

*Projekat iz predmeta Projektovanje digitalnih sistema*

## **WEB pristup akvizicionom hardveru**

Dejan Lukić 82 / 2014

Predmetni profesor:

**Lazar Saranovac**

Predmetni asistenti:

**Strahinja Janković**

Elektrotehnički fakultet

Univerzitet u Beogradu

Letnji semestar 2017/2018

# 1 Uvod

## 2 Uvod

Diplomski

socfpga linux vhdل de1soc cyclone5 alati

Komunikacija linux operativnog sistema prema hardveru realizovanom u FPGA

U ovom radu je na DE1-SoC razvojnom sistemu implementiran jednostavan hardver u FPGA, portovan je Linux operativni sistem i napisan je drajver za pristup registrima i prihvatanje prekida iz FPGA. (Ovo u zakljucak: Izrada ovog rada je bila motivisana zeljom da se upoznaju konkretni alati i postupak projektovanja sa SoCFPGA sistemima. Izvestaj o radu je napisan tako da pruži kratak pregled bitnih pojmova i detaljna uputstva za reprodukciju rezultata sa osvrtom na usputne probleme. Steceno znanje moze olaksati slusanje predmeta na master studijama, ili biti osnova za resavanje konkretnog problema (npr. ubrzavanje algoritama kompresije i obrade slike), a autoru je olaksalo snalazenje na novom radnom mestu. Autor izrazava veliku zahvalnost profesoru Lazaru Saranovcu i asistentu Strahinji Jankovicu za podrsku i savete prilikom izrade diplomskog rada.

Sa porastom mogucnosti namenskih sistema doslo je do popularizacije sistema na cipovima (SoC - System on Chip) koji integrišu mikroprocesore sa više jezgara, memorije na cipu, mnogobrojne periferije i transivere, kao i FPGA (Field Programmable Gate Array).

Ova tehnologija daje dizajneru sistema veliku slobodu i mogucnosti. Mogucnosti klasicnog pristpa projektovanju namenskih sistema su proširene punom snagom FPGA. Uz to se ostvaruje veća integracija, manja potrošnja, manja površina štampane ploče (PCB - Printed Circuit Board) i veći protok podataka između procesora i FPGA dela.

Uobicajena primena ovih sistema je implementacija specifičnih akceleratora koji ubrzavaju izvršavanje algoritama i implementacija specifičnih programabilnih interfejsa ka spoljnom svetu. Nove tehnologije kao što su OpenCL, (navedi jos)omogućavaju kompatibilnost dizajna softvera na visokom nivou i implementiranog hardvera na niskom nivou.

SoCFPGA sistemi najčešće sadrže ARM mikroprocesor. Aplikacije na mikroprocesoru bez operativnog sistema (baremetal) nude jednostavno pisanje koda i uštedu na resursima. Za kompleksnije aplikacije koriste se operativni sistemi (OS) i time se olakšava integrisanje mrežnih protokola, rad sa multimedijalnim sadržajima, kriptografskim bibliotekama kao i mnoge druge mogucnosti koje su dostupne kao open-source softver. Kada je potrebno garantovati reakciju u određenom vremenu na neki spoljni događaj veliki operativni sistemi nisu dobro rešenje i koriste operativni sistemi u realnom vremenu (RTOS).

Hardver u FPGA se projektuje upotrebom nekog dva popularna jezika za opis hardvera (Verilog i VHDL - Very High Speed Integrated Circuit Hardwer Description Language) i softverskih alata za specifični uređaj. Dodatno ovi alati olakšavaju dizajn upotrebom IP( Intellectual Property) blokova kao i generisanjem raznih izlaznih fajlova koji opisuju projektovani hardver na standardni način i koriste se prilikom razvoja softvera.

U ovom radu korišćen je DE1-SoC razvojni sistem koji se vrlo često upotrebljava u edukativne svrhe. Razvojni sistem je zasnovan na cipu iz familije Cyclone 5 kompanije Altera (sada Intel) sa dodatom DDR3 memorijom kapaciteta 1GB i hardverom za audio, video, mrežu i ostalo. Korišćeni alati za razvoj su Quartus, Altera EDK(Embedded Development Kit), i mnogi drugi koji će biti pomenuti u uputstvu.

