VGA grafički upravljač s FPGA

Horvat, Matej

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:452841

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-24**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstvo

VGA GRAFIČKI UPRAVLJAČ S FPGA

Završni rad

Matej Horvat

Osijek, 2019.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. VGA I PS2 2
2.1. VGA 2
2.1.1. VGA Standard 3
2.1.2. Vremenski intervali VGA 5
2.2. PS/2 7
2.2.1. Komunikacija tipkovnice s PS/2 protokolom 7
2.2.2. Karakteristike PS/2 protokola 8
3. FPGA 10
3.1. Povijest FPGA 10
3.2. Građa FPGA 10
3.3. Upotreba FPGA 11
3.3.1. Dizajn 12
3.3.2. Simulacija dizajna 12
3.3.3. Sinteza dizajna 12
3.3.4. Implementacija dizajna 12
3.3.5. Programiranje uređaja 13
3.3.6. Verifikacija dizajna 13
4. IMPLEMENTACIJA 14
4.1. PS/2 modul 15
4.2. RAM Modul 17
4.3. VGA modul 18
4.4. Testiranje 19
5. ZAKLJUČAK 22
LITERATURA 23

SAŽETAK 24
ABSTRACT 25
ŽIVOTOPIS 26
PRILOZI 27

1. UVOD

Grafički upravljač (engl. *VGA controller*) obrađuje grafiku primljenu od strane računala i kreira točke te linije na ekranu radi generiranja slike. Tema završnog rada je implementacija VGA grafičkog upravljača na FPGA (engl. *field-programmable gate array*) razvojnom sustavu. FPGA koji je korišten prilikom implementacije je Nexys 3 Spartan-6 FPGA Trainer Board, tvrtke Digilent. VGA grafički upravljač je implementiran uz pomoć VHDL-a (engl. *Very High Speed Integrated Circuit Hardware Description Language*), jezika za opisivanje sklopovlja. Konačni sustav sastoji se od 3 modula.

Jedan od modula je PS/2 (engl. *Personal Systems/2*), protokol za komunikaciju računalne periferije s računalom, a u ovom slučaju za komunikaciju s FPGA je korištena tipkovnica. Sustav sadrži RAM (engl. *Random Access Memory*) koji pohranjuje informacije, odnosno znakove koje korisnik unosi pomoću tipkovnice. Uneseni znakovi od strane korisnika su prikazani na monitoru pomoću VGA modula. Korisniku je dozvoljeno unositi samo znakove i brojeve, odnosno ukoliko pritisne neku drugu tipku na tipkovnici (s iznimkom dvije koje su naknadno objašnjene u 4. poglavlju) osim brojeva ili slova, na ekran će se ispisati razmak.
Za izradu sustava korišten je ISE Design Suite: System Edition koji omogućava sintezu jezika za opisivanje skopovlja te njegovu simulaciju, implementaciju, povezivanje s uređajem te JTAG (engl. *Joint Test Action Group*) programiranje samog FPGA integriranog sklopa. Za implementaciju, osim FPGA, od sklopovlja korišteni su tipkovnica i monitor. Rezolucija monitora koja je korištena u VGA standardu je 640x480 piksela.

1.1. Zadatak završnog rada

Zadatak završnog rada je implementirati i testirati VGA grafički upravljač, koristeći FPGA razvojni sustav. Grafički upravljač potrebno je napraviti VHDL jezikom. Osim toga, potrebno je implementirati PS/2 protokol za komunikaciju s tipkovnicom te RAM.

2. VGA I PS2

Oba protokola prvi put su predstavljena 1987. godine u IBM-ovoj seriji osobnih računala proizvedenih pod nazivom IBM Personal Systems/2 iz čijeg naziva potječe i samo ime PS/2 protokola. PS/2 je protokol za komunikaciju računalne periferije (tipkovnice i miša) sa samim računalom dok VGA (engl. *Video Graphics Array*) predstavlja standardizirano sučelje za povezivanje računala s analognim monitorima [1].

2.1. VGA

Dio VGA standarda koji se odnosi na samo računalo naziva se grafički upravljač, dok se dio koji se odnosi na monitor naziva upravljač prikaza. VGA monitori mogu biti CRT (engl. *cathode ray tubes*) ili LCD (engl. *liquid crystal displays*). LCD se razlikuje od CRT-a po tome što su potpuno digitalni te je njihovo standardno sučelje DVI (engl. *Digital Video Interface*), no usprkos tome u mnogo slučajeva sadrže VGA utor tako da se VGA standard može koristiti.

Tablica 2.1 : Prikaz imena, rezolucija, omjera te frekvencija VGA standarda [2].

STANDARD	REZOLUCIJA	OMJER	FREKVENCIJA OSVJEŽAVNJA (Hz)
VGA	640 x 480	4:3	60, 72, 75, 85
SVGA	800 x 600	4:3	60, 72, 75, 85
XGA	1024 x 768	4:3	60, 70, 75, 85
SXGA	1280 x 1024	5:4	60, 75, 85, 100

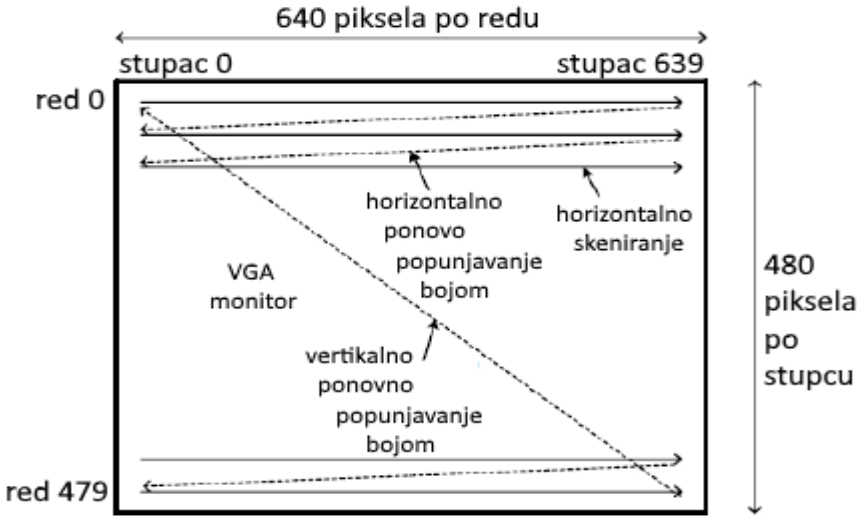
Osnovna rezolucija VGA monitora je 640x480 x 60 Hz. To znači da se sastoji od 640 redaka i 480 stupaca piksela te se slika osvježava 60 puta po sekundi. Osim klasičnog VGA, postoje standardi veće rezolucije kao što su SVGA (engl. *super VGA*), XGA *(engl. extended graphics array*), SXGA (engl. *super extended graphics array*) te ostali. Prikaz tih standarda te njihovih rezolucija, omjera te frekvencija na kojima rade nalaze se na tablici 2.1.

2.1.1. VGA Standard

Prikaz slike VGA standardom izvodi se pomoću 5 signala za prikaz slike. Ti signali su signal za horizontalnu sinkronizaciju, signal za vertikalnu sinkronizaciju te signali za prikaz plave, zelene i crvene boje. VGA monitor funkcionira na principu emitiranja energije u crvenom, plavom te zelenom spektru vidljivih boja ovisno o jačini napona koju određuju signali za crvenu, plavu i zelenu boju.

Na monitoru se pikseli prikazuju od gornjeg ljevog ugla prema desno po horizontalnoj liniji, te se na taj način liniju po liniju kreću prema dnu ekrana kao što je prikazano na slici 2.1. Prilikom tog procesa signal za horizontalnu sinkronizaciju sinkronizira svaku novu liniju. Kada dosegne dno ekrana, signal za vertikalnu sinkronizaciju uzrokuje početak ponovnog procesa prikaza slike od gornjeg lijevog vrha ekrana. Grafički upravljač na izlaz stalno daje vrijednosti signala zaduženih za crvenu, plavu te zelenu boju uz istovremenu sinkronizaciju horizontalnih linija uz okomito ispunjavanje bojom.

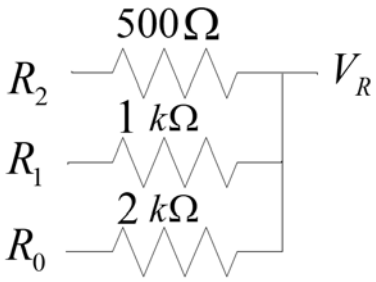
Za spajanje signala s monitorom koristi se DB15 priključak (VGA priključak). VGA priključak se naziva DB15 te kao što mu ime govori sastoji se od 15 pinova (slike 2.2 i 2.3). Određeni pinovi ovog priključka odnose se na signale kao što su signali za prikaz boja te signali



Slika 2.1 : Prikaz piksela na ekranu.

Slika 2.2 : „Muški“ VGA priključak.

Slika 2.3 : „Ženski“ VGA priključak.



Slika 2.4 : Električni krug [3].

za vertikalno i horizontalno sinkroniziranje, dok se određeni pinovi odnose na samu identifikaciju te podešavanje monitora.

Pinovi na VGA priključku monitora za vrijednosti crvene, plave te zelene boje su analogni. Problem koji nastaje prilikom korištenja FPGA za prikaz boja je taj što su signali na izlaznim pinovima FPGA digitalni. Iz tog razloga digitalni izlazi moraju biti pretvoreni u analogne vrijednosti. U nastavku je prikazan jedan od načina na koji je moguće dobiti analogne vrijednosti od digitalnih korištenjem određenog digitalno-analognog pretvornika [2].

