



Laboratorijska vježba 5

Korištenje PS2 sučelja s Microblaze procesorom



Opis vježbe

U ovoj vježbi studenti će biti upoznati s korištenjem PS2 sučelja. Korištenje PS2 sučelja bit će objašnjeno kroz primjer spajanja tipkovnice s Microblaze [1] procesorom. Kontroler koji će se koristiti za komunikaciju s tipkovnicom zove se *XPS PS2 Controller* [2] i dostupan je unutar *Xilinx Platform Studio*-a (XPS-a). Ovaj kontroler omogućava komunikaciju s tipkovnicom u *polled* i *interrupt* načinima rada, a na ovoj vježbi studenti će se upoznati s oba načina rada.

Računalni sustav dizajniran korištenjem XPS-a bit će implementiran i testiran na Spartan 6 zasnovanom Nexys 3 razvojnom sustavu [3]. Navedeni razvojni sustav nema PS2 sučelje, nego se tipkovnica spaja preko USB sučelja, a na razvojnom sustavu postoji integrirani sklop koji pretvara USB protokol na PS2 protokol.

Cilj vježbe

Nakon uspješno odrađene vježbe studenti će naučiti:

- dodati *XPS PS2 Controller* u računalni sustav,
- napisati programski kod za komunikaciju s tipkovnicom u *polled* načinu rada korištenjem PS2 sučelja,
- napisati programski kod za komunikaciju s tipkovnicom u *interrupt* načinu rada korištenjem PS2 sučelja,
- napisati programski kod koji obavlja određeni zadatak u ovisnosti o pritisnutoj tipki/tipkama na tipkovnici

Priprema za vježbu

- Proučiti upute za vježbu dane u poglavlju Rad na vježbi.
- Proučiti upute za Nexys3 razvojni sustav s naglaskom na poglavlja *Memory*, *Keyboard*. Upute su dostupne na internet adresi:
https://www.digilentinc.com/Data/Products/NEXYS3/Nexys3_rm.pdf.
- Proučiti upute za korištenje *XPS PS2 Controller*-a [2] za komunikaciju s tipkovnicom preko PS2 sučelja. Upute su dostupne na Internet adresi:
http://www.xilinx.com/support/documentation/ip_documentation/xps_ps2.pdf.
- Proučiti predloške koda dostupne na linku www.tinyurl.com/DRS-LV5-templates.

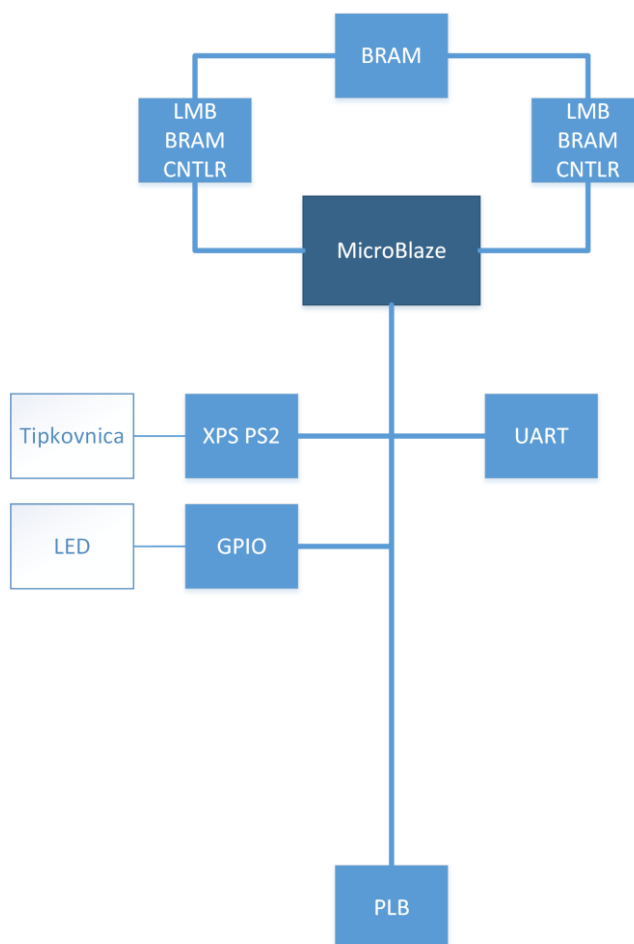


Rad na vježbi

Na ovoj laboratorijskoj vježbi će korištenjem XPS-a i BSB-a biti dizajniran računalni sustav koji se sastoji od sljedećih dijelova:

