

Univerza v Ljubljani  
Fakulteta *za elektrotehniko*



## POROČILO VGA IGRA

**Avtor:**

Jure Jelovčan (64170238)

**Mentor:**

prof. dr. Andrej Trost

Ljubljana, Januar 2020

## VSEBINA

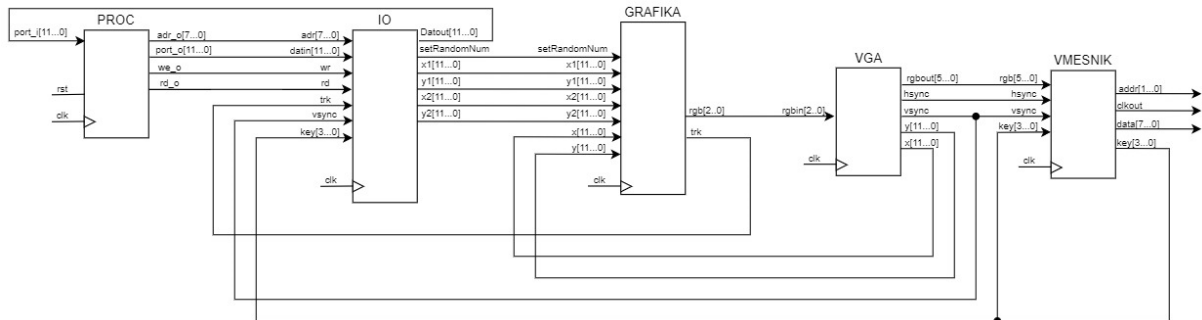
KAZALO SLIK .....	1
1 UVOD.....	1
2 ELEMENTI IN POVEZAVE V DIGITALNEM VEZJU.....	2
2.1 PROCESOR.....	2
2.3.1 VHODI/IZHODI PROCESORJA.....	3
2.2 VHODNO - IZHODNI VMESNIK.....	3
2.2.1 VHODI IN IZHODI V IO VMESNIK.....	4
2.2.2 NADGRADNJA VHODNO – IZHODNEGA VMESNIKA.....	4
2.3 GRAFIČNI PROCESOR .....	5
2.3.1 VHODI/IZHODI IZ GRAFIČNEGA PROCESORJA.....	5
2.3.2 NADGRADNJA GRAFIČNEGA PROCESORJA.....	5
2.4 BLOKA VGA IN VMESNIK.....	6
2.4.1 POMEMBNEJŠI SIGNALI ZA VGA BLOK.....	6
2.4.2 POMEJBNEJŠI SIGNAL ZA VMESNIK.....	6
2.5 BLOKOVNA SHEMA.....	7
3 DELOVANJE VGA IGRE .....	8
4 POVZETEK SINTEZE.....	9
4.1 RTL SHEMA.....	9

## KAZALO SLIK

Slika 1: Celotno vezje .....	1
Slika 2: Glavni elementi vezja .....	2
Slika 3: Blokovna shema.....	7
Slika 4:Igra Izmikalec.....	8
Slika 5: Primer trka.....	8
Slika 6: Rtl shema.....	9
Slika 7: Rtl shema povečana 1/2.....	9
Slika 8: Rtl shema povečana 2/2.....	10

# 1 UVOD

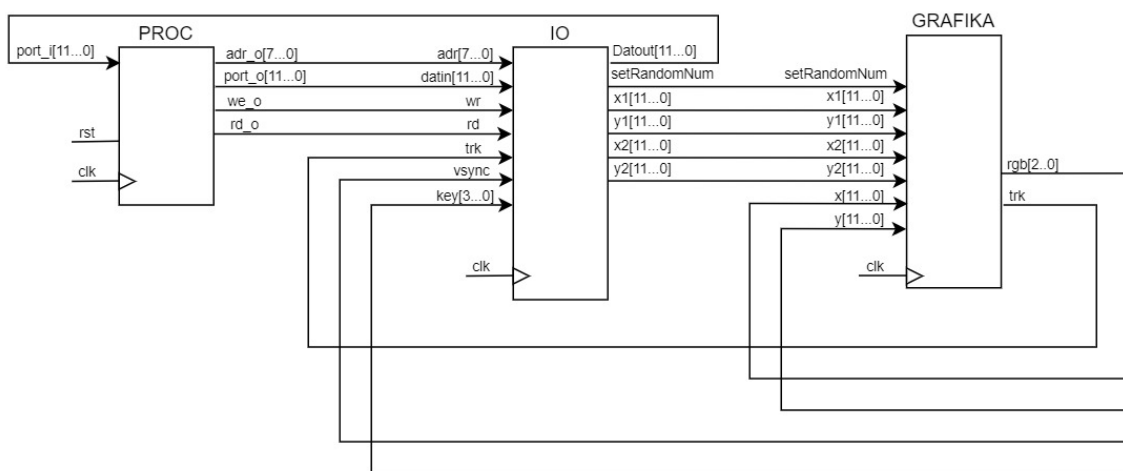
Naloga je bila samostojna nadgradnja digitalnega vezja iz laboratorijskih vaj. Odločil sem se, da naredim igro *Izmikalec*. Kjer je naloga igralca, da se čim dlje izmika oviram, katere naključno prihajajo iz strani.



