



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ψηφιακά Συστήματα VLSI

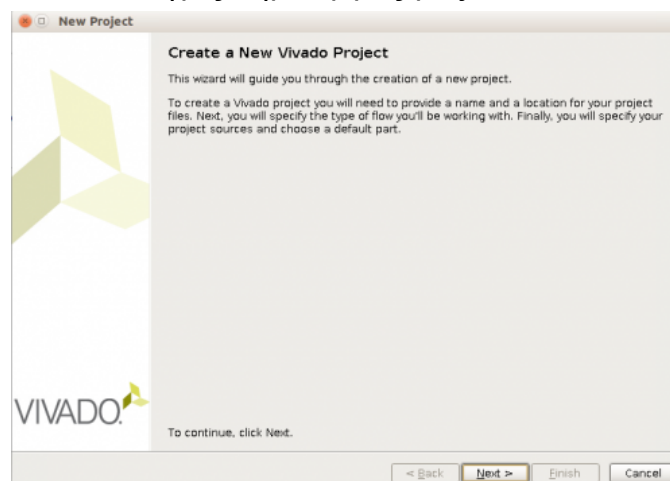
Οδηγός Δημιουργίας Συστήματος Υλικού/Λογισμικού στο ZYBO

I. Δημιουργία νέου Project

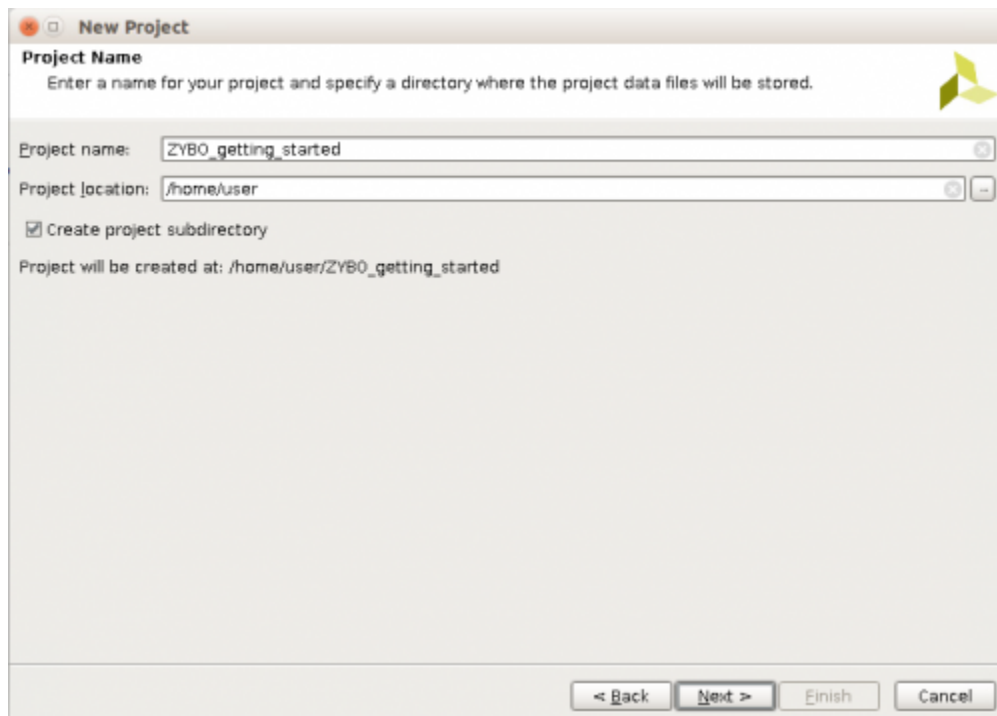
- Κάντε κλικ στο **Create New Project**.



- Θα παρουσιαστεί ο οδηγός δημιουργίας project. Κάντε κλικ στο κουμπί **Next**.



- Εισάγετε ένα όνομα και το path του project και κάντε κλικ στο κουμπί **Next**.