- MicroBlaze procesor
- MDM (engl. *MicroBlaze Debug Module*)
- UART (za serijsku komunikaciju)
- BRAM (engl. *Block RAM*)
- LMB (engl. *Local Memory Bus*) kontroleri za BRAM
- GPIO (engl. *General Purpose Input Output*)
- XPS PS2 Controller
- Tipkovnica
- Monitor
- LED (LE diode na razvojnom sustavu)

Blok dijagram kompletnog računalnog sustava koji će biti dizajniran na ovoj vježbi prikazan je na Slici 1.



Slika 1. Računalni sustav koji će biti dizajniran na ovoj laboratorijskoj vježbi.



1. Kreirati projekt korištenjem Base System Builder-a

1.1. Pokrenuti *Xilinx Platform Studio* i kreirati novi projekt. Koristiti *Base System Builder* za dizajniranje računalnog sustava.

Računalni sustav dizajnirati korištenjem *Base System Buildera* na temelju znanja naučenih na prvoj laboratorijskoj vježbi. Postavke koje treba odabrati pri dizajniranju sustava su:

- EDK projekt spremi u folder *D:\DRS\LV3_prezime\EDK*
- PLB sabirnica
- jednoprocesorski Microblaze sustav
- frekvencija procesora 66.67 MHz
- lokalna memorija procesora 32 KB
- ukloniti sve periferne sklopove, osim LE dioda (LEDs_8Bits), PS2 kontrolera (PS2_Mouse_Keyboard), UART kontrolera (RS232_Uart_1), LMB kontrolera za BRAM (dlmb_cntlr, ilmb_cntlr),
- ostale postavke ostaviti na predefinirane vrijednosti.

1.2. Generirati .bit datoteku i eksportati dizajnirani računalni sustav u SDK.

Ovaj korak je potrebno napraviti na temelju znanja stečenih na prvoj laboratorijskoj vježbi

2. Kreirati aplikaciju za komunikaciju tipkovnice i Microblaze procesora u polled načinu rada

2.1. U SDK kreirati praznu aplikaciju (*Empty Application*), kreirati novu .c datoteku i napisati programski kod za komunikaciju s tipkovnicom u polled načinu rada. (Za pomoć pogledati datoteke *xps2_sinit.c* i *xps2.c*)

2.1.1. U SDK kliknuti na *File* → *New* → *Application Project*. Upisati naziv projekta te kliknuti na *Next* i u sljedećem koraku odabrati *Empty Application*.

2.1.2. Desnom tipkom miša kliknuti na *src* direktorij unutar projekta i odabrati opciju *New* → *File* i pod *File name* upisati naziv datoteke s nastavkom .c (pr. *main.c* ili *ps2_keyboard_polled.c*).

2.1.3. Kopirati kod iz datoteke *lv5_ps2_polled.c* u kreiranu .c datoteku.

2.1.4. Napisati liniju **82** kojom se poziva funkcija *XPps2_LookupConfig(...)* s odgovarajućim parametrom. Povratnu vrijednost funkcije spremi u varijablu *Ps2ConfigPtr*. Ova funkcija služi za dohvaćanje konfiguracije PS2 kontrolera. (TO DO 1)

2.1.5. Napisati liniju **95** kojom se poziva funkcija *XPps2_CfgInitialize(...)* s odgovarajućim parametrima. Ova funkcija služi za inicijalizaciju PS2 kontrolera. **POMOĆ:** Treći parametar postaviti korištenjem *Ps2ConfigPtr* pokazivača (*Ps2ConfigPtr→BaseAddress*). (TO DO 2)



