

PYNQ-Z2 Vitis Tutorijal

Student: Bačanin Ivana 26-2021

Profesor: Aleksandar Peulić

Sadržaj

1. Uvod.....	2
2. Zahtevi.....	2
3. Izvoz hardvera iz Vivado-a.....	2
5. Kreiranje Hello World aplikacije	3
6. Programiranje FPGA i pokretanje aplikacije	4
7. Podesavanje serijskog terminala.....	4
8. Kontrola LED dioda putem GPIO-a	5
10. Rezime i sledeći koraci.....	6

1. Uvod

Ovaj vodič pruža detaljno, korak-po-korak objašnjenje kako da napravite jednostavan projekat saradnje između CPU-a i FPGA-e koristeći Vivado i Vitis na PYNQ-Z2 ploči. Obuhvata izvoz hardvera, konfiguraciju Vitis platforme (BSP), pokretanje Hello World aplikacije, verifikaciju UART serijske komunikacije i kontrolu LED dioda putem GPIO-a.

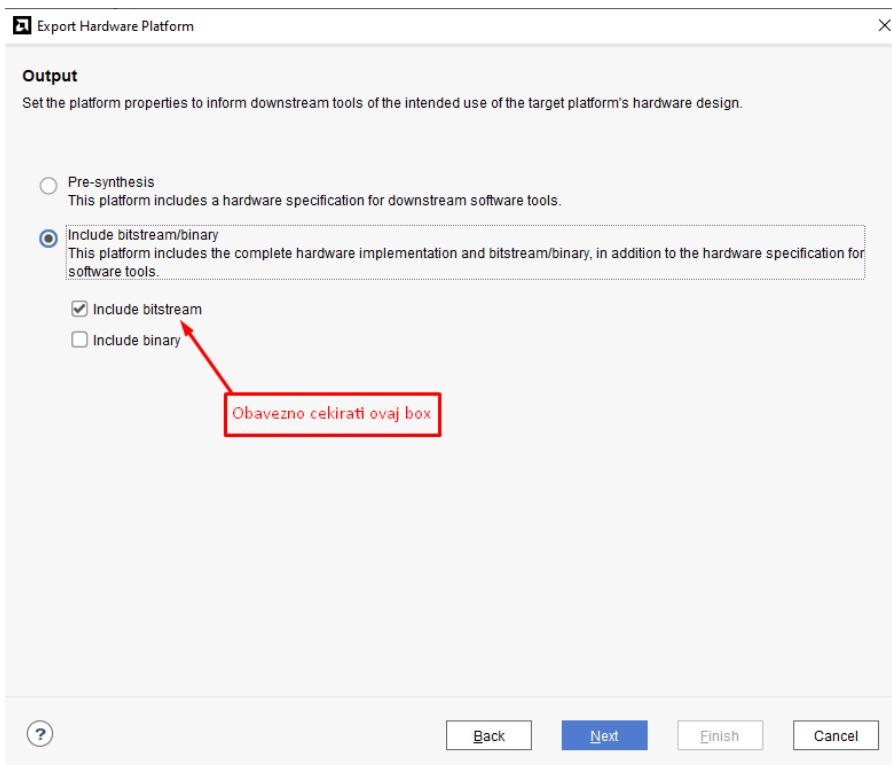
2. Zahtevi

- Vivado (verzija 2020.2 ili novija)
- Vitis IDE (ista verzija kao Vivado)
- PYNQ-Z2 ploča (ili slična Zynq-7000 ploča)
- USB kabl za JTAG i UART povezivanje
- Serijski terminal (ugrađeni Vitis terminal)
- Računar sa Windows ili Linux operativnim sistemom i instaliranim drajverima za ploču

3. Izvoz hardvera iz Vivado-a

U Vivado-u, hardverska platforma definiše FPGA i povezane periferije. Ove informacije se izvoze kao XSA fajl, koji Vitis koristi da razume koji hardver je dostupan

- Kreirajte novi Vivado projekat i izaberite PYNQ-Z2 board preset.
- Otvorite Block Design i dodajte 'Zynq Processing System' IP.
- Dodajte AXI GPIO blok i povežite ga sa LED diodama ili prekidačima.
- Pokrenite Connection Automation da Vivado automatski poveže signale.
- Pokrenite Block Automation da se podese podrazumevana PS podešavanja.
- Validirajte dizajn i generišite bitstream.
- Nakon završetka generisanja bitstream-a, idite na File → Export → ExportHardware i izaberite 'Include Bitstream'.



