Projektovanje procesora – osnovne komponente

# Standardne komponente

U datoteci *reg.vhd* realizovati 16-bit registar sa dozvolom upisa. Prolazi registra su dati u Tabeli 1-1.

Tabela 1-1. Prolazi registra

|  |  |  |
| --- | --- | --- |
| **Prolaz** | **Smer** | **Funkcija** |
| iCLK | in | signal takta |
| inRST | in | signal reseta, aktivan na visokom logičkom nivou |
| iD [15:0] | in | ulazni podatak |
| iWE | in | signal dozvole upisa |
| oQ [15:0] | out | vrednost registra |

U datoteci *cnt.vhd* realizovati 16-bit brojač sa dozvolom upisa i brojanja. Prolazi brojača su dati u Tabeli 1-2.

Tabela 1-2. Prolazi brojača

|  |  |  |
| --- | --- | --- |
| **Prolaz** | **Smer** | **Funkcija** |
| iCLK | in | signal takta |
| inRST | in | signal reseta, aktivan na visokom logičkom nivou |
| iD [15:0] | in | ulazni podatak |
| iEN | in | signal dozvole brojača, ako je na niskom nivou brojač zadržava vrednost |
| iLOAD | in | signal dozvole upisa, ako je na visokom nivou brojač preuzima vrednost sa ulaza ID, a ako je na niskom nivou brojač uvećava vrednost za 1 |
| oQ [15:0] | out | vrednost brojača |

U datoteci *mux.vhd* realizovati 16-bit9:1 multiplekser. Prolazi multipleksera su dati u Tabeli 1-3.

Tabela 1-3. Prolazi multipleksera

|  |  |  |
| --- | --- | --- |
| **Prolaz** | **Smer** | **Funkcija** |
| iD0 [15:0] – iD8 [15:0] | in | devet 16-bit ulaza multipleksera |
| iSEL [3:0] | in | selekcija multipleksera |
| oQ [15:0] | out | izlaz multipleksera |

# Memorija

U datotekama *instr\_rom.vhd* i*data\_ram.vhd* realizovati kombinacionu ROM i sinhronu RAM memoriju kapaciteta 32 reči. ROM memorija ima širinu reči 15 bita, a RAM memorija širinu reči 16 bita. ROM memorija će se koristiti za instrukcije procesora, a RAM memorija za podatke. ROM memorija ima omogućeno samo čitanje, a RAM memorija ima omogućeno i čitanje i pisanje. Prolazi memorija su dati u Tab. 2-1.

Tabela 2-1. Prolazi memorije

|  |  |  |
| --- | --- | --- |
| **Prolaz** | **Smer** | **Funkcija** |
| iCLK | in | signal takta, **samo u RAM memoriji** (ROM memorija je kombinaciona mreža) |
| inRST | in | signal reseta aktivan na niskom logičkom nivou, **samo u RAM memoriji** |
| iA [4:0] | in | adresa |
| iD [15:0] | in | ulazni podatak, **samo u RAM memoriji** |
| iWE | in | dozvola upisa podatka, **samo u RAM memoriji** |
| oQ [15:0]  oQ [14:0] | out  out | izlazni podatak**RAM memorije**  izlazni podatak **ROM memorije** |

Memoriju za podatke opisati pomoću nizova u VHDL jeziku, a ROM memoriju kao kombinacionu mrežu. Vrednosti u ROM memoriji postaviti proizvoljno. Početna vrednost RAM memorije neka bude 0x0000 na svim lokacijama.

# Aritmetika u procesoru

U datoteci *alu.vhd* realizovati aritmetičko-logičku jedinicu (ALU) procesora. Prolazi ALU su dati u Tabeli 3-1.

Tabela 2-1. Prolazi ALU

|  |  |  |
| --- | --- | --- |
| **Prolaz** | **Smer** | **Funkcija** |
| iA [15:0] | in | prvi ulazni operand |
| iB [15:0] | in | drugi ulazni operand |
| iSEL [3:0] | in | selekcija operacije |
| oC [15:0] | out | rezultat operacije |
| oZERO | out | statusni bit, rezultat jednak nuli |
| oSIGN | out | statusni bit, rezultat negativan |
| oCARRY | out | statusni bit, rezultat ima prenos |

ALU realizovati kao kombinacionu mrežu, podržavajući operacije navedene u Tabeli 3-2. Sve operacije rade sa brojevima u označenoj predstavi II komplementa.

Tabela 3-2. Operacije ALU

|  |  |
| --- | --- |
| **Kod operacije** | **Operacija** |
| 0000 | A |
| 0001 | A + B |
| 0010 | A – B |
| 0011 | A and B |
| 0100 | A or B |
| 0101 | not (A) |
| 0110 | A + 1 |
| 0111 | A – 1 |
| 1000 | shl (A) |
| 1001 | shr (A) |
| 1010 | ashl (A) |
| 1011 | ashr (A) |

Statusni biti trebaju biti realizovani kombinaciono, po sledećim pravilima:

* sZERO je aktivan ukoliko je rezultat operacije jednak nuli,
* sSIGN je aktivan ukoliko je rezultat operacije negativan,
* sCARRY je aktivan ukoliko rezultat operacije ima prenos.

