

VEŽBA 2

Upoznavanje sa VGA spregom

Potrebno predznanje

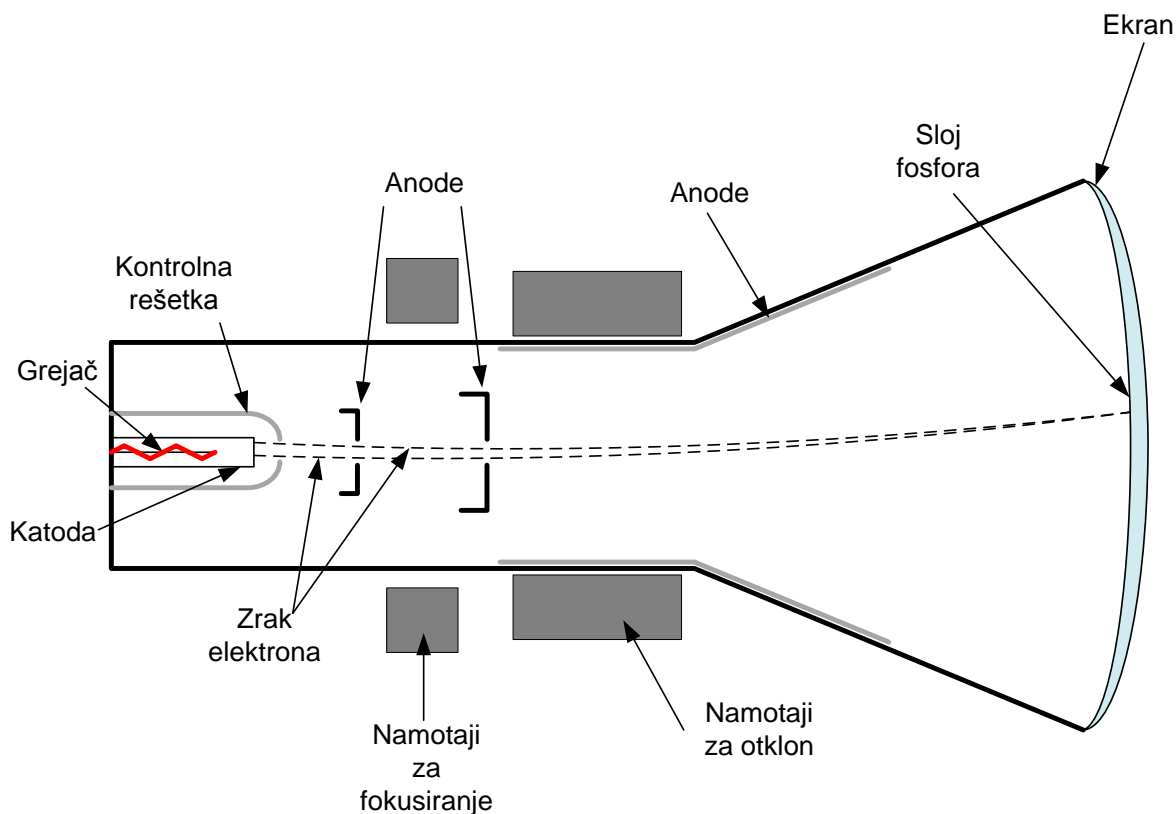
- Urađena vežba 1

Cilj vežbe

- Upoznavanje sa VGA spregom i načinom generisanja upravljačkih signala VGA sprege.
- Upoznavanje sa memorijski mapiranom grafikom.
- Upoznavanje sa tekstualnim grafičkim režimom.
- Podsećanje na realizaciju upravljačkih struktura u VHDL-u.

Ekran sa katodnom cevi

Ekran sa katodnom cevi je vakumska cev koja sadrži elektronski top, ekran na koji je nanesen sloj fosfora i namotajima za fokusiranje i usmeravanje snopa elektrona. Elektroni koji izleću iz elektronskog topa se prvo fokusiraju u jedan vrlo uzak snop. Snop se potom usmerava, po horizontali i vertikali, sa dva para namotaja. Tako usmereni elektroni pogađaju fosfor nanesen na stakleni ekran. Fosfor se koristi jer ima osobinu da emituje svetlost (fotone) kada se pobudi elektronima. Na taj način se iscrtava jedna tačka na ekranu. Presek ekrana sa katodnom cevi je dat na slici 2.



Slika 1: Presek ekrana sa katodnom cevi

Ekrani koji prikazuju sliku u boji koriste tri elektronska topa, za tri boje: crvenu, zelenu i plavu. Da bi se izbeglo mešanje snopova različitih boja ispred ekrana, sa unutrašnje strane, se postavlja rešetka. Oblik i svojstva rešetke zavise od proizvođača do proizvođača.

