



1. Uvod

U ovom projektnom zadatku potrebno je dizajnirati računalni sustav zasnovan na MicroBlaze procesoru koji na monitoru realizira kviz znanja s proizvoljnim pravilima. Unos od strane korisnika realizirati korištenjem tipkovnice, a prikaz pitanja i status igre prikazivati na monitoru. Administrator treba imati mogućnost dodavanja novih pitanja.

Kroz rad će se razmotriti i odabrati tehnologije kojima je moguće riješiti zadani problem. Nakon odabira tehnologija pojasniti će se realizacija sustava.



2. Primijenjene tehnologije i alati

2.1. Nexys 3 razvojni sustav

Nexys 3 [1], [2] je razvojni sustav koji sadrži Xilinx Spartan-6 LX16 FPGA (engl. *Field Programmable Gate Array*) [3], 48MB vanjske memorije i dovoljno ulazno/izlaznih uređaja i priključaka za dizajn različitih digitalnih sustava. Namijenjen je za stjecanje iskustva s Xilinx tehnologijama i pogotovo je koristan učenicima i studentima za dizajniranje različitih rješenja bez briga o složenim vanjskim sučeljima; moguće je kreirati potpune digitalne sustave, uključujući kontrolere i ugradbena rješenja temeljena na Xilinx MicroBlaze [4] procesorima. MicroBlaze procesor je 32-bitni programski opisan (engl. *soft core*) RISC (engl. *reduced instruction set computer*) procesor optimiziran za implementaciju na Xilinx FPGA integriranim sklopovima. Procesor ima odvojene adrese za podatke i instrukcije.

Nexys 3 je u potpunosti kompatibilan sa svim Xilinx alatima, uključujući Xilinx Platform Studio i Xilinx Software Development Kit.



Slika 1: Nexys 3 razvojni sustav.

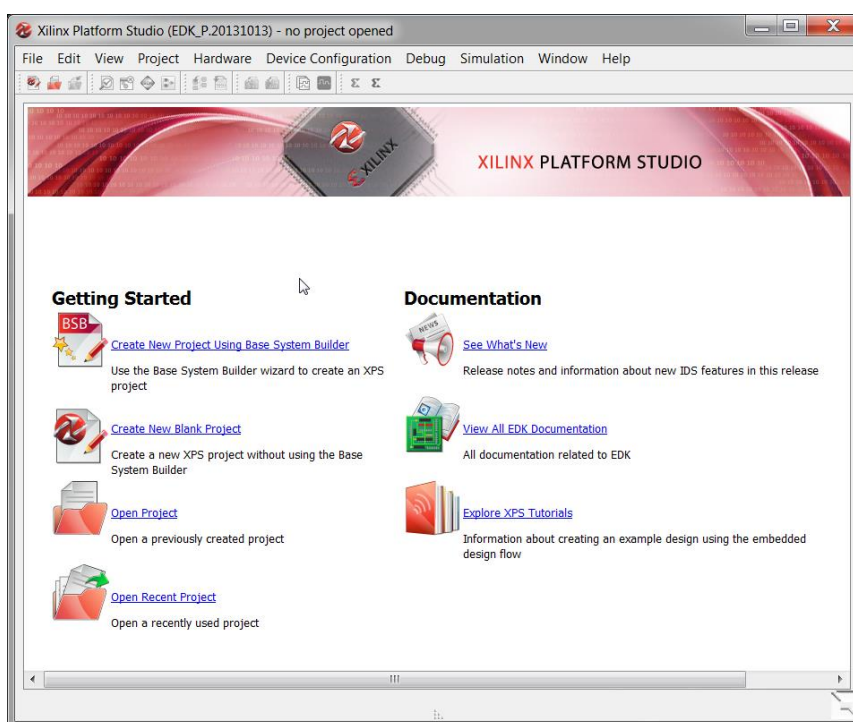


2.2. Xilinx Embedded Development Kit

Xilinx Embedded Development Kit (EDK) [5] je integrirano razvojno okruženje (engl. *Integrated Development Environment (IDE)*) za dizajniranje ugradbenih sustava. Sadrži alate i dokumentaciju potrebnu za dizajniranje potpunih ugradbenih procesorskih sustava za implementaciju na Xilinx FPGA uređajima. Xilinx EDK među ostalim uključuje Xilinx Platform Studio (XPS) i Xilinx Software Development Kit (SDK). Instaliranje Xilinx EDK zahtjeva i instaliranje Xilinx integriranog programskog okruženja (engl. *Integrated Software Environment (ISE)*) [6], koje služi za implementaciju dizajna na Xilinx programabilne logičke uređaje.

2.2.1. Xilinx Platform Studio

Xilinx Platform Studio (XPS) [7] [5] je programski paket koji služi za dizajniranje računalnih sustava. Sastoji se od grafičkog integriranog razvojnog okruženja (engl. *Integrated Development Environment (IDE)*) te svih alata potrebnih za razvoj hardverskih i softverskih komponenti ugradbenih sustava zasnovanih na MicroBlaze procesorima. XPS omogućuje pomoć pri svakom koraku stvaranja sustava, te omogućava provjeravanje stvorenog sustava. Također sadrži različita sučelja za što jednostavniju konfiguraciju arhitektura, sabirnica i perifernih jedinica ugradbenih sustava.