Projektovanje procesora – osnovne komponente

# Prazna upravljačka jedinica

U datoteci *control\_unit.vhd*realizovati upravljačku jedinicu procesora. Upravljačka jedinica unutar procesora kontroliše rad svih ostalih delova procesora. U našem slučaju, upravljačka jedinica treba da definiše vrednosti svih kontrolnih signala unutar procesora – dozvola upisa u registre i brojače, dozvolu brojanja, selekciju multipleksera, selekciju operacije, itd. Ovi signali se postavljaju u zavisnosti od instrukcije koja se trenutno izvršava. Upravljačka jedinica ima prolaze date u Tabeli 4-1.

Tabela 4-1. Prolazi upravljačke jedinice

|  |  |  |
| --- | --- | --- |
| **Prolaz(i)** | **Smer** | **Funkcija** |
| iCLK | in | signal takta |
| inRST | in | signal reseta, aktivan na niskom logičkom nivou |
| iINSTR [14:0] | in | trenutno izvršavana instrukcija |
| oREG\_WE [7:0] | out | dozvole upisa u registre opšte namene |
| oA\_WE, oB\_WE, oC\_WE | out | dozvole upisa u registre oko ALU |
| oIR\_WE | out | dozvola upisa u instrukcijski registar |
| oPC\_WE | out | dozvola upisa u programski brojač |
| oPC\_LOAD | out | dozvola brojanja programskog brojača |
| oMUXA\_SEL [3:0] | out | selekcija za multiplekser prvog operanda |
| oMUXB\_SEL [3:0] | out | selekcija za multiplekser drugog operanda |
| oALU\_SEL [3:0] | out | selekcija operacije ALU |
| oMEM\_WE | out | dozvola upisa u memoriju za podatke |

Pošto u ovoj vežbi ne definišemo rad procesora sa instrukcijama, za sada treba da napravimo praznu upravljačku jedinicu. Alat za sintezu će primetiti da naš procesor ništa ne radi, pa ga neće ni sintetisati. Sinteza će moći da se izvrši tek u narednoj vežbi, kada definišemo rad upravljačke jedinice. Za sada postaviti sve izlaze na proizvoljnu vrednost.

# Vrh hijerarhije procesora

Komponente realizovane u ranijim delovima vežbe ćemo iskoristiti kako bi realizovali procesor. Slika 5-1 prikazuje arhitekturu procesora koji realizujemo. Vaš zadatak je da instancirate sve potrebne komponente, prema Slici 5-1, kao i da ih povežete internim signalima takođe prema Slici 5-1. Komponente procesora su:

* R0 : R7 – osam registara opšte namene,
* A, B – registri za ulazne operande u ALU,
* C – registar za rezultat računanja ALU,
* ZSC – tri flip-flopa za registrovanje statusnih bita (realizovati ih u jednom procesu u vrhu hijerarhije),
* IR – instrukcijski registar, u kome se smešta tekuća izvršavana instrukcija,
  + **Napomena:** pošto je instrukcijski registar širine 15 bita, za razliku od ostalih registara, potrebno je uraditi sledeće:
    - Promeniti *reg.vhd* tako da širina registara bude ulazni parametar (**generic**) podrazumevane vrednosti 16,
    - Pri instanciranju IR registra, postaviti vrednost parametra širine na 15 (**generic map**).
* PC – programski brojač, u kome se smešta adresa trenutno izvršavane instrukcije,
* MUXA, MUXB – multiplekseri koji selektuju operande za smeštanje u registre A i B,
* ALU – aritmetičko-logička jedinica,
* CTRL\_UNIT – upravljačka jedinica.



Slika 5‑1. Arhitektura procesora

Vrh hijerarhije procesora realizovati u datoteci *cpu\_top.vhd*. Prolazi procesora su dati u Tabeli 5-1.

Tabela 5-1. Prolazi vrha hijerarhije procesora

|  |  |  |
| --- | --- | --- |
| **Prolaz(i)** | **Smer** | **Funkcija** |
| iCLK | in | signal takta |
| inRST | in | signal reseta, aktivan na niskom logičkom nivou |
| iINSTR [14:0] | in | instrukcija iz memorije za instrukcije |
| iDATA [15:0] | in | podatak iz memorije za podatke |
| oPC [15:0] | out | vrednost programskog brojača, za adresiranje memorije za instrukcije |
| oDATA [15:0] | out | vrednostzaupisumemorijuzapodatke |
| oADDR [15:0] | out | adresa za memoriju za podatke |
| oWE | out | dozvola upisa u memoriju za podatke |