VGA sprega

U ovom poglavlju će biti objašnjeni principi funkcionisanja VGA (eng. Video Graphics Array) ekrana. Poglavlje će se bazirati na standardnoj VGA rezoluciji od 640x480 tačaka. Izneseni principi se mogu lako primeniti na bilo koju drugu od podržanih rezolucija: 800x600, 1024x768, 1152x864 i 1280x1024 koristeći tabelu 1, u kojoj su navedeni svi bitni parametri za svaku rezoluciju. Različite rezolucije u VHDL kodu se odabiraju prosleđivanjem različitih vrednosti parametra *resolution_type* modulu VGA unutar vrha hijerarhije. Na osnovu ovog parametra se određuju parametri na osnovu kojih će se praviti kontrolni signali i koji će od DCM modula biti instanciran.

Radi jednostavnije simulacije sistema postoji i rezolucija 64x48 koja je podržana samo u simulaciji.

Table 1: Parametri VGA kontrolnih signala za sve podržane rezolucije

Rezolucija	Frekvencija takta (MHz)	Hsync parametri (tačke)	front porch	sync puls	back porch	Vsync parametri (linije)	front porch	sync puls	back porch	Ostali parametri	Frekvencija horizontalne sinhronizacije (kHz)	Frekvencija vertikalne sinhronizacije (Hz)	Frekvencija osvežavanja slike (Hz)	Ukupan broj tačaka po horizontali	Ukupan broj tačaka po vertikali	vrednost parametra resolution_type
64x48	25.17		2	2	2		2	2	2		-	-	-	-	-	0
640x480	25.17		16	96	40		11	2	31		31.46	60.04	60.04	800	524	1
800x600	50		56	120	64		37	6	23		48.08	72.19	72.19	1040	666	2
1024x768	75		24	136	144		3	6	29		56.48	70.07	70.07	1328	806	3
1152x864	108		64	128	256		1	3	32		67.5	75	75	1600	900	4
1280x1024	108		48	112	248		1	3	38		63.98	60.02	60.02	1688	1066	5