Kratak opis Cyclone 5 sa procesom boot-ovanja. Detaljan opis projekta i svih delova Opis DE1-SoC

U nastavku su navedene samo osobine razvojnog sistema koje se tiču ovog rada, a detaljniji opis se moze pronaći u dokumentu DE1-SoC User Manual [] - Sistem na cipu Cyclone V SoC 5CSEMA5F31 - Memorija 1GB (2x256Mx16) DDR3 SDRAM povezana na HPS - Slot za Micro

SD karticu povezan na HPS - UART na USB (USB Mini-B konektor) - 5 debaunsiranih tastera (FPGA x4, HPS x1) - 11 LE dioda (FPGA x10, HPS x 1) - 12V DC napajanje

(dodati slike tastera, diodica, itd?)

[<https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=205&> DE1-SoC User Manual(rev.E Board) Terasic

Opis Altera Cyclone 5 Altera Cyclone 5 je SoC FPGA koji se sastoji od dva dela(slika): procesorski deo (HPS - Hard processor System) i programabilni FGPA deo. HPS se sastoji od MPU (Microprocessor unit) sa ARM Cortex-A9 MPCore sa dva jezgra i sledećih modula: kontroleri memorije, memorije, periferije, sistem interkonekcije, debug moduli, PLL moduli. FPGA deo se sastoji od sledećih delova: FPGA programabilna logika (look-up tabele, RAM memorije, množači i rutiranje), kontrolni blok, PLL, kontroler memorije.

Svaki pin kućista je povezan na samo jedan od ova dva dela sistema, tako da HPS deo i FPGA deo ne mogu međusobno razmenjivati pinove.

Konfigurisanje FPGA i pokretanje HPS Pri pokretanju HPS (boot) može da učitava program iz FPGA dela, iz eksterne flash memorije ili preko JTAG. FPGA ima mogućnost da se konfiguriše softverski iz HPS korišćenjem periferije FPGA Manager ili spoljnim programatorom. Kombinacije ovih mogućnosti daju nekoliko scenarija: - nezavisno konfigurisanje FPGA i pokretanje HPS - konfigurisanje FPGA, zatim pokretanje HPS iz memorije koja se nalazi u FPGA - pokretanje HPS, zatim konfigurisanje FPGA iz HPS DE1-SoC razvojni sistem dolazi sa integrisanim programatorom kojem se pristupa preko USB porta. Moguće je podesiti konfigurisanje FPGA spolja ili iz HPS upotrebom prekidača MSEL, dok se HPS uvek pokreće iz flash memorije SD kartice. (dodati tabeau 3-2 iz de1soc user guide)

HPS-FPGA interfejsi HPS-FPGA interfejsi su komunikacioni kanali između HPS i FPGA dela. U nastavku su nabrojani i opisani HPS-FPGA interfejsi: FPGA-to-HPS bridge - magistrala visokih preformansi konfigurabilne širine od 32,64 ili 128 bita. Na ovoj magistrali je FPGA master. Ovaj interfejs otkriva FPGA masterima ceo adresni prostor HPS dela. HPS-to-FPGA bridge - magistrala visokih preformansi konfigurabilne širine od 32,64 ili 128 bita. Na ovoj magistrali je HPS master a u FPGA se nalazi slave. Lightweight HPS-to-FPGA - magistrala širine 32 bita. HPS je master na ovoj magistrali. Ovaj interfejs manjeg protoka je namenjen za pristup statusnim i kontrolnim registrima periferijama implementiranim u FPGA delu. FPGA manager - HPS periferija koja komunicira sa FPGA delom prilikom konfiguracije ili pokretanja (boot) Prekidi - mogućnost povezivanja prekida iz FPGA na HPS kontroler prekida HPS debug interfejs - omogućava da se debug mogućnosti prošire i na FPGA deo Interfejsi koji su produžetak AXI magistrale na FPGA deo su FPGA-to-HPS bridge, HPS-to-FPGA bridge i Lightweight HPS-to-FPGA. Za povezivanje na ovu magistralu sa strane FPGA koristi se Avalon magistrala, stoga je neophodan AXI-Avalon bridge.