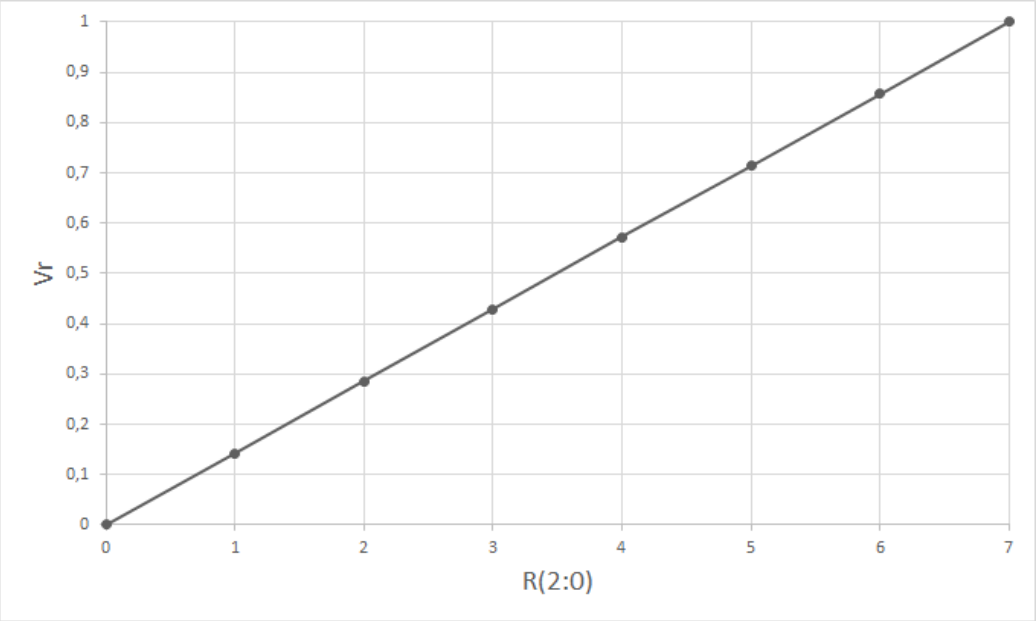
Na slici 2.4 prikazana je slika električnog kruga s 3 otpornika različitih vrijednosti. U ovom slučaju svaka žica predstavlja bit crvene boje, ali se ovaj princip koristi i za ostale dvije boje čijim intenzitetom VGA direktno manipulira (plava i zelena). Raspisuje se Kirchoffov zakon za struje za ovaj krug te se dobije izraz (1). Množenjem tog izraza, te prebacivanjem napona s lijeve strane jednadžbe, a otpora s desne strane i dijeljenja s konstantom dolazi se do izraza (4). Uz pomoć ovog izraza moguće je manipulirati sa 8 različitih intenziteta crvene boje, pomoću digitalnih izlaza, na način prikazan u tablici 2.2. Na slici 2.5 prikazan je odnos vrijednosti izlaznog napona u odnosu na digitalnu vrijednost ulaza [3].

$$\frac{V_R - R_2}{0.5k\Omega} + \frac{V_R - R_1}{1k\Omega} + \frac{V_R - R_0}{2k\Omega} = 0 \tag{1}$$

$$4V_R - 4R_2 + 2V_R - 2R_1 + V_R - R_0 = 0 \tag{2}$$

$$7V_R = 4R_2 + 2R_1 + 1R_0 \tag{3}$$

$$V_R = \frac{4}{7}R_2 + \frac{2}{7}R_1 + \frac{1}{7}R_0 \tag{4}$$



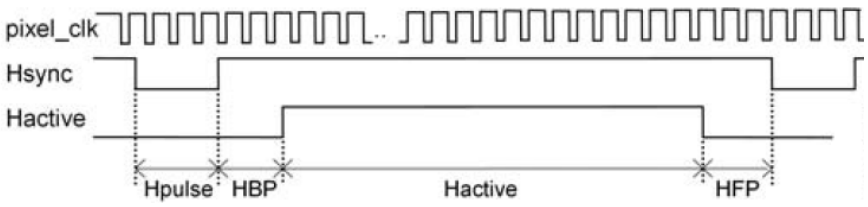
Slika 2.5: Ovisnost analognog izlaznog napona o digitalnom ulazu [3].

Tablica 2.2: Prikaz osam različitih intenziteta boja [3].

R_2	R_1	R_0	V_R
0	0	0	0
0	0	1	1/7
0	1	0	2/7
0	1	1	3/7
1	0	0	4/7
1	0	1	5/7
1	1	0	6/7
1	1	1	1

2.1.2. Vremenski intervali VGA

Standardna frekvencija VGA monitora je 59.94 Hz što znači da se ekran osvježava 60 puta u sekundi. Pomoću te informacije, moguće je odrediti potrebna vremena izvođenja za svaki piksel, horizontalne linije te okomita popunjavanja bojom. Prvi korak je određivanje signala takta (engl. *pixel clock*) kojim radi VGA grafički upravljač ove rezolucije, a to je 25.175 MHz. Nakon toga, određuju se vremenski intervali signala za horizontalnu sinkronizaciju (slika 2.6) koji se sastoji od četiri dijela: pulsa sinkronizacije (Hpulse), stražnjeg praga (HBP – engl. *horizontal back porch*), aktivanog prikaza (Hactive) te prednjeg praga (HFP – engl. *horizontal front porch*). Puls sinkronizacije označava početak nove horizontalne linije te spušta signal za horizontalnu sinkronizaciju u logičku '0'. Taj isti signal se zatim vraća u logičku '1' za stražnji prag, a on predstavlja dio vremena u kojem pikseli još nisu prikazani na ekranu. Nakon stražnjeg praga, signal ostaje u logičkoj '1' tijekom perioda aktivnog prikaza prilikom kojeg su



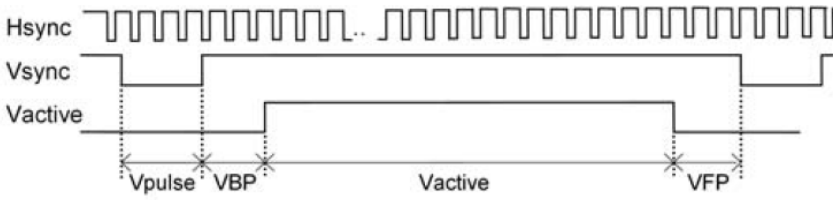
Slika 2.6: Prikaz signala za horizontalno generiranje [2].

Tablica 2.3: Prikaz horizontalnih vremena sinkronizacije [2].

VGA standard		Osvježavanje (Hz)	Signal takta (MHz)	Horizontalna vremena sinkronizacije (pikseli)				
Ime	**Rezolucija**			**Hpulse**	**HBP**	**Hactive**	**HFP**	**Broj piksela**
VGA	640 x 480	60	25.175	96	48	640	16	800
VGA	640 x 480	75	31.5	96	48	640	16	800
SVGA	800 x 600	60	40	128	88	800	40	1056
SVGA	800 x 600	75	49.5	80	160	800	16	1056
SVGA	800 x 600	85	56.25	64	152	800	32	1048
XGA	1024 x 768	60	65	136	160	1024	24	1344
XGA	1024 x 768	85	94.5	96	208	1024	48	1376
SXGA	1280 x 1024	60	108	112	248	1280	48	1688

pikseli ispisani s lijeva prema desno. Također, signal za horizontalnu sinkronizaciju ostaje u logičkoj '1' i tijekom prednjeg praga prilikom kojeg se pikseli ne ispisuju po ekranu, nakon čega pada u logičku '0'. Trajanje pojedinih elemenata horizontalne sinkronizacije za pojedinu rezoluciju vidljivi su na tablici 2.3.

Vrlo slično tom procesu izvodi se proces vertikalne sinkronizacije (slika 2.7). Proces je praktički jednak prethodnom, osim što se parametri više ne odnose na horizontalnu sinkronizaciju već vertikalnu. Sastoji se od četiri dijela: pulsa sinkronizacije (Vpulse), stražnjeg praga (VBP – engl. *vertical back porch*), aktivnog prikaza (Vactive) te prednjeg praga (VFP – engl. *vertical front porch*). Trajanja pojedinih elemenata vertikalne sinkronizacije za pojedinu rezoluciju vidljivi su na tablici 2.4 [2] [3].



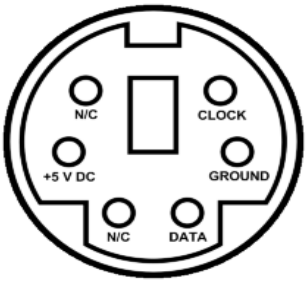
Slika 2.7: Prikaz signala za vertikalno generiranje [2].

Tablica 2.4: Prikaz vertikalnih vremena sinkronizacije [2].