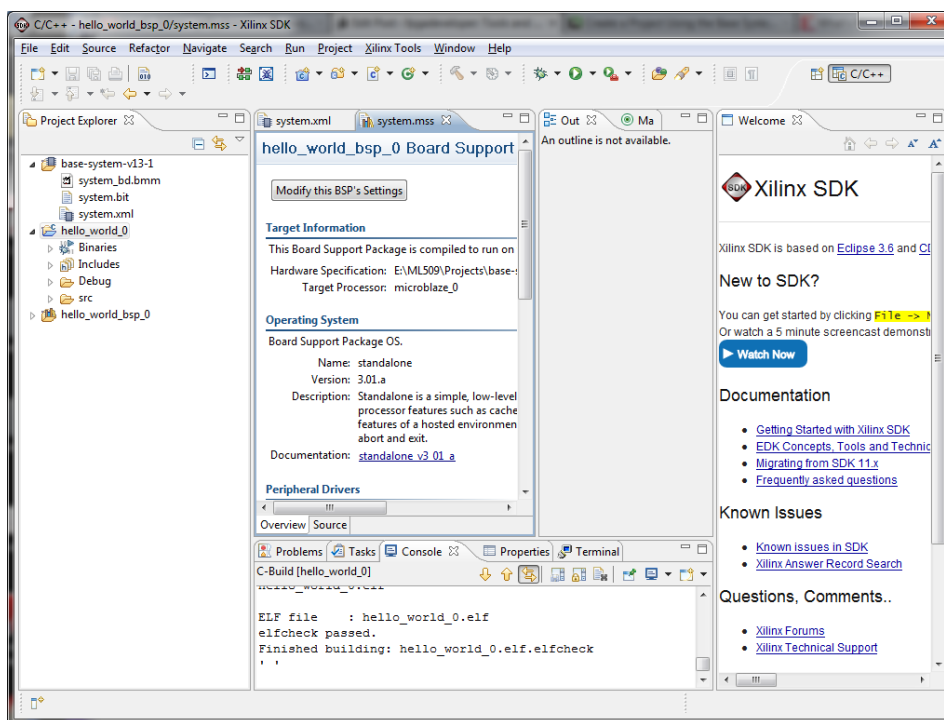


Slika 2: XPS početni zaslon.



2.2.2. Xilinx Software Development Kit

Xilinx Software Development Kit (XSDK) [8], [5] je dopunsko grafičko korisničko sučelje (engl. *graphical user interface (GUI)*) za Xilinx Platform Studio i pruža razvojno okruženje za programske aplikacije. XSDK se temelji na Eclipse standardu otvorenog koda (engl. *open source*). XSDK uključuje C/C++ uređivač i prevoditelj koda, rukovoditelj projekta, pomoć pri greškama, provjera verzija izvornog koda i slično.

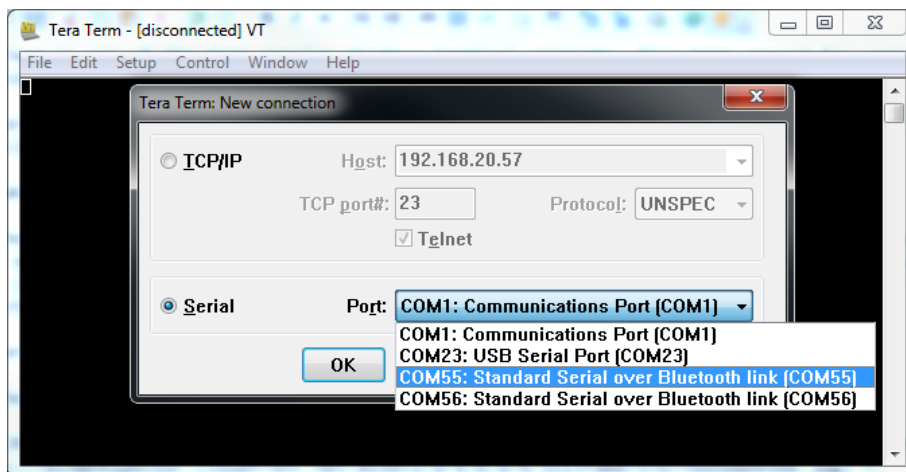


Slika 3: XSDK primjer zaslona.



2.3. Tera Term

Tera Term [9] je *terminal emulator* otvorenog koda za Microsoft Windows operacijske sustave koji podržava serijsku i TCP/IP (telnet, SSH1, SSH2) veze. Ima ugrađeni Macro skriptni jezik i često se koristi za automatizaciju zadataka povezanih s udaljenim vezama pokrenutih od računala. Omogućava slanje i primanje serijskih naredbi.



Slika 4: Tera Term primjer zaslona i dodavanja konekcije.

3. Realizacija sustava

Prije početka potrebno je spojiti tipkovnicu i monitor na Nexys 3 razvojni sustav. Tipkovnica se spaja putem USB priključka, a monitor se spaja putem VGA priključka. Nakon toga potrebno je povezati računalo i Nexys 3 putem UART priključka i priključka za napajanje. Raspored priključaka Nexys 3 razvojnog sustava može se pronaći u literaturi [2].

3.1. Dizajn računalnog sustava

Kako je za realizaciju zadatka potrebno korištenje tipkovnice i monitora, tj. PS2 i VGA sučelja, potrebno je dodati odgovarajuće kontrolere. Kontroler korišten za komunikaciju s tipkovnicom zove se *XPS PS2 Controller* i dostupan je unutar XPS-a, a omogućava komunikaciju s tipkovnicom u polled i interrupt načinima rada. Kontroler korišten za komunikaciju s monitorom zove se *XPS Thin Film Transistor (TFT) Controller* i također je dostupan unutar XPS-a. Kako bi se omogućio ispravan rad TFT kontrolera potrebno je osigurati minimalno 2 MB video memorije.



Nexys 3 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. Također, Nexys 3 razvojni sustava omogućava korištenje maksimalno 64 KB *BRAM* memorije, a kako je TFT kontroleru potrebno osigurati minimalno 2 MB video memorije, u računalni sustav potrebno je dodati *Micron RAM* memoriju čija je maksimalna veličina 16 MB.