- 2.1.6. Napisati liniju **116** kojom se poziva funkcija `XPs2_Send (...)` na način da se tipkovnici pošalje vrijednost varijable koja je parametar funkcije `Ps2SendData ()`. Povratnu vrijednost funkcije spremi u varijablu `BytesSent`. (TO DO 3)
- 2.1.7. Napisati liniju **132** kojom se poziva funkcija `XPs2_Recv (...)` na način da se podatak primljen s tipkovnice spremi u varijablu `RxBuffer`. Povratnu vrijednost funkcije spremi u varijablu `BytesReceived`. (TO DO 4)
- 2.1.8. Napisati liniju **161** kojom se poziva funkcija `XPs2_Recv (...)` na način da se podatak primljen s tipkovnice spremi u varijablu `RxBuffer`. Povratnu vrijednost funkcije spremi u varijablu `BytesReceived`. (TO DO 5)
- 2.1.9. Napisati linije **41** kojom se poziva funkcija `Ps2SendData (...)` na način da se putem PS2 sučelja pošalje vrijednost varijable `SendCODE`. (TO DO 6)
- 2.1.10. Napisati liniju **42** kojom se poziva funkcija `Ps2SendData (...)` na način da se putem PS2 sučelja pošalje vrijednost varijable `i`. (TO DO 6)
- 2.1.11. Napisati liniju **55** kojom se poziva funkcija `Ps2ReceiveData (...)` s odgovarajućim parametrima. Povratnu vrijednost funkcije spremi u varijablu `Status`. (TO DO 7)
- 2.1.12. Napisati liniju **67** kojom se poziva funkcija `xil_printf (...)` na način da se u heksadecimalnom obliku ispiše vrijednost primljena od tipkovnice. (TO DO 8)
- 2.1.13. Napisati liniju **9** kojom se definira konstanta `KEYBOARD_ACK` i pridružuje joj se odgovarajuća vrijednost. (TO DO 9).

PITANJE:

Što se postiže slanjem vrijednosti konstante `SendCODE` i nakon toga cjelobrojne vrijednosti iz intervala `[0, 7]` putem PS2 sučelja?

2.2. Implementirati dizajnirani računalni sustav i napisani programski kod na razvojni sustav i testirati rad sustava.

- 2.2.1. Podesiti postavke kabela klikom na *Xilinx Tools* → *Configure JTAG Settings* (Odabрати *Digilent USB Cable*).
- 2.2.2. Podesiti postavke kabela klikom na *Xilinx Tools* → *Configure JTAG Settings* (Odabрати *Digilent USB Cable*).
- 2.2.3. Pokrenuti program za nadzor serijske komunikacije (npr. *Tera Term* ili *Terminal* unutar SDK) i testirati napisani programski kod.



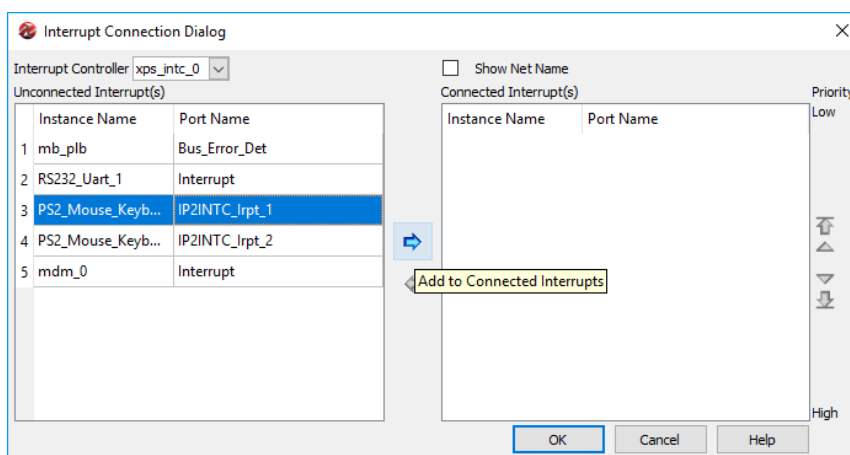
3. U postojeći računalni sustav dodati upravljač prekidima te ga povezati s XPS PS2 kontrolerom

3.1.1. Dodati *XPS Interrupt Controller* (nalazi se u kategoriji *Clock, Reset and Interrupt*) s predefiniranim postavkama.