Proces pokretanja HPS (boot) Pokretanje HPS je proces koji se obavlja u više koraka. Nakon izvršavanja svakog koraka se učitava i pokreće sledeći. Ovo je proces je sličan kod svih ARM procesora, a u nastavku je ukratko opisan za konkretnu platformu. Boot ROM -> Preloader -> bootloader -> Linux Pri izlazu iz reset stanja procesor počinje izvršavanje sa reset vektora iz memorije na čipu. Na adresi reset vektora je upisan Boot ROM program. Ovo je prvi korak u pokretanju HPS. BootROM izvršava osnovna podešavanja procesora i dohvata Preloader iz NOR flash memorije, NAND flash memorije ili SD/MMC flash memorije. Očitavaju se BSEL pinovi na osnovu kojih se određuje gde je smešten Preloader, zatim se inicijalizuje taj interfejs i učitava i pokreće Preloader. Boot ROM softver proizvođača i ne može se menjati.

Preloader je prvi korak u pokretanju koji može da se konfiguriše. Preloader koji se koristi u ovom radu je zasnovan na SPL (Secondaz Program Loader) framework koji je deo U-Boot projekta, što znači da Preloader i U-Boot dele dosta izvornog koda, kao što je mnoštvo pouzdanih drajvera. Preloader obično izvršava inicijalizaciju SDRAM, dodatna podešavanja sistema, inicijalizaciju flash kontrolera koji sadrži sledeći program (NAND, SD/MMC, QSPI) i zatim

učitavanje programa u RAM memoriju i pokretanje. [slika]

Softver koji sledi nakon Preloader-a može biti baremetal aplikacija ili bootloader. Preloader i svi prethodni programi se izvršavaju na prvom jezgru procesora dok je drugo u reset stanju. Naredni koraci mogu inicijalizovati drugo jezgro.

Bootloader ima zadatak da podesi promenljive OS okruženja, dohvati operativni sistema (sa flash memorije, putem Ethernet preko TFTP protokola ili USB), konfigurise FPGA pruži konzolu za korisničke operacije. Neki od popularnih open-source bootloadera su U-Boot i Barebox.

Alati U nastavku će ukratko biti opisani korišćeni alati sa izdvojenim najvažnijim mogućnostima. Quartus je softverski alat kompanije Intel za razvoj hardvera na FPGA. Deo Quartus softverskog paketa je Platform Designer (ranije Qsys). Upotrebom ovog alata se u dizajn uključuje HPS, IP blokovi i definiše se povezanost ovih delova. Platform Designer takodje prilikom povezivanja Avalon IP blokova na AXI magistralu automatski generise AXI-Avalon bridge. EDS se koristi Git se koristi Kroskompajler se koristi Make se koristi Integracija pogledaj GSRD

Softver U-Boot, Linux, drajver

Opis projekta U ovom radu je implementiran jednostavan sistem koji demonstrira osnovne mogućnosti u dizajniranju sistema na SoC FPGA. DE1-SoC System Builder je jednostavan softver kompanije Terasic koji generise prazan projekat sa povezanim pinovima. Pocevsi od ovog projekta, u Quartus Platform Designeru se dodaje HPS i definiše interfejs ka DDR3 memoriji, zatim se dodaje izlazni PIO IP blok za upravljanje LE diodama i ulazni PIO za očitavanje tastera i slanje prekida.

EDS se koristi za generisanje Preloader-a, Device Tree fajla i konverziju FPGA konfiguracionog fajla u odgovarajući format.

Uobicajeni izbor za bootloader-a je U-Boot. Zatim se preuzima, konfigurise i kompajlira U-Boot i Linux kernel, pri cemu U-boot nakon pokretanja učitava konfiguracionu fajl u RBF formatu i konfigurise FPGA. Zatim učitava binarni fajl za opis hardverske platforme (DTB Device Tree Blob) i binarni fajl kernela operativnog sistema. Konfiguracioni argumenti se prosledjuju pri prepustanju toka izvršavanja kernelu.