VGA standard		Osvježavanje (Hz)	Signal takta (MHz)	Vertikalna vremena sinkronizacije (linije)				
Ime	**Rezolucija**			**Vpulse**	**VBP**	**Vactive**	**VFP**	**Broj linija**
VGA	640 x 480	60	25.175	2	33	480	10	525
VGA	640 x 480	75	31.5	2	33	480	10	525
SVGA	800 x 600	60	40	4	23	600	1	628
SVGA	800 x 600	75	49.5	3	21	600	1	625
SVGA	800 x 600	85	56.25	3	27	600	1	631
XGA	1024 x 768	60	65	6	29	768	3	806
XGA	1024 x 768	85	94.5	3	36	768	1	808
SXGA	1280 x 1024	60	108	3	38	1024	1	1066

2.2. PS/2

Priključak za PS/2 standard sastoji se od 6 pinova koji su prikazani na slici 2.8. Priključak sadrži dvije žice za komunikaciju (signal takta – CLOCK i podaci – DATA). Podaci se prenose serijski u paketima od 11 bitova. Signal takta tipkovnice određuje kad je podatak spreman za prijenos (na padajući brid).



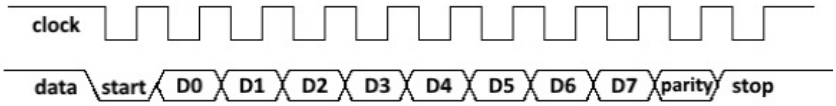
Slika 2.8 : PS/2 „ženski“ priključak s označenim pinovima [4].

Unatoč što prilikom komunikacije s uređajem i tipkovnica i miš šalju podatke u istom obliku, podatci su interpretirani od strane uređaja na drugi način. Dva oblika komunikacije PS/2 standarda su prijenos informacija s računalne periferije na uređaj (engl. *device-to-host communication*) i prijenos informacija s uređaja na računalnu periferiju (engl. *host-to-device communication*). Takvu vrstu komunikacije omogućuje obostrana sabirnica (engl. *bi-directional bus).* Drugi način komunikacije potreban je kad uređaj treba podesiti određene parametre na računalnoj periferiji (tipkovnici i/ili mišu).

2.2.1. Komunikacija tipkovnice s PS/2 protokolom

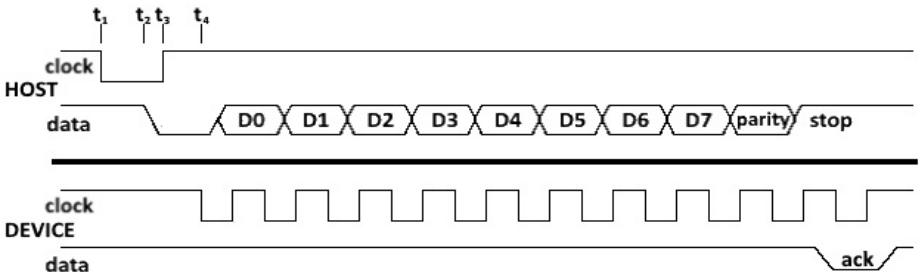
Tipkovnica s uređajem, kao što je prethodno navedeno, komunicira putem slanja paketa. Dijagram vremena prijenosa paketa informacija s tipkovnice na uređaj prikazan je na slici 2.9. Paketi u kojima se prenose informacije sastoje se od 11 bitova u koje spadaju: startni bit, iza kojeg slijedi 8 podatkovnih bitova, zatim bit za provjeru pariteta te stop bit. Na padajući brid signala takta tipkovnice označeno je da je podatkovni paket na žici za prijenos podataka spreman za prijenos. Ukratko, svih 11 bitova bilo kakve su vrste (startni, podatkovni, paritetni

ili stop) šalju se pojedinačno serijski na padajući brid signala takta tipkovnice. Tipkovnica radi na frekvencijama od 10 kHz do 16.7 kHz (period signala takta je između 60 μs i 100 μs), a podatkovni signal je stabilan barem 5 μs prije i poslje padajućeg brida signala takta [3].



Slika 2.9 : Prikaz prijenosa informacija s tipkovnice na uređaj PS/2 protokolom [3].

Uređaj započinje komunikaciju na način da spušta signal takta u logičku '0' u trenutku t_1 na barem 100 us. Zatim, u trenutku t_2 spušta i podatkovni signal u '0' , što predstavlja startni bit. U trenutku t_3 signal takta vraća se u logičku '1' što omogućava tipkovnici čitanje startnog bita. Nakon toga uređaj spušta signal takta u logičku '0' u trenutku t_4 te tada šalje prvi podatkovni bit (D0) podatkovnom sabirnicom. Nakon toga tipkovnica čita poslani bit na rastući brid signala takta te zatim vraća vrijednost signala takta u logičko '0'. Na padajući brid signala takta uređaj šalje sljedeći podatkovni bit te se ovaj proces nastavlja ponavljati dok uređaj nije poslao sve podatkovne bitove, paritetni te stop bit. Ovaj proces prikazan je na slici 2.10 [3] [5].



Slika 2.10: Prikaz procesa komunikacije računala s tipkovnicom [3].

2.2.2. Karakteristike PS/2 protokola

Prilikom pritiska tipke, procesor tipkovnice stvara *make code* te stvara *break code* prilikom puštanja navedene tipke. Ti kodovi se sastoje od 8 bitova (podatkovni bitovi PS/2 paketa podataka). PS/2 kôd u većini slučajeva sastoji se od tri kodne riječi. Tablica kodnih riječi korištenih u završnom radu prikazana je na tablici 2.6. *Make code* je jednak kao i *break code*, osim što se razlikuju po tome što *break code* u sebi sadrži heksadekadsku vrijednost F0. Ovaj proces je prikazan uz pomoć sljedećeg primjera: ukoliko je na tipkovnici pritisnuto slovo 'a', poslat će *make code* 1C te će se prilikom puštanja tipke poslati *break code* F0,1C. Uz pomoć vrijednosti F0, uređaj prepoznaje puštanje tipke.

Tablica 2.6 : Make i break kodovi PS/2 protokola [2].

TIPKA	„MAKE“ KOD	„BREAK“ KOD	TIPKA	„MAKE“ KOD	„BREAK“ KOD	TIPKA	„MAKE“ KOD	„BREAK“ KOD
A	1C	F0,1C	M	3A	F0,3A	Y	35	F0,35
B	32	F0,32	N	31	F0,31	Z	1A	F0,1A
C	21	F0,21	O	44	F0,44	0	45	F0,45
D	23	F0,23	P	4D	F0,4D	1	16	F0,16
E	24	F0,24	Q	15	F0,15	2	1E	F0,1E
F	2B	F0,2B	R	2D	F0,2D	3	26	F0,26
G	34	F0,34	S	1B	F0,1B	4	25	F0,25
H	33	F0,33	T	2C	F0,2C	5	2E	F0,2E
I	43	F0,43	U	3C	F0,3C	6	36	F0,36
J	3B	F0,3B	V	2A	F0,2A	7	3D	F0,3D
K	42	F0,42	W	1D	F0,1D	8	3E	F0,3E
L	4B	F0,4B	X	22	F0,22	9	46	F0,46
BACKSPACE	66	F0,66	ENTER	5A	F0,5A			

Ne sastoje se svi *make code-ovi* od jednog bajta. Iako u završnom radu nisu korištene, neke tipke imaju *make code* sastavljen od 2 bajta, gdje je prvi bajt jednak heksadekadskoj vrijednosti E0, dok se *make code* ponekih tipaka sastoji i od četiri te osam bajtova. Prilikom slanja slova s tipkovnice, ako se šalju velika slova, računalo će to prepoznati po *make code-u shift* tipke. Odnosno, ukoliko je pritisnuto malo slovo 'a', tipkovnica će poslati sljedeći niz bajtova: 12 F0 12. U suprotnom, ako je na tipkovnici pritisnuto veliko slovo 'A' tipkovnica će poslati sljedeći niz bajtova : 12 1C F0 1C F0 12 [3].

Prilikom držanja određene tipke na tipkovnici, nakon odgode od 0.25 s do 1 s tipkovnica će nastaviti slati *make code* pritisnute tipke intenzitetom od 2 do 30 puta po sekundi. Ukoliko se vrijeme odgode ili intenzitet slanja *make code-a* prilikom držanja tipke želi promjeniti, to može biti učinjeno slanjem komande heksadekadske vrijednosti F3 s računala na tipkovnicu praćeno bajtom koji sadrži informacije o novoj vrijednosti odgode te intenziteta slanja [2] [3].

3. FPGA

FPGA predstavlja integrirani elektronički sklop koji se sastoji od konfigurabilnih logičkih blokova, programibilnih interkonekcija te konfigurabilnih izlaznih i ulaznih blokova. Najčešče se podešavaju na način da su opisivane uz pomoć jezika za opis sklopovlja (HDL – engl. *Hardware Description Language*). Sastoji se od niza programibilnih logičkih blokova koji mogu biti podešeni na različite načine od toga da izvode najjednostavnije operacije kao što su NE,I i ILI do složenih i kompleksnih funkcija [6].

Nastale su iz razloga velike kompleksnosti i nepraktičnosti rekonfiguracije gotovog sklopovlja radi ispunjenja korisnikovih željenih zahtjeva. FPGA su praktične što ih korisnik može programirati i reprogramirati te ih naknadno mijenjati na način koji želi. Konfiguracija FPGA se gubi svaki put kad nisu spojene na izvor napajanja [8].