New Project

Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: ZYBO_getting_started

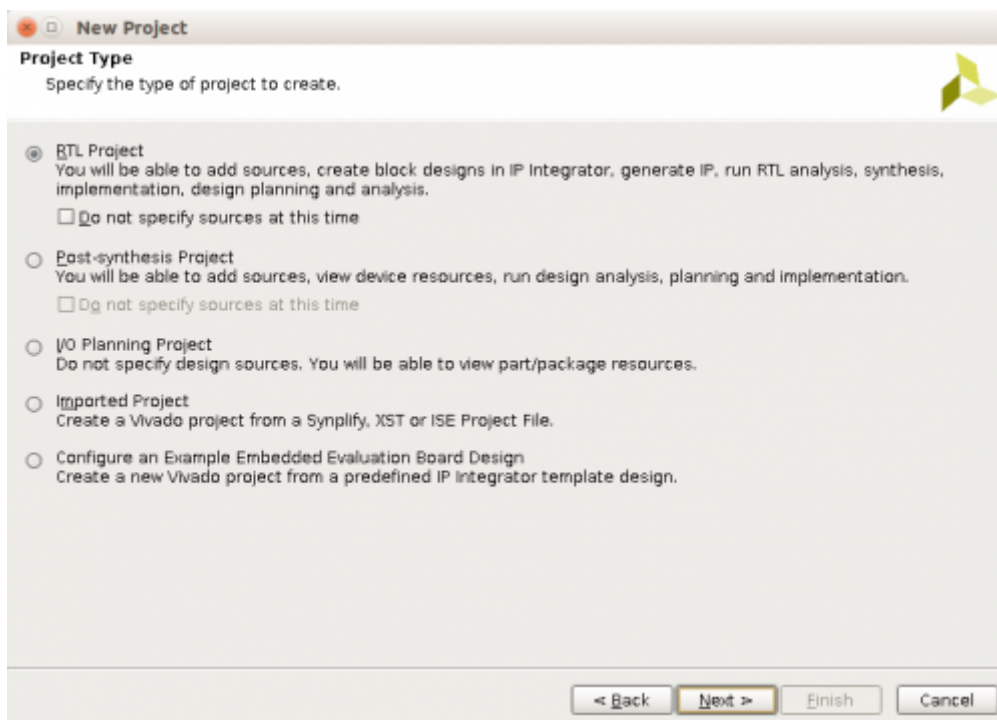
Project location: /home/user

☒ Create project subdirectory

Project will be created at: /home/user/ZYBO_getting_started

< Back Next > Finish Cancel

- Επιλέξτε **RTL Project** και κάντε κλικ στο κουμπί **Next**.



New Project

Project Type
Specify the type of project to create.

☒ **RTL Project**
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
☐ Do not specify sources at this time

☐ **Post-synthesis Project**
You will be able to add sources, view device resources, run design analysis, planning and implementation.
☐ Do not specify sources at this time

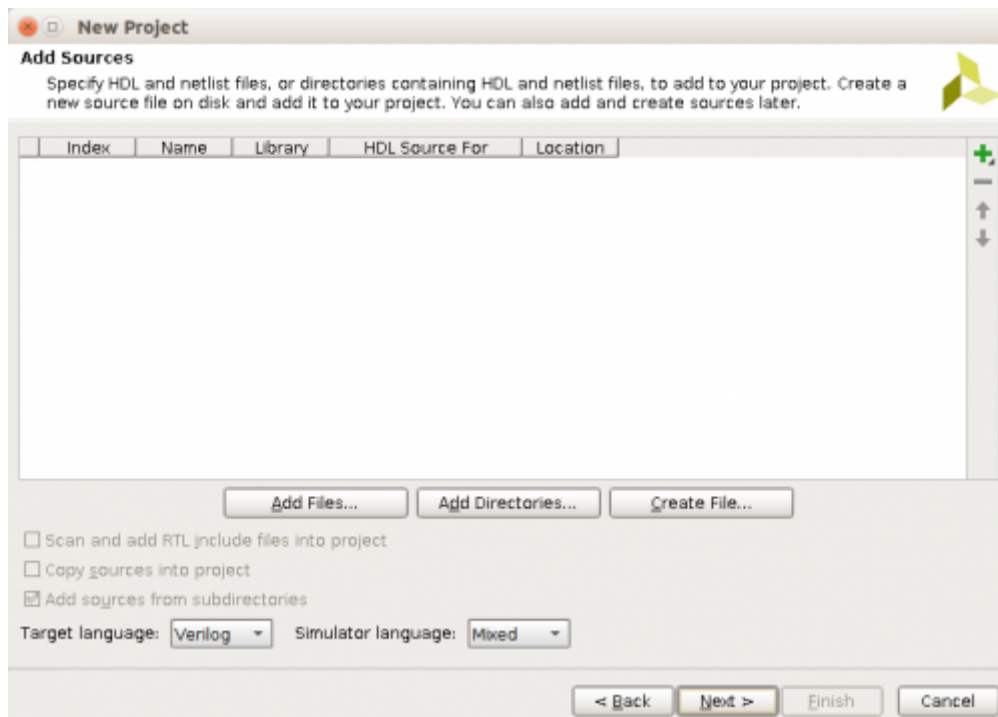
☐ **I/O Planning Project**
Do not specify design sources. You will be able to view part/package resources.

☐ **Imported Project**
Create a Vivado project from a Synplify, XST or ISE Project File.