### 3 Uputstvo

1. U Windows okruženju pokrenuti DE1SoC\_SystemBuilder [dostupan gde?]
2. Izabrati konfiguraciju kao na slici 1 i izabrati Generate

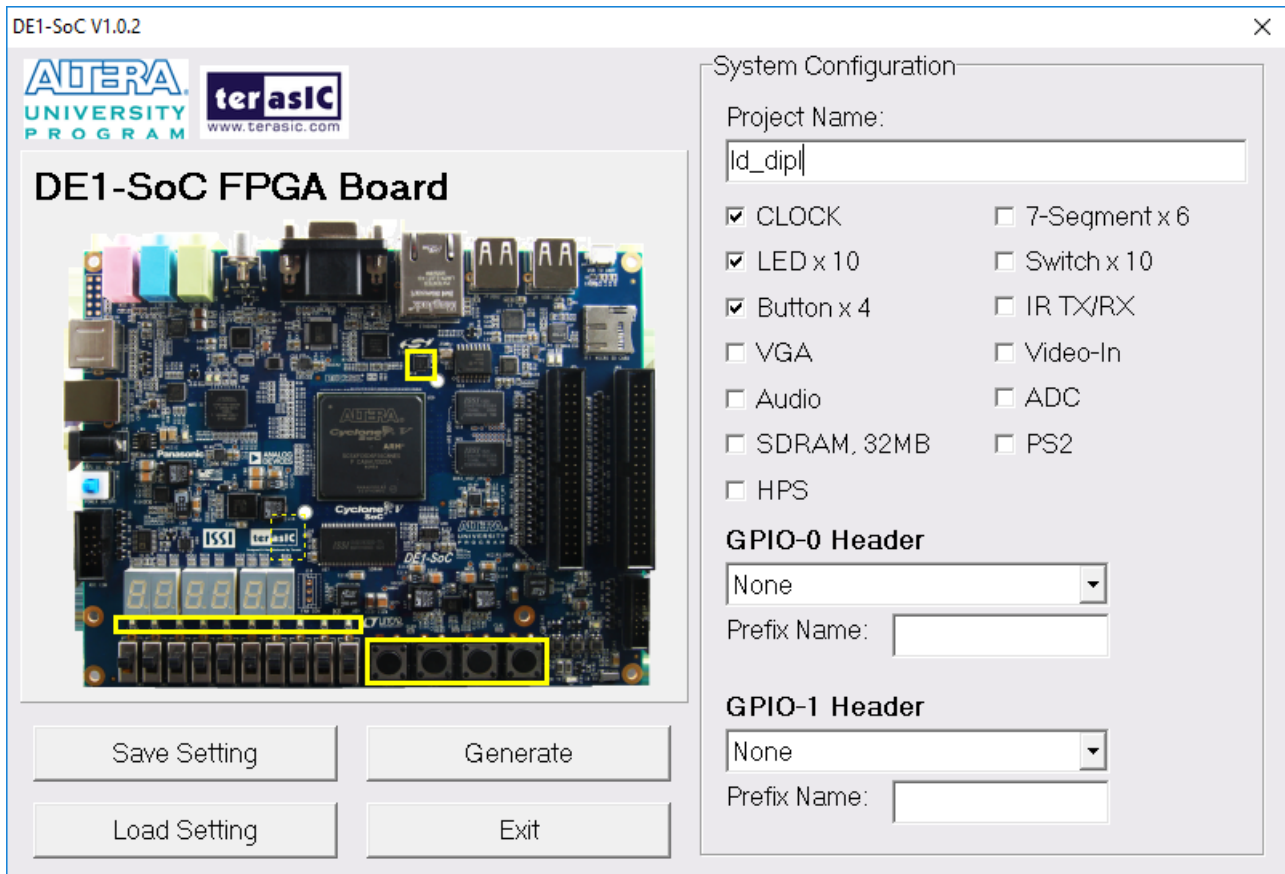


Figure 1: Podešavanja DE1\_Soc Bulder-a

Od generisanih fajlova značajni su:

- .qpf - projektni fajl za Quartus
- .qsf - skripta za podešavanje pinova
- .sdc - skripta za podešavanje takta
- .v - Verilog HDL fajl

3. Izbristati ld\_dipl.v fajl (kasnije će biti napravljen ld\_dipl.vhd fajl za VHDL)
4. Kopirati fajlove u Ubuntu u radni folder ~/ld\_dipl/hw/
5. Pokrenuti Quartus (Quartus Prime 18.0) Lite Edition
6. Otvoriti projekat komandom File > Open Project... i izabrati ~/ld\_dipl/hw/ld\_dipl.qpf
7. U prozoru Tasks izabrati Edit Settings, u novom prozoru pod Timing Analyser ispod teksta SDC files to include in the project klikom na dugme '...' izabrati fajl ld\_dipl.sdc

