

DOMAĆI ZADATAK 1

Kalkulator

U VHDL jeziku za opis fizičke arhitekture realizujte 4-bitni digitalni kalkulator koji ima sledeće prolaze:

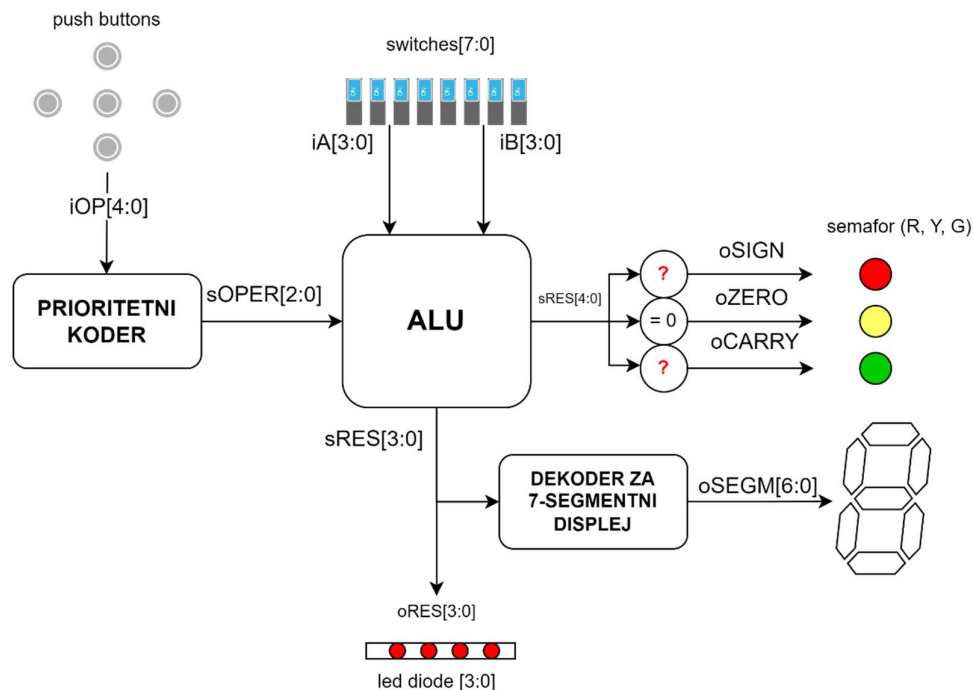
Ulazi sistema:

- ***iA[3:0]*** – prvi operand, označen u komplement II predstavi
- ***iB[3:0]*** – drugi operand, označen u komplement II predstavi
- ***iOP[4:0]*** – operacija koju je potrebno izvršiti nad ulaznim operandima

Izlazi sistema:

- ***oRES[3:0]*** – rezultat
- ***oZERO*** – statusni bit za indicaciju da li je rezultat jednak nuli
- ***oSIGN*** – statusni bit za indicaciju da li je rezultat pozitivan ili negativan
- ***oCARRY*** – statusni bit za indicaciju da li se desio prenos u toku sabiranja
- ***oSEGM[6:0]*** – signal za prikaz rezultata na 7-segmentnom displeju

Arhitektura sistema kalkulatora je prikazana na slici:



Sistem obuhvata sledeće komponente:

- **Prioritetni koder višeg prioriteta** za određivanje željene operacije,
- **4-bitna ALU jedinica** (Aritmetičko-logička jedinica) za izvršavanje logičke ili aritmetičke operacije,
- **Komponente** za određivanje statusnih bita za **ZNAK, NULU i PRENOS**,
- **Dekoder** za prikaz na **7-segmentni displej**.

Korisnik zadaje 4-bitne operande **IA** i **IB**, kao i željenu operaciju preko signala **iOP**. Ulaz **iOP** je 5-bitni signal koji se **prioritetnim koderom višeg prioriteta** pretvara u trobitni selekcionni signal **sOPER**, koji zatim određuje operaciju koju je potrebno izvršiti, u skladu sa pozicijom aktivnog bita signala **iOP**:

| Prioritet | iOP[4:0] | Taster | sOPER[2:0] | Operacija |
|-----------|----------|----------------|------------|-----------------------------------|
| 1 | „10000“ | UP | „100“ | Sabiranje |
| 2 | „01000“ | DOWN | „011“ | Oduzimanje |
| 3 | „00100“ | CENTAR | „010“ | Logička AND operacija |
| 4 | „00010“ | LEVO | „001“ | Pomeranje iA ulevo za jedno mesto |
| 5 | „00001“ | DESNO | „000“ | OPERACIJA PO IZBORU |
| - | „00000“ | NIJEDAN TASTER | „111“ | BEZ OPERACIJE i sRES = 0 |

NAPOMENA: Potrebno je realizovati koder višeg prioriteta tako da ukoliko korisnik pritisne više tastera treba da se izvrši operacija najvišeg prioriteta. Prioritet operacija je označen brojevima od 1 do 5, gde najviši prioritet predstavlja taster UP, a najniži taster DESNO.