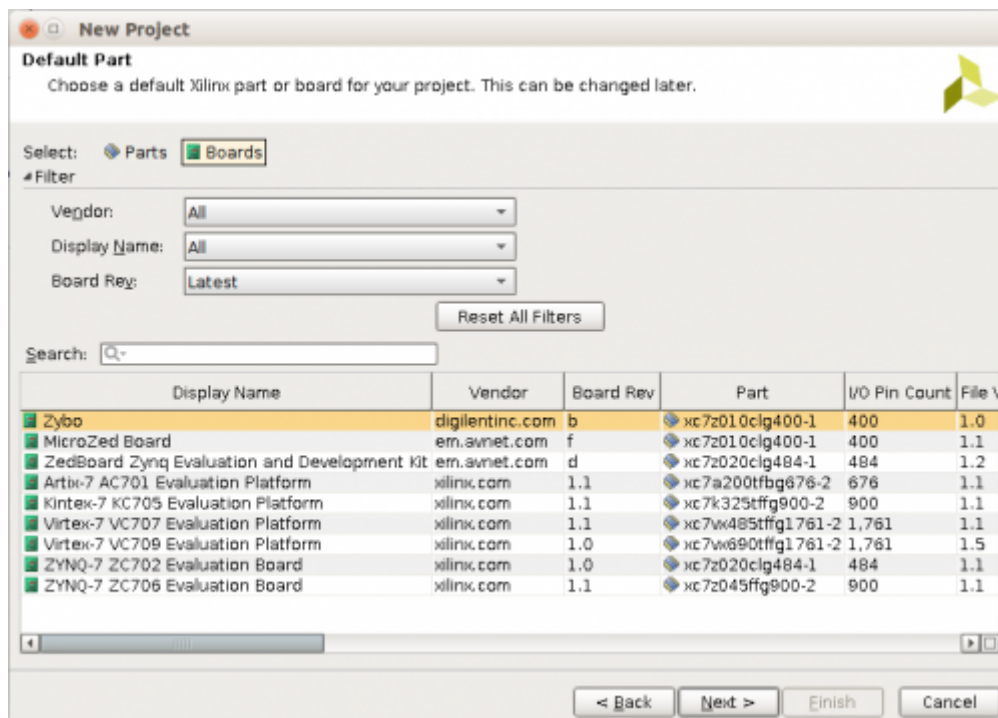
☐ **Configure an Example Embedded Evaluation Board Design**
Create a new Vivado project from a predefined IP Integrator template design.

< Back Next > Finish Cancel

- Αυτό το demo δεν χρησιμοποιεί source files (.vhd), IP cores ή constraint files. Πατήστε **Next** στις επόμενες τρεις οθόνες.

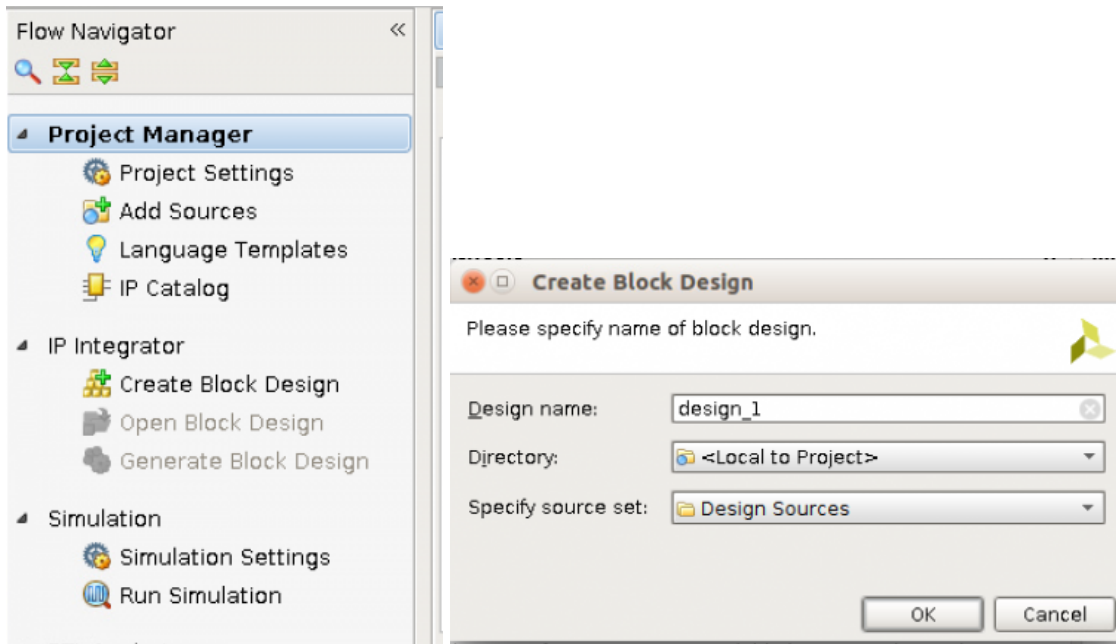


- Επιλέξτε **Boards** και επιλέξτε το αρχείο του **Zybo**. Κάντε κλικ στο κουμπί **Next** και έπειτα στο κουμπί **Finish**.