8. Pokrenuti Platform Designer (Tools > Platform Designer)
9. Iz prozora IP Catalog izabrati Processors and Peripherals > Hard Processor Systems > ArriaV/Cyclone V Hard Processor System
10. Ovim se otvara meni za podešavanje HPS modula
11. Pod tabom FPGA interfaces izvršiti sledeće izmene:
  - U opštim podešavanjima isključiti opciju Enable MPU standbz and event signals
  - U podešavanjima AXI Bridges podesiti FPGA-to-HPS interface i HPS-to-FPGA interface na unused, a Lightweight HPS-to-FPGA na 32-biti
  - U podešavanjima FPGA-to-HPS SDRAM Interface izabrati f2h\_sdram0 i zatim isključiti pritiskom na dugme '-'
  - U podešavanjima Interrupts uključiti opciju Enable FPGA-to-HPS interrupts
12. Pod tabom Peripheral Pins izvršiti sledeće izmene
  - U podešavanjima SD/MMC Controller postaviti SDIO pin na HPS I/O Set 0 i SDIO mode na 4-bit Data
  - U podešavanjima UART Controllers postaviti UART0 pin na HPS I/O Set 0 i UART mode na No Flow Control

*U ovom tabu je za potrebe nekog drugog projekta moguće uključiti ostale periferije: CAN Controller, Ethernet Media Access Controller, I2C Controller, SPI Controller, QSPI Flash Controller, NAND Flash Controller, Trace Port Intefrace Unit, GPIO za podesavanja pogledati []*

13. Pod tabom HPS Clocks ostaviti podešavanja na podrazumevanim vrednostima
14. Pod tabom SDRAM podesiti:
  - SDRAM Protocol: DDR3
  - PHY Settings:
    - Clocks:
      - \* Memory clock frequency: 400.0 MHz
      - \* PLL reference clock frequency: 25.0 MHz
    - Advanced PHY Settings:
      - \* Supply Voltage: 1.5V DDR3
  - Memory Parameters:
    - Memory vendor: Other
    - Memory device speed grade: 800.0 MHz
    - Total interface width: 32
    - Number of chip select/depth expansion: 1
    - Number of clocks: 1
    - Row address width: 15
    - Column address width: 10
    - Bank-address width: 3

- Uključiti DM pins
- Uključiti DQS#
- Memory Initialization Options:
  - \* Mirror Addressing: 1 per chip select: 0
  - \* Burst Length: Burst chop 4 or 8 (on the fly)
  - \* Read Burst Type: Sequential
  - \* DLL precharge power down: DLL off
  - \* Memory CAS latency setting: 11
  - \* Output drive strength setting: RZQ/7
  - \* ODT Rtt nominal value: RZQ/4
  - \* Auto selfrefresh method: Manual
  - \* Selfrefresh temperature: Normal
  - \* Memory write CAS latency setting: 8
  - \* Dynamic ODT (Rtt\_WR) value: RZQ/4
- Memory Timing:
  - tIS (base): 180 ps
  - tIH (base): 140 ps
  - tDS (base): 30 ps
  - tDH (base): 65 ps
  - tDQSQ: 125 ps
  - tQH: 0.38 cycles
  - tDQSCK: 255 ps
  - tDQSS: 0.25 cycles
  - tQSH: 0.4 cycles
  - tDSH: 0.2 cycles
  - tDSS: 0.2 cycles
  - tINIT: 500 us
  - tMRD: 4 cycles
  - tRAS: 35.0 ns
  - tRCD: 13.75 ns
  - tRP: 13.75 ns
  - tREFI: 7.8 us
  - tRFC: 260.0 ns
  - tWR: 15.0 ns
  - tWTR: 4 cycles
  - tFAW: 30.0 ns
  - tRRD: 7.5 ns
  - tRTP: 7.5 ns
- Board Settings:
  - Setup and Hold Derating:
    - \* Use Altera's default settings
  - Channel Signal Integrity:
    - \* Use Altera's default settings