Slika 1: Celotno vezje

## 2 ELEMENTI IN POVEZAVE V DIGITALNEM VEZJU

Vezje je sestavljeno iz petih blokov procesorja, vhodno-izhodnega vmesnika, grafičnega procesorja, generatorja signalov za monitor VGA in vmesnika za razvojno ploščo. Vsi bloki so med seboj sinhronizirani z sistemsko uro 50 MHz, med njimi pa gredno vodila oz. signali, ki služijo za komunikacijo.



Slika 2: Glavni elementi vezja

### 2.1 PROCESOR

Jedro celega sistema je procesor (blok PROC) v njem se nahajata cpu enota za izvajanje programa in pomnilnik RAM v katerem so zapisane spremenljivke in program. Naloga procesorja je, da določa pozicijo in premikanje predmetov na ekranu. Prav tako pa mora skrbeti za vse dogodke, kot so na primer trki, konec igre, odzivi na pritiske tipk...

V mojem primeru je bila naloga procesorja, da je skrbel za premikanje Izmikalca, zaznaval mejo z robom ekrana, reagiral na trk, dal zahtevo za generiranje naključnega števila in reagiral na pritiske tipk.

### 2.3.1 VHODI/IZHODI PROCESORJA

VHODI/IZHODI	ŠTEVILO BITOV	OPIS
<b>adr_o</b>	8	Uporablja se za naslavljanje registrov v drugih blokih
<b>port_o</b>	12	Preko te povezave lahko pišemo v registre ostalih blokov
<b>we_o</b>	1	Če je signal na 1, pomeni da želimo pisati v registre drugih blokov
<b>re_o</b>	1	Če je signal na 1, pomeni da želimo brati vsebino registrov drugih blokov
<b>port_i</b>	1	Preko te povezave lahko beremo registre ostalih blokov

### 2.2 VHODNO - IZHODNI VMESNIK

Naloga vhodno-izhodnega vmesnika (IO blok), je da skrbi za komunikacijo in sinhronizacijo procesorja s sistemom. V njem se nahajata dva registra vhodni in izhodni register. V vhodnem registru so vse tiste vrednosti, katere pošlje procesor in so namenjene za ostale bloke. Izhodni register pa vsebuje vse vrednosti, katere pridejo iz ostalih blokov in so namenjena za procesor.

Za sinhronizacijo pa poskrbimo z prekinitvami. Prekinitve izvedemo tako, da signal zamaknemo za eno periodo ure in gledamo kdaj je vrednost originalnega signal na ena in zamaknjen signala na nič (pozitivna fronta signala) , takrat vrednost zapišemo v register.

### 2.2.1 VHODI IN IZHODI V IO VMESNIK

VHODI/IZHODI	ŠTEVILO BITOV	OPIS
<b>adr</b>	7	Naslavljanje registrov.
<b>datin</b>	11	Vrednost iz tega vodila se prenese v vhodni register na mesto kamor kaže adr.
<b>wr</b>	1	Omogoči pisanje v vhodni register.
<b>rd</b>	1	Omogoči branje iz izhodnega registra.
<b>trk</b>	1	Signal, ki pove da se je zgodil trk med objekti.
<b>vsync</b>	1	Signal, ki se postavi na 1 ob vsakem začetku prikazovanja slike.
<b>key</b>	4	Preko tega vodila beremo stanje tipk.
<b>datout</b>	12	Pošilja vrednost iz izhodnega registra v procesor.
<b>setRandomNum</b>	1	Grafiki sporoči naj generira novo naključno vrednost.
<b>x1,y1,x2,y2</b>	12	Pozicije objektov na sliki.