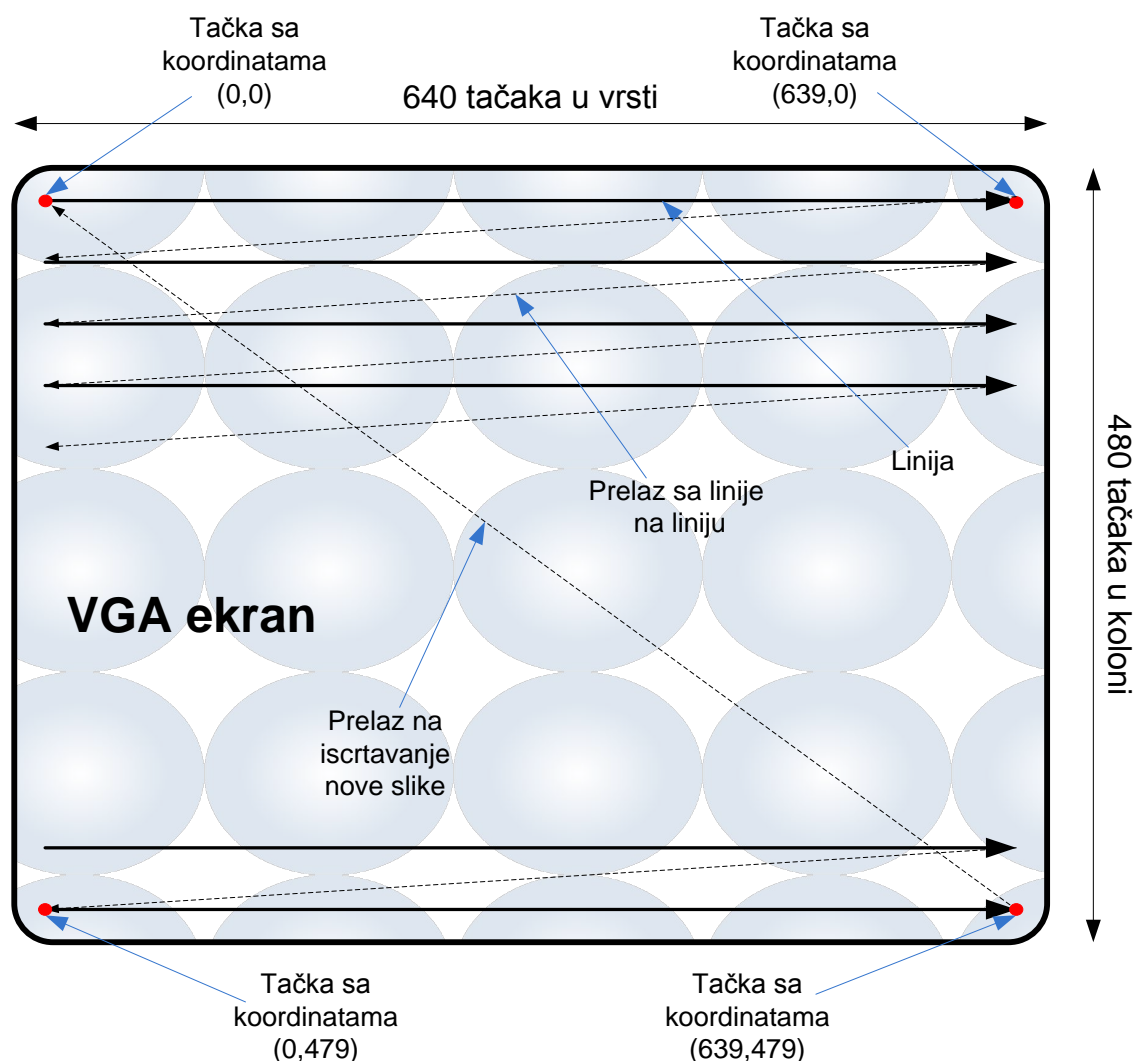
Formiranje slike

Slika na ekranu se formira iscrtavanjem pojedinih tačaka na ekranu. Formiranje slike počinje u gornjem levom uglu iscrtavanjem tačke sa koordinatama (0,0), iscrtavanje se nastavlja do kraja vrste završno sa tačkom sa koordinatama (639,0) i prelazi u narednu vrstu. Kada se iscrta i tačka donjem desnom uglu slike, tačka sa koordinatama (639,479), završeno je iscrtavanje jedne slike i prelazi se na iscrtavanje tačke sa koordinatama (0,0) sledeće slike. Slika 2 prikazuje proces formiranja slike.

Signali VGA sprege

VGA sprema se sastoji od 5 signala: 2 digitalna i 3 analogna. Digitalni signal su signali vertikalne i horizontalne sinhronizacije (u daljem tekstu Vsync i Hsync), a analogni signali opisuju boju svake pojedine tačke na ekranu.

Boja svake tačke na ekranu se formira kombinovanjem tri boje: crvene, zelene i plave (eng. RGB). Svaka od tri boje je predstavljena sa 8 bita, odnosno, boja svake tačke se opisuje sa 24 bita (na primer: 0x00 odsustvo boje, 0xFF pun intenzitet, tj. ako su sve tri boje 0x00 onda je taj piksel crn ili ako je jedna boja 0xFF a dve 0x00 onda je taj piksel te boje). Za pretvaranje 8 bita boje u napon od 0V do 0.7V, za svaku od boja, na E2LP ploči je zaduženo integrisano kolo ADV7125 proizvođača Analog Devices.



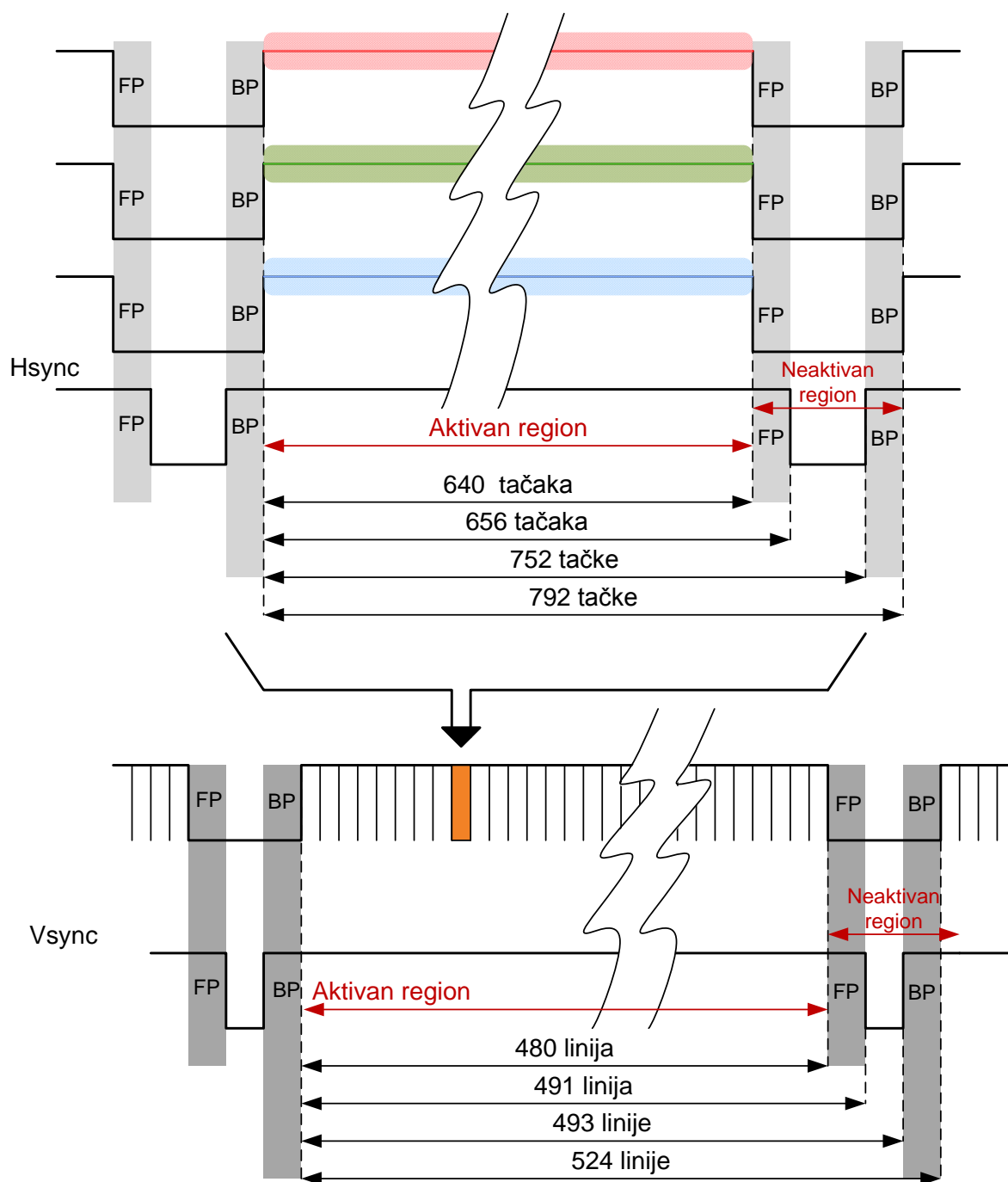
Slika 2: Proces iscrtavanja slike na ekranu