Računalni sustav je dizajniran u Xilinx Platform Studio-u pomoću Base System Builder-a [5]. Postavke odabrane pri dizajniranju sustava su:

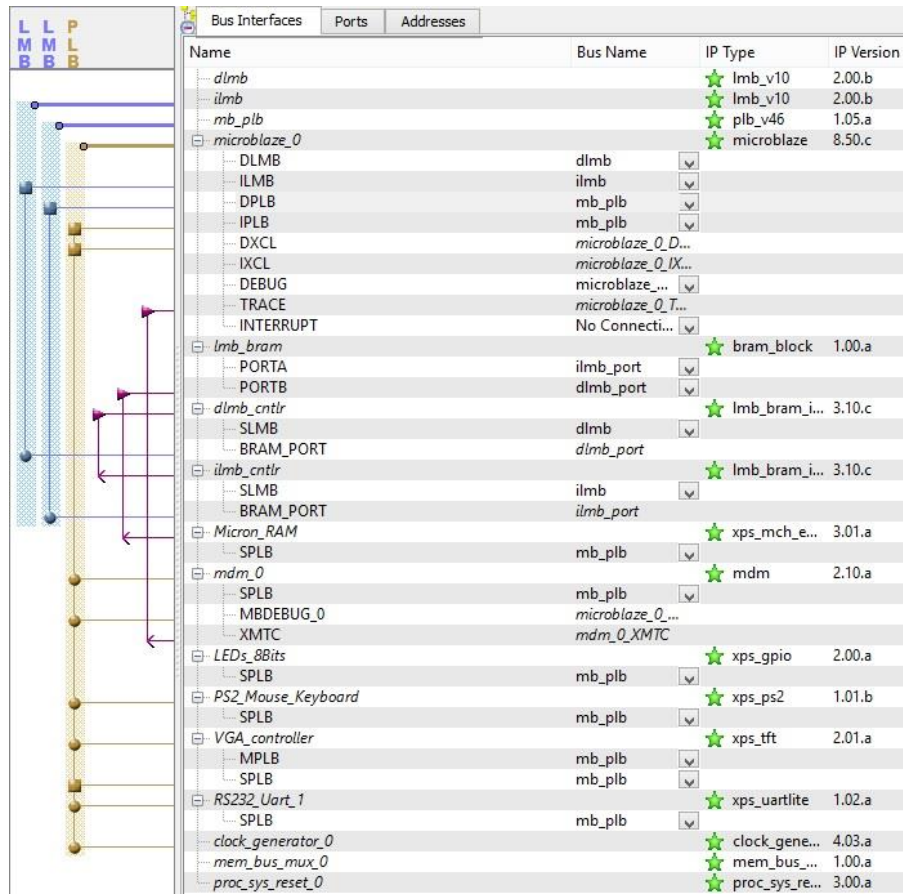
- PLB sabirnica,
- jednoprocesorski MicroBlaze sustav,
- frekvencija procesora 66.67 MHz,
- lokalna memorija procesora 32 KB,
- odabrani periferni sklopovi: LE diode (LEDs_8Bits), Micron RAM memorija (Micron RAM), PS2 kontroler (PS2_Mouse_Keyboard), UART kontroler (RS232_Uart_1), LMB kontroler za BRAM (dlmb_cntlr, ilmb_cntlr),
- ostale postavke ostavljene na predefiniranim vrijednostima.

Detaljne upute sa svim koracima za dizajn korištenog računalnog sustava mogu se pronaći u predlošcima za laboratorijske vježbe [10] [11].

Slika 5 prikazuje kompletno stablo komponenti računalnog sustava. U kompletnom stablu moguće je vidjeti preko koje sabirnice je spojena određena komponenta računalnog sustava.

Slika 6 prikazuje kompletno stablo vanjskih konekcija računalnog sustava.

Slika 7 prikazuje osnovne adrese (Base Address) i maksimalne adrese (High Adress) svake komponente računalnog sustava.



Slika 5: System Assembly View - Bus Interfaces.