### 2.2.2 NADGRADNJA VHODNO – IZHODNEGA VMESNIKA

Nadgradnja vhodno – izhodnega vmesnika je bila, da posreduje ukaz o generiranju naključnega števila od procesorja do grafičnega procesorja. Največja težava je bila, da procesor ne more posodabljati podatkov v IO vmesniku z hitrostjo sistemske ure 50 MHz. Da sem dobil signal, kateri traja en cikel sistemske ure, sem moral gledati, kdaj se vrednost v registru spremeni iz 0 v 1 in ob tej spremembi generirati pulz, kateri je povzročil izbiro novega naključnega števila.

Prikaz VHDL kode, katera je skrbela za sinhronizacijo signala in generiranje naključnega števila.

```
--Preberem vrednost iz registra
if(wantNewRandom = 1) then
    ranOn <= '1';
elsif(wantNewRandom = 0) then
    ranOn <= '0';
end if;
--Zamknem signal za en cikelj ure
ranOnD <= ranOn;

--Gledam kdaj se pojavi pozitivna fronta ure
if(ranOn = '1' AND ranOnD = '0') then
    setRandomNum <= '1';
else
    setRandomNum <= '0';
end if;
```

## 2.3 GRAFIČNI PROCESOR

Grafični procesor (blok GRAFIKA) skrbi za določanje barve posameznemu slikovni točki, shranjevanju sličic objektov, katere se prikazujejo na ekranu in zaznavanje trkov med objekti .

Grafični procesor deluje tako, da od vga bloka dobi x in y pozicijo trenutne slikovne točke, katero hoče izrisati, ta pa mu vrne podatek kakšne barve mora biti točka. Barva točke pa je odvisna od tega ali točka pripada objektu ali ozadju. Za zaznavanje trkov pa gledamo, če se na eni slikovni točki dodeljena dva ali več objektov (dva ali več podatkov za barvo dodelimo eni točki).

Primer kode za iskanje trkov (**dataX** je signal, ki pove da se na slikovni tički nahaja nek objekt)

```
if (data = '1' AND (data1 = '1' or data2 = '1' or data3 = '1' or  
data4 = '1' or data5 = '1' or data6 = '1' or  
data7 = '1' or data8 = '1'))
```

### 2.3.1 VHODI/IZHODI IZ GRAFIČNEGA PROCESORJA

VHODI/IZHODI	ŠTEVILO BITOV	OPIS
<b>setRandomNum</b>	1	Pulz, kateri traja en cikel ure, in sporoča izbiro novega naključnega števila.
<b>x1, y1, x2, y2</b>	12	Koordinate, katere določajo pozicijo objektov.
<b>x, y</b>	12	Koordinati slikovne točke, za katero je potrebno določiti barvo.
<b>rgb</b>	3	Izhod, kateri pove barvo željene slikovne točke.
<b>trk</b>	1	Če pride do trka, se signal postavi na ena.

### 2.3.2 NADGRADNJA GRAFIČNEGA PROCESORJA

Od vseh blokov je grafični procesor dobil največ nadgradenj. Najprej sem v pomnilnik od grafičnega procesorja dodal nov sličico trikotnika (objekt), kateri je v igri služil kot ovira. Naslednji korak je bil postavitvev Izmikalca in ovir na ekran, pozicijo za vseh osem ovir sem krmilil samo z eno koordinato (y). Ko je bila postavitvev narejena, je bilo potrebno dodati generator naključnih števil. Slednjega sem izvedel z hitrim števcem, ki je štel od 1 do 254 in ob zahtevi vrednost prepisal v register. Ta register pa določa, katere od ovire se prikažejo na ekranu. Zadnja nadgradnja, katero sem naredil je zaznavanje trka med prikazanimi objekti in Izmikalcem (Primer kode za trk, ki sem jo uporabil je podan zgoraj).