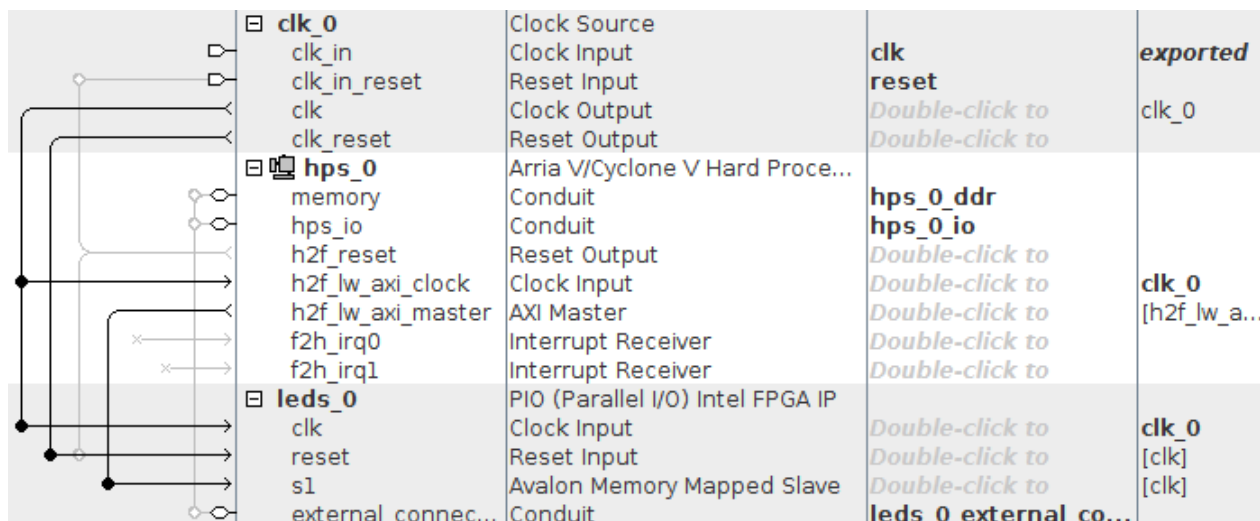


Figure 3: Povezivanje leds\_0 bloka

23. U podešavanjima PIO IP bloka pod Edge capture register uključiti opciju Synchronously capture, Edge Type podesiti na ANY, i uključiti bit-clearing for edge capture register.
24. U podešavanjima PIO IP bloka pod Interrupt uključiti opciju Generate IRQ i izabrati IRQ Type: EDGE
25. Preimenovati PIO blok u keys\_0. Duplim klikom u Export koloni eksportovati signale external\_connection i podesiti ime keys\_0\_external\_connection.
26. Povezati keys\_0 blok sa izvorom takta i resetom, zatim povezati Avalon Memory Mapped Slave pod imenom s1 sa hps\_0 interfejsom h2f\_lw\_axi\_master i na kraju povezati irq signal na f2h\_irq0 interfejs hps\_0 bloka, kao što je prikazano na slici 4

