

# Prikaz slike na VGA monitoru pomoću osnovnih logičkih modula

---

**Buršić, Mateo**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:213944>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-07-24**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET  
Prijediplomski studij elektrotehnike

Završni rad

**PRIKAZ SLIKE NA VGA MONITORU  
POMOĆU OSNOVNIH LOGIČKIH  
MODULA**

Rijeka, srpanj 2023.

Mateo Buršić  
0069089537

SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Prijediplomski studij elektrotehnike

Završni rad

**PRIKAZ SLIKE NA VGA MONITORU  
POMOĆU OSNOVNIH LOGIČKIH  
MODULA**

Mentor: doc.dr.sc. Ivan Volarić

Rijeka, srpanj 2023.

Mateo Buršić  
0069089537

Rijeka, 20. ožujka 2023.

Zavod: **Zavod za automatiku i elektroniku**  
Predmet: **Digitalna elektronika**  
Grana: **2.03.03 elektronika**

## ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Mateo Buršić (0069089537)**  
Studij: **Sveučilišni prijediplomski studij elektrotehnike**

Zadatak: **Prikaz slike na VGA monitoru pomoću osnovnih logičkih modula /  
Displaying to a VGA monitor using the basic logic modules**

### Opis zadatka:

U sklopu završnog rada potrebno je izraditi sklop za generiranje signala za horizontalnu i vertikalnu sinkronizaciju za 800x600 SVGA rezoluciju i 40Mhz radnim taktom piksela. Za izradu sklopa potrebno je dizajnirati 10-bitno asinkrono brojilo, iz kojeg će se određena stanja dekodirati, te koja će služiti kao signali za horizontalnu i vertikalnu sinkronizaciju. Izlaze iz brojila je također potrebno spojiti na adresne ulaze 28C256 EEPROMa u kojem je pohranjena slika relativno male rezolucije, gdje je svaki piksel opisan sa šest bitova, tj. dva bita za crveni, zeleni i plavi kanal. Podatkovne izlaze EEPROM-a potrebno je spojiti na RGB ulaze VGA signala preko jednostavnog D/A pretvornika izrađenog pomoću šest otpornika.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

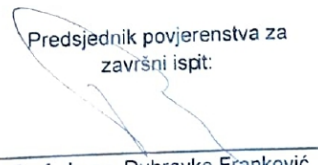
  
Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor:



Doc. dr. sc. Ivan Volarić

Predsjednik povjerenstva za  
završni ispit:

  
Prof. dr. sc. Dubravko Franković

## Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, srpanj 2023.

-----  
Ime Prezime

# Zahvala

Zahvaljujem se roditeljima i braći na podršci tijekom cijelog školovanja.  
Zahvaljujem se profesoru Volariću na korisnim savjetima i strpljenju tijekom pisanja ovog rada.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Osnovni logički sklopovi</b>	<b>3</b>
2.1	Booleova algebra . . . . .	3
2.1.1	Teoremi i aksiomi . . . . .	3
2.1.2	Logička vrata . . . . .	5
2.2	Bistabil . . . . .	8
2.2.1	S-R bistabil . . . . .	9
2.2.2	Brojilo . . . . .	10
2.3	Integrirani krugovi . . . . .	10
2.3.1	SN74LS161AN . . . . .	10
2.3.2	SN74LS04N . . . . .	11
2.3.3	SN74LS00N . . . . .	12
2.3.4	SN74LS08N . . . . .	12
<b>3</b>	<b>Princip rada sklopa</b>	<b>13</b>
3.1	Rezolucija monitora . . . . .	13
3.2	Sklop za horizontalnu sinkronizaciju . . . . .	15
3.3	Sklop za vertikalnu sinkronizaciju . . . . .	21

## *Sadržaj*

<b>4</b>	<b>Generiranje RGB signala</b>	<b>26</b>
4.1	AT28C256 . . . . .	26
4.2	VGA standard . . . . .	29
4.3	Zapis podataka u EEPROM memoriju . . . . .	32
<b>5</b>	<b>Zaključak</b>	<b>36</b>
	<b>Bibliografija</b>	<b>37</b>
	<b>Popis slika</b>	<b>39</b>
	<b>Popis tablica</b>	<b>41</b>
	<b>Pojmovnik</b>	<b>41</b>
	<b>Sažetak</b>	<b>42</b>



# Poglavlje 1

## Uvod

VGA (eng. *Video Graphics Array*) je jedan od najpoznatijih i najdugotrajnijih standarda za prikazivanje grafike na računalnim monitorima. Uveden još davne 1987. godine kao dio IBM-ove druge generacije (IBM PS/2) [1], VGA je postao izuzetno popularan zbog svoje sposobnosti prikaza grafičkog sadržaja u visokoj kvaliteti. VGA je zamijenio EGA (eng. *Enhanced Graphics Adapter*) koji se najviše koristio od 1984. do 1987. godine. Super VGA (SVGA) je razvijen godinu kasnije od VGA. Tijekom godina, VGA je postao standardni priključak za povezivanje računala i monitora, a korisnici širom svijeta oslanjali su se na ovaj standard kako bi dobili sliku na svojim monitorima.

Kroz godine VGA je proširen i na veće rezolucije kao što su 800x600, 1024x768, itd. VGA monitori podržavaju različite standarde, kao što su: SVGA, XGA, WXGA, ali ti standardi nisu toliko poznati, pa se za sve standarde koristi grupni naziv VGA. VGA monitori koriste analogni signal za prijenos podataka, što ih čini kompatibilnima s različitim računalima i uređajima. VGA je povećao grafičku memoriju na 256 kB, dok je kod EGA ona iznosila 64 kB. Također, značajno se povećala paleta boja. EGA je imao paletu od 64 boje, a VGA od 262144 boje. Svaka boja (crvena, zelena, plava) predstavljena je pomoću 6 bitova, čime se dobiva 64 razine ( $64^3 = 262144$ ).

Još jedna značajna prednost VGA monitora je jednostavnost upotrebe. Većina računala ima VGA priključak, pa je povezivanje monitora jednostavno. Ovo je osobito korisno kada korisnici žele dodati dodatni monitor ili zamijeniti stari monitor s

## Poglavlje 1. Uvod

VGA priključkom.

Kroz godine, VGA je postupno ustupio mjesto novijim sučeljima poput HDMI-a (eng. *High-Definition Multimedia Interface*) i *DisplayPort*-a, koji pružaju veće rezolucije i digitalni prijenos podataka. Unatoč napretku u tehnologiji prikaza, VGA monitori nisu u potpunosti izumrli. Mnoge industrijske aplikacije, kao što su industrijski kontrolni sustavi ili strojevi, koriste VGA sučelje za prikaz podataka [2]. VGA monitori su u upotrebi zbog svoje robusnosti na različite šumove u industrijskom okruženju. Ova trajnost i univerzalnost osiguravaju da su VGA monitori ostali popularni među korisnicima koji zahtijevaju pouzdanost i jednostavnost prikaza. Također, postoje adapteri i konvertori pomoću kojih je omogućeno povezivanje VGA monitora s modernijim sučeljima.

# Poglavlje 2

## Osnovni logički sklopovi

Budući da se u ovom završnom radu koriste različiti logički sklopovi, potrebno je definirati principe na kojima oni funkcioniraju, to jest potrebno je definirati osnove digitalne elektronike.

### 2.1 Booleova algebra

Booleova algebra koristi se za analizu i sintezu logičkih sklopova. Time se logička svojstva digitalnih sklopova mogu iskazati i algebarskim ili logičkim jednadžbama. Naziv je dobila po engleskom matematičaru Georgeu Booleu. On je definirao teoreme koji omogućuju pojednostavljenje složenih logičkih izraza [3]. U nastavku su opisana osnovna logička vrata i sklopovi koji su baza integriranih krugova.

#### 2.1.1 Teoremi i aksiomi

Operacije Booleove algebre zasnovane su na aksiomima koje je postavio E.V. Huntington.

Za  $x, y, z \in A$  vrijedi:

##### 1. Zatvorenost:

$$x + y \in A \tag{2.1a}$$

*Poglavlje 2. Osnovni logički sklopovi*

$$x * y \in A \quad (2.1b)$$

**2. Komutacija:**

$$x + y = y + x \quad (2.1c)$$

$$x * y = y * x \quad (2.1d)$$

**3. Distribucija:**

$$x + (y * z) = (y + x) * (x + z) \quad (2.1e)$$

$$x * (y + z) = (y * x) + (x * z) \quad (2.1f)$$

**4. Neutralnost (identičnost):**

$$x + 0 = x \quad (2.1g)$$

$$x * 1 = x \quad (2.1h)$$

**5. Komplement:**

$$x + \bar{x} = 1 \quad (2.1i)$$

$$x * \bar{x} = 0 \quad (2.1j)$$

**6. Dualnost:**

$$x + (y * z) = (y + x) * (x + z) \quad (2.1k)$$

$$x * (y + z) = (y * x) + (x * z) \quad (2.1l)$$

Neki od najbitnijih teorema Boolove algebre:

**1. Zakon idempotencije:**

$$x + x = x \quad (2.2a)$$

$$x * x = x \quad (2.2b)$$

**2. Dominacija:**

$$x + 1 = 1 \quad (2.2c)$$

$$x * 0 = 0 \quad (2.2d)$$

### 3. Apsorpcija:

$$x + x * y = x \quad (2.2e)$$

$$x * (x + y) = x \quad (2.2f)$$

### 4. Jedinstvenost komplementa:

Neka su  $\bar{x}_1$  i  $\bar{x}_2$  komplementi x tako da vrijedi:

$$x + \bar{x}_1 = x + \bar{x}_2 = 1 \quad (2.2g)$$

$$x * \bar{x}_1 = x * \bar{x}_2 = 0 \quad (2.2h)$$

Kombiniranjem ovih jednadžbi može se dokazati da vrijedi:  $\bar{x}_1 = \bar{x}_2$

### 5. Involicija:

$$\overline{\bar{x}} = x \quad (2.2i)$$

### 6. De Morganov zakon:

$$\overline{x + y} = \bar{x} * \bar{y} \quad (2.2j)$$

$$\overline{x * y} = \bar{x} + \bar{y} \quad (2.2k)$$

## 2.1.2 Logička vrata

### Logička operacija I:

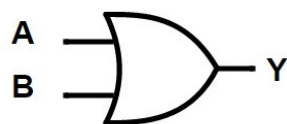
Logička vrata I (eng. *AND gate*) obavljaju logičku operaciju I (konjukcija). Ova vrata mogu imati dva ili više ulaza. Izlaz je u stanju 1 samo ako su svi ulazi u stanju 1 [3]. Tablica istine za I vrata prikazana je u tablici 2.1, a simbol je prikazan na slici 2.1.