Signali vertikalne i horizontalne sinhronizacije kontrolišu iscrtavanje tačaka na ekranu. Signal Hsync određuje prelazak na sledeću liniju trenutne slike dok signal Vsync određuje prelazak na iscrtavanje nove slike. Na slici (Slika 3) su prikazani signali VGA sprege i njihov međusobni odnos.

Kada je signal Hsync na niskom naponskom nivou prelazi se na iscrtavanje sledeće vrste slike, a kada je signal Vsync na niskom naponskom nivou prelazi se na iscrtavanje naredne slike. Period u kome su ova dva signala na visokom naponskom nivou je podeljen u tri faze:

- zadnji trem (eng. back porch, u daljem tekstu BP),
- aktivan region i
- prednji trem (eng front porch, u daljem tekstu FP).

Kada je signal Hsync unutar aktivnog regiona iscrtavaju se tačke, a kada je Vsync unutar aktivnog regiona iscrtavaju se vrste slike. Kada se signali nalaze u FP ili BP delu ništa se ne iscrtava na ekranu. Pomenuta dva dela signala su uvedeni da bi se omogućilo pomeranje slike po ekranu po vertikali i horizontali.



Slika 3: Signali VGA sprege i njihov međusobni odnos

Tekstualni režim rada

Tekstualni režim podrazumeva prikazivanje karaktera unapred definisanog oblika. Oblici se čuvaju u memoriji ROM tipa. Svaki znak se predstavlja matricom tačaka 8x8. Tabela 2 prikazuje primer realizacije znaka A.

Table 2: Primer realizacije slova A u ROM memoriji

Adresa	Podaci o znaku
000001000	00011000
000001001	00111100

000001010	01100110
000001011	01111110
000001100	01100110
000001101	01100110
000001110	01100110
000001111	00000000

Modul `char_rom` objedinjuje `char_rom_def` modul, u kome se nalazi memorija ROM tipa sa unapred definisanim oblicima znakova (u daljem tekstu CROM), i logiku za određivanje da li trenutna tačka treba da se razlikuje od pozadine. Datoteka `char_rom_def_mem.coe` sadrži inicijalne vrednosti ili podatke o znakovima za ROM. Da bi se odabrala prava vrsta iz matrice koja opisuje znak, adresa za CROM se pravi kombinovanjem adrese znaka i tri bita brojača vrste (trenutne vrednosti vrste).

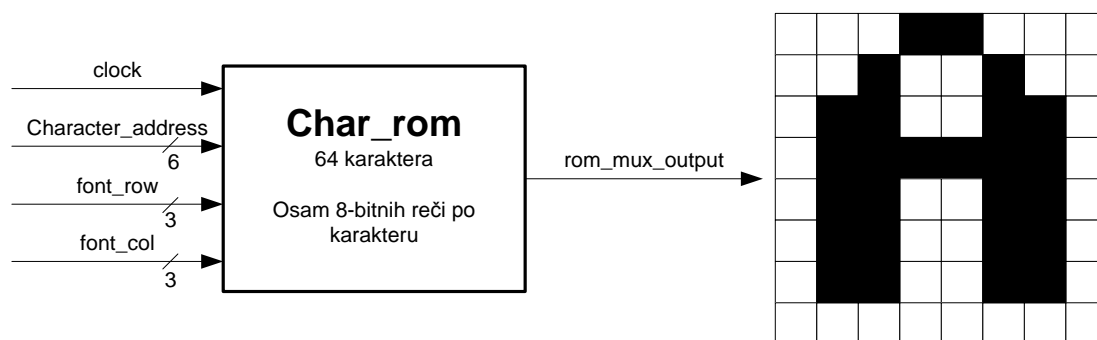
```
rom_address <= character_address & font_row;
```