Name	Connected Port	Direction	Range	Class	Frequency(Hz)	Reset Polarity
External Ports						
VGA_Blue	VGA_controller::[tft_0]:TFT_VGA_B	0	[5:0]	NONE		
VGA_Green	VGA_controller::[tft_0]:TFT_VGA_G	0	[5:0]	NONE		
VGA_HSYNC	VGA_controller::[tft_0]:TFT_HSYNC	0		NONE		
VGA_Red	VGA_controller::[tft_0]:TFT_VGA_R	0	[5:0]	NONE		
VGA_VSYNC	VGA_controller::[tft_0]:TFT_VSYNC	0		NONE		
VGA_controller_TFT_DE_pin	VGA_controller::[tft_0]:TFT_DE	0		NONE		
VGA_controller_TFT_DPS_pin	VGA_controller::[tft_0]:TFT_DPS	0		NONE		
VGA_controller_TFT_VGA_CLK_pin	VGA_controller::[tft_0]:TFT_VGA_CLK	0		NONE		
fpga_0_LEDs_8Bits_GPIO_IO_O_pin	LEDs_8Bits::[gpio_0]:GPIO_IO_O	0	[0:7]	NONE		
fpga_0_PS2_Mouse_Keyboard_PS2_1_CLK_pin	PS2_Mouse_Keyboard::[ps2_0]:PS2_1_CLK	IO		NONE		
fpga_0_PS2_Mouse_Keyboard_PS2_1_DATA_pin	PS2_Mouse_Keyboard::[ps2_0]:PS2_1_DATA	IO		NONE		
fpga_0_PS2_Mouse_Keyboard_PS2_2_CLK_pin	PS2_Mouse_Keyboard::[ps2_0]:PS2_2_CLK	IO		NONE		
fpga_0_PS2_Mouse_Keyboard_PS2_2_DATA_pin	PS2_Mouse_Keyboard::[ps2_0]:PS2_2_DATA	IO		NONE		
fpga_0_RS232_Uart_1_RX_pin	RS232_Uart_1::[uart_0]:RX	1		NONE		
fpga_0_RS232_Uart_1_TX_pin	RS232_Uart_1::[uart_0]:TX	0		NONE		
fpga_0_clk_1_sys_clk_pin	clock_generator_0::CLKIN	1		CLK	100000000	
fpga_0_mem_bus_mux_0_DQ_pin	mem_bus_mux_0::[mem_bus_mux_0]:DQ	IO	[0:15]	NONE		
fpga_0_mem_bus_mux_0_FLASH_ADDR_pin	mem_bus_mux_0::[mem_bus_mux_0]:FLASH_ADDR	0	[5:7]	NONE		
fpga_0_mem_bus_mux_0_FLASH_CEN_O_pin	mem_bus_mux_0::[mem_bus_mux_0]:FLASH_CEN_O	0		NONE		
fpga_0_mem_bus_mux_0_FLASH_RPN_O_pin	mem_bus_mux_0::[mem_bus_mux_0]:FLASH_RPN_O	0		NONE		
fpga_0_mem_bus_mux_0_MEM_ADDR_pin	mem_bus_mux_0::[mem_bus_mux_0]:MEM_ADDR	0	[0:22]	NONE		
fpga_0_mem_bus_mux_0_MEM_OEN_pin	mem_bus_mux_0::[mem_bus_mux_0]:MEM_OEN	0		NONE		
fpga_0_mem_bus_mux_0_MEM_WEN_pin	mem_bus_mux_0::[mem_bus_mux_0]:MEM_WEN	0		NONE		
fpga_0_mem_bus_mux_0_MOSI_QUAD_SPL_pin	mem_bus_mux_0::[mem_bus_mux_0]:MOSI_QUAD	IO		NONE		
fpga_0_mem_bus_mux_0_QUAD_SPL_C_O_pin	mem_bus_mux_0::[mem_bus_mux_0]:QUAD_SPL_C_O	0		NONE		
fpga_0_mem_bus_mux_0_QUAD_SPL_S_O_pin	mem_bus_mux_0::[mem_bus_mux_0]:QUAD_SPL_S_O	0		NONE		
fpga_0_mem_bus_mux_0_RAM_BEN_O_pin	mem_bus_mux_0::[mem_bus_mux_0]:RAM_BEN_O	0	[0:1]	NONE		
fpga_0_mem_bus_mux_0_RAM_CEN_O_pin	mem_bus_mux_0::[mem_bus_mux_0]:RAM_CEN_O	0		NONE		
fpga_0_rst_1_sys_rst_pin	clock_generator_0::RST	1		RST		1
proc_sys_reset_0:Ext_Reset_In						

Slika 6: System Assembly View – Ports.



Bus Interfaces Ports Addresses								
Instance	Base Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name	Lock	
microblaze_0's Address Map								
dlmb_cntlr	C_BASEADDR	0x00000000	0x00007FFF	32K	SLMB	dlmb		
ilmb_cntlr	C_BASEADDR	0x00000000	0x00007FFF	32K	SLMB	ilmb		
LEDs_8Bits	C_BASEADDR	0x81400000	0x8140FFFF	64K	SPLB	mb_plb		
RS232_Uart_1	C_BASEADDR	0x84000000	0x8400FFFF	64K	SPLB	mb_plb		
mdm_0	C_BASEADDR	0x84400000	0x8440FFFF	64K	SPLB	mb_plb		
Micron_RAM	C_MEM0_BASE...	0x85000000	0x85FFFFFF	16M	SPLB	mb_plb		
PS2_Mouse_Keyboard	C_BASEADDR	0x86A00000	0x86A0FFFF	64K	SPLB	mb_plb		
IP Type : xps_mch_emc	C_SPLB_BASEA...	0x86E00000	0x86E0FFFF	64K	SPLB	mb_plb		

Slika 7: System Assembly View – Addresses.

Dizajnirani računalni sustav sastoji se od sljedećih dijelova:

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

3.2. Programsko rješenje sustava

Programsko rješenje realizirano je u Xilinx Software Development Kit-u. U ovom potpoglavlju opisani su dijelovi koda koji obavljaju specifične zadatke. Kompletan programski kod priložen je na CD-u zajedno sa cjelokupnim rješenjem sustava.

Prije funkcije *main()* navedene su deklaracije korištenih funkcija, definirane globalne varijable te definirane konstante sa heksadekadskim ili dekadskim vrijednostima.

Na početku programa potrebno je inicijalizirati monitor i tipkovnicu. To se obavlja funkcijama prikazanim u programskom kodu 1 i u programskom kodu 2. U glavnom programu se funkcijama predaju heksadekadske *ID* vrijednosti monitora i tipkovnice.

**Linija Kod**

```
1:  int TftInitialize(u32 TftDeviceId) {
2:      int Status;
3:      XTft_Config *TftConfigPtr;
4:      TftConfigPtr = XTft_LookupConfig(TftDeviceId);
5:      if (TftConfigPtr == (XTft_Config *)NULL) {
6:          return XST_FAILURE; }
7:      XTft_CfgInitialize(&TftInstance, TftConfigPtr,
8:          TftConfigPtr->BaseAddress);
9:      if (Status != XST_SUCCESS) {return XST_FAILURE;}
10:     while (XTft_GetVsyncStatus(&TftInstance) !=
11:         XTFT_IESR_VADRLATCH_STATUS_MASK);
12:     XTft_SetFrameBaseAddr(&TftInstance, TFT_FRAME_ADDR);
13:     return XST_SUCCESS; }
```

Programski kod 1: Definicija funkcije za inicijalizaciju VGA kontrolera.