3.1. Povijest FPGA

Prvi FPGA (EP300) je napravljen 1984. godine od strane tvrtke Altera (danas su pod vlasništvom Intela). Xilinx Ross Freeman i Bernard Voderschmitt izumili su prvi komercijalni FPGA 1985. godine imena XC2064. Sastojao se od 64 konfigurabilna logička bloka s troulaznim LUT-ovima (engl. *Look Up Table*). Sve do danas Xilinx i Altera (danas pod vlasništvom Intela) ostali su vodeći proizvođači FPGA-ova. Danas se FPGA kombinira s ugrađenim mikroprocesorima te čine takozvani „sustav na čipu“ (SoC – engl. System on Chip). Primjeri takvih tehnologija su Xilinxov Zynq-7000 te Alterina Arria V FPGA. Također, tvrtke kao Microsoft koriste FPGA-ove za ubrzanje njihovih usluga kao što su Azure cloud te internetski pretraživač Bing [6].

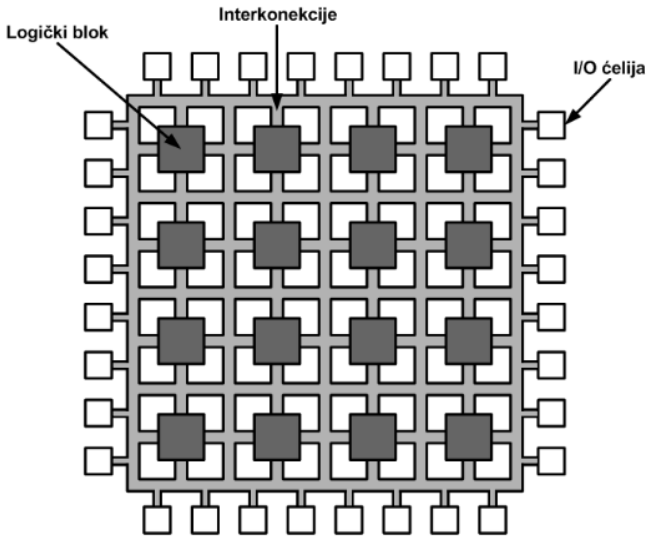
3.2. Građa FPGA

FPGA se sastoji od integriranih sklopova koji mogu biti programirani na način da implementiraju digitalni dizajn. Glavne tri komponente dizajna FPGA su: programirljivi logički blokovi koji implementiraju logičke funkcije, poveznice između tih logičkih blokova (konfigurabilne interkonekcije) koje implementiraju funkcije te ulazni i izlazni blokovi koji se koriste za povezivanje sa vanjskim uređajima (slika 3.1). Programirljivi logički blokovi sastoje se od više logičkih blokova (logičkih ćelija) .

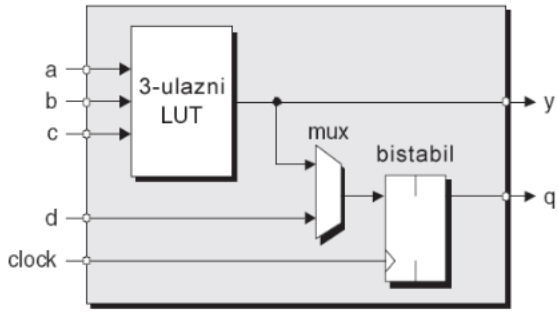
Logički blokovi (slika 3.2) se sastoje od n-ulaznog LUT-a (engl. *Look Up Table*), D bistabila i multipleksera. LUT je $2^n x1$ memorija koja omogućuje implementaciju bilo kojeg n-ulaznog kombinacijskog sklopa, dok bistabil omogućuje implementiranje sekvencijalnog

sklopa. Multiplekser prosljeđuje izlaz iz LUT-a ili ulaz logičkog bloka. Zatim, programibilne interkonekcije služe za uspostavljanje veze između logičkih blokova i ulazno-izlaznih blokova. Sastoje se od tranzistora te pojačala s tri stanja. Ulazno-izlazni blokovi služe za povezivanje logičkih blokova s vanjskim komponentama. Izlazi mogu biti programirani na način da budu aktivni u visokom ili niskom naponskom stanju, a na izlazima se obično nalaze *pull up* otpornici, ali ponekad i *pull down* otpornici [8].

Većina električnih krugova unutar FPGA su sinkroni te zahtijevaju korištenje signala takta. Kompleksni dizajnovi FPGA mogu koristiti više signala takta s različitim frekvencijama i odnosima između faza. Arhitektura FPGA u današnje vrijeme razvija se kroz dodavanje više specijaliziranih blokova kao što su aritmetičko-logičke jedinice, multipleksori te mikroprocesori [8].



Slika 3.1.: Građa FPGA [7].



Slika 3.2.: Građa logičkog bloka s 3-ulaznim LUT-om [7].

3.3. Upotreba FPGA

FPGA se može koristiti za implementiranje mekih mikroprocesora (jezgra mikroprocesora koja može u potpunosti biti implementirana logičkom sintezom). Primjer takvog mikroprocesora je Xilinx-ov MicroBlaze. Također se koriste prilikom ubrzavanja sklopovlja, na način da se FPGA koristi da ubrza određene dijelove algoritma i odradi dio računanja računalnog procesora. Koriste se i prilikom ubrzavanja umjetnih neuronskih mreža te pri strojnome učenju [6].

Također, FPGA ima i sljedeće upotrebe kao što su: obrada zvuka i videa te prepoznavanje govora, auto industrija, bioinformatika, medicinska oprema, itd. Prosječan FPGA je u početku svog razvoja imao oko par tisuća logičkih elemenata, dok ih neki FPGA-ovi danas imaju i preko 5 milijardi [8].

3.3.1. Dizajn

Dizajn digitalnih sklopova za FPGA (slika 3.3) omogućen je jezicima za opisivanje sklopovlja (HDL). Jezici za opisivanje sklopovlja koriste se iz razloga što omogućuju jednostavno opisivanje većih i kompleksnih elektroničkih sklopova. Za jezike za opisivanje sklopovlja postoje razna razvojna sučelja u kojima korisnik može provjeriti točnost sintakse te simulirati ponašanje digitalnog dizajna. Najpoznatiji i najčešće korišteni jezici za opisivanje sklopovlja su VHDL, Verilog te System Verilog [6] [8].

3.3.2. Simulacija dizajna

Napravljeni dizajn potrebno je simulirati i provjeriti funkcionira li na željeni način. Poznavajući dizajn potrebno je napisati testni uzorak te znati kakve rezultate upotrebom tog testnog uzorka je željeno postići.

3.3.3. Sinteza dizajna

Prilikom sinteze dizajna kod jezika za opis sklopovlja prevodi se u krug s logičkim elementima. Tijekom ovog procesa provjerava se točnost sintakse koda i vrši se analiza hijerarhije dizajna arhitekture. Gore opisani proces zapravo prevodi kod u listu elektroničkih komponenata u krugu te listu čvorova na koju su povezani te se spremaju u format datoteke NGC [8].

3.3.4. Implementacija dizajna

Implementacija dizajna sastoji se od tri osnova dijela : prevođenja, mapiranja te smještanja i usmjeravanja. Prevođenje se sastoji od kombiniranja svih ulaznih te izlaznih listi elektroničkih komponenata u datoteku logičkog dizajna (NGD format datoteke). Ovdje su svi ulazi i izlazi FPGA dodjeljeni fizičkim elementima kao što su prekidači i tipkala odnosno LED-ice i pinovi. To je spremljeno u UCF tipu datoteke.

Mapiranje odvaja krug u podblokove iz razloga da mogu stati u logičke blokove FPGA. Na taj način ovaj proces ubacuje logiku definiranu u NGD datoteci u kombinacijske logičke blokove i ulazno-izlazne blokove te stvara NCD datoteku. Ona predstavlja dizajn kruga mapiran u komponente same FPGA. Proces smještanja i usmjeravanja se sastoji od smještanja pod-

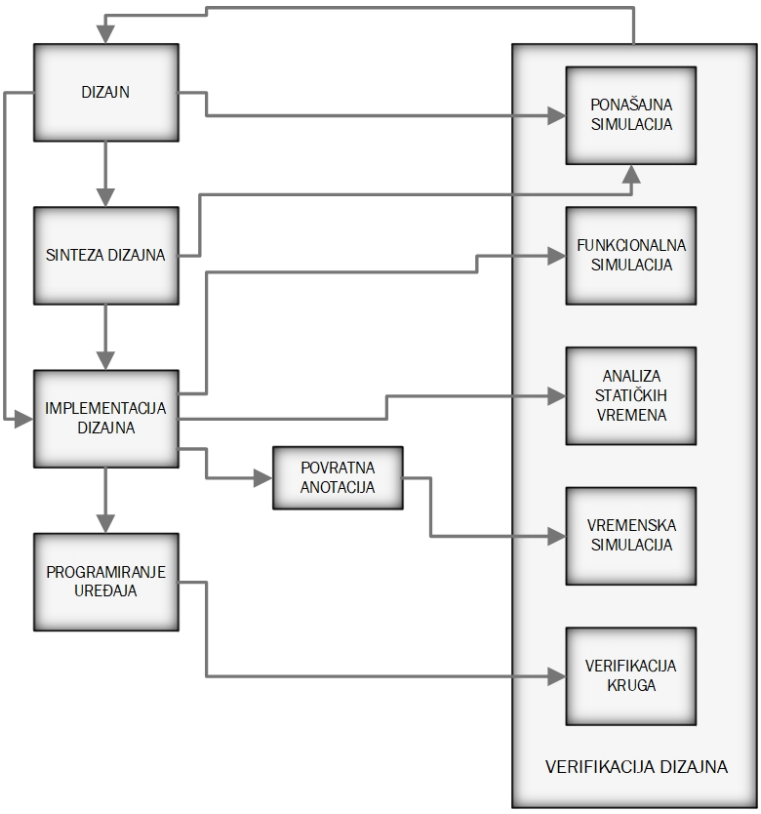
blokova nastalih tijekom procesa mapiranja. Zatim se prema uputama danima u dizajnu ti logički blokovi spajaju [8].