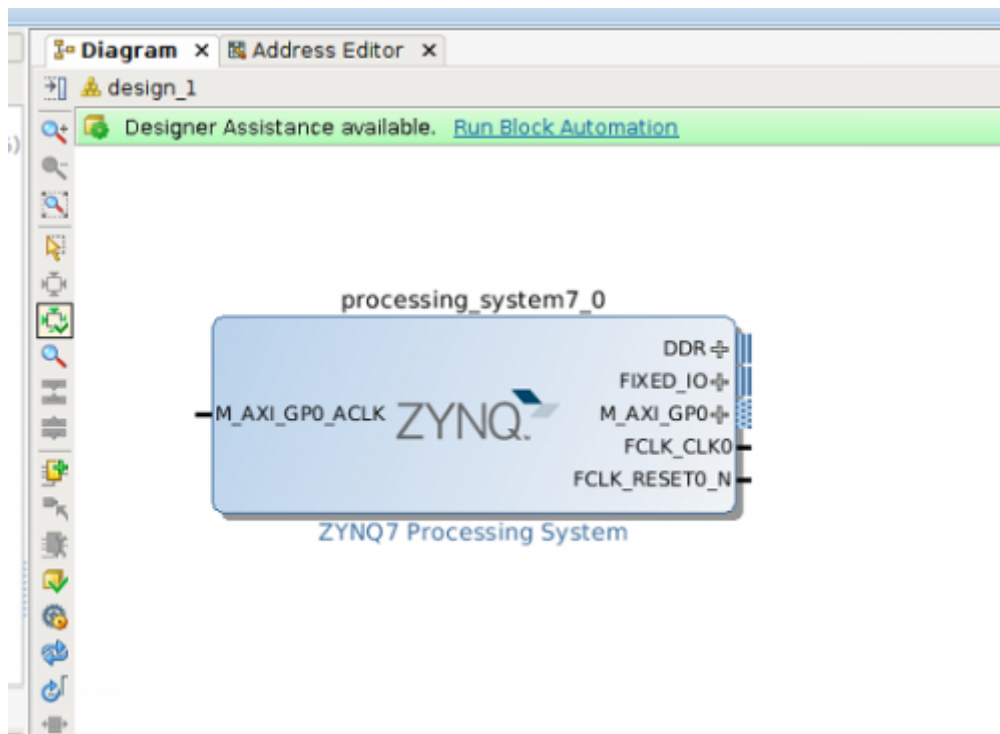
II. Δημιουργία νέου Block Design

- Μόλις ολοκληρωθεί η διαδικασία, κάντε κλικ στην επιλογή **Create Block Design** στο **Flow Navigator**. Στο παράθυρο που θα ανοίξει κάντε κλικ στο **OK**. Θα ανοίξει ένα κενό **Block Design**.



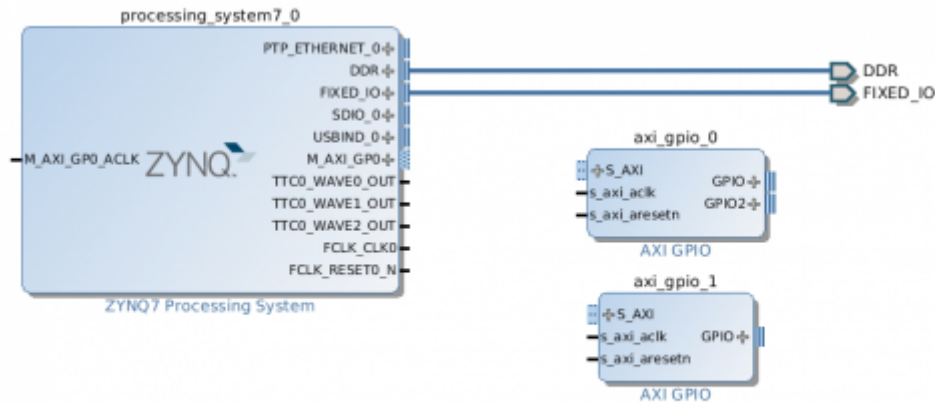
III. Προσθέστε τα μπλοκ Zynq IP & AXI GPIO

- Κάντε κλικ στο κουμπί  **Add IP** και αναζητήστε **ZYNQ**. Κάντε διπλό κλικ στο **ZYNQ7 Processing System** για να τοποθετηθεί στο **Block Design**.



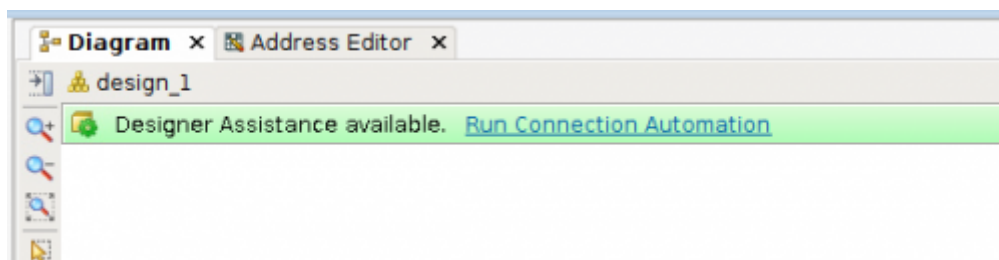
- Κάντε κλικ στο σύνδεσμο **Run Block Automation**.
- Κάντε κλικ στο κουμπί  **Add IP** και αναζητήστε **gpio**. Κάντε διπλό κλικ στο **AXI GPIO** για να τοποθετηθεί στο **Block Design**.