### Koda za prikaz 1 objekta:

```
-- xt1 in yt1 se preračunata za vsako oviro posevaj
if(xt1 < 75 and yt1 < 75 and rr(0) = '1') then
    adr1 <= resize(yt1 * to_unsigned(75,7),13) + xt1;
    data1<= rom1(to_integer(adr1));
else
    data1 <= '0';
end if;
```

### Generator naključnih števil:

```
if(randomNum > 254 OR randomNum = 0) then
    randomNum <= to_unsigned(1,8);
else
    randomNum <= randomNum + 1;
end if;
```

## 2.4 BLOKA VGA IN VMESNIK

VGA blok skrbi za generiranje signalov, ki omogočajo prikazovanje slike velikosti 800x600. Prav tako, pretvori naš tri bitni signal za barvo v signal ki mogoča zapis s 6-bitno barvno globino. Vmesnik pa je namenjen za povezavo z razvojno ploščo, na kateri so tipke za upravljanje naše igre in vga konektor.

### 2.4.1 POMEMBNEJŠI SIGNALI ZA VGA BLOK

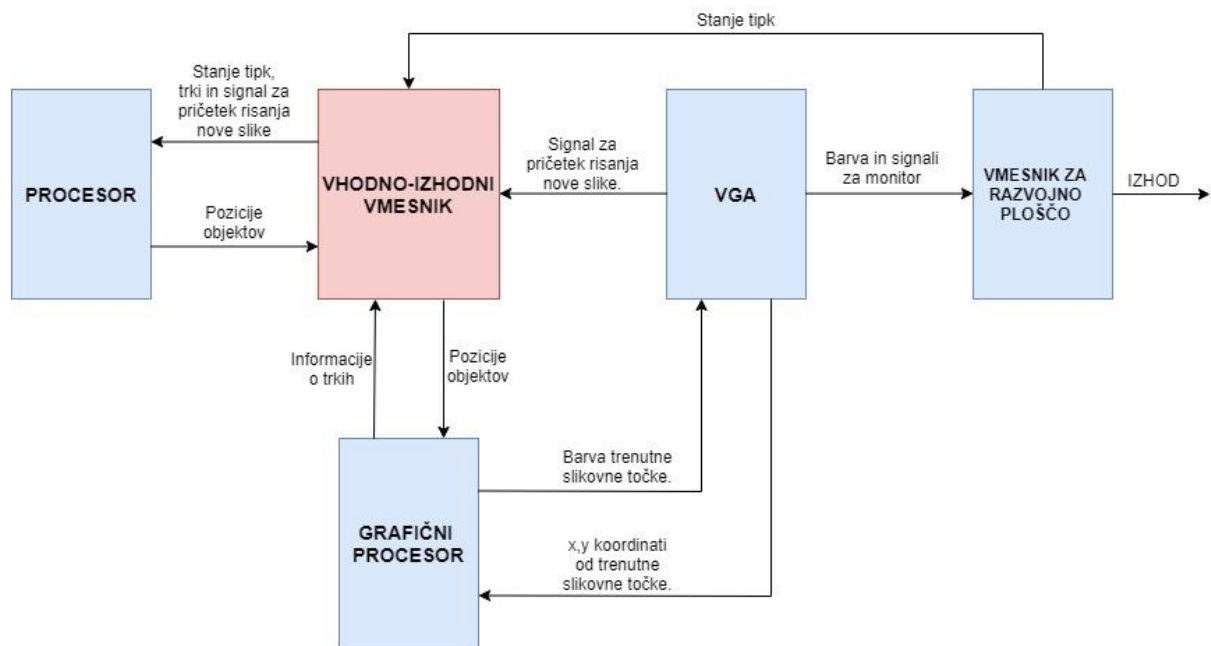
VHODI/IZHODI	ŠTEVILO BITOV	OPIS
<b>rgbin</b>	3	Barva slikovne točke, ki je trenutno izbran za prikaz.
<b>rgbout</b>	6	Izhod za prikaz slikovne točke z 6-bitno barvno globino
<b>x,y</b>	12	X in y koordinat slikovne točke, ki je trenutno v obdelavi.
<b>vSync</b>	1	Signal, ki je vedno na 1 samo pred začetkom risanja nove slike se za kratek čas postavi na 0.

### 2.4.2 POMEJBNEJŠI SIGNAL ZA VMESNIK

VHODI/IZHODI	ŠTEVILO BITOV	OPIS
<b>key</b>	3	Signal, ki sporoča, katera tipka je pritisnjena. Zaporedna številka bita, je enaka številki tipke.