Linija Kod

```
1:  int Ps2Initialize(u16 Ps2DeviceId) {
2:      XPs2_Config *Ps2ConfigPtr;
3:      Ps2ConfigPtr = XPs2_LookupConfig(Ps2DeviceId);
4:      if (Ps2ConfigPtr == NULL) {
5:          print("Config Lookup FAILED!\r\n");
6:          return XST_FAILURE; }
7:      else print("Config Lookup SUCCESS!\r\n");
8:      XPs2_CfgInitialize(&Ps2Inst, Ps2ConfigPtr,
9:          Ps2ConfigPtr->BaseAddress);
10:     print("PS2 Initialize SUCCESS!\r\n");
11:     return XST_SUCCESS; }
```

Programski kod 2: Definicija funkcije za inicijalizaciju PS2 kontrolera.

Za ispis znakova koristi se funkcija koja piše znak po znak sve dok ne dođe do *null* znaka ili kraja niza. Definicija funkcije je prikazana u programskom kodu 3.

Definicija funkcije za slanje podataka putem PS2 sučelja prikazana je u programskom kodu 4. Funkcija prima podatak koji treba poslati.

Definicija funkcije za primanje podataka putem PS2 sučelja prikazana je u programskom kodu 5.



Linija Kod

```
1: int TftWriteString(XTft *InstancePtr, const u8 *CharValue){
2:     while (*CharValue != 0) {
3:         XTft_Write(InstancePtr, *CharValue);
4:         CharValue++; }
5:     return XST_SUCCESS; }
```

Programski kod 3: Definicija funkcije za ispis znakova.

Linija Kod

```
1: int Ps2SendData(u8 data){
2:     u32 StatusReg;
3:     u32 BytesSent;
4:     u32 BytesReceived;
5:     int ACK_WAIT = 10000;
6:     BytesSent = XPs2_Send(&Ps2Inst, &data, 1);
7:     if( BytesSent != 1) {
8:         print("Data Send FAILED!\r\n");
9:         return XST_FAILURE; }
10:    do {
11:        StatusReg = XPs2_GetStatus(&Ps2Inst);
12:        ACK_WAIT--;
13:    }while((StatusReg & XPS2_STATUS_RX_FULL) == 0 &&
14:        ACK_WAIT > 0);
15:    BytesReceived = XPs2_Recv(&Ps2Inst, &RxBuffer, 1);
16:    if (RxBuffer != KEYBOARD_ACK) {
17:        print("Nije primljen ACK byte!\r\n");
18:        return XST_FAILURE; }
19:    else xil_printf("Data %x Send SUCESS!\r\n", data);
20:    return XST_SUCCESS; }
```

Programski kod 4: Definicija funkcije za slanje podataka putem PS2 sučelja.

**Linija Kod**

```
1:  int Ps2ReceiveData() {
2:      u32 StatusReg;
3:      u32 BytesReceived;
4:      do {
5:          StatusReg = XPs2_GetStatus(&Ps2Inst);
6:      }while((StatusReg & XPS2_STATUS_RX_FULL) == 0);
7:      BytesReceived = XPs2_Rcv(&Ps2Inst, &RxBuffer, 1);
8:      return XST_SUCCESS; }
```

Programski kod 5: Definicija funkcije za primanje podataka putem PS2 sučelja.

Za unos jednog od četiri ponuđena odgovora ili odabir u izborniku koristi se funkcija koja prima četiri moguća parametra koji su heksadekadske vrijednosti definirane na početku. Definicija funkcije je prikazana u programskom kodu 6. Heksadekadske vrijednosti su dobivene spajanjem razvojnog sustava i računala putem UART sučelja te prikaza pomoću Tera Term-a. Za ispis heksadekadske vrijednosti pritisnute tipke u konzolu Tera Term-a se koristi funkcija `xil_printf()`.

Linija Kod

```
1:  void input (int a, int b, int c, int d) {
2:      int Status;
3:      while (1){
4:          while(RxBuffer != 0xF0) Ps2ReceiveData();
5:          Status = Ps2ReceiveData();
6:          if (Status != XST_SUCCESS){
7:              return XST_FAILURE;
8:              print("Data receiving FAILED!\r\n"); }
9:          else{
10:             xil_printf("HEX vrijednost = %x\r\n", RxBuffer);
11:             if(RxBuffer == a || RxBuffer == b ||
12:                RxBuffer == c || RxBuffer == d)    break;  } } }
```

Programski kod 6: Definicija funkcije unos odabira.

Pitanja su definirana u funkciji `questions()`. Za generiranje nasumičnih pitanja koristi se funkcija `rand()` koja vraća cjelobrojnu vrijednost između 0 i broja definiranih pitanja. Kako bi se postavila početna vrijednost na temelju koje će se stvarati slučajni brojevi, koristi se funkcija `srand()`, koja prima cijeli broj (bez predznaka) i predstavlja generator slučajnih brojeva.



Zapravo, postavlja se vrijednosti na temelju koje će se u funkciji *rand()* izračunati nasumičan broj kojeg će vratiti. Poželjno je da vrijednost koja se predaje funkciji *srand()* bude također slučajna. Stoga se često koristi trenutno sistemsko vrijeme kojeg je moguće dobiti pozivom funkcije *time()*.

Kako Nexys 3 razvojni sustav nema nikakvo sistemsko vrijeme, potrebno je omogućiti generiranje nasumičnog broja. Definirana je konstanta *DATE* koja prima brojčanu vrijednost datuma u obliku DDMMYYYY. Na početku programa od korisnika traži unos broja 1 - 9. Vrijednost koja se predaje funkciji *srand()* dobivena je ostatkom cjelobrojnog dijeljenja vrijednosti konstante *DATE* sa unesenim brojem.