3.1.2. Povezati upravljač prekidima na *PLB* sabirnicu kao *slave* uređaj.

3.1.3. Generirati adrese za upravljač prekidima.

3.1.4. Na *Port* tabu *System Assembly View*-a kliknuti na *Connected Port* (za *Intr* port) *xps_intc_0* komponente i dva puta kliknuti na nepovezani *IP2INTC_Irpt_1* port *PS2_Mouse_Keyboard* komponente iz kategorije *Unconnected Interrupt(s)* (Slika 2). Nakon toga port prelazi na desnu stranu u polje *Connected Interrupt(s)*. Kliknuti na *OK*.



Slika 2. Povezivanje ps2 kontrolera i upravljača prekidima.

3.1.5. Povezati port *Irq* *xps_intc_0* komponente s *microblaze_0 INTERRUPT* portom. Ukoliko prilikom kreiranja projekta (korištenjem čarobnjaka) nisu odabrani prekidi ni za jednu komponentu, onda je ovo povezivanje potrebno napraviti u *MHS* datoteci dodavanjem sljedećih linija koda:

- *PORT INTERRUPT* = *microblaze_0_Interrupt* (unutar dijela *BEGIN microblaze*)
- *PORT Irq* = *microblaze_0_Interrupt* (unutar dijela *xps_intc*)

3.1.6. Generirati *.bit* datoteku klikom na *Generate BitStream*.

3.1.7. Eksportati generiranu *.bit* datoteku i pokrenuti SDK.

4. Napisati programski kod za komunikaciju tipkovnice i Microblaze procesora u interrupt načinu rada

4.1. Kreirati novu *.c* datoteku i napisati programski kod za komunikaciju s tipkovnicom u interrupt načinu rada.



- 4.1.1. Unutar SDK odabrati opciju *Project* → *Clean...*
- 4.1.2. Desnom tipkom kliknuti na postojeću .c datoteku te odabrati opciju *Properties*. U stablu s lijeve strane odabrati opciju *C/C++ Build* i nakon toga u prozoru s desne strane odabrati opciju *Exclude resource from build*.
- 4.1.3. Desnom tipkom miša kliknuti na *src* direktorij unutar projekta i odabrati opciju *New* → *File* i pod *File name* upisati naziv datoteke s nastavkom .c (pr. *main.c* ili *ps2_keyboard_polled.c*).
- 4.1.4. Kopirati kod iz datoteke *lv5_ps2_interrupt.c* u kreiranu .c datoteku.
- 4.1.5. Napisati linije **7** do **9** kojima se uključuju *header* datoteke *xparameters.h*, *xintc.h* i *xps2.h*. (TO DO 1)
- 4.1.6. Napisati linije **15** do **17** kojima se definiraju konstante *PS2_DEVICE_ID*, *INTC_DEVICE_ID* i *INTR_ID* te im se pridružuju odgovarajuće vrijednosti iz datoteke *xparameters.h*. **POMOĆ:** Konstanti *INTR_ID* pridružiti vrijednost oblika *XPAR_INTC_0_***_VEC_ID*. (TO DO 2)
- 4.1.7. Napisati linije **33** i **34** kojima se deklariraju varijable *Ps2Inst* i *IntcInst* koje su tipa *XPps2* i *XIntc*. (TO DO 3)
- 4.1.8. Napisati liniju **49** kojom se deklarira varijabla *RxBuffer* koja je tipa *u8*. (TO DO 4)
- 4.1.9. Napisati liniju **59** kojom se poziva funkcija *Ps2IntrExample(...)* s odgovarajućim parametrima. Povratnu vrijednost funkcije spremiti u varijablu *Status*. (TO DO 5)
- 4.1.10. Napisati liniju **96** kojom se poziva funkcija *XPps2_LookupConfig(...)* s odgovarajućim parametrom. Povratnu vrijednost funkcije spremiti u varijablu *ConfigPtr*. Ova funkcija služi za dohvaćanje konfiguracije PS2 kontrolera. (TO DO 6)
- 4.1.11. Napisati liniju **105** kojom se poziva funkcija *XPps2_CfgInitialize(...)* s odgovarajućim parametrima. Ova funkcija služi za inicijalizaciju PS2 kontrolera. **POMOĆ:** Treći parametar postaviti korištenjem *ConfigPtr* pokazivača (*ConfigPtr* → *BaseAddress*). (TO DO 7)
- 4.1.12. Napisati liniju **117** kojom se poziva funkcija *Ps2SetupIntrSystem(...)* s odgovarajućim parametrima. Povratnu vrijednost funkcije spremiti u varijablu *Status*. (TO DO 8)
- 4.1.13. Napisati linije **129** i **130** kojima se pozivaju funkcije *XPps2_IntrEnable(...)* i *XPps2_IntrGlobalEnable(...)* na način da se omoguće prekidi za prijem (*XPS2_IPIXR_RX_ALL*) te da se globalno omoguće prekidi u XPS PS2 kontroleru. (TO DO 9)