Slika 2.1 Logička vrata I - simbol (Izvor:[4]).

**Logička operacija ILI:**

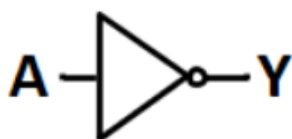
Logička vrata ILI (eng. *OR gate*) obavljaju logičku operaciju ILI (disjunkcija). ILI vrata mogu imati dva ili više ulaza. Izlaz je u stanju 1 ako barem jedan od ulaza ima stanje 1 [3]. Tablica istine za ILI vrata prikazana je tablicom 2.2, a simbol je prikazan na slici 2.2.



Slika 2.2 Logička vrata ILI - simbol (Izvor [4]).

**Logička operacija NE:**

Logička vrata NE (eng. *NOT gate*) obavljaju logičku operaciju NE (negacija, inverzija). Ova vrata imaju jedan ulaz. Kada je na ulazu stanje 1, na izlazu je stanje 0 i obrnuto [3]. Tablica istine za NE vrata prikazana je u tablici 2.3, a simbol je prikazan na slici 2.3.



Slika 2.3 Logička vrata NE - simbol (Izvor: [4]).

**Logička operacija NI:**

Logička vrata NI (eng. *NAND gate*) obavljaju logičku operaciju NI. NI vrata mogu imati dva ili više ulaza. Na izlazu ima stanje 1 ako je na bilo kojem ulazu stanje 0 [3]. Tablicom 2.4 prikazana je tablica istine za NI vrata, a simbol je prikazan na slici 2.4.

Poglavlje 2. Osnovni logički sklopovi



Slika 2.4 Logička vrata NI - simbol (Izvor: [4]).

**Logička operacija NILI:**

Logička vrata NILI (eng. *NOR gate*) obavljaju logičku operaciju NILI. Ova vrata mogu imati dva ili više ulaza. Na izlazu ima stanje 1 samo kada su svi ulazi u stanju 0 [3]. Tablica istine za NE vrata prikazana je u tablici 2.5, a na slici 2.5 prikazan je simbol NILI vrata.



Slika 2.5 Logička vrata NILI - simbol (Izvor: [4]).

Tablica 2.1 Tablica stanja logičkih vrata I.

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Tablica 2.2 Tablica stanja logičkih vrata ILI.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

## Poglavlje 2. Osnovni logički sklopovi

Tablica 2.3 Tablica stanja logičkih vrata  
NE.

A	Y
0	1
1	0

Tablica 2.4 Tablica stanja logičkih vrata  
NI.

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Tablica 2.5 Tablica stanja logičkih vrata NILI.

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

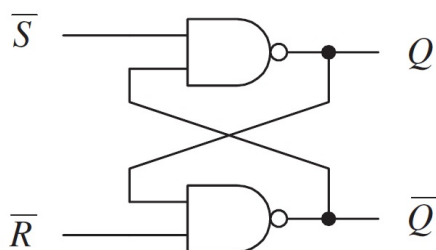
## 2.2 Bistabil

Bistabili predstavljaju osnovu sekvencijske logike. Imaju 2 stabilna stanja (SET i RESET). Sekvencijski sklopovi su oni sklopovi čiji izlaz ovisi ne o samo o ulazima, već i o unutaršnjem stanju sklopa [3]. Bistabili uz ulaz na koji se dovodi upravljački impuls ( $CLK$ ), mogu imati jedan ili dva ulaza. Bistabili imaju dva izlaza, od kojih je jedan izlaz  $Q$ , a drugi izlaz predstavlja njegov komplement  $\bar{Q}$ . Primjenjuju se kod brojala, registara i memorije. Neki od tipova bistabila su: S-R, J-K, D i T. U nastavku će biti opisano kako radi S-R bistabil, jer se taj bistabili koristi u ovom radu.



### 2.2.1 S-R bistabil

S-R bistabil ima dva ulaza:  $\bar{S}$  (*set* - postavlja izlaz u stanje 1) i  $\bar{R}$  (*reset* - resetira izlaz, tj. postavlja ga u stanje 0). Shema S-R bistabila prikazana je na slici 2.6, a tablica istine prikazana je tablicom 2.6.



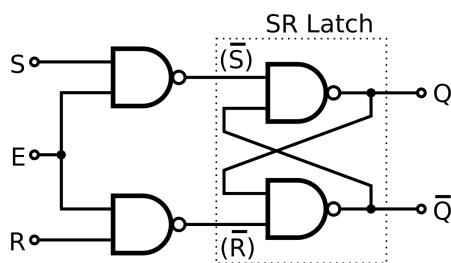
Slika 2.6 S-R bistabil (Izvor: [5]).

Tablica 2.6 Tablica stanja S-R bistabila.

$\bar{S}$	$\bar{R}$	$Q$	$\bar{Q}$
0	0	?	?
0	1	1	0
1	0	0	1
1	1	bez promjene	bez promjene

Ako je ulaz  $\bar{R}$  u stanju 0, prema tablici za logička vrata NI (tablica 2.4) može se zaključiti da će izlaz  $\bar{Q}$  biti 1. Također, ako je ulaz  $\bar{S}$  u stanju 0, pomoću iste tablice može se zaključiti da će i  $Q$  biti u stanju 1. No, budući da su  $Q$  i  $\bar{Q}$  komplementarna stanja, ona ne mogu imati istu vrijednost. Time se na izlazu dobije neizvjesno stanje.

Ponekad je u praksi korisno kontrolirati S-R bistabil neovisno o  $\bar{S}$  i  $\bar{R}$  ulazima. Zbog toga se dodaje još jedan ulaz, koji se označava sa  $E$  (*Enable*). Na slici 2.7 prikazana je shema sinkronog S-R bistabila. Izlazi iz toga sklopa mogu se mijenjati samo kada je ulaz  $E$  u stanju 1.



Slika 2.7 Sinkroni S-R bistabil (Izvor: [6]).

## 2.2.2 Brojilo

Brojilo je skup bistabila pomoću kojega je moguće postići određen broj stanja, kao i redoslijed kojima se ta stanja pojavljuju. Brojila se dijele na sinkrona i asinkrona. Kod sinkronih brojila signal radnog takta ( $CLK$ ) dolazi do svih bistabila u isto vrijeme, tako da se i promjene u tim bistabilima događaju istovremeno. Kod asinkronih brojila promjene stanja bistabila ovise o promjeni stanja prethodnog bistabila. Prednost asinkronih brojila u odnosu na sinkrona je ta što je njihova izrada jeftinija, a nedostatak su to što okidanje bistabila ovisi o drugim bistabilima, zbog čega se javljaju određena kašnjenja. Također, nedostatak je i to što se ne može postići proizvoljan ciklus brojanja.

## 2.3 Integrirani krugovi

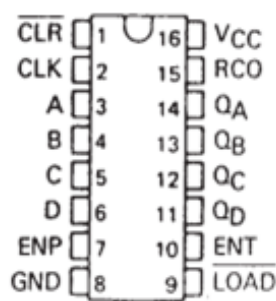
U ovoj dijelu opisani su integrirani krugovi koji su korišteni u ovom završnom radu. Njihova uloga u cijelom sklopu opisana je u sljedećem poglavlju.

### 2.3.1 SN74LS161AN

SN74LS161AN je 4-bitno sinkrono brojilo. Na slici 2.8 prikazan je raspored priključnica. Najbitnije priključnice za rad sklopa su  $VCC$  (napon napajanja) i  $GND$  (*ground*). Priključnica  $\overline{CLR}$  omogućava resetiranje sklopa, tj.  $Q_A - Q_D$  se postavljaju u stanje 0. Budući da oznaka priključnice  $\overline{CLR}$  sadrži komplement, resetiranje

## Poglavlje 2. Osnovni logički sklopovi

sklopa događa se kada je ona u stanju 0. Promjene na izlazu iz sklopa mogu se dogoditi samo kada  $CLK$  prelazi iz stanja 0 u stanje 1, tj. na uzlazni brid. Priključnica  $\overline{LOAD}$  u stanju 0 onemogućava rad brojila.

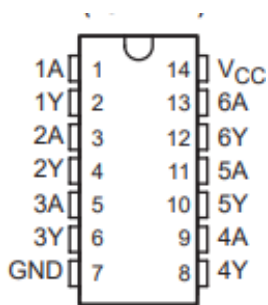


Slika 2.8 SN74LS161AN (Izvor: [7]).

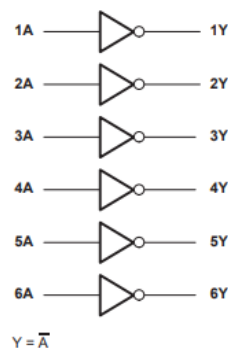
Priključnice  $ENP$  i  $ENT$  (*Enable* priključnice) omogućavaju rad brojila kada su u stanju 1.  $RCO$  (*ripple carry output*) omogućuje spajanje više ovakvih brojila kako bi se povećao broj stanja, jer se jednim brojiлом može brojati od 0 ( $0000_2$ ) do 15 ( $1111_2$ ).

### 2.3.2 SN74LS04N

SN74LS04N je integrirani krug koji sadrži šest NE vrata. Na slikama 2.9 i 2.10 prikazan je raspored priključnica i njihova funkcionalnost.



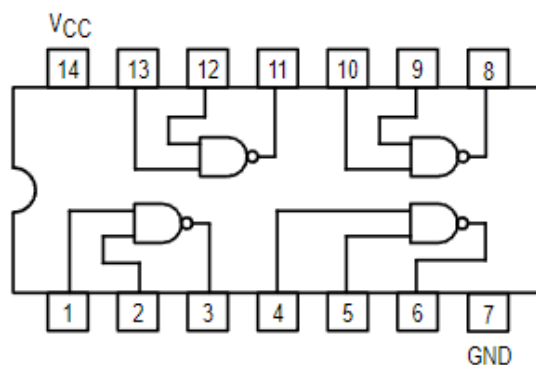
Slika 2.9 SN74LS04N (Izvor: [8]).



