project. Στο παράθυρο που θα ανοίξει, επιλέγουμε Platform και πατάμε Next.

Στο επόμενο παράθυρο θα πρέπει να εισάγουμε την τοποθεσία του DSA αρχείου που δημιουργήσαμε στο εργαστήριο 2. Επίσης επιλέγουμε Import Software Platform Components για να εισάγουμε τα αρχεία που δημιουργήσαμε παραπάνω και στη συνέχεια επιλέγουμε Finish.

Στη συνέχεια πατάμε πάνω στο Define System Configuration και εισάγουμε τις πληροφορίες που φαίνονται στην πιο κάτω εικόνα.



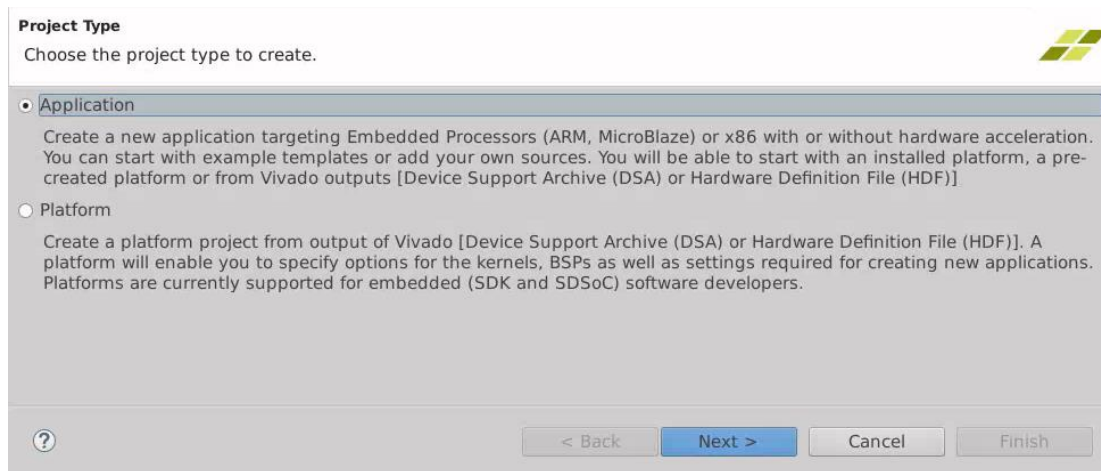
Έπειτα πατάμε OK για να κλείσουμε το παράθυρο, που φαίνεται στην παραπάνω εικόνα, και πατάμε στο Add Processor Group/Domain. Συμπληρώνουμε το παράθυρο που ανοίγει όπως φαίνεται στην πιο κάτω εικόνα:



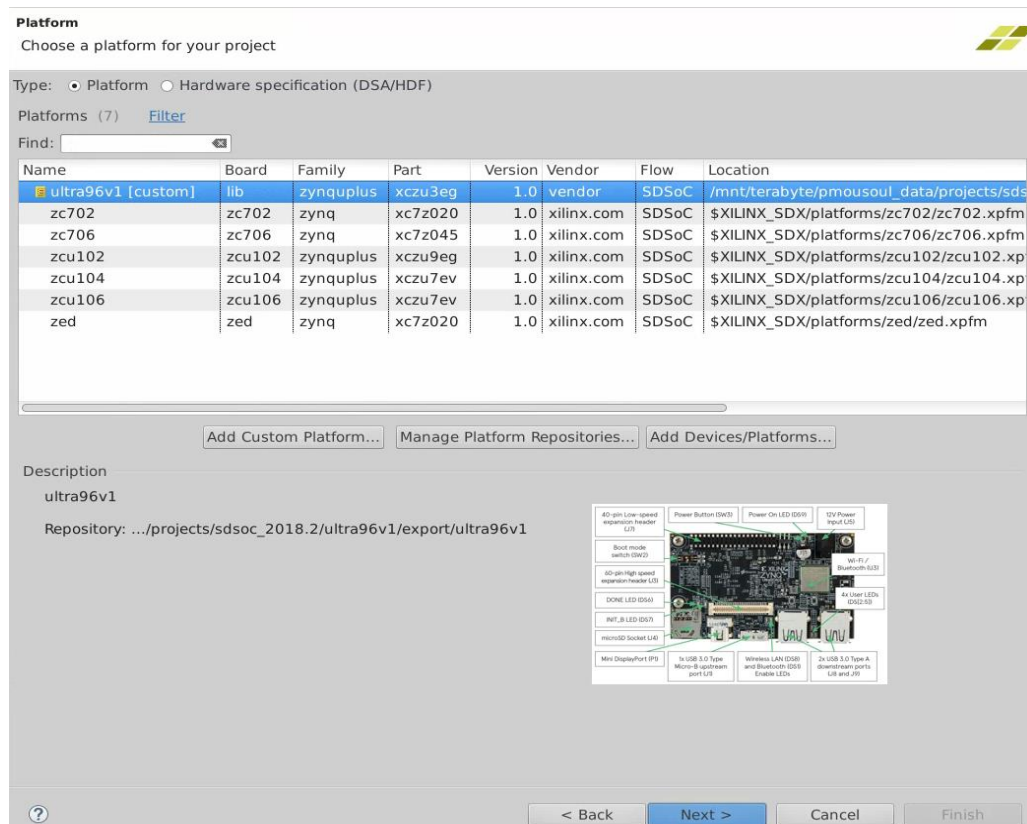
Στη συνέχεια επιλέγουμε το Generate Platform και πατάμε το OK στο παράθυρο που θα ανοίξει. Τέλος επιλέγουμε το Add to Custom Repositories για να κάνουμε την πλατφόρμα διαθέσιμη στο SDSoc και πατάμε OK. Σε αυτό το σημείο είμαστε έτοιμοι για να αναπτύξουμε την εφαρμογή που θα χρησιμοποιεί τον επιταχυντή που σχεδιάσαμε στο εργαστήριο 1.