ALU jedinica prihvata 2 ulazna operanda **IA** i **IB**, izvršava operaciju u zavisnosti od vrednosti signala **sOPER**:

- Sabiranje
- Oduzimanje
- Logička AND operacija
- Pomeranje ulevo za 1 mesto ($iA \ll 1$)
- **OPERACIJA PO IZBORU** (da nije logička operacija OR, NOT, XOR, itd.)

Rezultat iz ALU jedinice je **5-bitni signal sRES**, na osnovu kog je potrebno odrediti **statusne bite**:

- **oZERO** – treba da bude aktivan na vrednosti '1' ukoliko je rezultat jednak 0,
- **oSIGN** – treba da bude aktivan na vrednosti '1' ukoliko je rezultat negativan broj,
- **oCARRY** – treba da bude aktivan na vrednosti '1' ukoliko se desio prenos u toku operacije SABIRANJA.

Rezultat **sRES[3:0]** se takođe šalje i na **dekoder za prikaz na 7-segmentni displej**, koji ne radi u potpunosti kao klasičan dekodeer, ali omogućava dekodovanje ulaznog četvorobitnog broja na 7 svetlećih dioda 7-segmentnog displeja. Npr. da bi se prikazao broj 1, potrebno je da svetle segmenti b i c, što znači da se vrednost „0001“ treba dekodovati u „0110000“ za 7-segmentni displej. Potrebno je omogućiti ispis brojeva od 0 do 9.

Simulacija sistema:

Proveriti ponašanje sistema u simulaciji realizacijom **5 testnih slučajeva po izboru**, tako da se pokriju sve operacije ALU jedinice sa **različitim ulaznim vrednostima**. U okviru test bench datoteke tekstualno opisati u komentaru svaki test slučaj pre zadavanja signala u stimulus procesu.

Primer test slučaja: Simulirati operaciju sabiranja dva broja tako da rezultat na 7-segmentnom displeju bude 1 i da je aktivan statusni bit za prenos **oCARRY**.

Implementacija na ploču MAX1000:

Za implementaciju kalkulatora na platformu MAX1000, potrebno je povezati prolaze na sledeći način:

- Ulaz **IA** povezati na viša 4 prekidača, a ulaz **IB** na niža 4 prekidača (switch)
- Ulaz **iOP** povezati na 5 tastera redom za svaku operaciju kao što je definisano u prioritetnom koderu.
- Statusne bite **oSIGN**, **oZERO**, **oCARRY** povezati na semafor svetla kao što je prikazano na slici.
- Izlaz **oRES** povezati na 4 led diode.
- Izlaz **oSEGM** povezati na 7-segmentni displej

NAPOMENE ZA IZRADU ZADATKA:

- Rešenje realizovati u **dobijenoj postavci zadatka** (ne praviti novi projekat).
- **Nije dozvoljeno** menjati entitet sistema, kao i nazive već definisanih signala!
- Moguće je po potrebi dodati smislene interne signale.
- Na početku svake VHDL datoteke upisati **SVOJE PODATKE**.
- Podesite pinove za svaki ulaz i izlaz sistema na osnovu podataka iz datoteke **LPRS1_FPGA_pins.pdf**.
- Rešenje možete proveriti na MAX1000 platformi u dodatnim terminima koji će biti objavljeni na SOVI.

NAPOMENE ZA PREDAJU REŠENJA NA SOVU:

- Realizovano rešenje postaviti na **SOVU** u predviđenom roku za svoju grupu po sledećim pravilima:
 - SAMO VHDL datoteke (**lprs1_homework1.vhd** i **lprs1_homework1_tb.vhd**) ZIPOVATI u datoteku sa nazivom **LPRS1_Homework1_XX_YYY_QQQQ**, gde je
 - XX – modul (RA, PSI ili IN),
 - YYY – trocifren broj indeksa (npr. 003 ako je indeks 3/2023)
 - QQQQ – godina upisa (npr. 2023)
 - **Primer** za indeks RA 3/2023 -> **LPRS1_Homework1_RA_003_2023**
- **Predati zadaci sa pogrešnim nazivom neće biti pregledani!!!**

NAPOMENE ZA ODBRANU ZADATKA:

- Odbrana zadatka se realizuje **na redovnom terminu računarskih vežbi**, gde student nakon nekoliko teorijskih pitanja i dodatnog zadatka može ostvariti **0, 1, 2 ili 5 poena**. Student koji je odgovorio tačno na sva pitanja i ispravno uradio dodatni zadatak u predviđenom vremenu može dobiti maksimum poena.
- Koraci odbrane domaćeg zadatka:
 - Student treba da dođe u **svoj termin računarskih vežbi** po rasporedu.
 - **Preuzeti** sa SOVE **svoje rešenje i postavku** zadatka i **kopirati** svoje vhd datoteke u postavku.
 - Pokrenuti **simulaciju** i implementirati rešenje **na ploču**.
 - Nakon prethodno realizovanih koraka, pozvati asistenta za ispitivanje i dodatni zadatak.
- Ukoliko je rešenje predato na SOVU, a student se nije pojavio na odbrani zadatka, zadatak nosi **NULA** poena.
- Rešenja se provlače kroz alat za detekciju plagijarizma, gde svako rešenje sa velikim procentom sličnosti dobija **NULA** poena.