Slika 2.10 SN74LS04N - logički dijagram (Izvor: [8]).

### 2.3.3 SN74LS00N

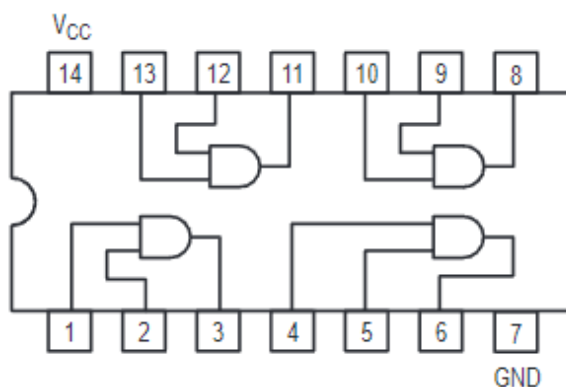
Integrirani krug SN74LS00N sadrži četiri logičkih NILI vrata s dva ulaza. Rad ovog sklopa opisan je tablicom 2.5, a raspored priključnica prikazan je na slici 2.11.



Slika 2.11 SN74LS00N (Izvor: [9]).

### 2.3.4 SN74LS08N

Integrirani krug SN74LS08N sadrži četiri I vrata s dva ulaza. Osnova rada ovog integriranog kruga je tablica 2.1, a raspored priključnica prikazan je na slici 2.12.



Slika 2.12 SN74LS08N Izvor: [10]).

## Poglavlje 3

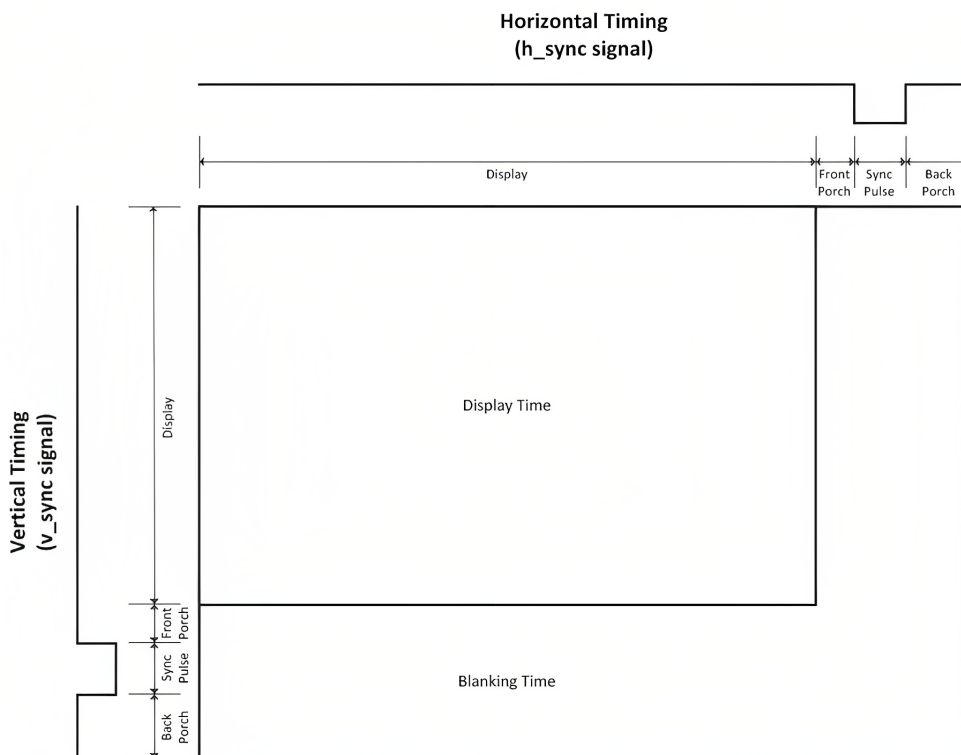
# Princip rada sklopa

U ovom poglavlju opisano je na koji je način potrebno povezati sklopove opisane u prethodnom poglavlju kako bi bilo moguće prikazati sliku na monitoru. Broj piksela koji se prikazuje u jednom redu pomnožen s brojem piksela koji se prikazuje u jednom stupcu predstavlja rezoluciju monitora. Postoji više rezolucija koje se koriste kod VGA monitora, a neke od njih su: 640x350, 640x480, 800x600, 1024x768, 1920x1440, itd.

### 3.1 Rezolucija monitora

Kako bi se prikazala slika na monitoru, potrebno je odrediti vremenske intervale tijekom kojih se slika može prikazivati, te intervale kada je to zabranjeno. Ti intervale prikazani su na slici 3.1. Kao što se može vidjeti na slici 3.1, vremenski intervale unutar jednog reda ili stupca dijele se na: *Display*, *Front Porch*, *Sync Pulse* i *Back Porch*. Unutar intervala *Sync Pulse* zraka elektrona pomiče se s kraja reda (ili stupca) na početak te se u tom intervalu na monitoru ne prikazuje ništa. Kada je riječ o horizontalnom vraćanju zrake, onda taj interval nazivamo *Horizontal Sync*, a kada je riječ o vertikalnom vraćanju zrake, taj se interval naziva *Vertical Sync*.

### Poglavlje 3. Princip rada sklopa



Slika 3.1 Prikazivanje slike na monitoru - vremenski dijagram [11].

Interval koji prethodi *Sync Pulse*-u naziva se *Front Porch*, a interval koji slijedi nakon naziva se *Back Porch*. *Front Porch* omogućava zraci elektrona da se smiri prije nego što se vrati na početak reda, tj. stupca te se u tom intervalu također ništa ne prikazuje. *Back Porch* daje elektronskoj zraci dovoljno vremena da se pozicionira na početak tako da niti u tom intervalu nema prikaza na monitoru. Trajanje opisanih intervala za rezoluciju 800x600 i frekvencijom prikaza piksela 40 MHz prikazano je tablicama 3.1 i 3.2.

U ovom završnom radu odabrana je rezolucija 200x600 i frekvencija prikazivanja piksela 10 MHz da bismo smanjili zahtjeve sklopovlja. Pošto je i frekvencija piksela i horizontalna rezolucija četiri puta manja, vremenski intervali ostaju isti kao i za rezoluciju 800x600, ali smo time povećali horizontalnu veličinu piksela četiri puta. Tako preskalirane vrijednosti prikazane su u tablici 3.3.

### Poglavlje 3. Princip rada sklopa

Tablica 3.1 Intervali unutar jednog reda za rezoluciju 800x600 i frekvenciju 40 MHz (Izvor: [12]).

Scanline Part	Pixels	Time[ $\mu s$ ]
Visible Area	800	20
Front Porch	40	1
Sync Pulse	128	3.2
Back Porch	88	2.2
Whole Frame	1056	26.4

Tablica 3.2 Intervali unutar jednog stupca za rezoluciju 800x600 i frekvenciju 40 MHz (Izvor: [12]).

Scanline Part	Pixels	Time[ $ms$ ]
Visible Area	600	15.84
Front Porch	1	0.0264
Sync Pulse	4	0.1056
Back Porch	23	0.6072
Whole Frame	628	16.5792

Tablica 3.3 Intervali unutar jednog reda za rezoluciju 200x600 i frekvenciju 10 MHz.

Scanline Part	Pixels	Time[ $\mu s$ ]
Visible Area	200	20
Front Porch	10	1
Sync Pulse	32	3.2
Back Porch	22	2.2

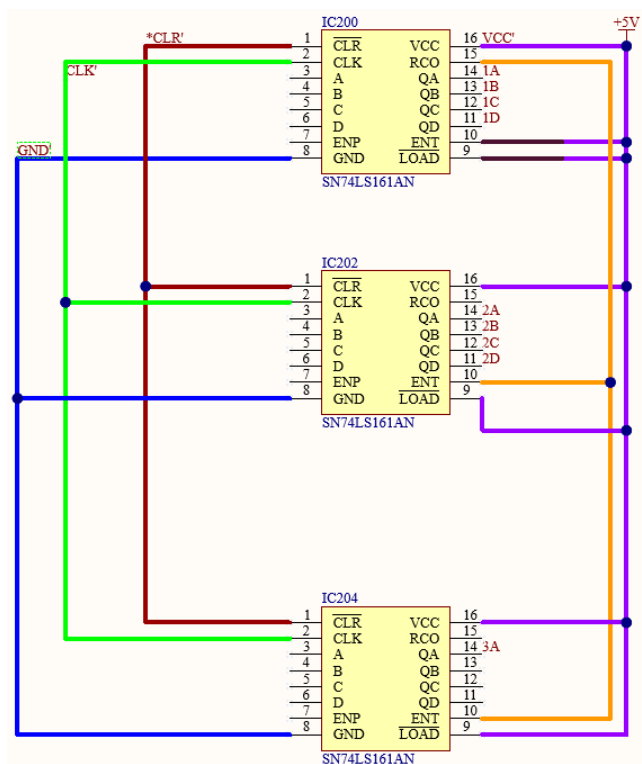
## 3.2 Sklop za horizontalnu sinkronizaciju

Unutar jednog reda prikazuje se 264 ( $200 + 10 + 32 + 22$ ) piksela, što znači da je potrebno dizajnirati mod 264 brojilo, tj. brojilo s 264 različita stanja. Pritom da bismo bili unutar zadanih specifikacija iz tablice 3.3, potrebno je dekodirati sljedeća stanja:  $200_{10} = 11001000_2$ ,  $210_{10} = 11010010_2$ ,  $242_{10} = 11110010_2$ ,  $264_{10} = 100001010_2$ .