4. Ανάπτυξη Εφαρμογής

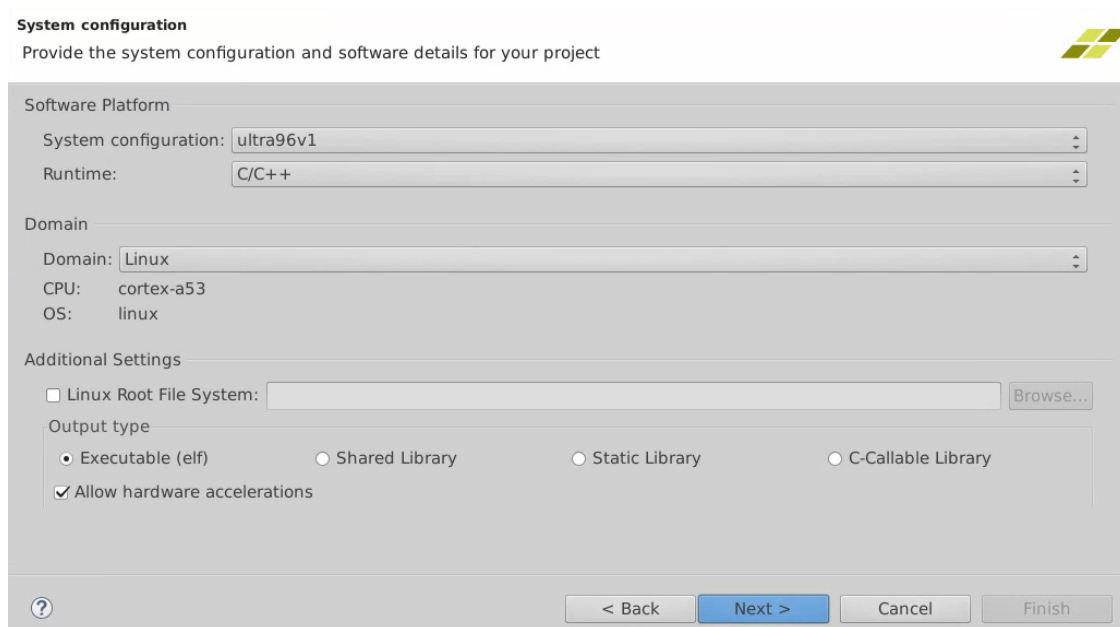
Πατώντας δεξί κλικ πάνω στο project της πλατφόρμας (ultra96v1) στον Project Explorer, επιλέγουμε Close Project. Στη συνέχεια, επιλέγουμε File -> New SDx Project. Στο παράθυρο που θα ανοίξει, επιλέγουμε Application και πατάμε Next.



Στο επόμενο παράθυρο, δίνουμε ένα όνομα στο project μας και επιλέγουμε Next. Έπειτα επιλέγουμε την πλατφόρμα που δημιουργήσαμε στα προηγούμενα βήματα.



Το επόμενο παράθυρο το αφήνουμε ως έχει και πατάμε στο Next για να αναπτύξουμε μια C/C++ εφαρμογή σε λειτουργικό Linux που θα τρέχει στην πλατφόρμα ultra96v1.



The image shows a 'System configuration' dialog box with the following sections:

- Software Platform**
 - System configuration: ultra96v1
 - Runtime: C/C++
- Domain**
 - Domain: Linux
 - CPU: cortex-a53
 - OS: linux
- Additional Settings**
 - ☐ Linux Root File System: [text field] [Browse...]
 - Output type**
 - ☒ Executable (elf)
 - ☐ Shared Library
 - ☐ Static Library
 - ☐ C-Callable Library
 - ☒ Allow hardware accelerations

At the bottom, there is a help icon (?) and four buttons: '< Back', 'Next >', 'Cancel', and 'Finish'.

Στο επόμενο παράθυρο επιλέγουμε Empty Application και πατάμε πάνω στο Finish για να αρχίσουμε την ανάπτυξη της εφαρμογής μας. Πατώντας δεξί κλικ πάνω στον κατάλογο src του project που δημιουργήσαμε μπορούμε να κάνουμε Import αρχεία κώδικα που περιγράφουν τον επιταχυντή μας καθώς και την εφαρμογή που τον χρησιμοποιεί (για τους σκοπούς του εργαστηρίου το testbench) από το 1ο εργαστήριο. Εναλλακτικά, πατώντας δεξί κλικ στον φάκελο src μπορούμε να επιλέξουμε New -> File και να δημιουργήσουμε επί τόπου τα αρχεία κώδικα του επιταχυντή και της εφαρμογής μας.