3.3.5. Programiranje uređaja

Nakon prijašnjih procesa, obrađeni dizajn mora se učitati u samu FPGA u formatu koji je podržan od strane određene FPGA. Zatim, spomenuta datoteka se učitava u program za generiranje BIT datoteke te se BIT datoteka stvara [8].

3.3.6. Verifikacija dizajna

Verifikacija dizajna se sastoji od ponašajne simulacije, funkcionalne simulacije te simulacije proračuna vremena. Prvo se izvodi ponašajna simulacija u kojoj se promatraju varijable, procedure i funkcije te se podešavaju točke prekida. Zatim slijedi funkcionalna simulacija koja nam govori o logičkom ponašanju dizajna. Na kraju dolazimo do simulacije proračuna vremena koji omogućava sažetak čitavog dizajna u smislu vremena izvođenja [8].



Slika 3.3: Dizajn FPGA [8].

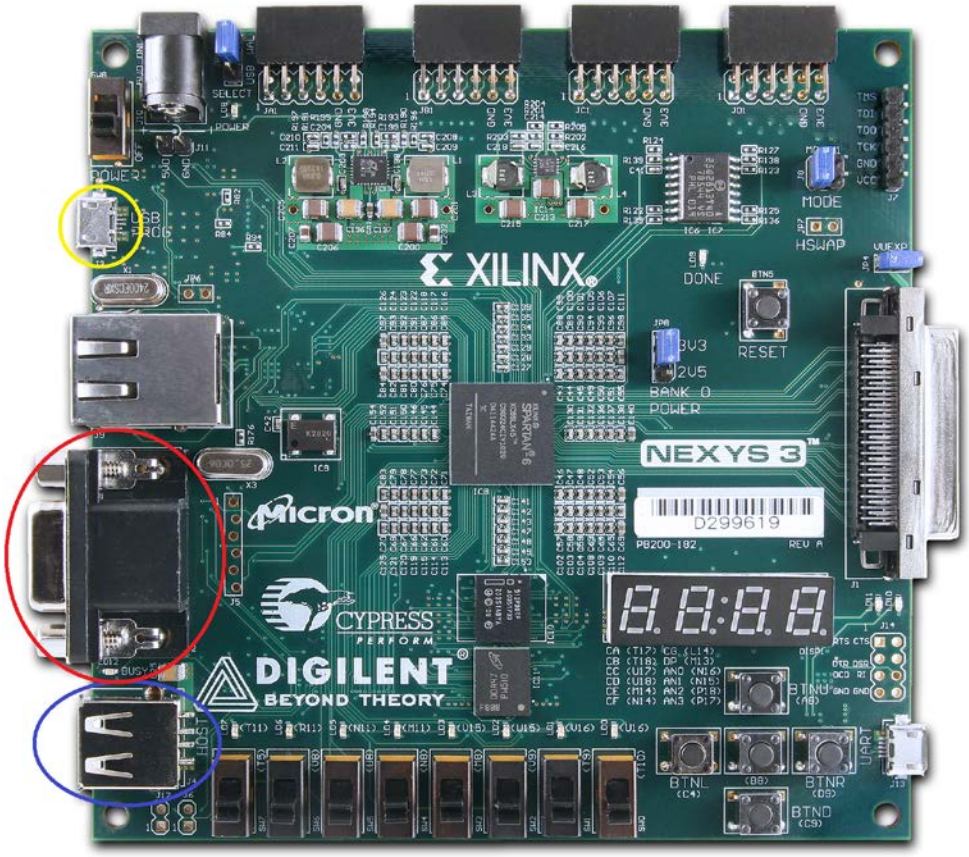
4. IMPLEMENTACIJA

VGA grafički upravljač implementiran je pomoću razvojnog sustava Nexys 3 Spartan-6 FPGA Trainer Board tvrtke Digilent, USB tipkovnice te monitora s VGA priključkom. Jezik za opisivanje sklopovlja koji je korišten je VHDL, a konačni dizajn sastoji se od 3 modula: PS/2, RAM i VGA. Korisnik pomoću tipkovnice unosi znakove, čiji se PS/2 *make code*-ovi prosljeđuju te spremaju u RAM. Zatim se ti podaci prosljeđuju VGA modulu te se ovisno o *make code*-u pojedinog znaka, ti znakovi ispisuju na ekran. Korisnik pomoću *backspace*-a može obrisati zadnje uneseni znak, a ukupni unos može obrisati pritiskom na tipku *enter*. Također, pomoću *generic* konstanti, moguće je namjestiti maksimalan broj znakova koji se može ispisati na ekran te koordinate ekrana na kojima se znakovi počinju ispisivati (gornji lijevi kut ekrana ima koordinate (0,0), a donji desni kut (639,479)). Ukoliko je uneseno više znakova od maksimalnog broja znakova, prepisat će se prethodno uneseni znakovi, počevši od prvog unesenog [5].

Neka od svojstva razvojnog sustava Nexys 3 Spartan-6 FPGA Trainer Board (slika 4.1):

- 2278 konfigurabilnih logičkih blokova koji sadrže 4 6-ulazna LUT-a te 8 bistabila
- 576 Kbita brzog *block RAM-a*
- 32 konfigurabilna logička bloka za obradu digitalnih signala
- *Ethernet* konektor
- 8-bitni VGA port
- USB-UART te USB-HID port
- 16 megabajtni *cellular* RAM
- 72 ulazno-izlaznih konektora za dodatne module
- 8 LED-ica, 5 tipkala, 8 prekidača te 4-znamenkasti sedam segmentni prikaz
- USB port za programiranje te USB2 kabel

Također, Nexys3 je kompatibilan sa svim Xilinx-ovim CAD (engl. *Computer-Aided Design*) alatima, uključujući ChipScope, EDK te ISE. Koristi Digilentov Adept USB2 sustav koji omogućava FPGA te ROM programiranje, automatizirane testove, virtualne ulazno-izlazne portove te pojednostavljeni prijenos podataka. Razvojno sučelje u kojem je završni rad rađen naziva se ISE Design Suite: System Edition [9].



Slika 4.1: Nexys 3 Spartan 6 FPGA Board [9].

Ulaz označen plavom bojom predstavlja USB (engl. *Universal Serial Bus*) ulaz na koji se spaja tipkovnica. Razlog zašto je tipkovnica s kojom je korišten PS/2 protokol spojena na USB utor FPGA je taj što koristi emulator (Microchip PIC24FJ192 je integrirani sklop korišten za emulaciju) tako da iako je tipkovnica povezana s FPGA pomoću USB porta, može se pretpostaviti da je tipkovnica spojena u FPGA pomoću PS/2 porta. VGA port na FPGA je označen crvenom bojom te je na njega spojen monitor na kojem trebaju biti prikazani znakovi koje korisnik unosi pomoću tipkovnice. Žuto označen ulaz predstavlja USB port s kojim je FPGA spojen s računalom te programiran [9].

4.1. PS/2 modul

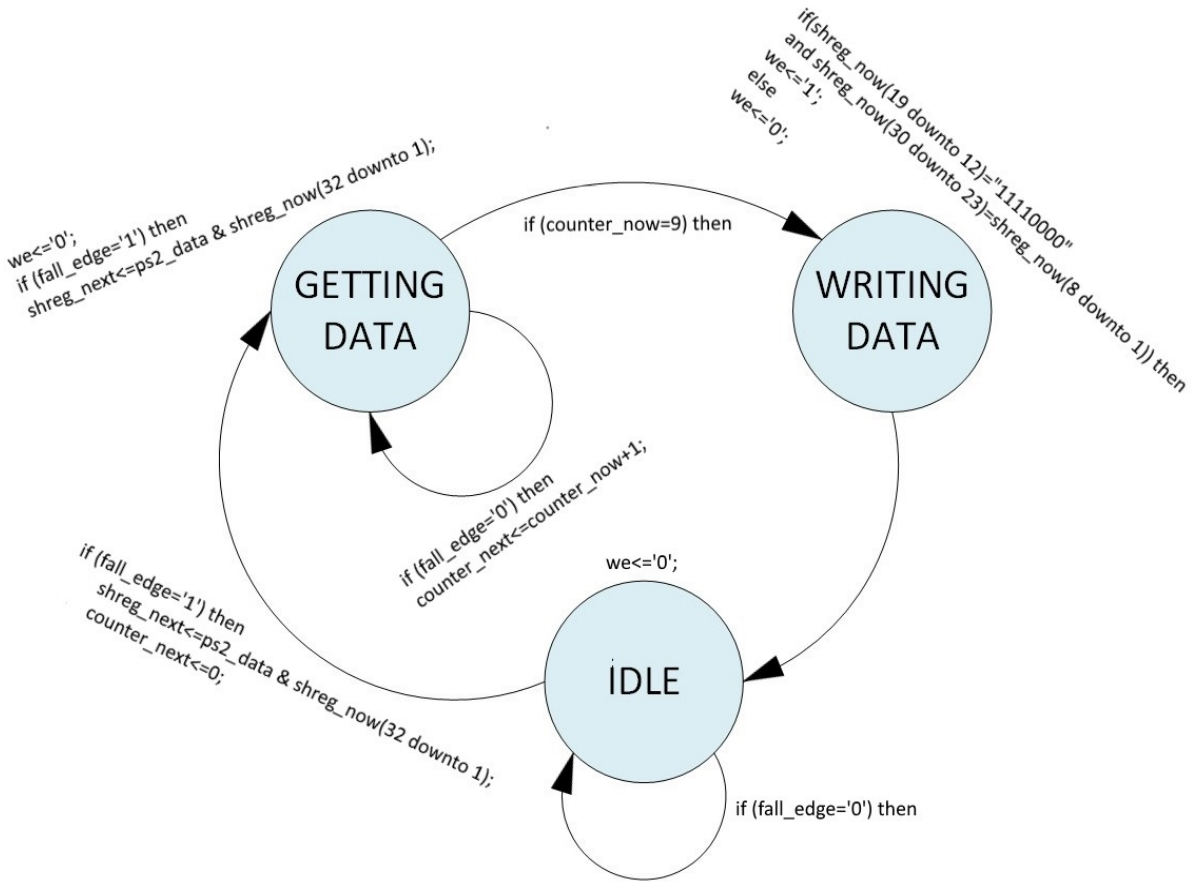
Ulazi u PS/2 modul su signal takta FPGA (clk), signal takta tipkovnice (ps2_clk) te podatkovna žica tipkovnice (ps2_data). Izlazi iz ovog modula su 8-bitni signal koji predstavlja *make code* znaka unesenog pomoću tipkovnice (data_out) i signal koji daje informaciju kad je *make code* znaka unesenog pomoću tipkovnice spreman za slanje u RAM (we). PS/2 u sebi