Shodno tome, možemo zaključiti da nam je potrebno 9-bitno brojilo. Budući da je brojilo koje koristimo (SN74LS161AN 2.3.1) 4-bitno, potrebna su nam tri integrirana kruga. Brojila se povezuju na način da se sva tri sklopa spoje na isti napon napajanja (5V) i uzemljenje. Na napon napajanja je spojena i priključnica *ENP* prvog brojila. Nadalje, *CLK* priključnice sva tri brojila spojene su zajedno jer se radi o sinkronom

### Poglavlje 3. Princip rada sklopa

brojilu. *ENP* priključnica drugog brojila povezuje se s *RCO* priključnicom prvog brojila, a *ENP* priključnica trećeg brojila povezuje se s *RCO* priključnicom drugog brojila. Razlog tomu je to što *RCO* priključnica brojila prijeđe u stanje 1 kada se brojilo prebacuje iz stanja  $1111_2$  ( $15_{10}$ ) u stanje  $0000_2$  ( $0_{10}$ ). Time se aktivira sljedeće brojilo, čime se povećava raspon do kojega je moguće brojati. Shematski prikazi brojila izrađen je u programu Altium Designer, te je prikazan na slici 3.2.



Slika 3.2 Shematski prikaz brojila (horizontalni smjer) u Altium Designeru.

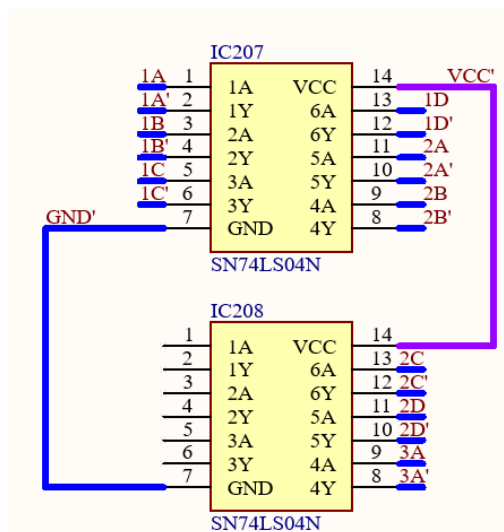
Devet izlaza iz brojila označeno je *Net Label*-ima. Pritom, oznaka 3A predstavlja najznačajniju znamenku (eng. MSB - *most significant bit*), a 1A predstavlja najmanje značajnu znamenku (eng. LSB - *least significant bit*).

Da bismo dekodirali već spomenuta stanja koristit ćemo I vrata, te NE vrata. Budući da SN74LS04N (2.3.2) sadrži 6 NE vrata, a nama je potrebno 8 za svaki od izlaza brojila, koristit ćemo dva integrirana kruga, kao što je prikazano na slici 3.3.



### Poglavlje 3. Princip rada sklopa

Negirani izlazi imaju oznake 1A', 1B',...,3A'.



Slika 3.3 Shematski prikaz spajanja dva sklopa SN74LS04N (horizontalni smjer).

Za dekodiranje svakog od spomenutog stanja potrebna su dva integrirana kruga SN74LS08N (2.3.4). Da bismo dekodirali određeno stanje brojlara, na ulaze I vrata potrebno je dovesti sve izlaze brojlara koji u tom stanju iznose 1, a one izlaze koji iznose 0, na ulaze I vrata je potrebno dovesti negirane.

#### Stanje 200:

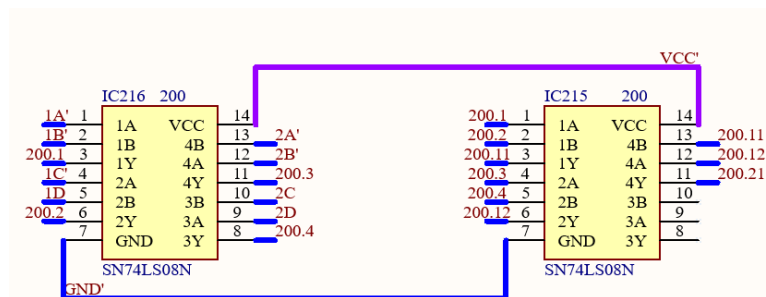
Kada je brojilo u stanju 200, izlazi 1D, 2C i 2D su u stanju 1, a izlazi 1A, 1B, 1C, 2A, 2B i 3A u stanju 0. Najznačajniji bit je u stanju 0, pa njega nije potrebno dovesti na ulaze I vrata. To znači da su ulazi u I vrata: 2D, 2C, 2B', 2A', 1D, 1C', 1B', 1A'. Sklop za dekodiranje stanja 200 prikazan je na slici 3.4.

Pošto su korištena dvo-ulazna vrata, potrebno je vršiti operaciju I postepeno u parovima, kao što je prikazano na slici 3.4. Pritom su međurezultati označeni s 200.1, 200.2, 200.3, 200.4, 200.11 i 200.12, dok konačno stanje ima oznaku 200.21.

$$200.1 = 1A' * 1B' \quad (3.1a)$$

$$200.2 = 1C' * 1D \quad (3.1b)$$

Poglavlje 3. Princip rada sklopa



Slika 3.4 Dekodiranje stanja 200.

$$200.3 = 2A' * 2B' \quad (3.1c)$$

$$200.4 = 2C * 2D \quad (3.1d)$$

$$200.11 = 200.1 * 200.2 \quad (3.1e)$$

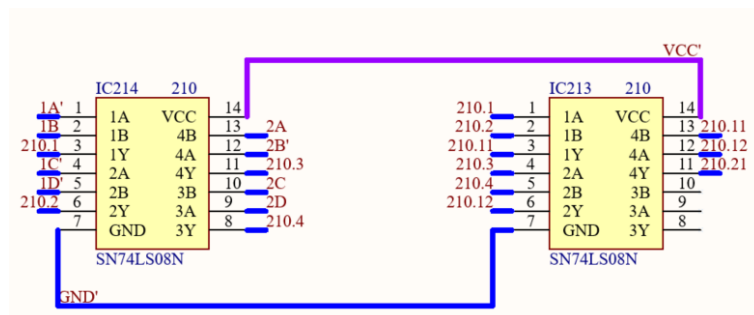
$$200.12 = 200.3 * 200.4 \quad (3.1f)$$

$$200.21 = 200.11 * 200.12 \quad (3.1g)$$

$$200.21 = 200.1 * 200.2 * 200.3 * 200.4 = 1A' * 1B' * 1C' * 1D' * 2A' * 2B' * 2C * 2D \quad (3.1h)$$

**Stanje 210:**

Kada je brojilo u stanju 210, četiri izlaza su u stanju 1, a njih pet je u stanju 0. Ni u ovom slučaju nije potrebno koristiti izlaz označen s 3A. Ulazi u I vrata su: 2D, 2C, 2B', 2A, 1D', 1C', 1B, 1A'. Sklop za dekodiranje stanja 210 prikazan je na slici 3.5.



Slika 3.5 Dekodiranje broja 210.

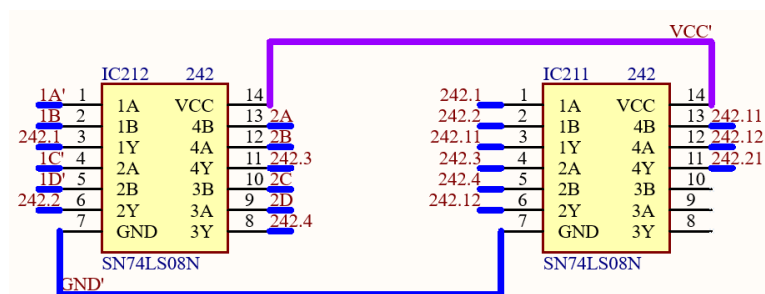
### Poglavlje 3. Princip rada sklopa

Analogno kao što je izveden izraz za dekodiranje stanja 200 (3.1), može se dobiti i izraz za dekodiranje stanja 210.

$$210.21 = 1A' * 1B * 1C' * 1D' * 2A * 2B' * 2C * 2D \quad (3.2)$$

#### Stanje 242:

U slučaju kada je stanje brojila 242, pet izlaza je u stanju 1, a njih četiri je u stanju 0. Ni u ovom slučaju nije potrebno koristiti izlaz označen s 3A. Ulazi u I vrata su: 2D, 2C, 2B, 2A, 1D', 1C', 1B, 1A'. Sklop kojim se dekodira stanje 242 prikazan je slikom 3.6.



Slika 3.6 Dekodiranje stanja 242.

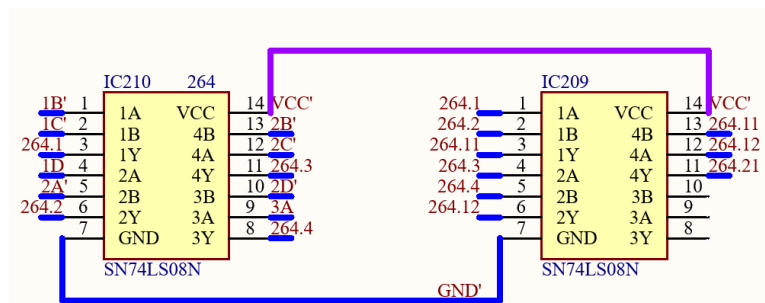
Analogno kao što je izveden izraz za dekodiranje broja 200 (3.1), može se dobiti i izraz za dekodiranje broja 242.

$$242.21 = 1A' * 1B * 1C' * 1D' * 2A * 2B * 2C * 2D \quad (3.3)$$

#### Stanje 264:

U slučaju kada je brojilo u stanju 264, samo dva izlaza su u stanju 1, a sedam ih je u stanju 0. Budući da je u ovom slučaju izlaz koji je 3A u stanju 1, a dva sklopa SN74LS08N zajedno imaju osam ulaznih priključnica, potrebno je izostaviti neki drugi izlaz iz brojila. U ovom slučaju izostavlja se najmanje značajna znamenka, tj. 1A, koja u ovom slučaju iznosi 0. Stoga, ulazi u I vrata su: 3A, 2D', 2C', 2B', 2A', 1D, 1C', 1B'. Sklop kojim se dekodira stanje 264 prikazan je na slici 3.7.