4.1.14. Napisati liniju **182** kojom se poziva funkcija `XPs2_Recv(...)` na način da se podatak primljen s tipkovnice spremi u varijablu `RxBuffer`. (TO DO 10)

4.1.15. Napisati liniju **255** kojom se poziva funkcija `XIntc_Initialize(...)` s odgovarajućim parametrima. Povratnu vrijednost funkcije spremi u varijablu `Status`. Ova funkcija služi za inicijalizaciju upravljača prekidima. (TO DO 11)

4.1.16. Napisati liniju **273** kojom se poziva funkcija `XIntc_Start(...)` na način da se mod rada upravljača prekidima postavi na mod u kojem su omogućeni samo sklopovski prekidi (`REAL_MODE`). (TO DO 12)

4.1.17. Napisati liniju **282** kojom se poziva funkcija `XIntc_Enable(...)` s odgovarajućim parametrima. Ova funkcija služi za omogućavanje rada upravljača prekidima. *POMOĆ*: Drugi parametar pri pozivu funkcije je ID prekida, a ne `DEVICE_ID`. (TO DO 13)

4.2. Implementirati dizajnirani računalni sustav i napisani programski kod na razvojni sustav i testirati rad sustava.

4.2.1. Podesiti postavke kabela klikom na *Xilinx Tools* → *Configure JTAG Settings* (Odabрати *Digilent USB Cable*).

4.2.2. Podesiti postavke kabela klikom na *Xilinx Tools* → *Configure JTAG Settings* (Odabрати *Digilent USB Cable*).

4.2.3. Pokrenuti program za nadzor serijske komunikacije (npr. *Tera Term* ili *Terminal* unutar SDK) i testirati napisani programski kod.



5. Zadaci za samostalni rad

- 5.1. Napisati programski kod kojim se realizira uključivanje i isključivanje LE dioda razvojnog sustava u ovisnosti o pritisnutim tipkama na tipkovnici. Tipke koje se koriste za paljenje određene LE diode odabrati proizvoljno.
- 5.2. Napisati programski kod kojim se na LE diodama prikazuje broj poziva prekidne rutine za XPS2 kontroler.
- 5.3. Napisati programski kod kojim se realizira ručno rotirajuće svjetlo na LE diodama razvojnog sustava. Pomicanje lijevo i desno treba omogućiti na pritisak proizvoljno odabranih tipki na tipkovnici.



Literatura

- [1] Xilinx, »MicroBlaze Processor Reference Guide - Embedded Development Kit EDK 10.1i,« Xilinx, 2008.
- [2] Xilinx, »LogiCORE IP XPS PS2 Controller (v1.01b),« Xilinx, 2010.
- [3] Digilent, »Nexys3 Board Reference Manual,« Digilent, Pullman, WA, 2013.
- [4] Xilinx, »Embedded System Tools Reference Manual - Embedded Development Kit,« Xilinx, 2008.
- [5] P. Marwedel, Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems, Springer Netherlands, 2011.
- [6] J. O. Hamblen, T. S. Hall i M. D. Furman, Rapid Prototyping of Digital Systems - SOPC Edition, Springer US, 2008.