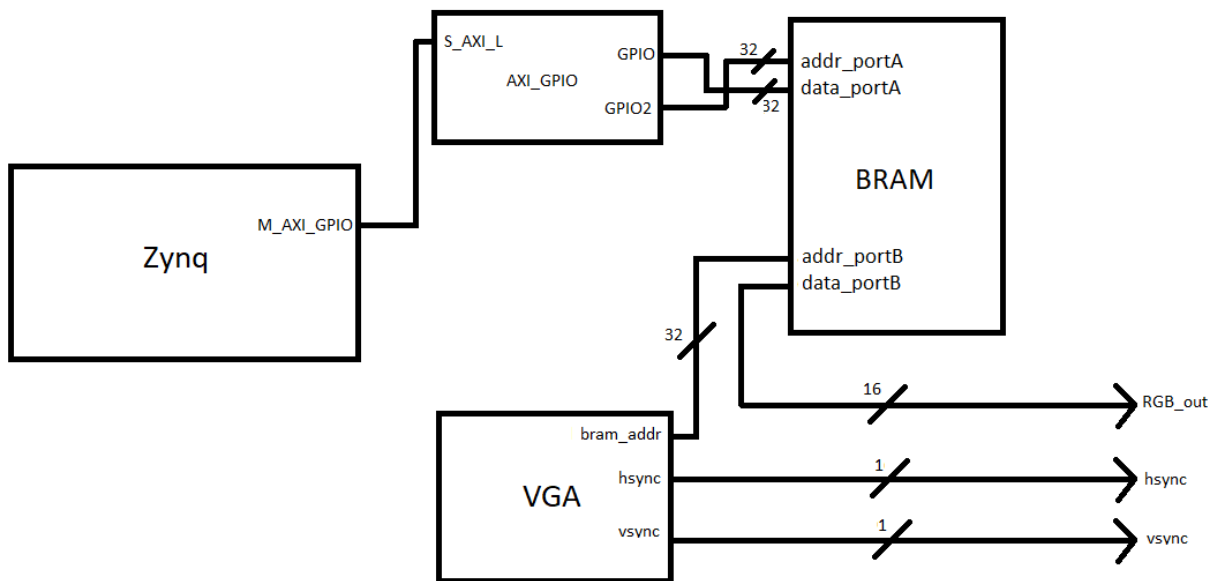


Implementacija VGA kontrolera na Zybo razvojnoj ploči

Da bi se napisao drajver za sistem čija je uloga da generiše sliku na VGA izlazu, mora se razumeti od kojih komponenti se taj sistem sastoji, kako se sa njim komunicira i koja su mu ograničenja. Na sledećoj slici je prikazana blok šema jedne implementacije takvog sistema:



Slika 1.

Kao što se sa slike Može videti sistem se sastoji iz sledećih komponenti:

- Zynq procesor
- AXI_GPIO
- BRAM
- VGA

Zynq procesor prosleđuje sliku piksel po piksel preko AXI_LITE interfejsa. Rezolucija slike koju procesor šalje iznosi 256x144, odnosno sastoji se iz 36864 piksela, pri čemu je svaki piksel predstavljen preko 16 bita (15:11-intenzitet crvene boje, 10:5-intenzitet zelene boje i 4:0 intenzitet plave boje). Način na koji procesor šalje piksele jeste da preko AXI_LITE interfejsa prvo prosledi adresu BRAM memoriji na koji hoće da upiše piksel, a nakon toga da preko istog AXI_LITE interfejsa prosledi vrednost piksela koja će biti upisana na prethodno datoj adresi (ovaj redosled je jako bitan).