sadrži filter signala takta PS/2 koji služi za eliminaciju potencijalnih smetnji i šumova. VHDL kod filtera prikazan je na slici 4.2.

```
30:      filter_next <= ps2_clk & filter_now(downto 1);
31:      fps2c_next <='1' when filter_now="11111111" else
32:                      '0 'when filter_now="00000000" else
33:                       fps2c_now;
34:      fall_edge <= fps2c_now and (not fps2c_next);
```

Slika 4.2: Prikaz VHDL koda PS/2 filtera [5].

PS/2 je izveden pomoću automata s konačnim brojem stanja (FSM – engl. *finite state machine*). Sastoji se od 3 stanja: idle, getting_data te writting_data (prikazanih na slici 4.3). Stanje idle predstavlja stanje mirovanja, prilikom kojeg se signal takta PS/2 nalazi u logičkoj '1' te je mogućnost slanja podatka u RAM onemogućena (we<='0') . Ukoliko je detektiran padajući brid, u posmačni registar na prvo mjesto ulazi startni bit podatka učitan s podatkovne žice PS/2. Brojač se postavlja u 0 te se prelazi u sljedeće stanje (getting_data). Prilikom stanja getting_data, na svaku detekciju padajućeg brida, u posmačni registar na prvo mjesto ulazi jedan bit podatka učitan s podatkovne žice tipkovnice. Ukoliko je brojač različit od 9, vrijednost mu se povećava za 1. U suprotnom, ukoliko je brojač dosegao 9, prelazi se u sljedeće i ujedno

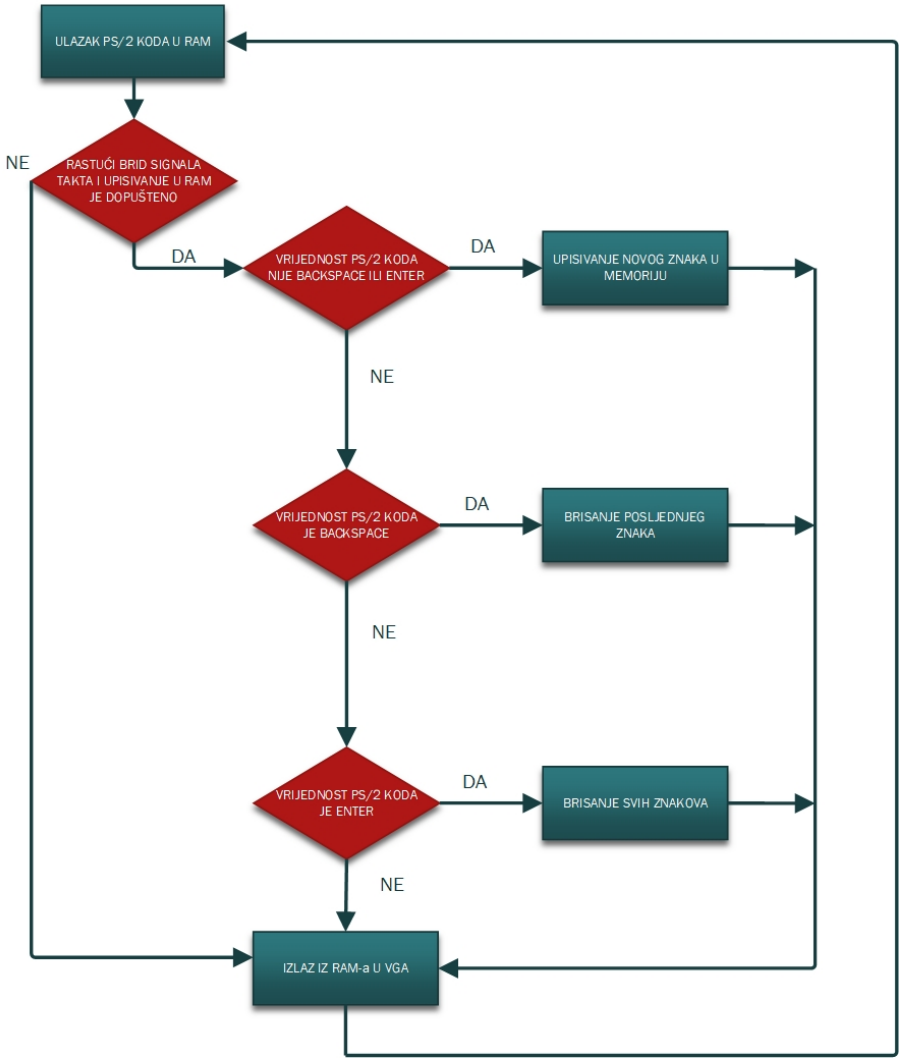


Slika 4.3: Prikaz stanja i prijelaza stanja implementacije PS/2 pomoću FSM.

posljednje stanje writting_data. Stanje writting_data, za razliku od prethodna 2 stanja, uvijek traje samo jedan signal takta FPGA. U svakom slučaju prelazi u sljedeće stanje (idle) te provjerava je li prvi bajt *break code-a* jednak heksadekadskoj vrijednosti F0 te jesu li *make code* te drugi bajt *break codea* jednaki. Ukoliko jedan od ovih uvjeta nije ispunjen, znači da je došlo do pogreške te slanje podatka u RAM neće biti omogućeno. Ukoliko su svi uvjeti ispunjeni, signal we je omogućen, odnosno nalazi se u logičkoj '1' te se *make code* pritisnute tipke tipkovnice šalje u RAM.

4.2. RAM Modul

RAM modul sadrži sljedeće ulaze: signal takta FPGA (clk), *make code* znaka unesenog preko tipkovnice (data_in) i signal koji ukoliko je u logičkoj '1', omogućuje zapis *make code*-a u RAM (we). Ovaj modul ima jedan izlaz (data_out), a on predstavlja niz od n *make code*-ova



Slika 4.4: Prikaz RAM modula.

koji se šalju u VGA modul da se, ovisno o vrijednosti pojedinog *make code*-a, uneseni znakovi ispisuju na ekran. Korisnik sam podešava maksimalni broj znakova za unos, odnosno n.

Ukoliko je korisnik pritisnuo tipku na tipkovnici, a ona nije jednaka *backspace-u ili enter-u,* tada se novi podatak upisuje u radnu memoriju. Ukoliko je uneseno više znakova od maksimalnog broja znakova, prepisat će se prijašnje uneseni znakovi, počevši od prvog unesenog. Ako je korisnik pritisnuo backspace, zadnje uneseni podatak u RAM, odnosno zadnje napisan znak na ekranu će se obrisati. Ukoliko korisnik pritisne *enter* na tipkovnici, svi podatci iz RAM-a, odnosno, svi znakovi s ekrana će se obrisati (slika 4.4).

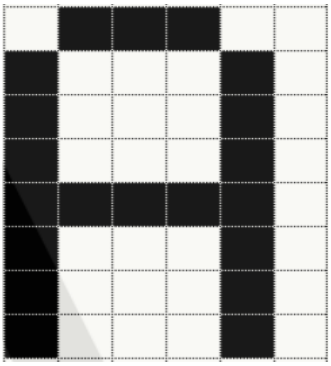
4.3. VGA modul

Ulazi u VGA modul su jednaki kao i ulazi u PS/2 modul, a oni su signal takta FPGA (clk), signal takta tipkovnice (ps2_clk) te podatkovna žica tipkovnice (ps2_data). Izlazi iz VGA modula su vrijednosti boja, a one su predstavljene 3-bitnim signalima R i G te 2-bitnim signalom B, iz razloga što FPGA razvojni sustav korišten za implementaciju sadrži 8-bitni VGA port (3+3+2). Uz to, sadrži niz *generic* konstanti pomoću kojih korisnik određuje rezoluciju, konstantu koja predstavlja maksimalan broj znakova koji može biti prikazan na ekranu (n), te dvije konstante koje određuju početnu koordinatu ispisivanja znakova na ekran (pix_x, pix_y).