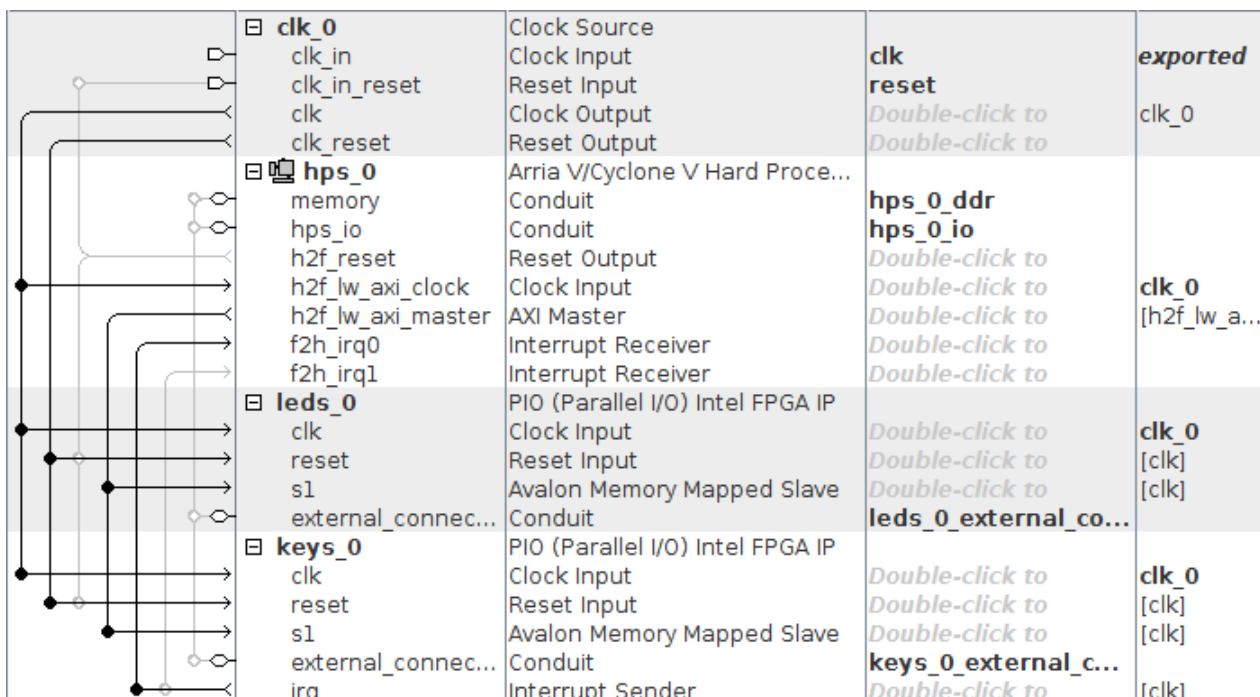


Figure 4: Povezivanje keys\_0 bloka

27. Duplim klikom u koloni **Base** podesiti adresu porta **s1** bloka **leds\_0** i porta **s1** bloka **keys\_0** kao na tabeli 1.
28. Sačuvati Platform Designer projekat izborom **File > Save** i sačuvati ga pod imenom **ld\_dipl\_system.qsys**
29. Trebalo bi da se pojavi obaveštenje **Save System: completed successfully**. Zatim odabrati iz menija **Generate > Generate HDL...** U novom prozoru podesiti **Create HDL design files for synthesis: VHDL** i isključiti opciju **Create block symbol file (.bsf)**. Pokrenuti generisanje klikom na **Generate**. Proces bi trebalo da se završi bez grešaka ali može imati upozorenja.
30. Zatvoriti Platform Designer. U prozoru Quartus-a izabrati **Project > Add/Remove Files in Project...** i u meniju klikom na **'...'** izabrati fajl **ld\_dipl\_system/synthesis/ld\_dipl\_s**
31. Izabrati **File > New VHDL File** i novi fajl nazvati **ld\_dipl.vhd**. Pod **Project Navigator > Files** desnim klikom na **ld\_dipl.vhd** izabrati **Set as Top-Level Entity**.
32. U ovom fajlu je potrebno instancirati HPS komponentu iz Platform Designer-a. Potrebno je ručno napiati ovaj fajl (primer za ovaj rad dat je u dodatku. Takođe među generisanim fajlovima nalazi se deklaracija komponente **ld\_dipl\_system** koja može biti od pomoći (fajl **~/ld\_dipl/hw/ld\_dipl\_system/ld\_dipl\_system.cmp**).
33. Izabrati **Processing > Start > Start Analysis and Synthesis**
34. Izabrati **Tools > Tcl Scripts...**

Izabrati **hps\_sdram\_p0\_pin\_assignments.tcl** i kliknuti **Run**. Ukoliko dođe do grešaka proveriti da li je izvršen prethodni korak.

35. Pokrenuti kompajliranje projekta izborom **Processing > Start Compilation**

Važno: Prozor koji se otvori mora da izgleda upravo kao na slici 5 (generisani fajlovi ne smeju biti duplirani). Ukoliko su fajlovi duplirani neophodno je zatvoriti Quartus i pokrenuti ponovo. Neke verzije Quartus-a imaju ovu grešku pri detekciji tcl skripti.