4. Kreiranje platformskog projekta (BSP) u Vitis-u

BSP (Board Support Package) obezbeđuje niskonivojske drajvere za periferije poput UART-a, GPIO-a i tajmera. Takođe definiše koji procesorski jezgro izvršava vašu aplikaciju.

- Otvorite Vitis IDE i kreirajte novi workspace direktorijum.
- Idite na File → New → Platform Project.
- Uvezite XSA fajl koji ste izvezli iz Vivado-a.
- Izaberite 'standalone' domen za bare-metal (bez operativnog sistema) okruženje.
- Kompletirati proces kako bi se automatski generisao izvorni kod BSP-a.

5. Kreiranje Hello World aplikacije

Možete koristiti ugrađeni 'Hello World' template ili kreirati prazan projekat i dodati sopstveni kod. Ovaj program proverava da li UART komunikacija i povezivanje BSP-a rade ispravno.

```
1 #include "xparameters.h"
2 #include "xil_printf.h"
3
4 int main() {
5     xil_printf("Hello World from Vitis!\r\n");
6     xil_printf("Successfully ran Hello World application.\r\n");
7     return 0;
8 }
9
```

1. Hello world kod

6. Programiranje FPGA i pokretanje aplikacije

- Povežite PYNQ-Z2 putem USB-a i *uključite napajanje*.
- U novoj verziji Vitis IDE-a programiranje FPGA je automatski pokriveno ukoliko je pri exportovanju .XSA fajla iz vivada bio ukjucen bitstream.
- Nakon programiranja, izaberite Run → Launch on Hardware (Single Application Debug).
- Ovaj korak učitava kompajliranu aplikaciju u ARM jezgro Zynq SoC-a i izvršava je dok komunicira sa FPGA logikom.

7. Podesavanje serijskog terminala

PYNQ-Z2 ploča prenosi *UART* podatke preko svog *USB* interfejsa. Da biste videli izlaz funkcije `'xil_printf'`, potrebno je da otvorite **serijski terminal**.

- U Vitis-u otvorite Vitis → Serial Monitor.
- Odaberite ispravan COM port ako koristite USB UART.
- Podesite Baud Rate = 115200.
- Kliknite OK da se povežete.

8. Kontrola LED dioda putem GPIO-a

Da biste proverili saradnju hardvera i softvera, možete kontrolisati LED diode povezane na AXI GPIO blok.

```

1  #include "xparameters.h"
2  #include "xgpio.h"
3  #include "xil_printf.h"
4
5  #define LED_DELAY 10000000    // crude delay
6
7  int main() {
8      XGpio LEDGpio;
9      int status;
10
11     // Initialize GPIO (use 0 because only 1 AXI GPIO instance exists)
12     status = XGpio_Initialize(&LEDGpio, 0);
13     if (status != XST_SUCCESS) {
14         xil_printf("GPIO Initialization Failed!\n");
15         return XST_FAILURE;
16     }
17
18     // Set LED GPIO direction to output (channel 1)
19     XGpio_SetDataDirection(&LEDGpio, 1, 0x0);
20
21     xil_printf("Starting 4-LED binary counter...\n");
22
23     unsigned int led_value = 0;
24
25     while (1) {
26
27         XGpio_DiscreteWrite(&LEDGpio, 1, led_value);
28         for (volatile int i = 0; i < LED_DELAY; i++);
29         led_value = (led_value + 1) & 0xF;    // 4 LEDs
30     }
31
32     return 0;
33 }
34

```

9. Najčešći problemi i rešenja

- Nema UART izlaza: Proverite da li je izabran ispravan COM port ili JTAG UART i da je flow control isključen.
- 'xil_printf.h' nije pronađen: Ponovo izgradite BSP ili proverite da li aplikacija koristi ispravnu platformu.
- LED diode ne reaguju: Proverite da li je GPIO smer podešen na izlaz (0x0) i da se koristi ispravan kanal.
- Ploča se ne programira: Ponovo izvezite hardver iz Vivado-a sa uključenim 'Include Bitstream'.

10. Rezime i sledeći koraci

Uspešno ste konfigurisali PYNQ-Z2 za interakciju između CPU-a i FPGA-e koristeći Vitis. Naučili ste kako da:

- Izvezete hardver iz Vivado-a i uvezete ga u Vitis.
- Konfigurišete i povežete BSP (Board Support Package).
- Verifikujete UART komunikaciju pomoću `xil_printf`.
- Kontrolišete LED diode putem AXI GPIO-a.

Sledeći koraci mogu uključivati:

- Čitanje ulaza sa tastera ili prekidača.
- Dodavanje GPIO prekida (interrupts).
- Prebacivanje računanja na prilagođene hardverske akceleratore.