

NAZIV VEŽBE:

Procesor.

ZADATAK VEŽBE:

Proširenje skupa instrukcija procesora realizovanog u prethodnoj vežbi

CILJ VEŽBE:

Analiza toka podataka unutar procesora i između procesora i memorije sa podacima.

1. Opis proširenja arhitekture procesora

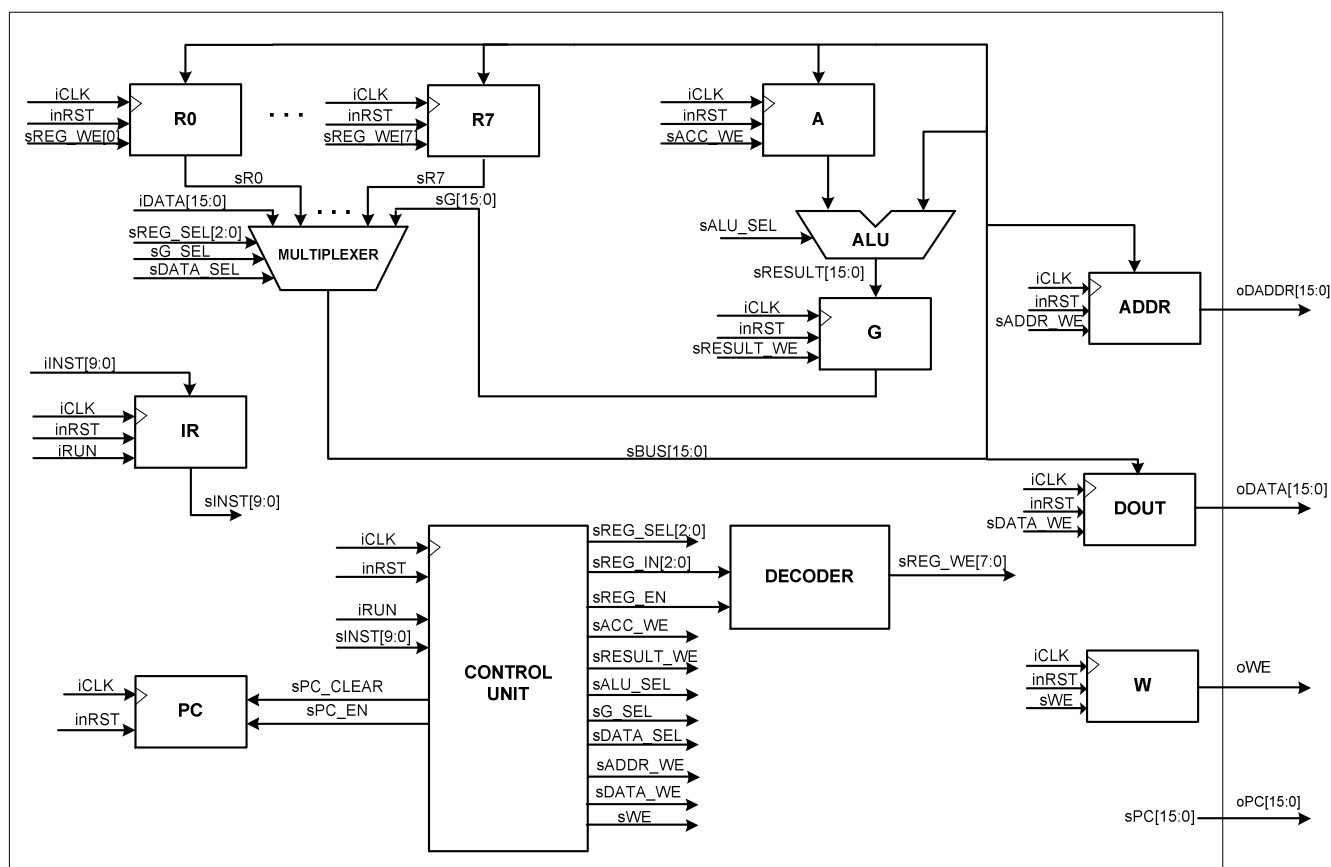
Procesor iz prethodne laboratorijske vežbe je potrebno proširiti sa skupom instrukcija koje omogućuju razmenu podataka između procesorskog jezgra i memorije sa podacima. Blok šemu arhitekture procesora koju treba realizovati prikazuje Slika 1.1.

Procesor treba proširiti sa tri nove instrukcije koje prikazuje Tabela 1.1. Instrukcija **ld** (engl. *load*) učitava podatke u registar RX iz spoljne memorije sa adrese koja se nalazi u registru RY. Upis podataka iz procesora u memoriju realizuje instrukcija **st** (engl. *store*), tako što podatak iz registra RX upisuje na memorijsku adresu sadržanu u registru RY. Na kraju, instrukcija **mvnz** (engl. *move if not zero*) dozvoljava izvršenje instrukcije kopiranja sadržaja registra u drugi registar samo ako je trenutni sadržaj registra G različit od nule.