Poglavlje 3. Princip rada sklopa



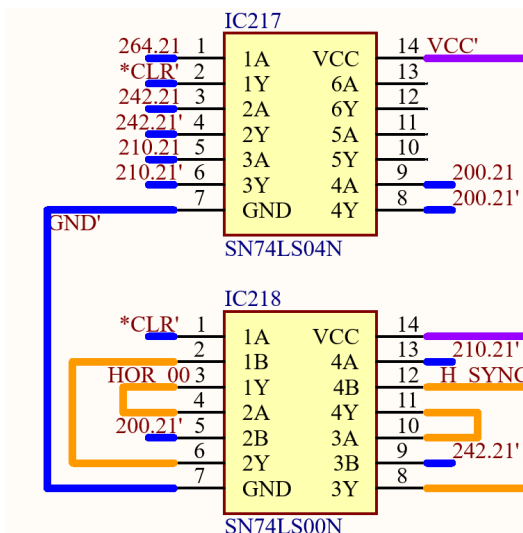
Slika 3.7 Prepoznavanje broja 264.

Analogno kao što je izveden izraz za prepoznavanje broja 200 (3.1), može se dobiti i izraz za prepoznavanje broja 264.

$$264.21 = 1B' * 1C' * 1D * 2A' * 2B' * 2C' * 2D' * 3A \quad (3.4)$$

Posljednji korak u ovom dijelu je generiranje dva ju signala: signal koji će biti u stanju 1 kada je potrebno slati piksele monitoru (HOR\_00), te signal za horizontalnu sinkronizaciju (H\_SYNC). Kao što je vidljivo iz tablice 3.1, HOR\_00 treba biti u stanju 1 između stanja 0 i 200, a H\_SYNC između stanja 210 i 242. Za to nam trebaju dva S-R bistabila čiji su izlazi spomenuti signali. To znači da je bistabil odgovoran za HOR\_00 potrebno setirati u stanju 0, a resetirati u stanju 200, a bistabil odgovoran za H\_SYNC je potrebno setirati u stanju 210, a resetirati u stanju 242. Sklop kojim se to postiže prikazan je na slici 3.8.

### Poglavlje 3. Princip rada sklopa



Slika 3.8 Sklop za generiranje signala HOR\_00 i H\_SYNC

## 3.3 Sklop za vertikalnu sinkronizaciju

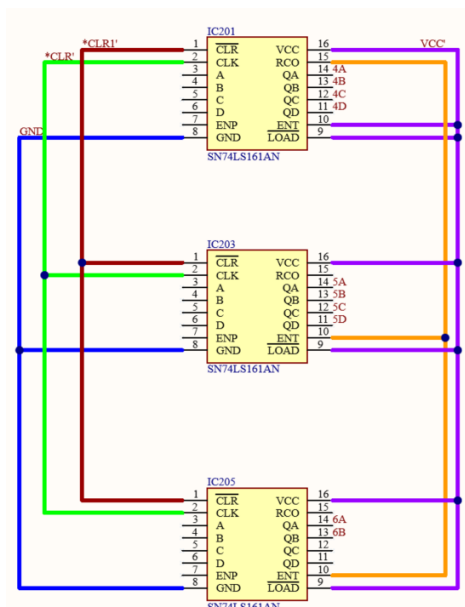
U prethodnom poglavlju, opisano je kako se broji unutar jednog reda. U ovom poglavlju opisana je funkcionalnost sklopa za vertikalnu sinkronizaciju, čiji je princip rada jednak horizontalnom, samo je potrebno dekodirati druga stanja brojila. Prema tablici 3.2, brojilo treba imati 628 ( $600 + 1 + 4 + 23$ ) stanja. Stanja koja je potrebno dekodirati su:  $600_{10} = 1001011000_2$ ,  $601_{10} = 1001011001_2$ ,  $605_{10} = 1001011101_2$ ,  $628_{10} = 1001110100_2$ .

Shodno tome, možemo zaključiti da nam je potrebno 10-bitno brojilo. Budući da je brojilo koje koristimo (SN74LS161AN 2.3.1) 4-bitno, potrebna su nam tri integrirana kruga. Brojila se povezuju na isti način kao što je opisano u prethodnom poglavlju. 6B označava MSB, a 4A označava LSB. Brojilo je prikazano na slici 3.9

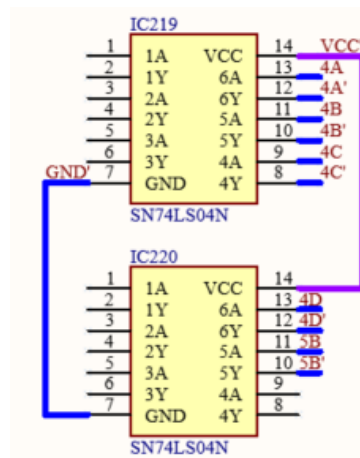
U sljedećem koraku potrebno je dekodirati prethodno definirana stanja. Za to je potrebno upotrijebiti I vrata, te NE vrata. Potrebno je iskoristiti dva sklopa SN74LS04N (2.3.2), kao što je prikazano na slici 3.10. Negirani izlazi označeni su s: 4A', 4B',...,6B'.

Da bi se dekodiralo svako od stanja, potrebna su dva integrirana kruga SN74LS08N.

Poglavlje 3. Princip rada sklopa



Slika 3.9 Shematski prikaz brojila (vertikalni smjer) u Altium Designeru.

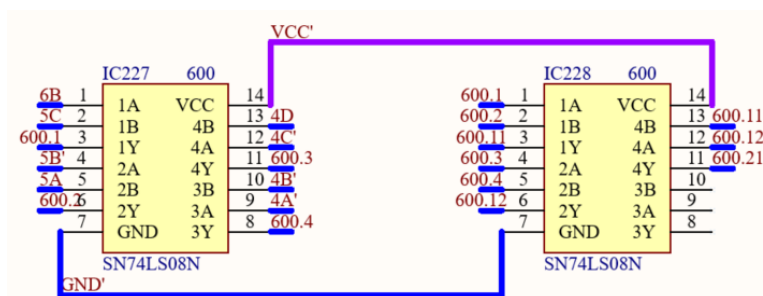


Slika 3.10 Shematski prikaz spajanja dva sklopa SN74LS04N (vertikalni smjer).

Izlazi iz brojila označeni s 5D i 6A (slika 3.9) su za svako od spomenutih stanja u stanju 0, pa te izlaze nije potrebno dovoditi na ulaze I vrata.

**Stanje 600:**

Kada je brojilo u stanju 600, četiri izlaza su u stanju 1, a šest ih je u stanju 0. Uspoređujući binarni zapis sa slikom 3.9, može se uočiti kako su izlazi koji su u stanju 1 označeni s 4D, 5A, 5C, 6B. Ulazi u sklop I vrata su: 6B, 5C, 5B', 5A, 4D, 4C', 4B', 4A'. Sklop pomoću kojeg se dekodira stanje 600 prikazan je slikom 3.11.



Slika 3.11 Dekodiranje stanja 600.

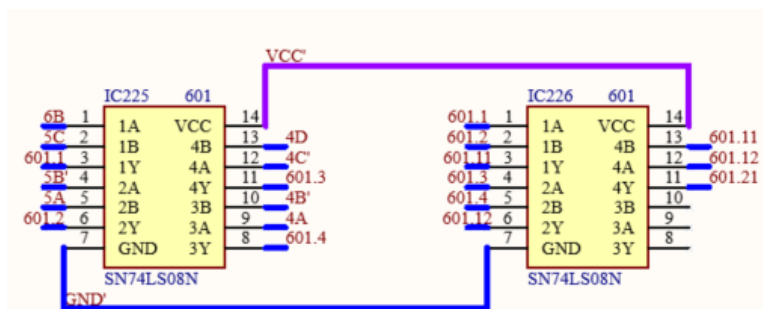
### Poglavlje 3. Princip rada sklopa

Analogno kao što u je izveden izraz za dekodiranje stanja 200 (3.1), može se dobiti i izraz za dekodiranje stanja 600.

$$600.21 = 4A' * 4B' * 4C' * 4D * 5A * 5B' * 5C * 6B \quad (3.5)$$

#### Stanje 601:

U slučaju kada je brojilo u stanju 601, pet izlaza je u stanju 1, a preostalih pet je u stanju 0. Ulazi u I vrata su: 6B, 5C, 5B', 5A, 4D, 4C', 4B', 4A. Sklop kojim se dekodira stanje 601 prikazan je slikom 3.11.



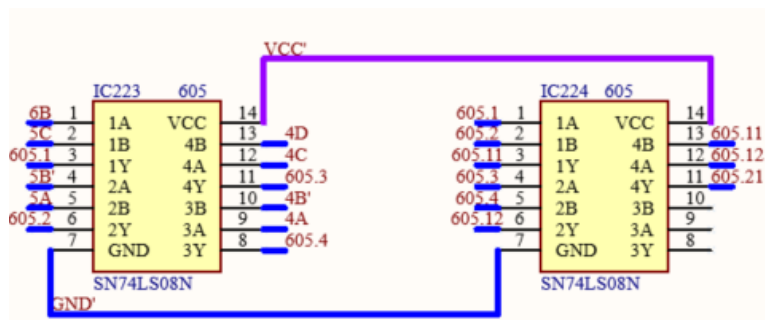
Slika 3.12 Dekodiranje broja 601.

$$601.21 = 4A * 4B' * 4C' * 4D * 5A * 5B' * 5C * 6B \quad (3.6)$$

#### Stanje 605:

Kada je brojilo u stanju 605, šest izlaza je u stanju 1, a četiri izlaza su u stanju 0. Na ulazima u I vrata su: 6B, 5C, 5B', 5A, 4D, 4C, 4B', 4A,. Sklop pomoću kojeg se dekodira stanje 605 prikazan je slikom 3.11.

Poglavlje 3. Princip rada sklopa

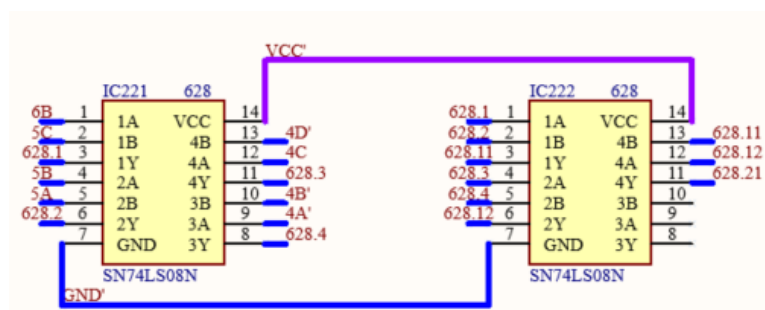


Slika 3.13 Dekodiranje broja 605.

$$605.21 = 4A * 4B' * 4C * 4D * 5A * 5B' * 5C * 6B$$

**Stanje 628:**

Kada je brojilo u stanju 628, pet izlaza je u stanju 1, te pet izlaza je u stanju 0. Ulazi u I vrata su: 6B, 5C, 5B, 5A, 4D', 4C, 4B', 4A'. Sklop koji služi za dekodiranje stanja 628 prikazan je na slici 3.11.