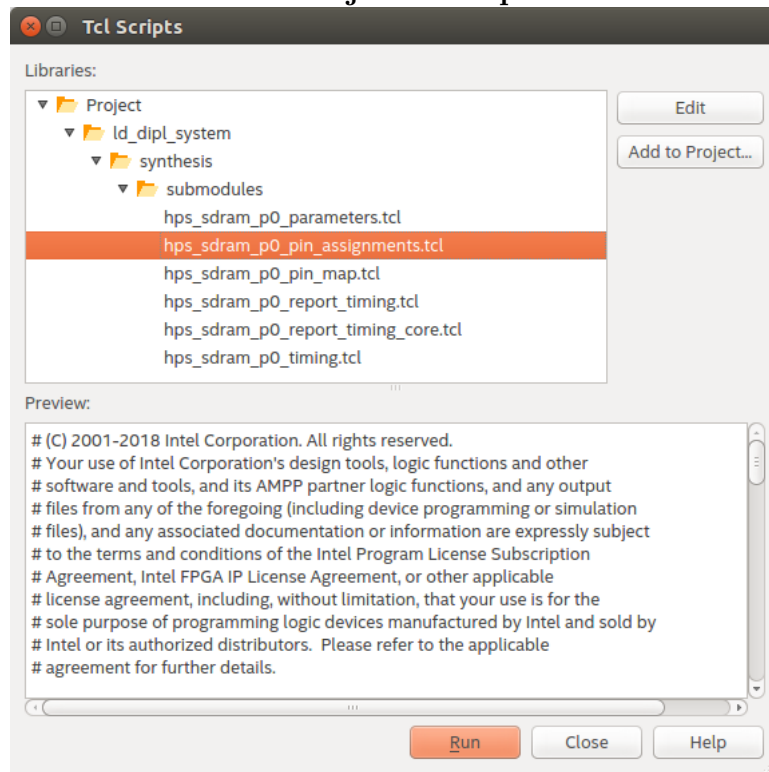


Figure 5: Ispravan izgled menija

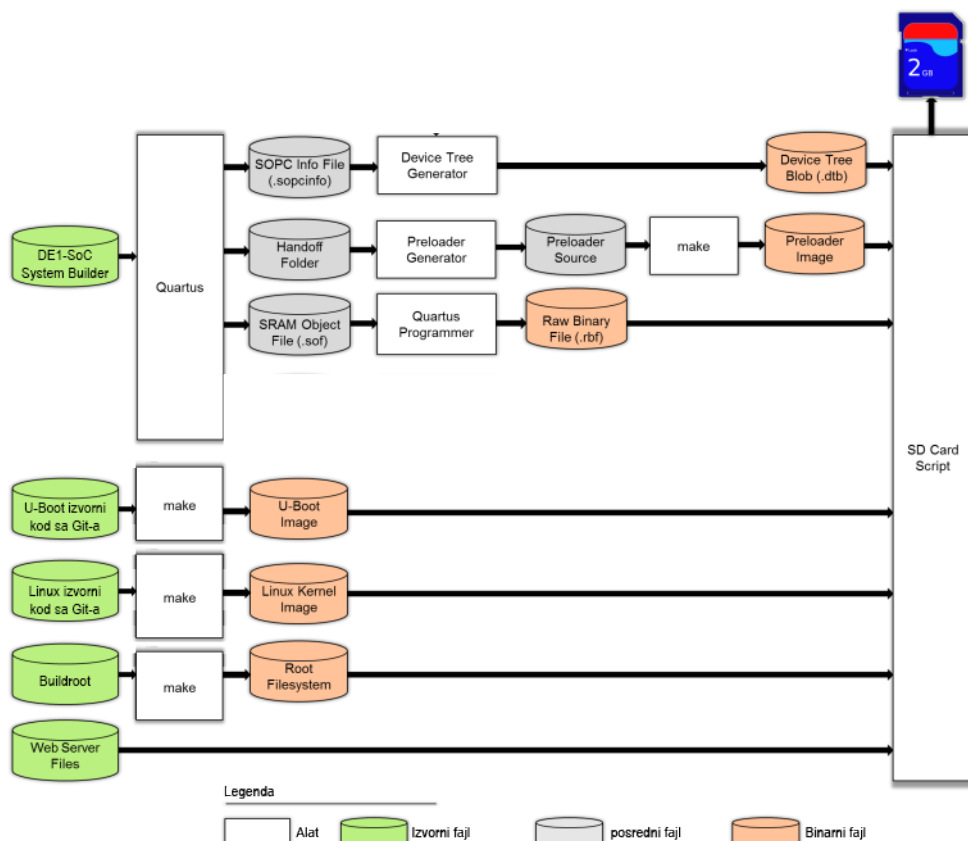


Figure 6: primer slike