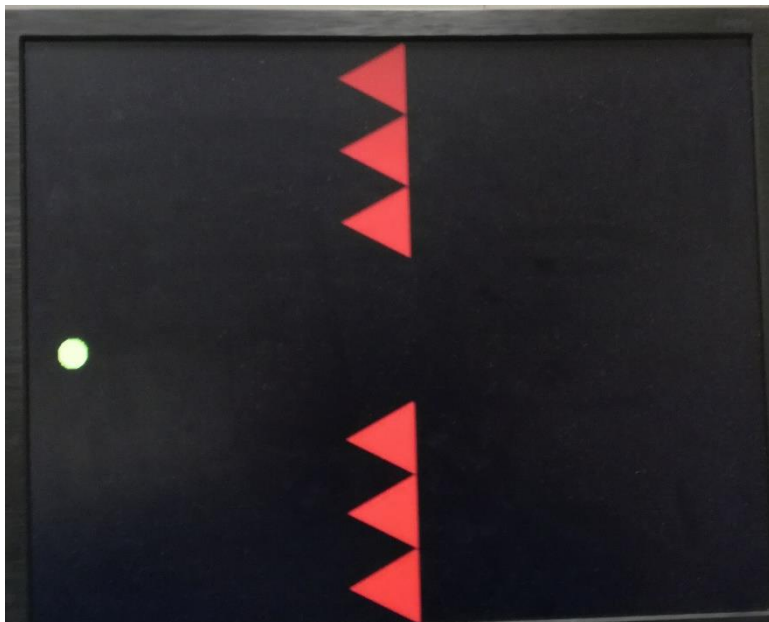


## 2.5 BLOKOVNA SHEMA



Slika 3: Blokovna shema

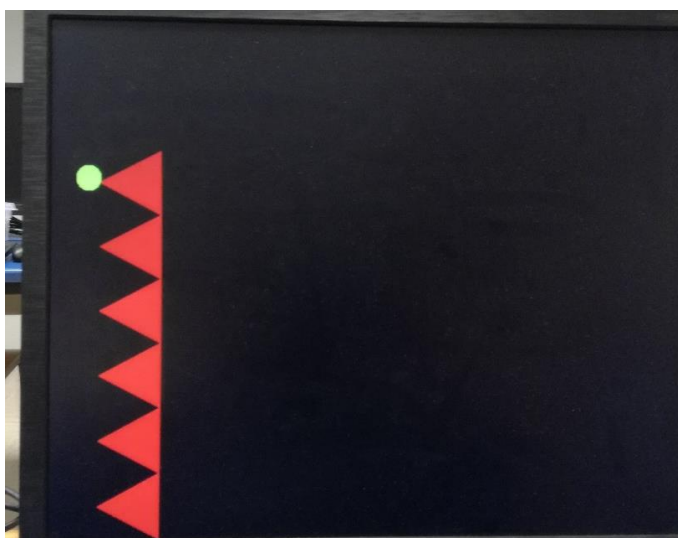
### 3 DELOVANJE VGA IGRE



Slika 4: Igra Izmikalec

Igralec igra v vlogi Izmikalca (rumena pika). Negov cilj je, da se čim dlje izmika vse hitreje bližajočim se oviram. Odprtine med ovirami se pojavljajo naključno, zato igra postane veliko bolj nepredvidljiva.

Za premikanje Izmikalca se uporabljata prva tipka (gor) in četrta (dol), če tipko za premikanje držiš Izmikalec dobi dodaten pospešek, kateri mu omogoča priti do odprtine na drugi strani pravočasno. V primer, da se igralec ne uspe izmakniti oviri, pride do trka in igre je konec. Za ponovno igro pa je potrebno pritisniti tipko 2(reset).