Nadalje, definiran je niz konstanti, korisnički definiranog tipa podatka digit6x8, korištenjem kojih se vrši prikaz znakova na ekran. Primjer jedne takve konstante te izgleda znaka prikazanog na ekran nalazi se na slikama 4.5 i 4.6.

```
constant a: digit6x8 := (
('0','1','1','1','0','0'),
('1','0','0','0','1','0'),
('1','0','0','0','1','0'),
('1','0','0','0','1','0'),
('1','1','1','1','1','0'),
('1','0','0','0','1','0'),
('1','0','0','0','1','0'),
('1','0','0','0','1','0'));
```

Slika 4.5: Prikaz definiranja konstante tipa digit6x8.



Slika 4.6: Prikaz znaka „A“ [10].

Za pretvaranje *make code*-ova unesenih pomoću tipkovnice u prethodno navedene konstante za prikaz pojedinih znakova na ekranu služi funkcija bin_to_digit6x8 [2]. Nakon toga, izvedeni su djelitelj frekvencije zbog generiranja signala takta na kojem radi određena rezolucija (pixel_clk), generiranje horizontalnih i vertikalnih signala te procesi za određivanje koordinata ekrana (x,y). Na samome kraju, nalazi se proces unutar kojeg se zadani znakovi iscrtavaju na ekran koji je naveden na slici 4.7. U ovom slučaju na ekran će se ispisivati znakovi bijele boje zbog kombiniranja svih boja (R,G i B).

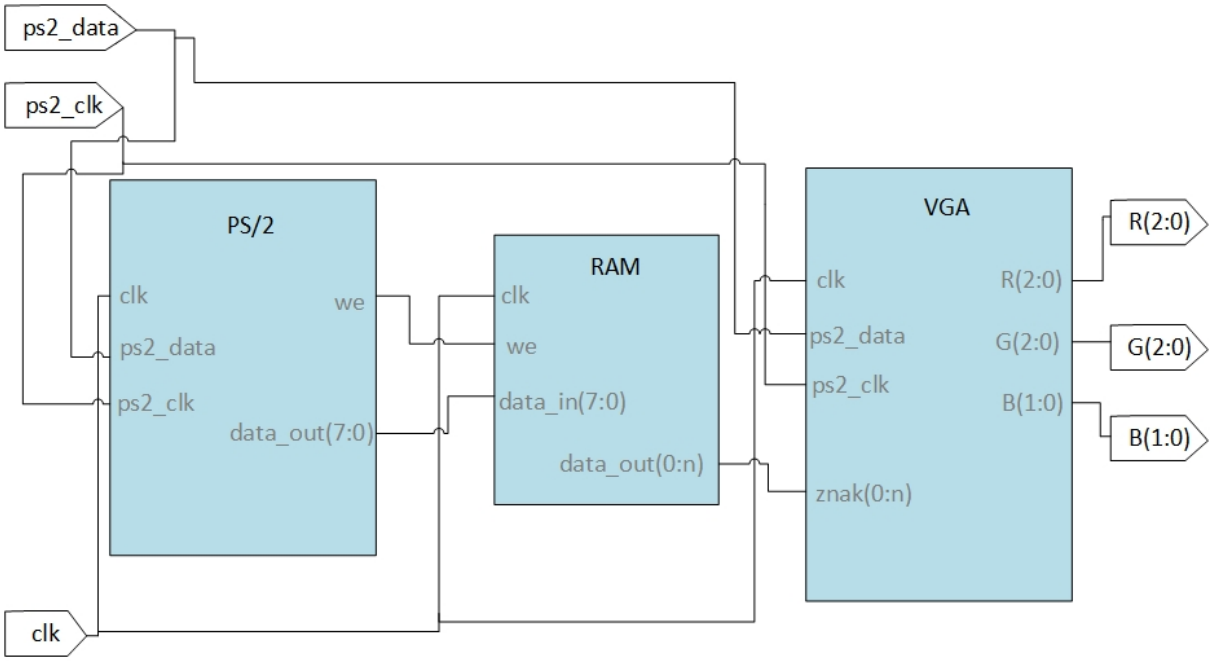
```
process (dena,znak,nacrtani_znak,x,y)
 begin
 for i in 0 to n loop
    nacrtani_znak(i)<=bin_to_digit6x8(znak(i));
 end loop;

 if(dena='1') then
    for i in 0 to n loop
 if (((x>=pix_x+i*6) and x<(pix_x+i*6+6)) and (y>=pix_y and y<pix_y+8)) then
    R <= (others=>nacrtani_znak(i)(y-pix_y, x-(pix_x+i*6)));
    G <= (others=>nacrtani_znak(i)(y-pix_y, x-(pix_x+i*6)));
    B <= (others=>nacrtani_znak(i)(y-pix_y, x-(pix_x+i*6)));
       end if;
    end loop;
 else
    R <= (others => '0');
    G <= (others => '0');
    B <= (others => '0');
 end if;
end process;
```

Slika 4.7: Proces generiranja znakova na ekran.

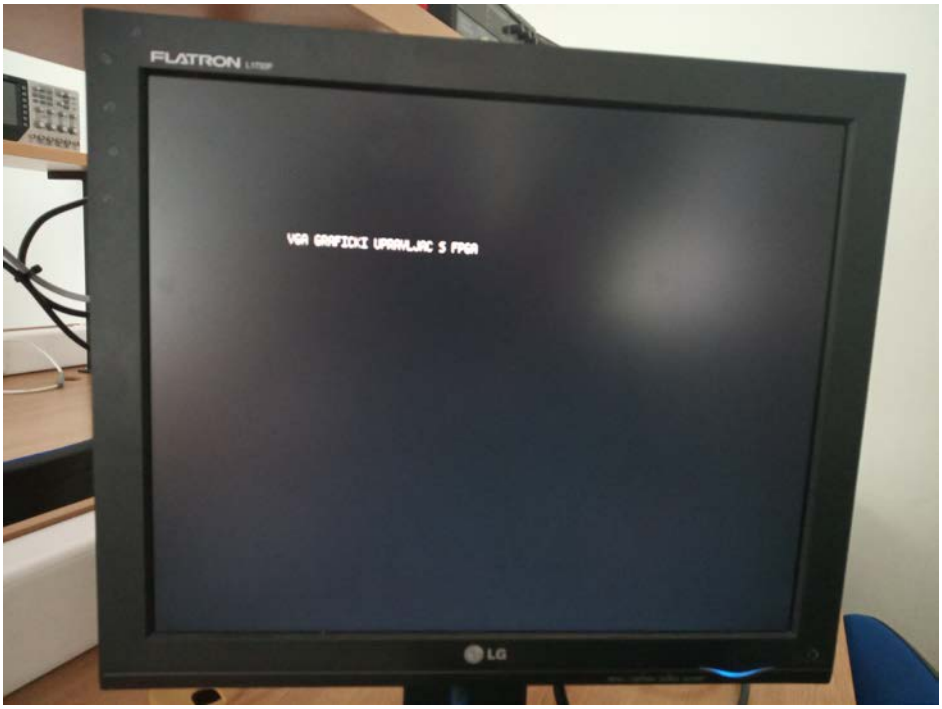
4.4. Testiranje

Nakon rješavanja inicijalnih problema, implementacija je uspješno izvedena. Prethodno objašnjeni parametri VGA modula n, pix_x te pix_y su postavljeni na vrijednosti 40, 120 i 120, što znači da je maksimalan broj znakova na ekran koji se može ispisati 40 te se prvi znak počinje ispisivati na koordinatama ekrana (120,120). Korisnik po želji može postaviti te parametre. Rezolucija ekrana na kojoj je izvedena implementacija je 640x480 piksela, ali se također može mjenjati. Ukoliko korisnik odluči promjeniti rezoluciju, mora promijeniti vrijednost konstanti koje predstavljaju broj piksela pojedine rezolucije (puls sinkronizacije, stražnji prag itd.). Također, morat će prilagoditi djelitelj frekvencije da daje traženu frekvenciju na kojoj radi određena rezolucija. Boja u kojoj se ispisuju znakovi po ekranu je bijela, ali se i ona može promijeniti. Ako korisnik želi ispisivati znakove u drugim bojama, to je moguće dobiti drugačijim načinom kombiniranja osnovnih boja VGA (crvena, plava i zelena). Na primjer, kombinacija crvene i zelena daju žutu. Unosom znakova preko tipkovnice moguće je ispisati bilo koje slovo (A-Z) i bilo koju znamenku (0-9) na ekran te implementacija funkcionira u