Može se posmatrati da je `character_address` bazna adresa, a `font_row` otklon (eng. offset). Kada smo odabrali odgovarajuću vrstu matrice koja opisuje znak treba da odaberemo i odgovarajuću kolonu. Odabiranje kolone se obavlja na osnovu tri bita brojača kolone (trenutne vrednosti kolone). Ova tri bita se koriste kao selekcionni ulazi za multiplexer čiji je izlaz informacija da li trenutna tačka treba da se razlikuje od pozadine, vrednost "1", ili ne treba, vrednost „0“.

```
CASE ( font_col ) IS
WHEN "000" => rom_mux_output <= rom_data(7);
WHEN "001" => rom_mux_output <= rom_data(6);
WHEN "010" => rom_mux_output <= rom_data(5);
WHEN "011" => rom_mux_output <= rom_data(4);
WHEN "100" => rom_mux_output <= rom_data(3);
WHEN "101" => rom_mux_output <= rom_data(2);
WHEN "110" => rom_mux_output <= rom_data(1);
WHEN "111" => rom_mux_output <= rom_data(0);
WHEN OTHERS => rom_mux_output <= '0';
END CASE;
```

Adresa karaktera "A" je "000001" binarno. Postavljanjem ove vrednosti na ulaz `character_address_i` modula `char rom`, vrši se selekcija bloka u memoriji u kome se nalaze podaci koji opisuju ovaj karakter, tj 8x8 bita kako je prethodno objašnjeno. Svaki od ovih 64 bita selektuje se uz pomoć `font_row` promenljive za izbor jedne od 8 reči koje se tiču ovog karaktera i `font_col` za izbor jednog bita iz izabrane reči.

Sve što je potrebno da bi se iscrtao znak na ekranu je da se, u zavisnosti od trenutne vrednosti vrste i kolone, modulu `char_rom` pošalje odgovarajuća vrednost adrese znaka definisanog u `char_rom_def_mem.coe` datoteci, sve ostale operacije su realizovane u modulu `char_rom` i skrivene od korisnika.



Slika 4: Šema korišćenja modula char_rom

Ovde treba napomenuti da se prilikom korišćenja neka tri bita brojača vrste i kolone (trenutne vrednosti vrste i kolone tačke koja se trenutno iscrtava) ekran deli na kvadratnu mrežu. Veličina i broj polja mreže zavise od toga koja se tri uzastopna bita brojača vrste i kolone odaberu. Na primer, ako se odaberu najniža tri uzastopna bita (biti 0-2), veličina znaka je 8x8 tačaka, ako se odaberu biti od 1 do 3 veličina znaka je 16x16 i tako dalje. Odabiranjem viših bita brojača veličina znaka se proporcionalno povećava, jer se svaka pojedinačna tačka matrice znaka iscrtava veći broj puta. U kodu je moguće podešavati veličinu znaka dodeljivanjem različitih vrednosti parametru *font_size*. Svi definisani znaci i njihove adrese su navedeni u tabeli Table 3

Adrese znakova se mogu navoditi pojedinačno u kodu ili se mogu postaviti u memoriju RAM tipa. Slika 7 prikazuje blok dijagram sistema u kome se adrese znakova nalaze u dvoprístupnoj memoriji. Ako se adrese znakova smeste u memoriju RAM tipa, umesto navođenja adresa (*char_addr_s*) treba adresirati RAM tj. davati odgovarajuću vrednost *txt_ram_addr_s* signalu. Iako ovo ne izgleda kao velika pogodnost prilikom ispisivanja malog broja karaktera, prilikom ispisivanja kompletnog ekrana je korisno, jer se u memoriju može smestiti kompletan sadržaj. S obzirom da je memorija dvoprístupana, njen sadržaj je moguće menjati.

Table 3: Spisak svih podržanih znakova i njihove adrese

Znak	Adresa (oktalna)	Adresa (dec)	Adresa (hexadec)
@	0	0	0
A	1	1	1
B	2	2	2
C	3	3	3
D	4	4	4
E	5	5	5
F	6	6	6
G	7	7	7
H	10	8	8
I	11	9	9
J	12	10	A
K	13	11	B
L	14	12	C
M	15	13	D
N	16	14	E
O	17	15	F
P	20	16	10
Q	21	17	11
R	22	18	12
S	23	19	13
T	24	20	14
U	25	21	15
V	26	22	16
W	27	23	17
X	30	24	18
Y	31	25	19
Z	32	26	1A

[33	27	1B
□	34	28	1C
]	35	29	1D
□	36	30	1E
□	37	31	1F
Razmak	40	32	20
!	41	33	21
"	42	34	22
#	43	35	23
\$	44	36	24
%	45	37	25
&	46	38	26
\	47	39	27
(50	40	28
)	51	41	29
*	52	42	2A
+	53	43	2B
,	54	44	2C
-	55	45	2D
.	56	46	2E
/	57	47	2F
0	60	48	30
1	61	49	31
2	62	50	32
3	63	51	33
4	64	52	34
5	65	53	35
6	66	54	36
7	67	55	37
8	70	56	38
9	71	57	39
A	72	58	3A
B	73	59	3B
C	74	60	3C
D	75	61	3D
E	76	62	3E
F	77	63	3F

Grafički režim rada

U grafičkom režimu formiranje slike se svodi na proveru pozicije tačke koja se trenutno iscrtava i menjanje vrednosti boja. Pozicija tačke je određena sa dva signala koji izlaze iz VGA modula: *pixel_row_o* i *pixel_column_o*. Ovi signali predstavljaju vrstu i kolonu tačke koja se trenutno iscrtava, a prave se u modulu VGA_SYNC.