Slika 3.14 Dekodiranje broja 628.

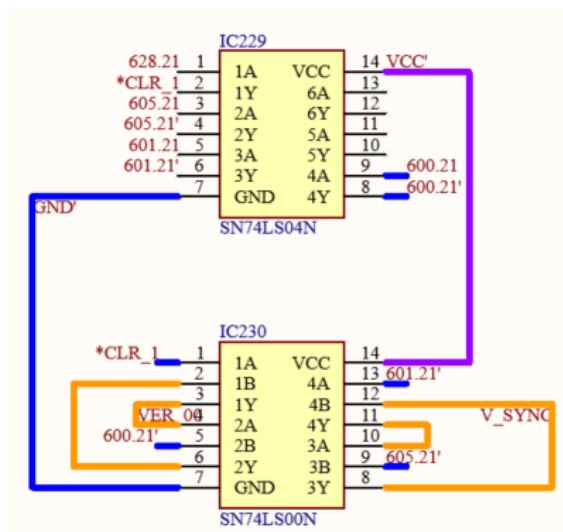
$$628.21 = 4A' * 4B' * 4C * 4D' * 5A * 5B * 5C * 6B$$

Posljednji korak je generiranje dvaju signala: signal koji će biti u stanju 1 kada je potrebno slati piksele monitoru (VER\_00), te signal za vertikalnu sinkronizaciju (V\_SYNC). Prema tablici 3.2, VER\_00 treba biti u stanju 1 između stanja 0 i 600, a V\_SYNC između stanja 601 i 605. Za to su potrebna dva S-R bistabila čiji su izlazi spomenuti signali. Bistabil odgovoran za VER\_00 potrebno je setirati u stanju



Poglavlje 3. Princip rada sklopa

0, a resetirati u stanju 600, a bistabil odgovoran za V\_SYNC je potrebno setirati u stanju 601, a resetirati u stanju 605. Sklop kojim se to postiže prikazan je na slici 3.15.



Slika 3.15 Sklop za generiranje signala VER\_00 i V\_SYNC.

# Poglavlje 4

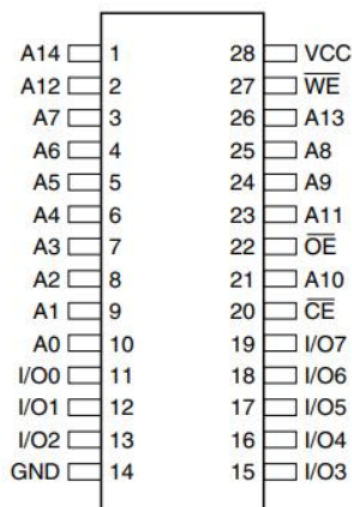
## Generiranje RGB signala

U ovom poglavlju objašnjeno kako se, koristeći memoriju, prikazuje slika na monitoru, te je opisan način na koji su podaci (pikseli) pohranjeni u memoriju. EEPROM (eng. *Electrically Erasable Programmable Read-Only Memory*) pripada ROM (eng. *Read-Only Memory*) skupini memorija. ROM memorija trajno ili polu-trajno pohranjuje podatke, čak i nakon što se izgubi napajanje. EEPROM memorija je također programibilna, te ju je moguće brisati. EEPROM memoriju je moguće reprogramirati jako velik broj puta prije nego dođe do oštećenja sklopa (10000 do 100000 puta [13]).

### 4.1 AT28C256

EEPROM memorija korištena u ovom završnom radu je AT28C256 proizvođača *AT-MEL*. Veličina memorije je 256 kb (32768 riječi \* 8 bitova) [14]. Na slici 4.1 prikazan je raspored priključnica memorije.

#### Poglavlje 4. Generiranje RGB signala



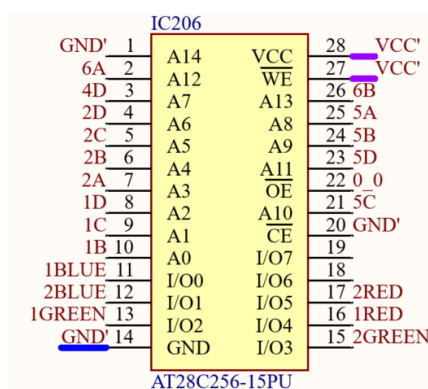
Slika 4.1 AT28C256 (Izvor[14]).

Priključnica  $VCC$  spaja se na napon napajanja, a  $GND$  na uzemljenje. Budući da je u radu sklopa potrebno samo čitati, potrebno je priključnicu  $\overline{CE}$  (*Chip Enable* - omogućava rad sklopa) uzemljiti, a priključnicu  $\overline{WE}$  (*Write Enable*) je potrebno spojiti na napon napajanja. Priključnice označene s  $A0, A1, \dots, A14$  su adresni ulazi, a  $I/O0, \dots, I/O7$  su podatkovni izlazi. Pomoću adresnih ulaza moguće je pristupiti memoriji zapisanoj na određenoj adresi. Zapis slike u memoriji opisan je u poglavlju 4.3. Slika se u horizontalnom smjeru prikazuje sve dok je  $HOR\_00$  aktivan, a u vertikalnom dok je  $VER\_00$  aktivan. Izlaze iz oba brojala potrebno je dovesti na adresne ulaze.

Međutim, zbog toga što je početna rezolucija monitora  $800 \times 600$ , a smanjena je na rezoluciju  $200 \times 600$ , slika koja bi se pojavila na monitoru bila bi jako izdužena u odnosu na sliku koja je pohranjena u memoriji. Također, problem je i to što AT28C256 ima samo 15 adresnih ulaza, a izlaza iz oba brojala ima sveukupno 18 (9 za horizontalno brojilo, 9 za vertikalno), to znači da je potrebno zanemariti neke od izlaza iz brojila, ali je također treba postići zadovoljavajuću rezoluciju koja će što bolje prikazati sliku. Budući da je početna rezolucija  $800 \times 600$ , znači da je potrebno rezoluciju nakon promjene održati što bliže omjeru  $800 : 600 = 4 : 3$ . Taj je omjer moguće održati ako se rezolucija smanji na  $100 \times 75$ . To se može ostvariti ako se na

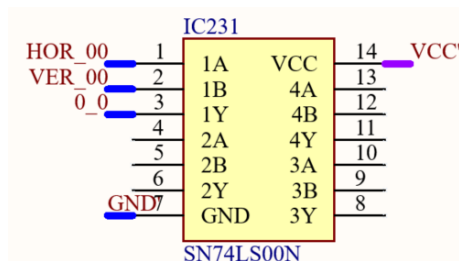
## Poglavlje 4. Generiranje RGB signala

adresne ulaze EEPROM-a dovedu 7 izlaza iz horizontalnog i 7 izlaza iz vertikalnog brojila. Sedam ulaza je potrebno jer je  $2^7 = 128 > 100 > 75$ . Neiskorišteni MSB adresni ulaz EEPROM-a A14 spojen je na masu. Kod horizontalnog brojila zane-mareni su MSB i LSB, a kod vertikalnog 3 LSB izlaza. Na slici 4.2 prikazano je na koji je način memorija povezana s ostatkom sklopa.



Slika 4.2 AT28C256 u Altium Designeru.

Priključnicu  $\overline{OE}$  (*Output Enable* - postoji izlaz iz sklopa) potrebno je uzemljiti kada se slika prikazuje. Slika se prikazuje kada su signali HOR\_00 i VER\_00 aktivni, stoga ih je preko I vrata potrebno spojiti na ovaj ulaz EEPROM-a. To je prikazano slikom 4.3.



Slika 4.3  $\overline{OE}$ .

RGB (*red, green and blue*) model boja je model kojim se pomoću tri boje (crvena, zelena, plava) mogu dobiti sve ostale boje. Broj boja koje je moguće dobiti ograničen je brojem bitova kojim se opisuje svaka od tri boje. Budući da AT28C256 ima 8

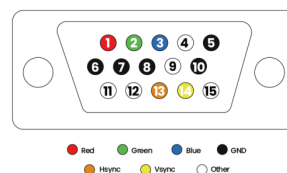
## Poglavlje 4. Generiranje RGB signala

izlaza, pomoću jednog bajta potrebno je opisati sve tri boje. Kako bi svaka boja imala jednak broj nijansi, dva izlaza iz EEPROM-a bit će uvijek u stanju 0, dok će ostalih šest služiti za prikaz tri boje. U ovom slučaju, svaka tri boje opisana je s dva bita, što znači da svaka boja ima 4 nijanse. Ukupan broj boja je  $2^2 * 2^2 * 2^2 = 2^6 = 64$ . U slučaju da su sve boje opisane s '00', na monitoru će se prikazati crni piksel, a kada su sve boje opisane s '11' na monitoru se prikazuje bijeli piksel.

## 4.2 VGA standard

Nakon što je definirano koji izlazi predstavljaju određenu boju, potrebno je te signale dovesti na ulaz VGA konektora koji će se povezati na monitor.

Na slici 4.4 prikazane su priključnice VGA konektora, a u tablici 4.1 ukratko je opisana funkcija svake priključnice. U ovom završnom radu, koriste se priključnice označene s 1, 2, 3, 5, 6, 7, 8, 9, 10, 13 i 14. Problem do kojeg dolazi je to što AT28C256 ima 5 V izlaze, a kako je vidljivo u tablici 4.1, napon je potrebno ograničiti na 0.7 V. To je moguće ostvariti naponskim djelilom. Također, može se vidjeti kako prve tri priključnice imaju unutarnji otpor od  $75 \Omega$ .



