

8. vaja: Mikroprocesor (2)

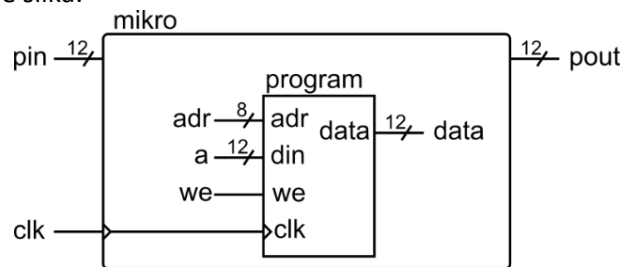
VHDL model s simboličnim ukazi

Model mikroprocesorja v jeziku VHDL bomo naredili bolj pregleden z uporabo simbolov za strojne operacije. Simbole definiramo kot 4-bitne nepredznačene konstante v paketu [procpak.vhd](#), ki ga vključimo na začetek opisa vezja:

```
library work;
use work.procpak.all;
```

- Popravi opis procesorja z ukazi LDA, ADD in JMP, da bo vseboval simbolične konstante in uporabi programski pomnilnik s testnim programom: [program.vhd](#). S stavkom **port map** poveži programski pomnilnik, kot prikazuje slika:

V modelu mikroprocesorja je potrebno deklarirati nov signal **we**, ki bo sprožil pisanje iz akumulatorja v pomnilnik.



- Dopolni opis pomnilnika, da bo omogočal shranjevanje vrednosti. Dodaj proces in stavek, ki ob fronti ure in **we='1'** prenese podatek iz **din** v pomnilnik **ram** na naslovu **adr**.
- V [model procesorja](#) dodaj opis ukaza STA: ob prejemu ukazu, naj postavi **we** na '1', da se bo vrednost zapisala v pomnilnik. Ugotovi v katerem stanju procesorja je potrebno to narediti in preveri delovanje s simulacijo v orodju ModelSim.
- Dodaj ukaz za odštevanje SBT, ki od akumulatorja odšteje podatek (**data**) in shrani rezultat v akumulator. Popravi testni program, da bo vseboval ta ukaz in preveri na simulaciji.

Vhodno-izhodni ukazi

Vhodno-izhodni ukazi omogočajo procesorju komunikacijo z okolico. Ukaz INP bere vrednost iz vhoda **pin** in ga shrani v akumulator, ukaz OUTP pa shrani vsebino akumulatorja v izhodni register **pout**.

- Dodaj v model procesorja oba vhodno-izhodna ukaza in preveri delovanje s testnim programom v katerem zamenjaj ukaz LDA z ukazom INP in STA z ukazom OUTP.

Ostali ukazi

Dopolni opis procesorja še z ostalimi operacijami (glej tabelo), napiši ustrezne testne programe in preizkusi delovanje s simulacijo.

- Logične operacije in pomikanje naredi po zgledu aritmetičnih, pogojne skočne ukaze pa podobno kot JMP.

ukaz	NOTA	JZE	JCS	ANDA	ORA	SHL	SHR
koda	0x0	0x5	0x6	0xC	0xD	0xE	0xF
pomen	negiraj akumulator	skoči, če je a=0	skoči, če je c=1*	logična IN	logična ALI	pomik akumulatorja v levo	pomik v desno

*c je prenosni bit, ki predstavlja prenos pri seštevanju, odštevanju ali pomikanju. Vrednost naj se ob izvedbi operacije shrani v flip-flop.