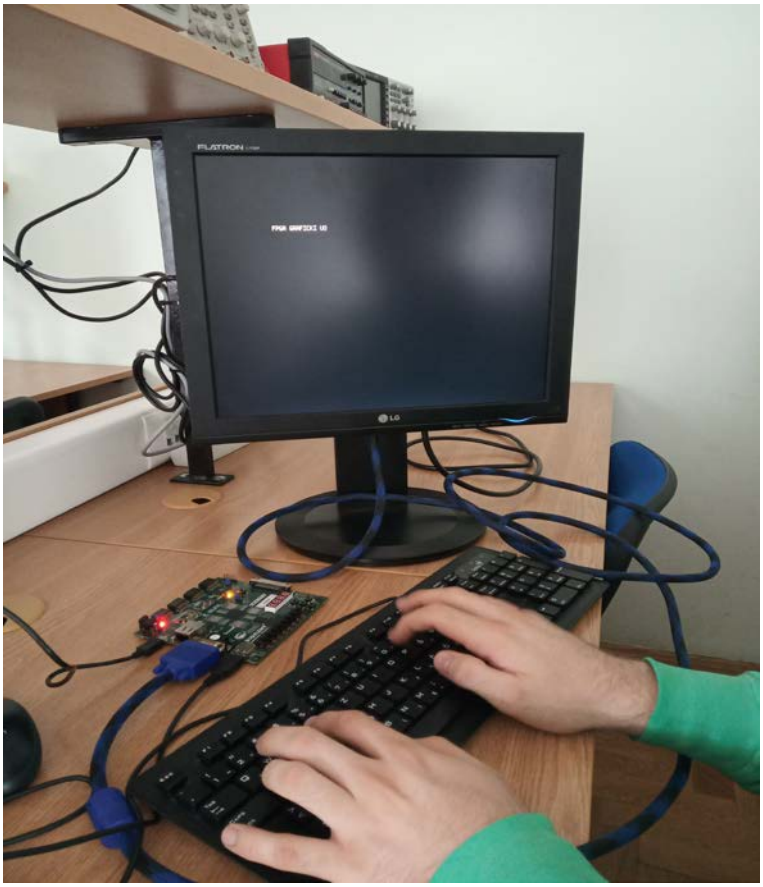
potpunosti. Shema implementacije nalazi se na slici 4.8. Na slikama 4.9 i 4.10 prikazano je testiranje implementacije.



Slika 4.8: Prikaz sheme implementacije.



Slika 4.9: Prikaz ispisa znakova na ekranu pomoću VGA grafičkog upravljača.



Slika 4.10: Prikaz implementacije VGA grafičkog upravljača.

5. ZAKLJUČAK

Prilikom izrade završnog rada, za implementaciju VGA grafičkog upravljača korištenjem FPGA, bilo je potrebno napisati kod pomoću jezika za opisivanje sklopovlja, koji je u ovom slučaju bio VHDL. Uz implementaciju VGA standarda, bilo je potrebno implementirati PS/2 protokol za komunikaciju tipkovnice s FPGA te radnu memoriju (RAM), u koju se spremaju korisnikovi unosi s tipkovnice da bi se prilikom prikaza znakova na ekran mogao pokazati određeni broj znakova koliki je korisnik odredio da bude prikazan.

U VGA modulu vrijednost konstante n, kojom je određen broj znakova koji mogu biti prikazani na ekranu je 40, ali baš zato što je *generic* vrijednost, korisnik ga može po želji mijenjati, kao i ostale *generic* konstante koje su potrebne za podešavanje rezolucije ekrana (pulsovi sinkronizacije, prednji i stražnji pragovi...) te po želji podesiti željenu rezoluciju ekrana. Da bi promjenio rezoluciju također mora prilagoditi i djelitelj frekvencije radi podešavanja frekvencije na kojoj radi VGA standard određene rezolucije. Uz to, promjenom konstanti pix_x te pix_y može se podesiti početno mjesto ispisivanja znakova na ekranu. Na sličan način na koji je kreiran ovaj VGA grafički upravljač, mogu se kreirati i vlastite jednostavne video igre. Svi moduli bi morali biti do neke mjere modificirani, ali ideja je da korisnik uz pomoć tipkovnice (može biti i miš te *joystick*, samo se treba implementirati PS/2 za njih) upravlja vlastitim likom u igri ili ostalim elementima igre. U ovom slučaju privremeni napredak u igri bi se spremao u RAM, ali moguće je kreirati i dodatni modul ROM ukoliko korisnik želi kompleksniju igru. Također, ako korisnik želi veću memoriju može koristiti dodatni vanjski memorijski modul za FPGA.

Na ovaj način, moguće je do neke mjere kreirati vlastitu igraču konzolu korištenjem FPGA te jezika za opisivanje sklopovlja. Korištenjem dodatnih modula kao što je kamera, moguće je uz pomoć VGA protokola korištenjem određenih FPGA-ova koji su kompatibilni s tim dodatnim modulom napraviti izravan video prijenos u stvarnom vremenu s kamere na ekran.

LITERATURA

[1] VGA i PS/2, https://en.wikipedia.org/wiki/Video_Graphics_Array, pristupljeno 23.5.2019.

[2] Volnei A. Pedroni, Circuit Design and Simulation with VHDL, The MIT Press, 2010.

[3] Richard E. Haskell; Darrin M. Hanna, Digital Design Using Digilent FPGA Boards VHDL/Active-HDL Edition, 2009.

[4] PS/2, https://wiki.osdev.org/PS/2, pristupljeno 25.5.2019.

[5] Pong P. Chu, FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version, Willey-Interscience, 2008.

[6] Povijest FPGA, Upotreba FPGA, https://en.wikipedia.org/wiki/Field-programmable_gate_array, pristupljeno 27.5.2019.

[7] Tomislav Matić, Auditorne vježbe 4, Digitalna elektronika

[8] Programiranje FPGA; Građa FPGA, https://www.edgefx.in/fpga-architecture-applications/, pristupljeno 22.5.2019.

[9] Nexys 3 Spartan 6 FPGA Trainer Board, https://reference.digilentinc.com/reference/programmable-logic/nexys-3/reference-manual, pristupljeno 26.5.2019.

[10] Web aplikacija za matrično crtanje znakova, Slika 4.6, https://www.pentacom.jp/pentacom/bitfontmaker2/

SAŽETAK

Implementacija VGA grafičkog upravljača korištenjem FPGA razvojnog sustava bila je cilj ovog završnog rada. Prilikom programiranja FPGA razvojnog sustava bilo je potrebno koristiti razvojno okruženje, a u ovom slučaju to razvojno okruženje bilo Xilinx-ov ISE Design Suite. FPGA koji je bio korišten je Nexys 3 Spartan-6 FPGA Trainer Board, koji posjeduje 8-bitni VGA priključak. Za ostvarenje tog cilja bilo je potrebno implementirati PS/2 protokol za komunikaciju tipkovnice s uređajem te radnu memoriju (RAM). Korisnik uz pomoć tipkovnice koja je spojena s FPGA unosi slova i/ili brojeve koji imaju vlastiti jedinstveni *make-code,* te se kôd, po kojem računalo prepoznaje o kojem je znaku riječ, sprema u radnu memoriju. Nakon toga se iz radne memorije uneseni podatci šalju VGA modulu, kako bi se željeni znakovi uspješno prikazali na ekranu jedan pored drugog.

Korisnik uz pomoć *backspace-a* može brisati jedan po jedan znak s ekrana, a ukoliko je pritisnut *enter* svi znakovi s ekrana bit će obrisani. Znakovi se na crnu pozadinu ekrana ispisuju bijelom bojom, dimenzija piksela 6x8 (široki su 6 piksela a visoki 8). Nadogradnjom i/ili izmjenama ovog projekta moguće je kreirati jednostavne video igre, slične nekadašnjim igrama za konzole. Korištenjem kamere kao dodatni modul te nekog drugog FPGA koji podržava taj modul, moguće je napraviti procesiranje videa u stvarnome vremenu.

Ključne riječi : FPGA, VHDL, VGA, PS/2, RAM, Xilinx

ABSTRACT

Implementation of VGA graphics controller using FPGA development board was the aim of this thesis. During the programming of FPGA development board it was necessary to use software tool and in this case it was Xilinx ISE Design Suite. The FPGA used here was Nexys 3 Spartan-6 FPGA Trainer Board which has 8-bit VGA port. To fulfill this task it was required to implement PS/2 communication protocol (device-to-host communication) and random-access memory module. By using the keyboard connected to FPGA the user inputs keys that have their unique make-code, by which the computer recognizes which key was pressed and stores this unique make-code in random access memory. Afterwards the information is forwarded from RAM to VGA module, so that each character is successfully shown on the screen in consecutive order.

By using backspace the user can delete the characters from the screen one by one and if enter button is pressed all the characters from the screen will be deleted. The characters on the black screen are written in white color, their dimensions being 6x8 (6 wide and 8 tall). By upgrading and/or changing this project it is possible to create simple video games, similar to old console games. By using the camera as an additional module and other types of FPGA's which support that module, it is also possible to process videos in real time.

Key words: FPGA, VHDL, VGA, PS/2, RAM, Xilinx

ŽIVOTOPIS

Matej Horvat rođen je 20. travnja 1996. godine u Našicama. Završio je Osnovnu školu Dore Pejačević, te Srednju školu Isidora Kršnjavoga u Našicama (prirodoslovno-matematičku gimnaziju). Tijekom srednjoškolskog školovanja sudjelovao je na brojnim natjecanjima iz područja matematike, informatike te fizike uz najbolje ostvareni rezultat 2. mjesto iz matematike. Trenutno je redovan student 3.godine preddiplomskog studija na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Tečan je govornik engleskog jezika. Nagrađen je od strane FERIT-a 2018. godine za uspješnost u studiranju. Radio je kao demonstrator iz predmeta Digitalna elektronika te radi kao instuktor iz stručnih predmeta iz područja računarstva i elektrotehnike u centru instrukcija Edukos.

PRILOZI

P1 VHDL kod nalazi se na CD-u

P2 Video snimka implementacije nalazi se na CD-u