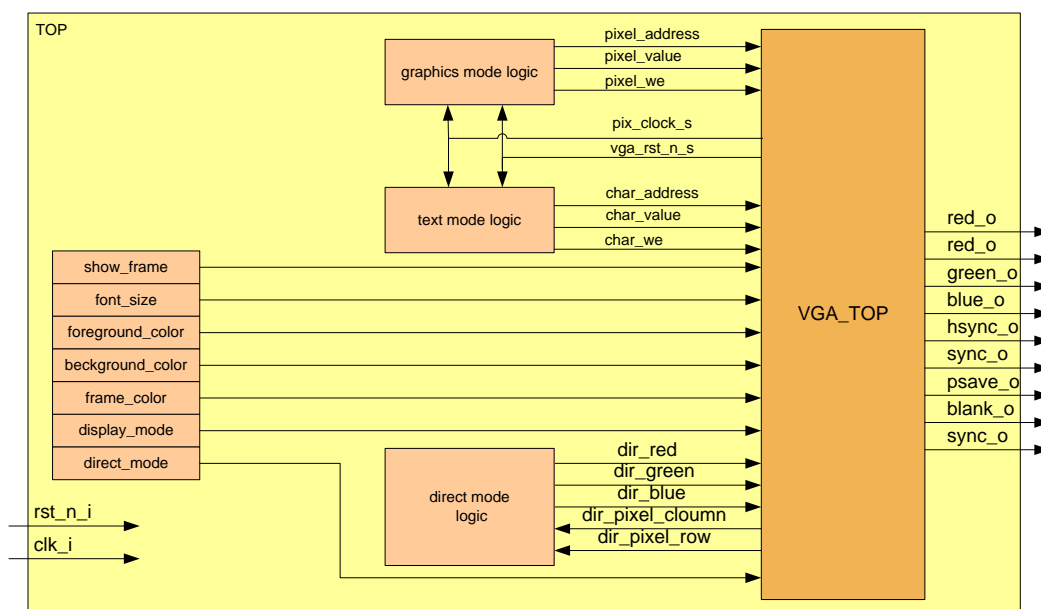
Radi pojednostavljenja, slika se formira u samo dve boje, boja objekta (*foreground_color*) i boja pozadine (*background_color*). Navedene boje se definišu preko odgovarajućih ulaza VGA_TOP modula. Informacija o tome koja se boja iscrtava na određenu poziciju na ekranu je se čuva u grafičkoj memoriji, pri čemu jedan bit u memoriji odgovara jednom pikselu na ekranu. Adresa jednobitne informacije o boji piksela iz memorije se formira na osnovu *pixel_row_o* i *pixel_column_o* signala. Grafička memorija je široka 32 bita i definiše sadržaj 32 uzastopna piksela. U slučaju da želimo da definišemo sadržaj u memoriji za jedan red slike pri rezoluciji 640x480, potrebno je upisati 20 reči (640/32). Definisane kompletne slike bi zahtevalo 480 ovakvih sekvenci upisa.

Implementacija

TOP modul

Slika 5 predstavlja blok dijagram TOP modula koji se nalazi na vrhu hijerarhije. On sadrži:

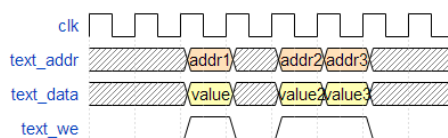
1. VGA_TOP modul koji upravlja VGA spregom i čiji su detalji dati kasnije u tekstu
2. Logiku za iscrtavanje teksta i grafike čiji je razvoj cilj ove vežbe



Slika 5: Blok dijagram TOP modula

Logika koju je potrebno realizovati se sastoji od tri nezavisne celine:

- *Direct mode logic* - predstavlja logiku koju je potrebno realizovati kako bi se definisao objekat (njegova RGB komponenta) na osnovu informacije o trenutnom redu i koloni tačke koja se iscrta (pogledati *Formiranje slike*).
- *Text mode logic* - predstavlja logiku koja upravlja ispisom teksta. Prema postojećoj implementaciji VGA_TOP modula, ispisom teksta se ne upravlja direktno već preko memorije. U okviru VGA_TOP modula realizovana je dvopristupna memorija TEXT_MEM i odgovarajuća logika koja na osnovu trenutne vrste i kolone tačke koja se iscrta adresira memoriju kao što je objašnjeno u poglavlju *Tekstualni režim rada*. Na korisniku je da definiše sadržaj memorije koristeći adresu (*text_addr*), podatak (*text_data*) i signal dozvole upisa (*text_we*).