Kako bi se spriječilo ponavljanja istih pitanja definirano je polje cijelih brojeva gdje su vrijednosti svih elemenata polja postavljene na 0. Nakon generiranja broja pitanja, vrijednost elementa na indeksu generiranog broja u polju se postavlja na 1 što znači da pitanje ne može više biti izvučeno. Za prikaz pitanja i odgovora koristi se naredba *switch* kojoj se predaje nasumično generirani broj. Navedeno je prikazano u programskom kodu 7.

Linija Kod

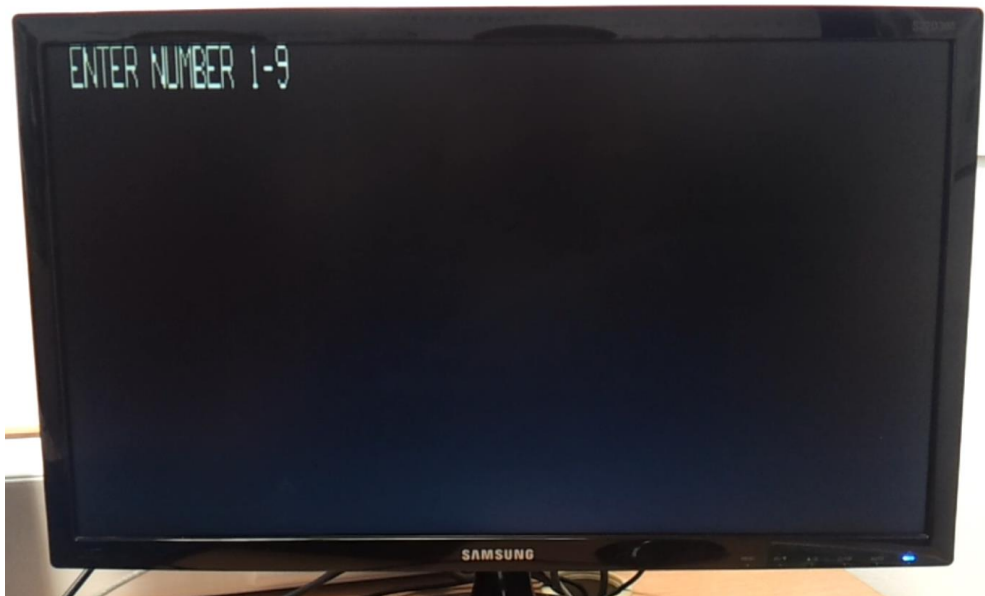
```
1:  int questions(void) {
2:      int i, current, counter = 0;
3:      int mq[21] = {0};
4:      for (i = 0; i < NUMBER_OF_MONEY_ROUND_QUESTIONS; i++) {
5:          current = rand() % NUMBER_OF_DEFINED_QUESTIONS + 1;
6:          while (mq[current] == 1)
7:              {current = rand() % NUMBER_OF_DEFINED_QUESTIONS+1;}
8:          mq[current] = 1;
9:          switch(current) {
10:              ...
```

Programski kod 7: Definicija funkcije za generiranje pitanja.



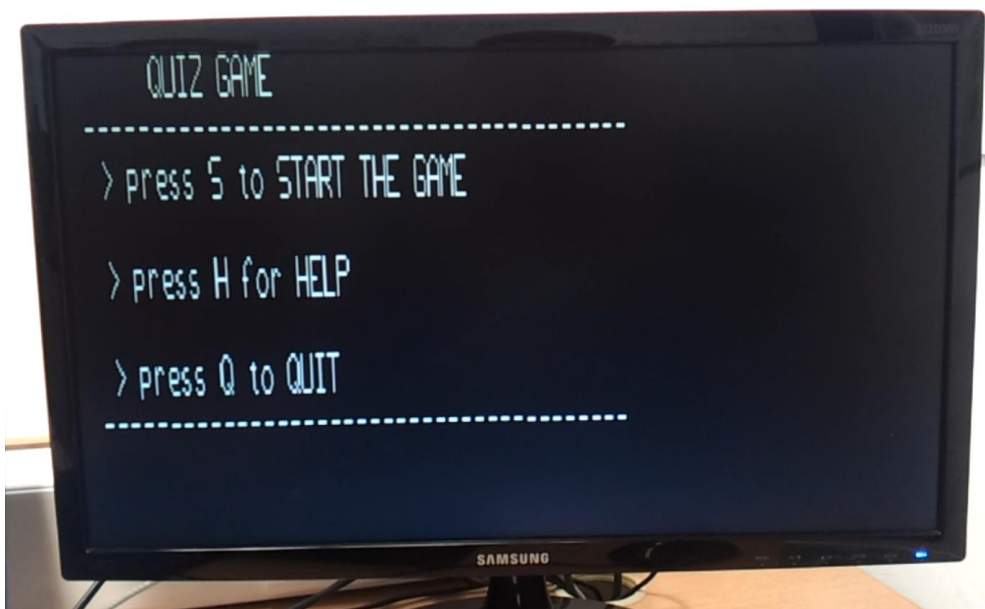
3.3. Prikaz rada sustava

Na početku se od korisnika zahtjeva unos broja 1 - 9 kao što je prikazano na slici 8.



Slika 8: Zahtjev za unosom broja 1 - 9.