AXI_GPIO komponenta omogućava ulazno/izlazni interfejs prema AXI_LITE interfejsu. U ovom slučaju konfigurisana je tako da ima dva izlazna kanala (GPIO i GPIO2). GPIO i GPIO2 su memorijski mapirani na adrese 0x0000 i 0x0008 respektivno. GPIO je povezan na *data_portA* BRAM memorije, a

GPIO2 je povezan na *addr_portA* BRAM memorije, tako da ako procesor želi da postavi adresu na koju želi da upiše piksel u BRAM memoriji, on mora preko AXI_LITE interfejsa da gađa GPIO2 koji je memorijski mapiran na 0x0008 adresi, a ako hoće da upiše vrednost piksela, on mora da gađa GPIO koji je memorijski mapiran na adresi 0x0000. *S_AXI_L* ulaz je povezan sa Zynq procesorom, odnosno preko njega procesor pogađa memorijski mapirane *GPIO* i *GPIO2* izlaze. Detaljni opis AXI_GPIO komponente se može naći na sledećem linku:

https://www.xilinx.com/support/documentation/ip_documentation/axi_gpio/v2_0/pg144-axi-gpio.pdf

U **BRAM** memoriju se smešta slika koju je potrebno prikazati na monitoru. Dubina BRAM memorije iznosi 36864 32-bitne lokacije, i razlog za to je rezolucija slike koja iznosi 256*144 piksela, a kako je jedan piksel predstavljen sa 16 bita, to znači da se 16 bita od 32 prilikom isčitavanja piksela iz bram memorije odbacuje, što se može videti na slici 1 (nižih 16 bita 32-bitne lokacije jesu vrednost piksela, a gornjih 16 se odbacuje). Zbog ovakvog načina organizacije memorije i zbog toga što je BRAM memorija bajt adresabilna, pojedinačni piksel je pomeren za četiri adresne lokacije u odnosu na piksele oko sebe, odnosno prvi piksel se upisuje na adresu 0x0000, drugi piksel se upisuje na adresu 0x0004, treći na 0x0008 itd. U ovom slučaju BRAM komponenta je konfigurisana kao dvoprístupna memorija (portA i portB), pri čemu se slika učitava preko portaA, a isčitava preko portaB. U memoriju se upisuje tako što se preko GPIO modula prvo prosledi adresa na koju se želi upisati podatak, a nakon toga sa šalje i sam podatak. Iz memorije isčitava tako što se pošalje adresa sa koje se želi isčitati, a u sledećem taktuse podatak prosledi na VGA port.

VGA komponenta je zadužena za generisanje hsync i vsync signala koji se prosleđuju na hsync i vsync portove VGA konektora. Ti signali su sinhronizacioni signali koji su neophodni VGA konektoru kako bi znao na kom mestu na monitoru da postavi određeni piksel. VGA komponenta ima i jedan port povezan sa BRAM-om koji generiše adrese piksela koje treba postaviti na određenu lokaciju na monitoru, a BRAM na osnovu tih adresa daje vrednosti piksela na portu data_portB i prosleđuje ih na RGB_out port VGA konektora kao što se može videti na slici 1. VGA komponenta generiše adrese piksela u skladu sa hsync i vsync signalima. Detaljno poznavanje rada ove komponente nije neophodno za pisanje drajvera ovog sistema, te se ova skripta neće upuštati u to. Za one koji žele da znaju više, to mogu naći na sledećem linku:

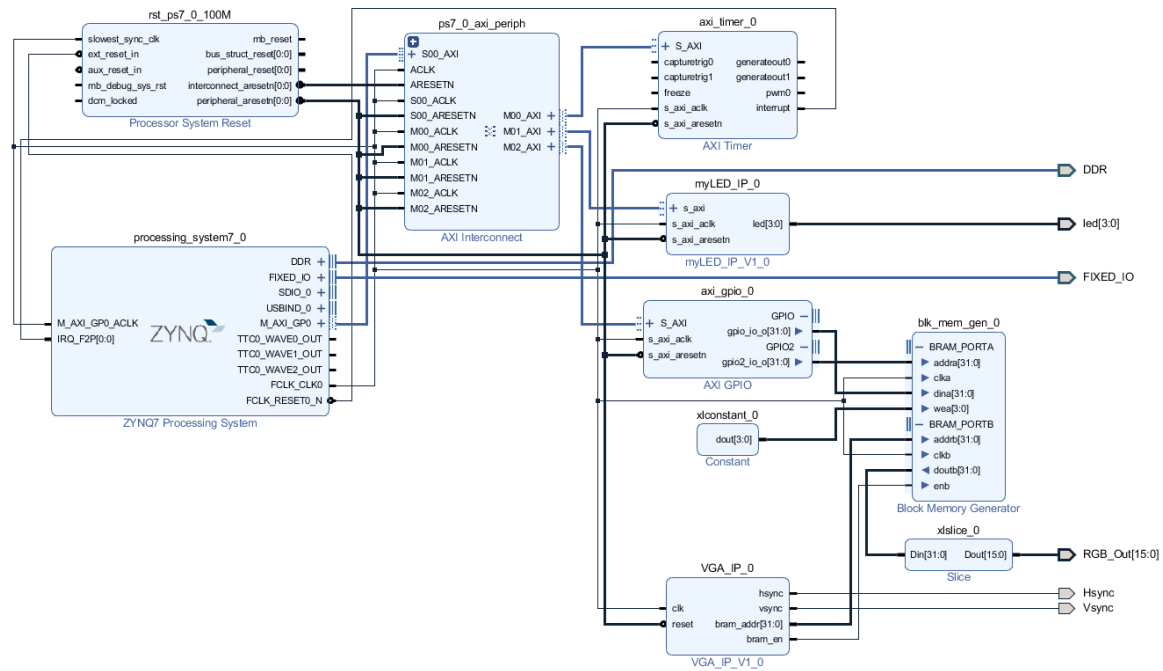
<https://www.elektronika.ftn.uns.ac.rs/uvod-u-mikroracunarsku-elektroniku/wp-content/uploads/sites/134/2018/03/Vezba-11-Projektovanje-slo%C5%BEenih-digitalnih-sistema-VGA.pdf>

Ograničenje ovog sistema je to da ne može da predstavi sliku čija je rezolucija veća od 256x144 piksela jer u programabilnoj logici ne postoji dovoljna količina BRAM-a, tako da iako VGA kontroler generiše vsync i hsync signali tako da se može predstaviti slika rezolucije 480x640, količina BRAM memorij na zbyo razbojnoj ploči u ograničava tu rezoluciju na 256x144. **Tako da prilikom pisanja drajvera treba voditi računa da se ne prosleđuje slika čiji je broj piksela veći od 36864.**

Napomena:

Prethodna blok šema (slika 1) predstavlja pojednostavljeni izgled sistema gde su određeni signali (klok, reset, ...), komponente i portovi izostavljeni jer nisu bitni za razumevanje njegovog rada.

Na sledećoj slici je prikazan sistem napravljen u IP integratoru Vivado alata:



Slika 2.

U ovom projektu pored sistema za generisanje slike na VGA konektoru, postoji i komponenta koja kontroliše rad određenih dioda (na slici 2. *myLED_IP_0*), kao i AXI_TIMER komponentna (na slici 2. *axi_timer_0*). Njihov način rada neće biti opisan.

Kako bi se projekat automatski generisao, potrebno je skinuti sledeće sa github-a:

https://github.com/DjordjeMiseljic/VGA_BRAM_Controller.git

Na ovom linku tcl skripta se nalazi u direktorijumu VGA_BRAM_Project, a u direktorijumu Driver_App se nalaze drajver i aplikacija za rad sa sistemom.

Prvo je potrebno pokrenuti vivado, u njemu pritisnuti tools->Run Tcl Script, navesti putanju do skinutog direktorijuma i u njemu dvoklikom pokrenuti Master.tcl fajl. Vivado na osnovu tog fajla generiše, sintetizuje, implementira projekat, i na osnovu toga generiše bitstream fajl (za sve ovo je potrebno 10-tak minuta).