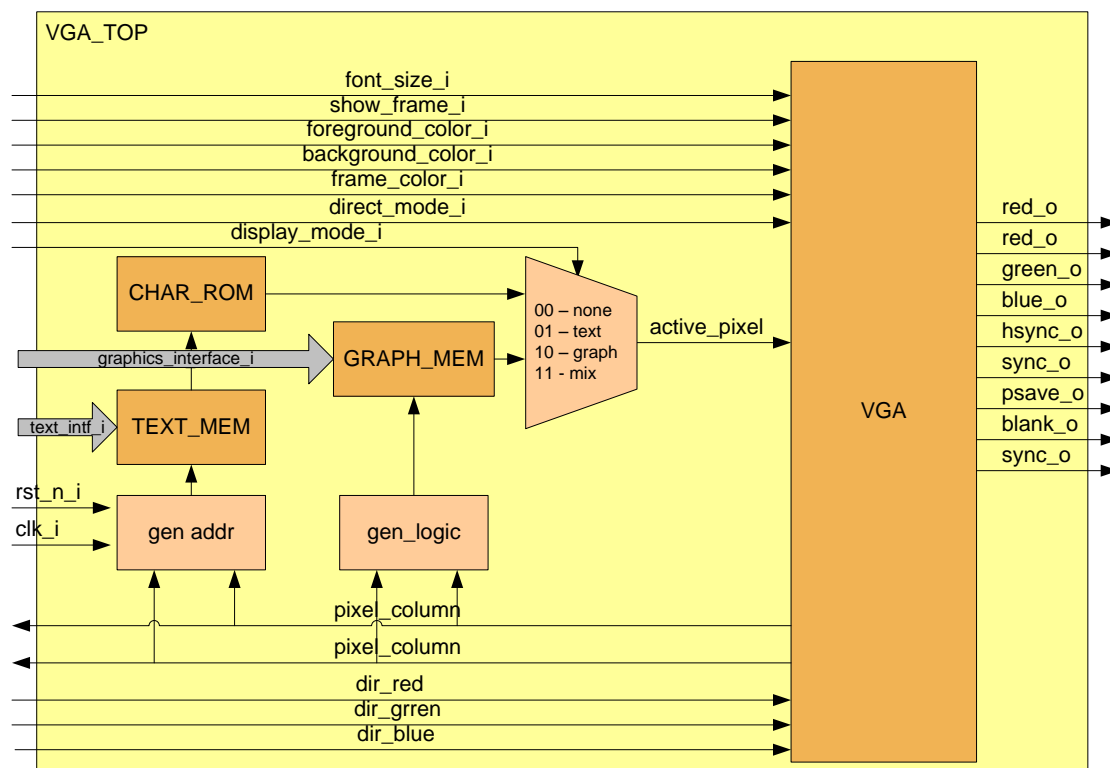


Slika 6: primer upisa u memoriju

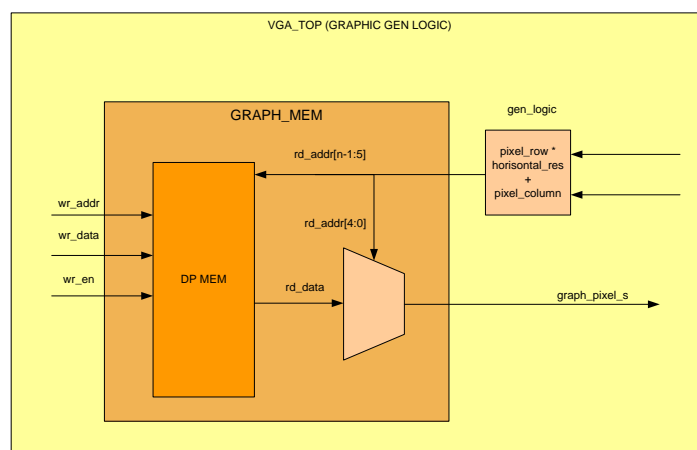
- *Graphics mode logic* - predstavlja logiku koju je potrebno realizovati za iscrtavanje slike. Kao i kod tekstualnog načina rada, iscrtavanje slike se realizuje posredno preko memorije Slika 8. Na korisniku

je da definiše korektan sadržaj grafičke memorije GRAPH_MEM uzimajući u obzir izabranu rezoluciju koja definiše veličinu memorije, kao i širinu podataka za pisanje (32 bita) prema dijagramu koji prikazuje Slika 6 samo sa signalima *graph_addr*, *graph_value*, *graph_we*.

Blok dijagram VGA_TOP modula je dat na slici .