- Κάντε διπλό κλικ στον νέο IP **axi_gpio_0** που μόλις προστέθηκε για να εμφανιστεί το παράθυρο προσαρμογής. Στην καρτέλα **IP Configuration**, επιλέξτε το **Enable Dual Channel**. Κάντε κλικ στο κουμπί **OK**.
- Προσθέστε ακόμα ένα **AXI GPIO** αλλά μην ενεργοποιήσετε το Dual Channel. Το Block Diagram θα πρέπει τώρα να μοιάζει με την παρακάτω εικόνα.

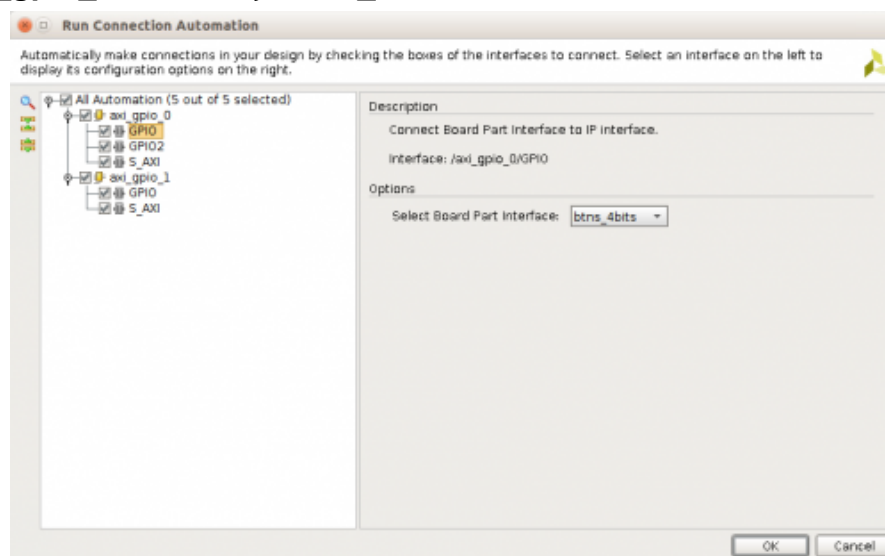


IV. Εκτελέστε το Connection Automation Tool

- Επιλέξτε **Run Connection Automation**. Το εργαλείο αυτόματης σύνδεσης θα προσθέσει τα απαραίτητα λογικά μπλοκ για το demo.




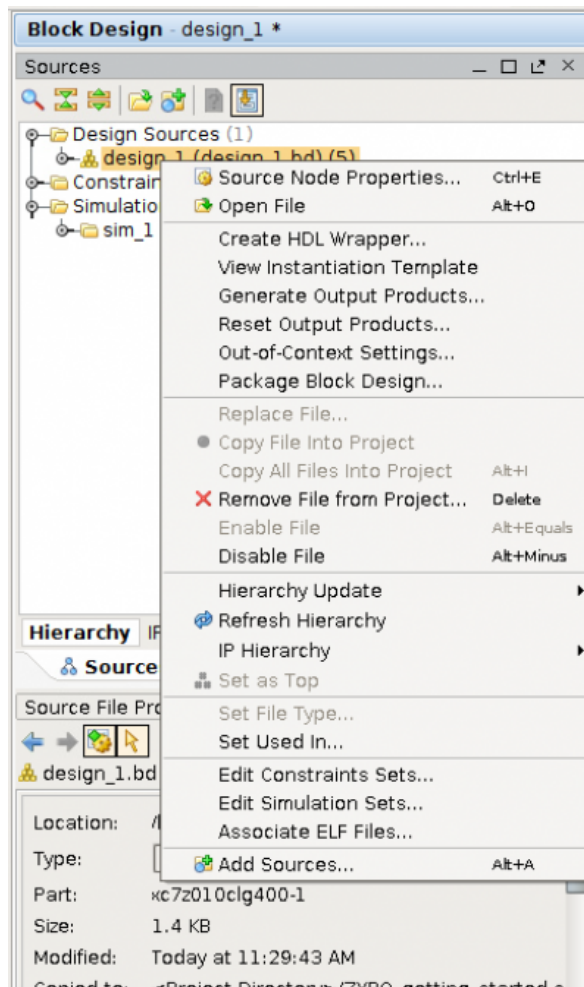
- Ελέγξτε το κουτί με την επιλογή **All Automation**. Επιλέξτε **GPIO** κάτω από το **axi_gpio_0** και επιλέξτε **btns_4bits** στο πλαίσιο **Select Board Part Interface**.