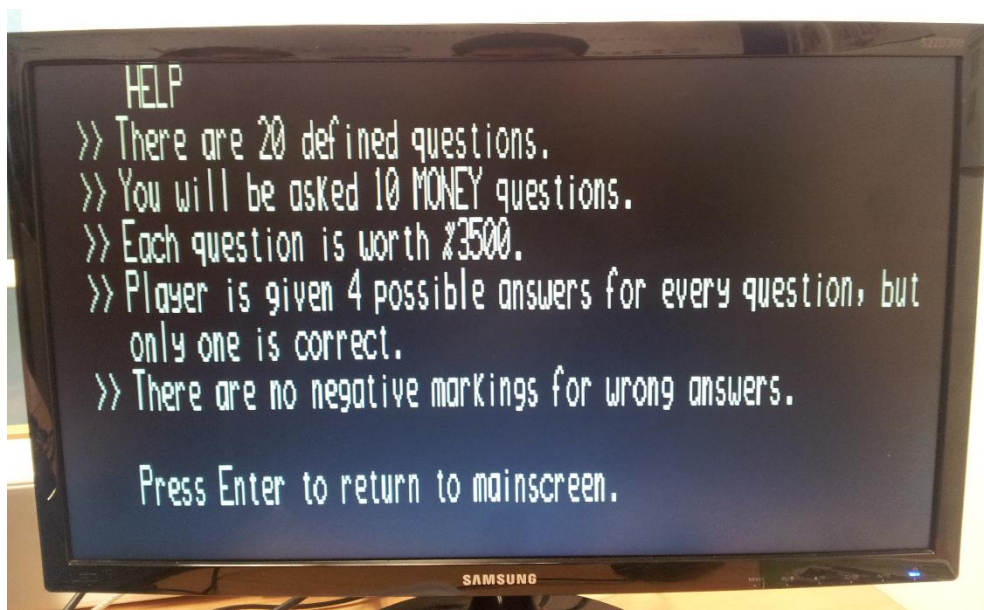
Glavni izbornik prikazan je na slici 9.



Slika 9: Glavni izbornik.

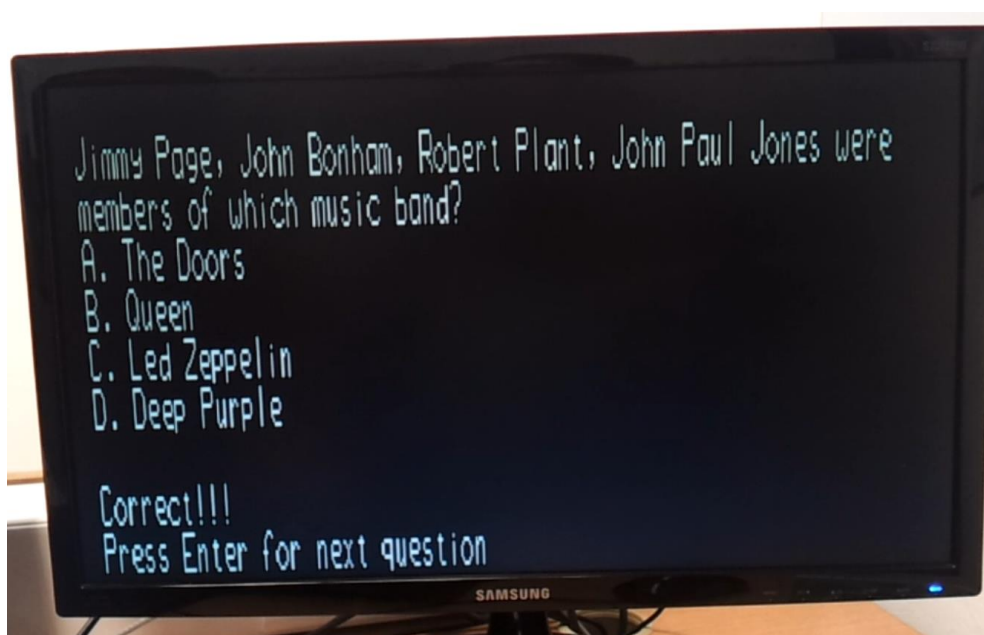


Pomoćni izbornik (engl. *help*) opisuje pravila igre i prikazan je na slici 10.



Slika 10: Pomoćni izbornik.

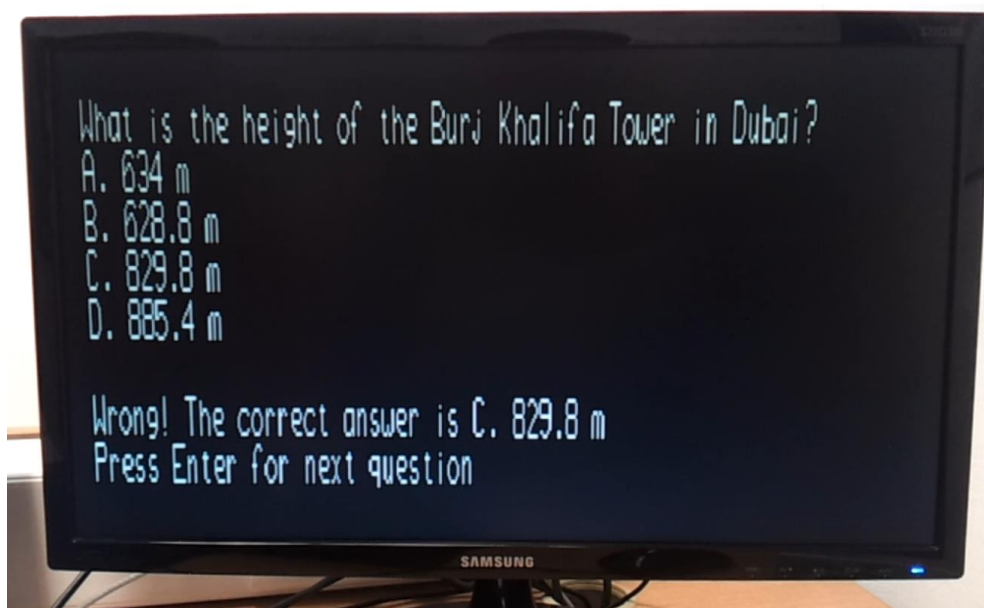
Primjer točno odgovorenog pitanja prikazan je na slici 11.