Slika 4.4 VGA priključnice (Izvor:[15]).

Poglavlje 4. Generiranje RGB signala

Tablica 4.1 VGA (Izvor: [16]).

Pin	Name	Description
1	RED	Red Video (75 ohm, 0.7 V p-p)
2	GREEN	Green Video (75 ohm, 0.7 V p-p)
3	BLUE	Blue Video (75 ohm, 0.7 V p-p)
4	RES	Reserved
5	GND	Ground
6	RGND	Red Ground
7	GGND	Green Ground
8	BGND	Blue Ground
9	+5V	+5 VDC
10	SGND	Sync Ground
11	ID0	Monitor ID Bit 0
12	SDA	DDC Serial Data Line
13	HSYNC or CSYNC	Horizontal Sync(or Composite Sync)
14	VSYNC	Vertical Sync
15	SCL	DDC Data Clock Line

Budući da svaka od osnovnih boja (crvena, zelena, plava) ima četiri nijanse, potrebno je ostvariti četiri naponske razine. Te razine su: 0 V, 0.233 V, 0.467 V i 0.7 V. Za 0 V obje priključnice koje pripadaju određenoj boji su u stanju 0. Shema otporničkog naponskog djelila je prikazana na slici 4.5, iz čega slijedi:

$$V_1 = I * (R_1 + R_2) \quad (4.1a)$$

$$V_{IZL} = I * R_2 \quad (4.1b)$$

$$I = \frac{V_1}{R_1 + R_2} \quad (4.1c)$$

$$I = \frac{V_{IZL}}{R_2} \quad (4.1d)$$

Izjednačavanjem izraza (4.1c) i (4.1d) dobije se:

$$\frac{V_1}{R_1 + R_2} = \frac{V_{IZL}}{R_2} \quad (4.2)$$

#### Poglavlje 4. Generiranje RGB signala

Iz ove jednadžbe možemo izraziti otpor  $R_1$ :

$$V_1 * R_2 = V_{IZL} * (R_1 + R_2) \quad (4.3a)$$

$$V_1 * R_2 = V_{IZL} * R_1 + V_{IZL} * R_2 \quad (4.3b)$$

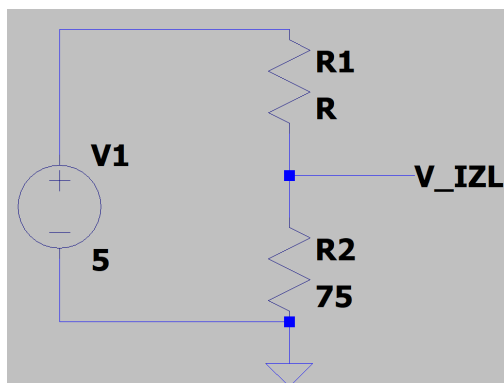
$$V_{IZL} * R_1 = V_1 * R_2 - V_{IZL} * R_2 \quad (4.3c)$$

$$R_1 = R_2 * \frac{V_1 - V_{IZL}}{V_{IZL}} \quad (4.3d)$$

U slučaju da je  $V_{IZL} = 0.233V$ , dobivamo  $R_1 = 1532.14\Omega$ . Za  $V_{IZL} = 0.467V$  dobije se  $R_1 = 728.57\Omega$ , a za  $V_{IZL} = 0.7V$ ,  $R_1 = 460.71\Omega$ . Budući da ne postoje otpori ovakvih vrijednosti, u završnom radu su korišteni otpori od  $1500\Omega$  i  $680\Omega$ . Također, spajanjem ta dva otpora u paralelu dobije se:

$$R = \frac{1500 * 680}{1500 + 680} = 467.89\Omega \quad (4.4)$$

Dobiveni otpor je po vrijednosti dovoljno blizu izračunatom otporu od  $460.71\Omega$ . Pomoću ovih šest otpornika (tri otpornika od  $1500\Omega$  i tri od  $680\Omega$ ) ostvaren je D/A pretvornik.



Slika 4.5 Naponsko djelilo u programu LTSpice.

## 4.3 Zapis podataka u EEPROM memoriju

Naposljetku, potrebno je napisati kod pomoću koje ćemo uobičajenu 8 - bitnu sliku pretvoriti u 2 - bitnu sliku, te ostaviti prazna mjesta na kraju svakog retka, tj. pripremiti podatke koje ćemo spremiti u EEPROM memoriju. Kod je napisan u Python-u te je prikazan na slici 4.6.

```
1  from PIL import Image
2  import math
3
4  def extract_color_bits(red, green, blue):
5      byte = math.floor(blue/64) + 4*math.floor(green/64) + 16*math.floor(red/64)
6      return byte
7
8
9  def process_image(image_path, output_path):
10     # Učitavanje slike
11     image = Image.open(image_path)
12     if image.mode != 'RGB':
13         image = image.convert('RGB')
14     width, height = image.size
15
16     # Otvaranje binarne datoteke za pisanje
17     with open(output_path, 'wb') as output_file:
18         # Pisanje bajtova u binarnu datoteku
19         for y in range(height):
20             for x in range(width):
21                 pixel = image.getpixel((x, y))
22                 if isinstance(pixel, int):
23                     red, green, blue = pixel, pixel, pixel
24                 else:
25                     red, green, blue = pixel[:3]
26                 byte = extract_color_bits(red, green, blue)
27                 output_file.write(byte.to_bytes(1, byteorder='big'))
28             for i in range(128 - width):
29                 output_file.write(b'\x00')
30
31     print("Izrada binarne datoteke je završena.")
32
33 # Primjer korištenja
34 input_image_path = 'C:\Users\Mateo\Desktop\završni\slike\riteh.png'
35 output_file_path = 'C:\Users\Mateo\Desktop\završni\izlazna_datoteka_novo.bin'
36 process_image(input_image_path, output_file_path)
```

Slika 4.6 Kod.



## Poglavlje 4. Generiranje RGB signala

Prva linija koda `from PIL import Image` omogućava rad sa slikama u Pythonu. Drugom linijom koda omogućava se upotreba određenih matematičkih funkcija. Prvo je definirana funkcija `process_image` koja ima dvije ulazne varijable: `image_path` - lokacija slike i `output_path` - lokacija izlazne datoteke.

Pomoću `image = Image.open(image_path)` otvara se slika te se sprema u varijablu `image`, a pomoću `image = image.convert('RGB')` se slika konvertira u RGB format. Nakon toga, pomoću naredbe `with open(output_path, 'wb') as output_file:` otvara se izlazna datoteka u koju se zapisuje heksadekadski zapis slike. Sljedeći korak je pomoću dvije ugniježdene `for` petlje dobiti 2D - polje kod kojega svaki element predstavlja jedan piksel. `for` petlje ograničene su širinom i visinom slike, a ti su podaci dobiveni preko: `width, height = image.size`.

Zatim se pomoću naredbe `pixel = image.getpixel((x, y))` pristupa pojedinačnom pikselu. Budući da je svaka od osnovnih boja u RGB formatu predstavljena s 8 bitova (vrijednosti od 0 do 255), a u ovom završnom radu s 2 bita, potrebno je smanjiti broj bitova. To se postiže unutar funkcije `extract_color_bits`. Pomoću `byte = math.floor(blue/64) + 4*math.floor(green/64) + 16*math.floor(red/64)` postiže se opis piksela jednim bajtom. Prva dva bita bit će 0, crvena komponenta piksela bit će opisana 3. i 4. bitom, zelena 5. i 6., a plava pomoću posljednja dva bita.

Pomoću `output_file.write(byte.to_bytes(1, byteorder='big'))` ostvaruje se zapis u binarnu datoteku. Također potrebno je, unutar svakog reda, zadnjih 28 piksela zapisati kao 00 (crna boja). Binarna se datoteka može otvoriti u programu HXD. Dio heksadekadskog zapisa slike prikazan je na slici 4.7.

## Poglavlje 4. Generiranje RGB signala

```

izlazna_datoteka_novo.bin
Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000010 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000020 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000030 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000040 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000050 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000060 3F 3F 3F 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????.....
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000080 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000090 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000000A0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000000B0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000000C0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000000D0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000000E0 3F 3F 3F 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 ????.....
000000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000100 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000110 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000120 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000130 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000140 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000150 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000160 3F 3F 3F 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 ????.....
00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000180 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000190 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000001A0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000001B0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000001C0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000001D0 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
000001E0 3F 3F 3F 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 ????.....
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000200 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000210 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000220 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000230 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000240 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000250 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000260 3F 3F 3F 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 ????.....
00000270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000280 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????
00000290 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F ?????????????????

```

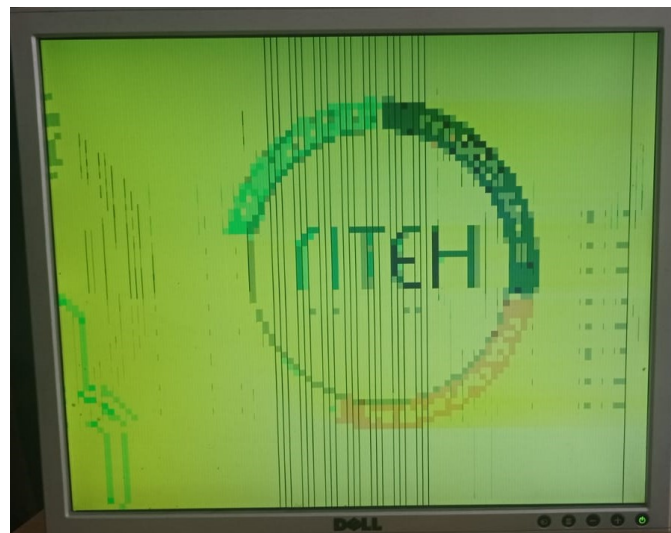
Slika 4.7 Binarna datoteka.

Svaki piksel opisan je s dvije heksadekadske znamenke, što odgovara zapisu od osam bitova. Ako se uzme za primjer prvi piksel, zapisan s 3F, njegov binarni zapis je: 0011 1111. Kako je u kodu definirano, crvena komponenta piksela opisana je 3. i 4. bitom, zelena 5. i 6., a plava pomoću posljednja dva bita. Ovime se može zaključiti da je taj piksel bijele boje. Sveukupno, ovaj zapis zauzima  $128 * 75 = 9600B = 9.6kB$ . Budući da AT28C256 ima memoriju od 32kB (256kb) ([14]), potrebno je ostatak memorije popuniti nulama. Tako obrađena slika prikazana je na slici 4.8, a konačni rezultat, prikaz na monitoru je prikazan na slici 4.9.