- Παρόμοια, επιλέξτε **GPIO2** κάτω από το **axi_gpio_0** και επιλέξτε **swts_4bits** στο πλαίσιο **Select Interface Part Interface**. Επιλέξτε **GPIO** κάτω από το **axi_gpio_1** και επιλέξτε **leds_4bits** στο πλαίσιο **Select Board Part Interface** και πατήστε **OK**.

V. Δημιουργία του HDL Wrapper και Validate Design

- Επιλέξτε  **Validate Design**. Αυτό θα ελέγξει για τα σφάλματα σχεδίασης και σύνδεσης
- Μετά το βήμα επικύρωσης του σχεδιασμού θα συνεχίσουμε με τη δημιουργία ενός HDL wrapper του συστήματος. Στο παράθυρο Block Design, στην καρτέλα **Design Sources**, κάντε δεξί κλικ στο block diagram. Το ονομάσαμε "design_1.bd" και επιλέξτε **"Create HDL Wrapper"**. Αυτό θα δημιουργήσει ένα top VHDL αρχείο και θα σας επιτρέψει να δημιουργήσετε το bitstream.



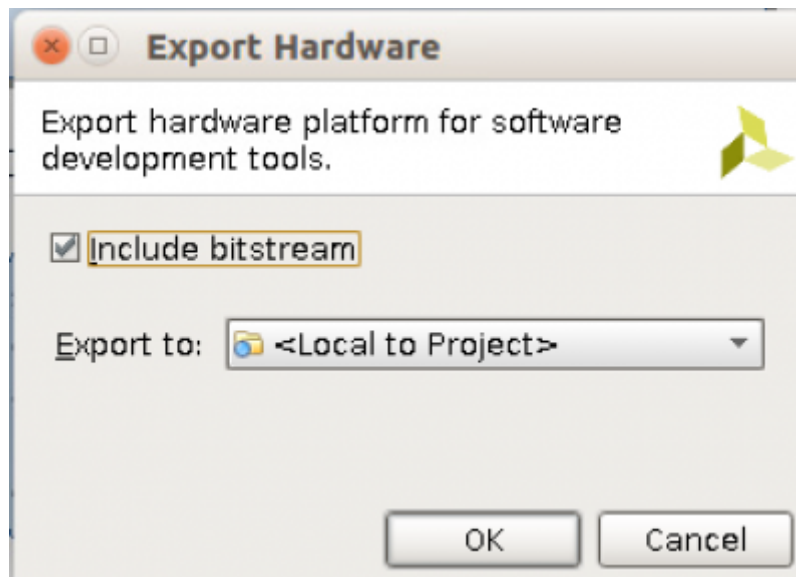
VI. Δημιουργήστε το Bitstream

- Κάντε κλικ στην επιλογή **Generate Bitstream** στο κάτω μέρος του Flow Navigator. Περιμένετε μέχρι να ολοκληρωθεί η διαδικασία και κάντε κλικ στο κουμπί **OK**.



VII. Εξαγωγή αρχείων υλικού για το SDK και εκκίνηση του SDK

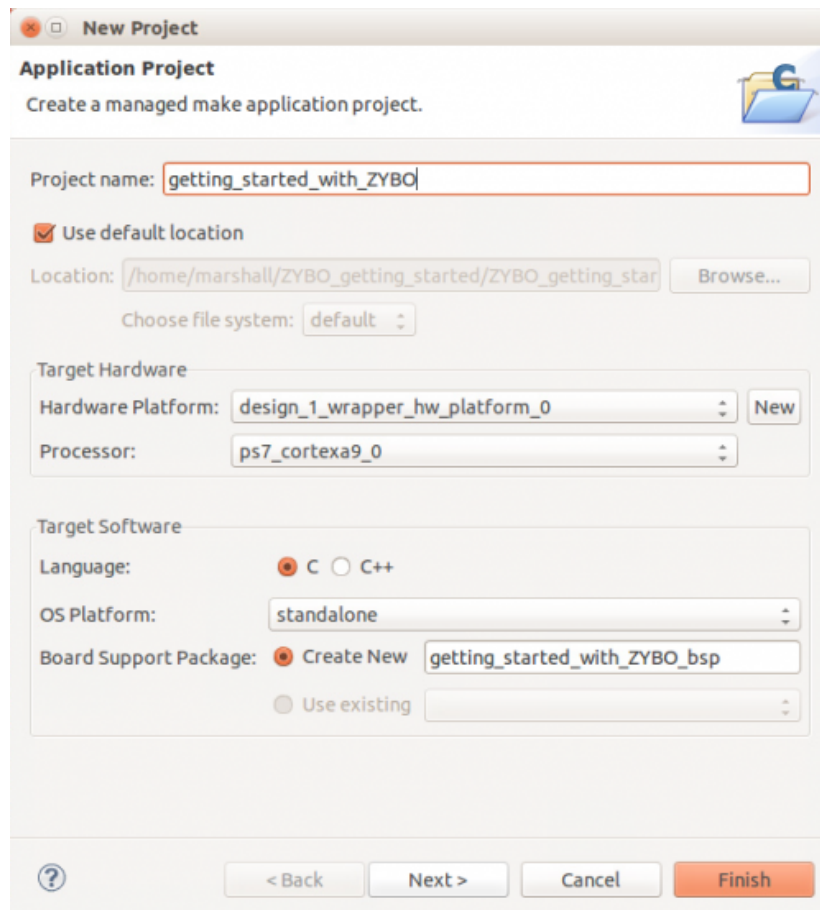
- Μεταβείτε στο **File→Export→Export Hardware...** Βεβαιωθείτε ότι έχετε επιλέξει το **Include bitstream** και στη συνέχεια κάντε κλικ στο κουμπί **OK**.



