

LABORATORIJSKA VEŽBA 7

Projektovanje procesora

Potrebno predznanje

- Instanciranje komponenti, rad procesora
- Arhitektura računara i assembler

Šta će biti naučeno tokom izrade vežbe?

U ovoj vežbi:

- Instanciraćete i uvezati nove komponente u procesor
- Upoznaćete se sa magistralama i vezom ulazno-izlaznih jedinica sa procesorom
- Pisaćete programsku podršku za ovaj sistem u assembleru
- Simuliraćete svoje rešenje i testirati ga na ploči

Apstrakt i motivacija

Sam procesor i memorija nisu dovoljni da bi se obavila računarskih zadataka. Takođe je potrebno imati ulazno-izlazne jedinice. Procesor komunicira sa kontrolerima ulazno-izlaznih jedinica, takozvanim ulazno-izlazni kontrolerima (IO Ctrl - Input/Output Controller) preko magistrale za podatke. Preko ulazno-izlaznih kontrolera, procesor dobija podatke od ulazno-izlazne jedinice. Na isti način procesor može da isporuči rezultate svojih računanja. Da bi ovaj sistem mogao da bude pravilno orkestriran, neophodna je dobra programska podrška.

Procesor i Magistrale

Datoteka *top.vhd* sada osim procesora, instrukcijskog ROM-a i RAM-a za podatke sada sadrži još 2 ulazno-izlazna kontrolera. *rgb_matrix_io_ctrl.vhd* je izlazni kontroler RGB matrice, koja služi za vizuelnu komunikaciju sa humanoidom, dok *push_buttons_dec_io_ctrl.vhd* prima naredbe od istog preko 4 tastera. Ovaj sistem je prikazan na Slici 1. Na istu magistralu podataka prikačeni su RAM za podatke i IO kontroleri. Na taj način je moguće pristupiti IO kontrolerima preko LOAD i STORE instrukcija. Magistrala za instrukcije je ostala ista.

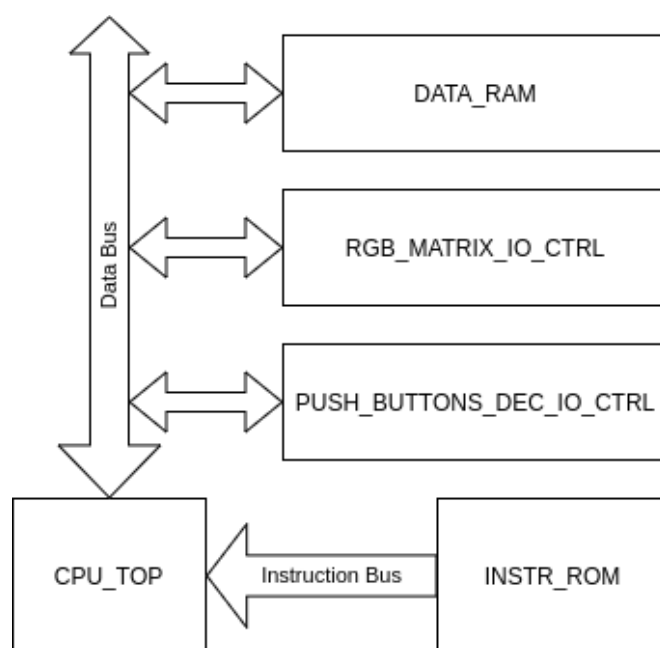
Memorijsko mapiranje

U Tabeli 1 je prikazana memorijska mapa sistema. ADDR[9:8] se koriste za selekciju jedinice, dok niži biti ADDR[7:0] služe za adresiranje registara unutar jedinice.

Realizacija magistrale podataka

Slika 2 prikazuje realizaciju magistrale podataka za više jedinica. Na *cpu_top* su vezana 4 signala magistrale podataka: sBUS_A, sBUS_RD, sBUS_WD, sBUS_WE. Signali sBUS_A[7:0] i sBUS_WD su zajednički za sve jedinice. Signal sBUS_A[9:8] se koristi za multipleksiranje više signala pročitanih podataka u jedan sBUS_RD. Isti