Slika 7: Blok dijagram VGA_TOP modula



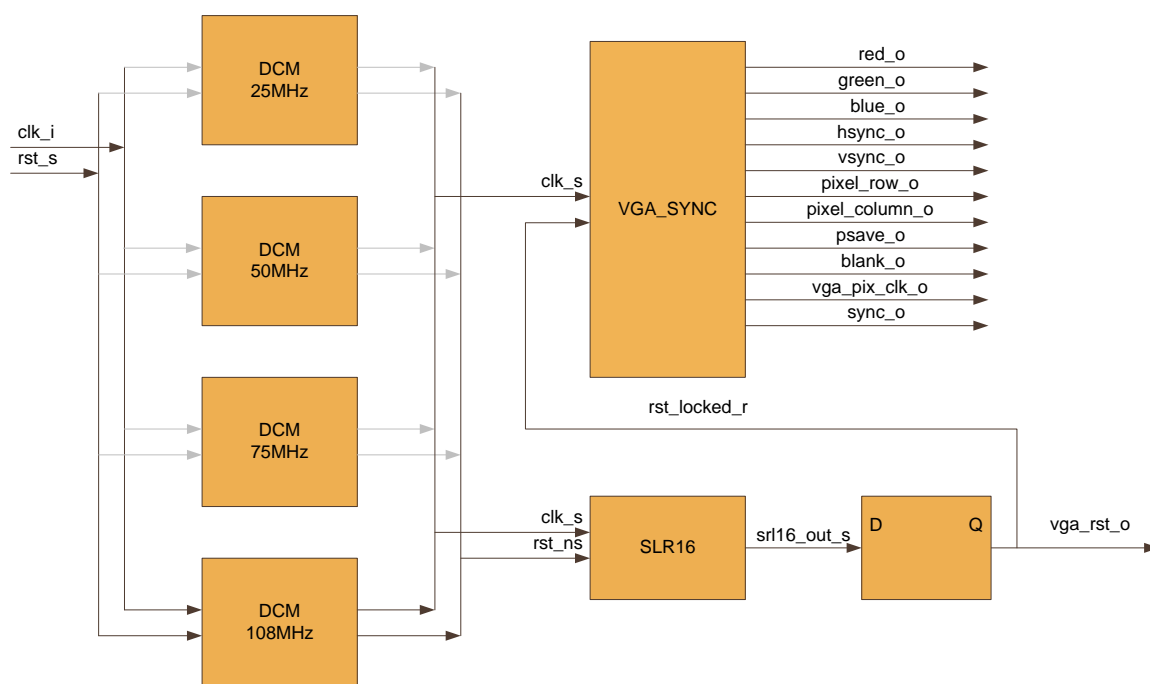
Slika 8: Blok dijagram GRAPH_MEM modula

U nastavku je dato objašnjenje preostalih portova VGA_TOP modula:

1. *direct_mode* - [b0 - off ,b1 - on] - Izbor načina generisanja slike. Aktivan *direct_mode* generiše sliku (direktnu vrednost RGB komponenti) na osnovu informacije o *dir_pixel_row* i *dir_pixel_column* iz *vga_top* modula U slučaju da *direct_mode* nije aktivan generisanje slike određuje *display_mode*.

2. *display_mode* - [b00 - nedefinisano, b01 - tekstualni način rada, b10 - grafički, b11 - tekstualni i grafički] - U slučaju da je isključen *direct_mode* omogućava izbor načina generisanja slike prema datoj specifikaciji.
3. *show_frame* - [0 - off ,1 - on] - iscrtava okvir po ivici ekrana koji je nezavisan od načina iscrtavanja slike.
4. *font_size* - [b00 - 8, b01 - 16, b10 - 32, b11 - 64] - odabir veličine fonta.
5. *foreground_color* - predstavlja boju objekta (tekst ili grafika). To je 24-bitna vrednost kod koje se na najvišim bitima [23:16] nalazi crvena, [15:7] zelena, i na [7:0] plava komponenta.
6. *background_color* - predstavlja boju pozadine sa istim mapiranjem boja.
7. *frame_color* - predstavlja boju okvira ako je aktivan. Koristi se isto mapiranje boja kao i kod prethodnih registara.

Zarad kompletnosti, dat je i blok dijagram VGA modula.



Slika 9: Blok dijagram VGA modula

Zadatak

1. Na ekranu iscrtati 8 vertikalnih pruga iste širine ali različitih boja (eng. color bar) koristeći *direct_mode* i povezati signale *direct_mode* i *display_mode* na prekidače sa E2LP platformi.
2. Realizovati logiku koja obezbeđuje ispis proizvoljnog teksta na proizvoljnoj poziciji na ekranu koristeći tekstualni režim rada VGA_TOP modula.
3. (Dodatni) Realizovati pomeranje ispisanog teksta sa definisanom dinamikom.

4. Realizovati logiku koja opisuje kvadrat proizvoljnih dimenzija na sredini ekrana koristeći grafičku memoriju VGA_TOP modula.
5. (Dodatni) Realizovati pomeranje kvadrata po horizontali.

Za sve zadatke potrebno je detaljno simulirati sistem i nacrtati odgovarajuće blok dijagrame.