Slika 11: Primjer točno odgovorenog pitanja.

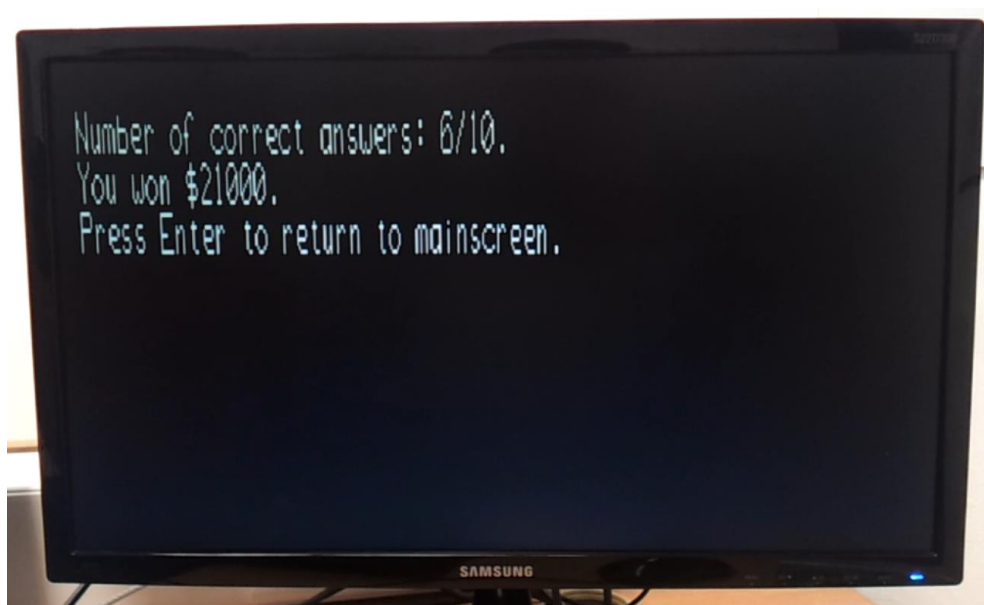


Primjer netočno odgovorenog pitanja prikazan je na slici 12.



Slika 12: Primjer netočno odgovorenog pitanja.

Nakon 10 pitanja prikazuje se broj točno odgovorenih pitanja i osvojeni iznos kao što je prikazano na slici 13.



Slika 13: Primjer zaslona nakon odgovorenih pitanja.

Pritiskom na tipku Q u glavnom izborniku dolazi do završavanja programa gdje dolazi do brisanja sadržaja ekrana i ostaje samo crni zaslon.



3.4. Problemi prilikom izrade projektnog zadatka

Glavni problem pri izradi projektnog zadatka predstavljao je nedostatak lokalne memorije Nexys 3 razvojnog sustava. Nakon dizajniranja računalnog sustava u Xilinx Platform Studio-u nije ostalo dovoljno memorije za planirano programsko rješenje u Xilinx Software Development Kit-u.

Zamišljena rješenja koja nisu realizirana:

- Dodavanje tipkala razvojnog sustava gdje bi se nasumična vrijednost predana funkciji *srand()* dobivala pritiskom na tipkalo, čime bi se omogućila bolja nasumičnost.
- Prikaz točnog odgovora zelenom, a netočnog odgovora crvenom bojom korištenjem funkcije *XTft_SetColor()*.
- Omogućavanje administratoru unosa novih pitanja pomoću tipkovnice; ostalo je memorije za kratka pitanja s kratkim odgovorima pa je ovaj dio preskočen.

Pokušano je prebacivanje cijelog dizajniranog sustava u Micron RAM memoriju razvojnog sustava, no time se smanjila brzina odziva, pa je i ovo rješenje preskočeno.



4. Zaključak

U ovom projektnom zadatku dizajniran je računalni sustav zasnovan na MicroBlaze procesoru koji na monitoru realizira kviz znanja s proizvoljnim pravilima. Sustav je uspješno realiziran i obavlja zamišljeni zadatak: omogućava korisniku prikaz pravila igre, te mogućnost odgovaranja na postavljeni broj pitanja, gdje svako pitanje vrijedi jednaki iznos novca. Jedina mana je nemogućnost realizacije dodavanja pitanja od strane administratora, kao i drugi problemi opisani u potpoglavlju 3.4. Unatoč tome, i ovakva realizacija pokazuje mogućnost samostalnog sličnog projekta na Nexys 3 razvojnom sustavu.



Literatura

- [1] <http://store.digilentinc.com/nexys-3-spartan-6-fpga-trainer-board-limited-time-see-nexys4-ddr/>
[Pokušaj pristupa Veljača 2017.]
- [2] Digilent, »Nexys3 Board Reference Manual,« Digilent, Pullman, WA, 2013.
- [3] <https://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html>
[Pokušaj pristupa Veljača 2017.]
- [4] Xilinx, »MicroBlaze Processor Reference Guide - Embedded Development Kit EDK 10.1i,« Xilinx, 2008.
- [5] Xilinx, »Embedded System Tools Reference Manual - Embedded Development Kit,« Xilinx, 2008.
- [6] <https://www.xilinx.com/products/design-tools/ise-design-suite.html>
[Pokušaj pristupa Veljača 2017.]
- [7] <https://www.xilinx.com/products/design-tools/xps.html>
[Pokušaj pristupa Veljača 2017.]
- [8] <https://www.xilinx.com/products/design-tools/embedded-software/sdk.html>
[Pokušaj pristupa Veljača 2017.]
- [9] <https://ttssh2.osdn.jp/index.html.en>
[Pokušaj pristupa Veljača 2017.]
- [10] I. Vidović, »Laboratorijska vježba 3,« Osijek, 2016.
- [11] I. Vidović, »Laboratorijska vježba 1,« Osijek, 2016.