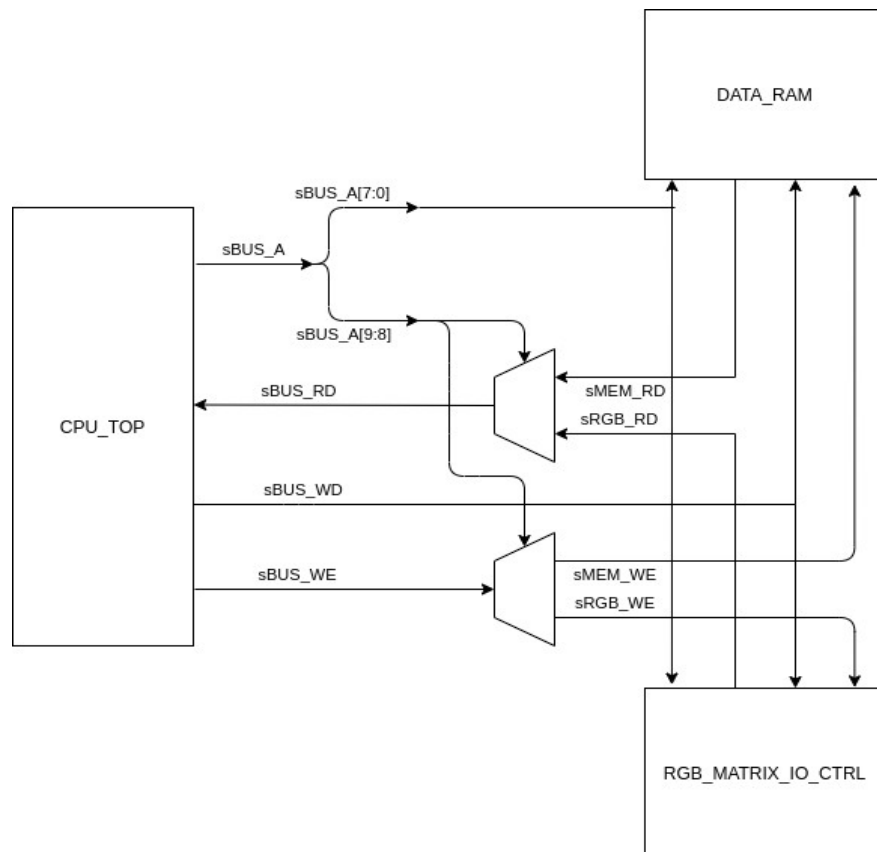
sBUS_A[9:8] se koristi za demultipleksiranje signala dozvole upisa sBUS_WE, kako bi se rezultat upisao u odgovarajuću jedinicu.



Slika 1. Vrh hijerarhije sistema

Tabela 1. Memorijska mapa

Jedinica	U assembleru	Početna adresa	Krajnja adresa
Data RAM	.data	00 0000 0000	00 1111 1111
RGB Matrix IO Ctrl	p_rgb_matrix	01 0000 0000	01 0011 1111
RGB Matrix IO Ctrl	p_frame_sync	01 0100 0000	01 0100 0000
Push Buttons Dec IO Ctrl	p_pb_dec	10 0000 0000	10 0000 0000