Operacija	Funkcija	Kod instrukcije
ld Rx, [[Ry]]	$RX \leftarrow [[RY]]$	1000
st [Ry], [Rx]	$[RY] \leftarrow [RX]$	1001
mvnz Rx, Ry	If G!=0 $RX \leftarrow [RY]$	1010

Tabela 1.1 Nove intrukcije procesora

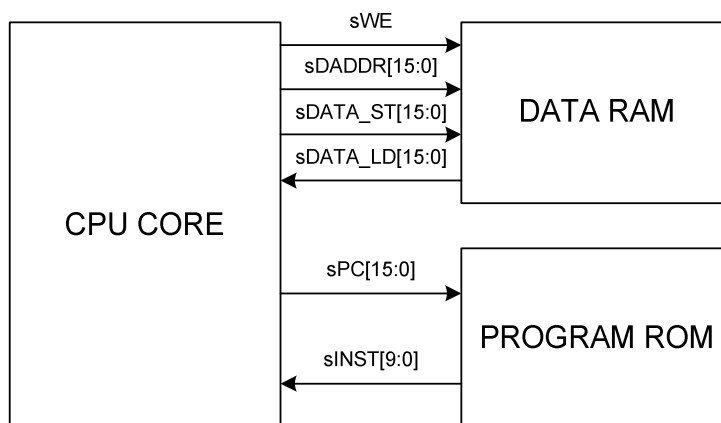
Prilikom proširenja skupa instrukcija procesora uzeti u obzir da je upravljačka jedinica procesora realizovana sa registrovanim izlazima, što prouzrokuje da svi upravljački signali kasne za jednu periodu takt signala.

**Slika 1.1 Blok šema arhitekture procesora**

Pored proširenja arhitekture samog procesorskog jezgra, potrebno je i izmeniti format memorije sa podacima. U prethodnoj vežbi memorija sa podacima je implementirana u formi stalne memorije (ROM). U ovom slučaju za potrebe izvršenja instrukcije **st** potrebno je omogućiti upis u memoriju sa podacima, čime ona postaje RAM memorija. Signal dozvole upisa generiše procesorsko jezgro i označen je sa **sWE** (Slika 1.1). Podaci koji se upisuju u memoriju treba da se nalaze u memorijskom registru podataka (**oDATA[15:0]**) a adresa na koju se upisuje u memorijskom adresnom registru (**oDADDR[15:0]**).

Ukoliko je signal dozvole pisanja neaktivan, na niskom naponskom nivou, na magistrali podataka od memorije ka procesorskom jezgru se nalazi sadržaj adresirane lokacije. Za operaciju čitanja, instrukcija **ld**, nije potreban dodatni signal dozvole, već je to podrazumevani način pristupa memoriji ako nije posebno naglašen proces upisa u memoriju.

Blok šemu povezivanja procesorskog jezgra i memorije prikazuje Slika 1.2.



Slika 1.2 Blok šema povezivanja procesorskog jezgra i memorije

2. Zadatak

Definisani procesor proširiti sa 3 nove instrukcije.

U okviru vežbe je potrebno:

- Formirati tabelu generisanja upravljačkih signala za nove instrukcije
- Formirati mikrokod na osnovu kojeg će se proveriti funkcionalnost novih instrukcija
- Izmeniti VHDL kod sa ciljem izvršavanja projektovanog mikrokoda
- Simulirati izvršenje mikroalgoritma pomoću testbencha koji sam proverava tačnost rezultata
Testbench treba da simulira rad modula opisanog u datoteci CPU.VHD
- Izvršiti sintezu procesora za TLL5000 demonstracionu ploču, pomoću datog VHDL modula za povezivanje sa potrebnim periferijama na TLL5000 ploči (CPU_SHELL.VHD)
- Izvršiti testiranje u realnom vremenu na TLL5000 demonstracionoj ploči

REALIZACIJA VEŽBE:

U okviru vežbe je potrebno kreirati projekat za realizaciju traženog digitalnog sistema i izvršiti njegovu simulaciju uz pomoć VHDL testbench-a.

PRIPREMA ZA VEŽBU:

Priprema za vežbu obuhvata sledeće:

- Formirati tabelu generisanja upravljačkih signala za nove instrukcije
- Formirati mikrokod na osnovu kojeg će se proveriti funkcionalnost novih instrukcija
- Izmenjen VHDL kod sa ciljem izvršavanja projektovanog mikrokoda

DODATNI ZADATAK:

1. Izmeniti VHDL kod tako da procesor sadrži statusni registar sa četiri statusna bita Z, N, O i C.
Stanje četiri statusna bita prikazati na LED diodama
2. Umesto takt signala generisanog putem tastera iskoristiti isti takt signal za procesor kao i za LCD displej
3. Formirati mikroalgoritam koji redom pali i gasi LED diode u intervalu od 0.5 sekundi