- Μεταβείτε στην επιλογή **File→ Launch SDK** και κάντε κλικ στο **OK**.

VIII. Δημιουργήστε ένα νέο Application Project στο SDK

- Στο SDK μεταβείτε στο **File→New →Application Project**. Εισάγετε τα στοιχεία του project.



- Το demo του Hello World είναι ένα καλό σημείο εκκίνησης. Κάντε κλικ στο κουμπί **Next**, επιλέξτε Hello World και κάντε κλικ στο κουμπί **Finish**. Αυτή η διαδικασία θα προσθέσει δύο directories στον Project Explorer.
- Ανοίξτε το **get_started_with_ZYBO**, στη συνέχεια ανοίξτε το **src** και κάντε διπλό κλικ στο "helloworld.c". Αυτός είναι ο C κώδικας του Hello World.
- Αντιγράψτε και επικολλήστε τον παρακάτω κώδικα στο αρχείο helloworld.c.

```
/******
```

Getting Started Guide for Zybo

This demo displays the status of the switches on the LEDs and prints a message to the serial communication when a button is pressed.

```
*****/
```

```
#include <stdio.h>
#include "platform.h"
#include <xgpio.h>
#include "xparameters.h"
#include "sleep.h"
```

```
int main()
{
```



```

XGpio input, output;
int button_data = 0;
int switch_data = 0;

XGpio_Initialize(&input, XPAR_AXI_GPIO_0_DEVICE_ID); // initialize input XGpio variable
XGpio_Initialize(&output, XPAR_AXI_GPIO_1_DEVICE_ID); // initialize output XGpio variable

XGpio_SetDataDirection(&input, 1, 0xF); // set first channel tristate buffer to input
XGpio_SetDataDirection(&input, 2, 0xF); // set second channel tristate buffer to input

XGpio_SetDataDirection(&output, 1, 0x0); // set first channel tristate buffer to output

init_platform();

while(1) {

    switch_data = XGpio_DiscreteRead(&input, 2); // get switch data

    XGpio_DiscreteWrite(&output, 1, switch_data); // write switch data to the LEDs

    button_data = XGpio_DiscreteRead(&input, 1); // get button data


    // print message dependent on whether one or more buttons are pressed
    if (button_data == 0b0000) {
        // do nothing
    } else if (button_data == 0b0001) {
        xil_printf("button 0 pressed\n\r");
    } else if (button_data == 0b0010) {
        xil_printf("button 1 pressed\n\r");
    } else if (button_data == 0b0100) {
        xil_printf("button 2 pressed\n\r");
    } else if (button_data == 0b1000) {
        xil_printf("button 3 pressed\n\r");
    } else {
        xil_printf("multiple buttons pressed\n\r");
    }
    usleep(200000); // delay
}

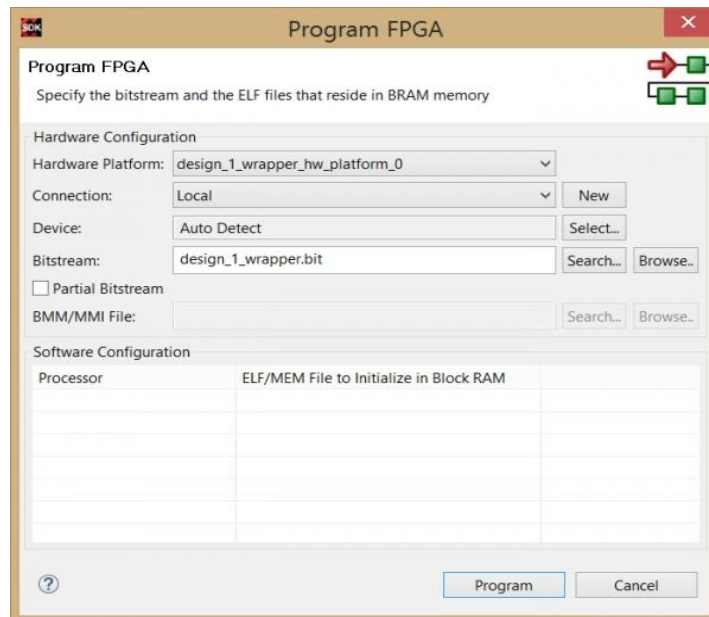
cleanup_platform();
return 0;
}