Poglavlje 4. Generiranje RGB signala



Slika 4.8 Slika koja se želi prikazati (Izvor: [17]).



Slika 4.9 Slika na VGA monitoru.

# Poglavlje 5

## Zaključak

Korištenjem osnovnih logičkih operacija ostvaren je prikaz slike na VGA monitoru. Frekvencija generatora takta povezana je s time koliko puta se slika *refresh*-a u sekundi, te s rezolucijom monitora. Generator takta u ovom slučaju je kristalni oscilator frekvencije 10 MHz. Rezolucija slike koja se prikazuje je 100x75, što u velikoj većini primjena ne bi bila zadovoljavajuća rezolucija. Rezoluciju bi se moglo povećati korištenjem EEPROM memorije koja ima više adresnih ulaza od memorije koja se koristi, te korištenjem generatora takta veće frekvencije.

Isti sklop je moguće ostvariti korištenjem manjeg broja integriranih sklopova. Na primjer, umjesto 8 SN74LS08N sklopova i jednog SN74LS04N sklopa koji se koristi kod prepoznavanja određenog broja na izlazu iz brojila, može se koristiti SN74LS30N (8-ulazni NILI sklop).

Kako bi se postigao još elegantniji izgled sklopa i smanjila njegova veličina, moguće ga je napraviti u jednom od programa za izradu tiskanih pločica (eng. PCB (*Printed Circuit Board*)).

# Bibliografija

- [1] Encyclopedia Britannica, “VGA,” ,srpanj 2023. , s Interneta, <https://www.britannica.com/technology/VGA>
- [2] HMIVision, “Why Do Industrial Computers Generally Use VGA as a Display Interface?” ,srpanj 2023. , s Interneta, <https://hmivision.com/why-do-industrial-computers-generally-use-vga-as-a-display-interface/>
- [3] S. Paunović, *Digitalna elektronika*. Školska knjiga.
- [4] All About Circuits, “Boolean algebra basics-an overview of boolean logic,” 2022, ,srpanj 2023. , s Interneta, <https://www.allaboutcircuits.com/technical-articles/boolean-basics/>
- [5] ElectronicsHub, “Sr flip flop-designing using gates and applications,” 2021, ,srpanj 2023. , s Interneta, <https://www.electronicshub.org/sr-flip-flop-design-with-nor-and-nand-logic-gates/>
- [6] Wikimedia Commons. , s Interneta, [https://commons.wikimedia.org/wiki/File:NAND\\_Gated\\_SR\\_Latch.png](https://commons.wikimedia.org/wiki/File:NAND_Gated_SR_Latch.png) ,srpanj 2023.
- [7] “SN74LS161AN,” Texas Instruments, 1988. , s Interneta, <https://www.ti.com/product/SN74LS161A>
- [8] “SN74LS04N,” Texas Instruments, 2004. , s Interneta, <https://www.ti.com/lit/ds/symlink/sn74ls04.pdf>
- [9] “SN74LS00N,” ON Semiconductor, 1999. , s Interneta, <https://pdf1.alldatasheet.com/datasheet-pdf/view/12606/ONSEMI/SN74LS00N.html>
- [10] “SN74LS08N,” Motorola Inc. , s Interneta, <https://pdf1.alldatasheet.com/datasheet-pdf/view/5638/MOTOROLA/SN74LS08N.html>
- [11] DigiKey, “Vga timing,” , lipanj 2023. , s Interneta, <https://forum.digikey.com/t/vga-controller-vhdl/12794>

## *Bibliografija*

- [12] “Vga,” , lipanj 2023. , s Interneta, <http://tinyvga.com/vga-timing>
- [13] R. Awati, “Eeprom (electrically erasable programmable read-only memory),” 2022., ,lipanj 2023. , s Interneta, <https://www.techtarget.com/whatis/definition/EEPROM-electrically-erasable-programmable-read-only-memory>
- [14] “AT28C256 EEPROM: Datasheet, pinout, applications.” Kynix, 2022., ,srpanj 2023. , s Interneta, <https://www.kynix.com/components/AT28C256-EEPROM-DATASHEET-PINOUT-APPLICATIONS.html#at28c256-pinout>
- [15] Ugreen, “A complete guide to vga: What you need to know,” 2021, ,lipanj 2023. , s Interneta, <https://blog.ugreen.com/vga-connector/>
- [16] B. Eater. Let’s build a video card! , s Interneta, <https://eater.net/vga> ,lipanj 2023.
- [17] “Sveučilište u Rijeci - Tehnički fakultet,” Facebook, ,srpanj 2023. , s Interneta, <https://www.facebook.com/TehnickiFakultetSveucilistaURijeci/>

# Popis slika

2.1	Logička vrata I - simbol (Izvor:[4]). . . . .	5
2.2	Logička vrata ILI - simbol (Izvor [4]). . . . .	6
2.3	Logička vrata NE - simbol (Izvor: [4]). . . . .	6
2.4	Logička vrata NI - simbol (Izvor: [4]). . . . .	7
2.5	Logička vrata NILI - simbol (Izvor: [4]). . . . .	7
2.6	S-R bistabil (Izvor: [5]). . . . .	9
2.7	Sinkroni S-R bistabil (Izvor: [6]). . . . .	10
2.8	SN74LS161AN (Izvor: [7]). . . . .	11
2.9	SN74LS04N (Izvor: [8]). . . . .	11
2.10	SN74LS04N - logički dijagram (Izvor: [8]). . . . .	11
2.11	SN74LS00N (Izvor: [9]). . . . .	12
2.12	SN74LS08N Izvor: [10]). . . . .	12
3.1	Prikazivanje slike na monitoru - vremenski dijagram [11]. . . . .	14
3.2	Shematski prikaz brojila (horizontalni smjer) u Altium Designeru. . .	16
3.3	Shematski prikaz spajanja dva sklopa SN74LS04N (horizontalni smjer). .	17
3.4	Dekodiranje stanja 200. . . . .	18
3.5	Dekodiranje broja 210. . . . .	18
3.6	Dekodiranje stanja 242. . . . .	19
3.7	Prepoznavanje broja 264. . . . .	20

## Popis slika

3.8	Sklop za generiranje signala HOR_00 i H_SYNC . . . . .	21
3.9	Shematski prikaz brojila (vertikalni smjer) u Altium Designeru. . . .	22
3.10	Shematski prikaz spajanja dva sklopa SN74LS04N (vertikalni smjer). .	22
3.11	Dekodiranje stanja 600. . . . .	22
3.12	Dekodiranje broja 601. . . . .	23
3.13	Dekodiranje broja 605. . . . .	24
3.14	Dekodiranje broja 628. . . . .	24
3.15	Sklop za generiranje signala VER_00 i V_SYNC. . . . .	25
4.1	AT28C256 (Izvor[14]). . . . .	27
4.2	AT28C256 u Altium Designeru. . . . .	28
4.3	$\overline{OE}$ . . . . .	28
4.4	VGA priključnice (Izvor:[15]). . . . .	29
4.5	Naponsko djelilo u programu LTspice. . . . .	31
4.6	Kod. . . . .	32
4.7	Binarna datoteka. . . . .	34
4.8	Slika koja se želi prikazati (Izvor: [17]). . . . .	35
4.9	Slika na VGA monitoru. . . . .	35



# Popis tablica

2.1	Tablica stanja logičkih vrata I. . . . .	7
2.2	Tablica stanja logičkih vrata ILI. . . . .	7
2.3	Tablica stanja logičkih vrata NE. . . . .	8
2.4	Tablica stanja logičkih vrata NI. . . . .	8
2.5	Tablica stanja logičkih vrata NILI. . . . .	8
2.6	Tablica stanja S-R bistabila. . . . .	9
3.1	Intervali unutar jednog reda za rezoluciju 800x600 i frekvenciju 40 MHz (Izvor: [12]). . . . .	15
3.2	Intervali unutar jednog stupca za rezoluciju 800x600 i frekvenciju 40 MHz (Izvor: [12]). . . . .	15
3.3	Intervali unutar jednog reda za rezoluciju 200x600 i frekvenciju 10 MHz. . . . .	15
4.1	VGA (Izvor: [16]). . . . .	30

# Sažetak

U ovom je završnom radu opisano kako se upotrebom jednostavnih integriranih krugova, prikazuje slika na VGA monitoru. Pomoću digitalnih brojila određeno je u kojim je intervalima potrebno slati podatke prema VGA standardu. Rezolucija slike ovisi o broju adresnih ulaza EEPROM memorije. Izlazi brojila spojeni su na adresne ulaze EEPROM memorije, u kojoj je na pripadnoj adresi zapisan podatak o boji piksela. Prikaz piksela ostvaruje se korištenjem naponskog djelila. Napon na izlazu naponskih djelila za crvenu, zelenu i plavu boju dovodi se na pripadajuće VGA ulaze, uz signale za horizontalnu i vertikalnu sinkronizaciju. Svaka boja je u EEPROM memoriji opisana sa dva bita, što je također limitirajući faktor koji je omogućio prikaz slika na samo 64 boje.

***Ključne riječi*** — VGA, brojilo, EEPROM, rezolucija

## Abstract

In this final thesis, it is described how the use of simple integrated circuits allows us to display a picture on a VGA monitor. Digital counters determine the intervals in which data needs to be sent according to the VGA standard. Resolution of a picture depends on the number of address pins of the EEPROM memory. The outputs of the counters are connected to the address inputs of the EEPROM that is used. Every address contains data about the color of a pixel at a specific location. Display of a pixel is achieved with the use of a voltage divider. The voltage at the output of the voltage dividers for red, green, and blue colors is connected to the corresponding VGA inputs, along with the horizontal and vertical synchronization signals. Each color in the EEPROM memory is represented by two bits, which is a limiting factor that enables the display of images in only 64 colors.

***Keywords*** — VGA, counter, EEPROM, resolution