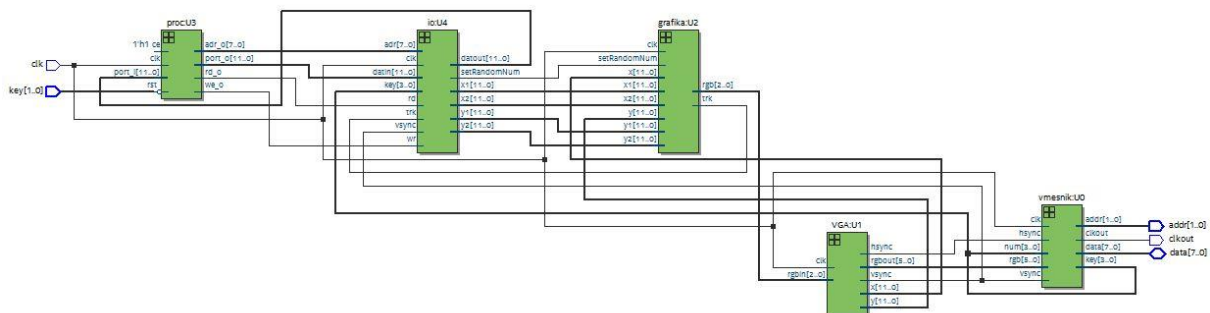


Slika 5: Primer trka

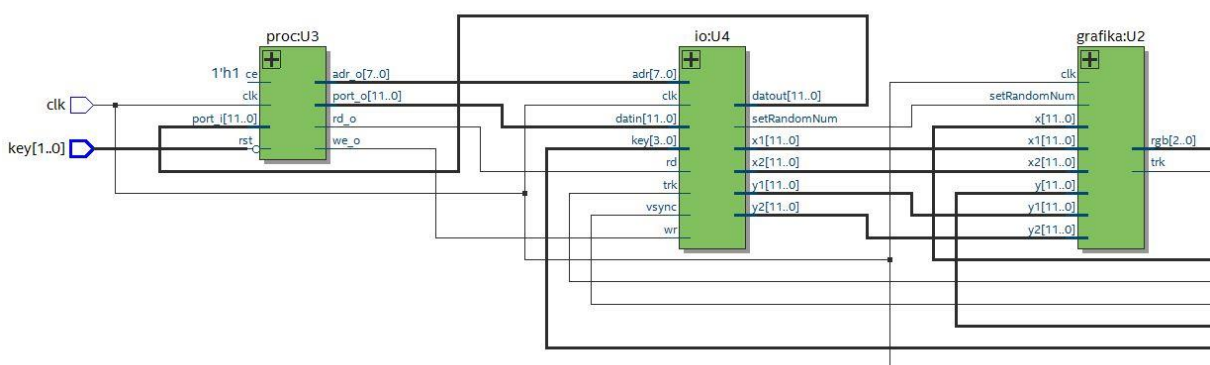
## 4 POVZETEK SINTEZE

ELEMENT	ŠTEVIL/ZASEDENOST
Logični elementi	2823/22230 (13%)
Registri	331
Pini	14/154 (9%)
Virtualni pini	0
Pomnilne celice	3072/606256 (< 1%)

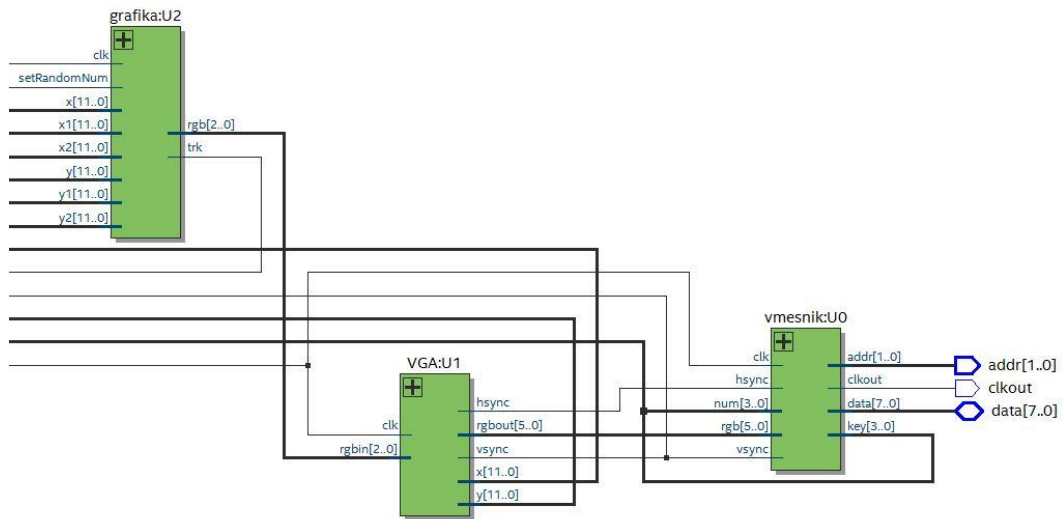
### 4.1 RTL SHEMA



Slika 6: Rtl shema



Slika 7: Rtl shema povečana 1/2



Slika 8: Rtl shema povečana 2/2