```

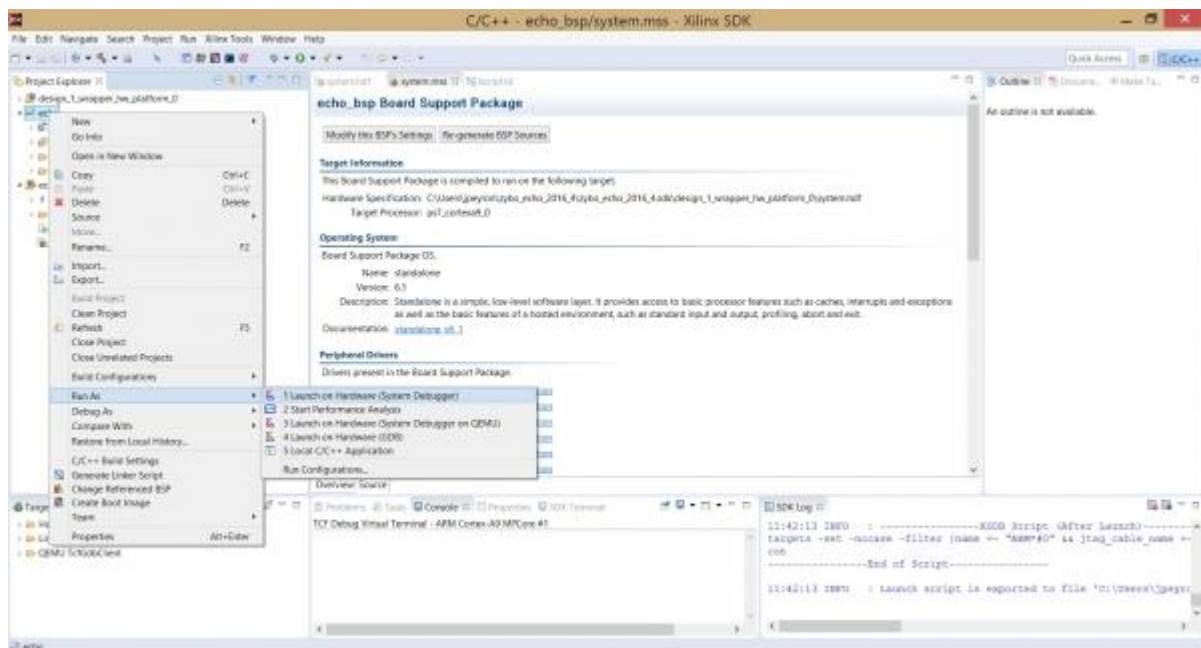
IX. Εκτελέστε την εφαρμογή

- Βεβαιωθείτε ότι το Zybo είναι συνδεδεμένο με τον κεντρικό υπολογιστή μέσω της θύρας USB UART και ότι το JP5 έχει οριστεί σε JTAG. Για να

προγραμματίσετε το FPGA, στην επάνω γραμμή εργαλείων, κάντε κλικ στο κουμπί  **Program FPGA**.



- Αποθηκεύστε το project. Το project θα γίνει build αυτόματα. Κάντε δεξί κλικ στο **getting_started_with_ZYBO** και επιλέξτε **Run as→Launch on Hardware (GDB)**.



- Το demo θα τρέξει στο Zybo. Δοκιμάστε να παίζετε με τους 4 διακόπτες (με την ένδειξη SW0-SW3). Κάτι τέτοιο πρέπει να ανάψει το αντίστοιχο LED. Επίσης μέσω της σειριακής θύρας, πατώντας κάθε κουμπί (με την ένδειξη BTN0-BTN3) θα εμφανιστεί το μήνυμα "button x pressed". Για τη σειριακή επικοινωνία το Tera Term ή οποιοδήποτε σειριακό τερματικό θα μπορούσε να

χρησιμοποιηθεί ως κονσόλα για την εμφάνιση της εξόδου των BTN (στο SDK υπάρχει ενσωματωμένη κονσόλα). **Σημείωση:** Προτείνεται η σειριακή επικοινωνία να προετοιμαστεί πριν την εκτέλεση της εφαρμογής στο Zybo. Στην εικόνα που ακολουθεί δίνεται πως πρέπει να γίνει setup το πρόγραμμα για την σειριακή επικοινωνία μεταξύ του κεντρικού υπολογιστή και του Zybo.

Tera Term: Serial port setup

Port: COM13

Baud rate: 115200

Data: 8 bit

Parity: none

Stop: 1 bit

Flow control: none

OK

Cancel

Help

Transmit delay

0 msec/char 0 msec/line