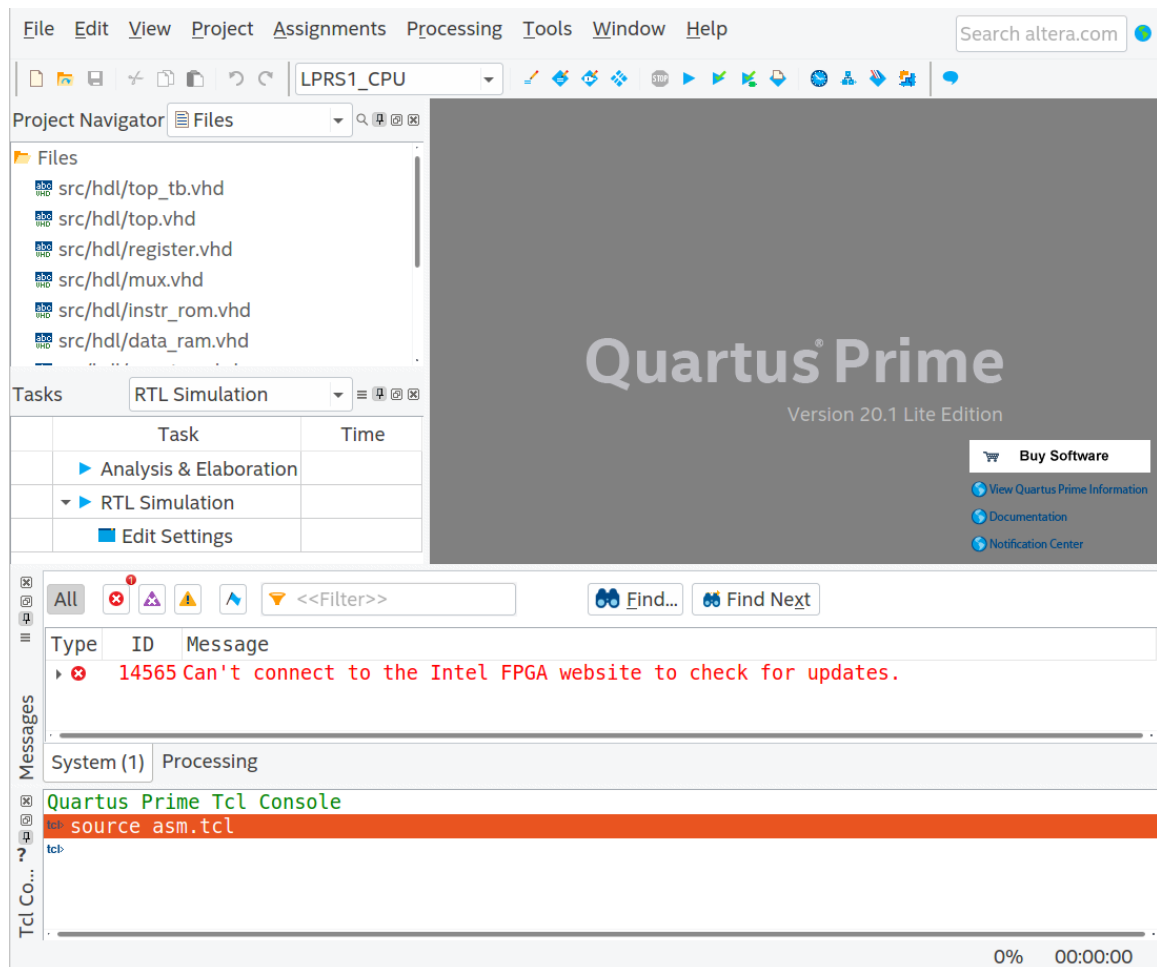
Slika 2. Realizacija magistrale podataka

Asembliranje

U slučaju izmene programa u datoteci snake.asm, potrebno je ponovno izvršiti asembliranje. Radi pokretanja asemblera, otvoriti Tcl Console preko prečice **Alt+2**. U toj istok konzolici upisati komandu praćenu enterom:

```
source asm.tcl
```

Slika 3 prikazuje izgled Quartus-a pri asembliranju.



Slika 3. Asembliranje

Radi asembliranja nekog drugog fajla, izmeniti promenljivu `asm_file` u `asm.tcl` datoteci.

ZADATAK

Opšti zadatak je realizovati igru zmijice.

1. Instancirati `push_buttons_dec_io_ctrl` u `top` i povezati ga na magistralu.
2. Realizovati asemblerski kod za iscrtavanje zmejice na RGB matricu. Koristiti `p_food_and_snake` i `p_rgb_matrix`. Rešenje proveriti u simulaciji.
3. Realizovati jedenje hrane. Kada glava zmijice se poklopi sa hranom, generisati novu hranu. Takođe zmijica kad pojede hranu, treba da naraste. Ovo isto proveriti u sintezi.
4. Realizovati čekanje brojanje frejmova do jednog otkucaja. Trenutno se zmijica pomeri pri svakom iscrtanom frejmu, ali to je prebrzo za realan hardver. Koristiti `frames_cnt` i `frames_per_heartbeat` promenljive.

- a) Umesto 75 za *frames_per_heartbeat*, koristiti 1, pa proveriti rešenje u simulaciji. Posle toga proveriti rešenja sa 2 za *frames_per_heartbeat*.
 - b) Uraditi sintezu i proveriti rešenje na platformi u realnom vremenu. Koristiti 75 za *frames_per_heartbeat*. Time će otkucaj biti jednak jednoj sekundi.
- 5. Putem istog *frames_per_heartbeat* vršiti ubrzavanje zmijice.
 - 6. Napraviti detekciju kolizije sa zidovima i zmijice sa samom sobom.
 - 7. Garantovati da se hrana en generiše na zmijici.
 - 8. Napraviti da zmijica prolazi kroz zidove i pojavi se na suprotnoj strani.
 - 9. Obrnuti igricu: Imati zmijicu dužine 64. Dokaz okačiti na Youtube.