



### 3. vaja: Sistem s procesorjem

```
0=> lda & x"07",  
1=> outp & x"00",  
2=> lda & x"08",  
3=> add & x"09",  
4=> sta & x"08",  
5=> outp & x"01",  
6=> jmp & x"03",  
7=> x"500",  
8=> x"000",  
9=> x"001"
```

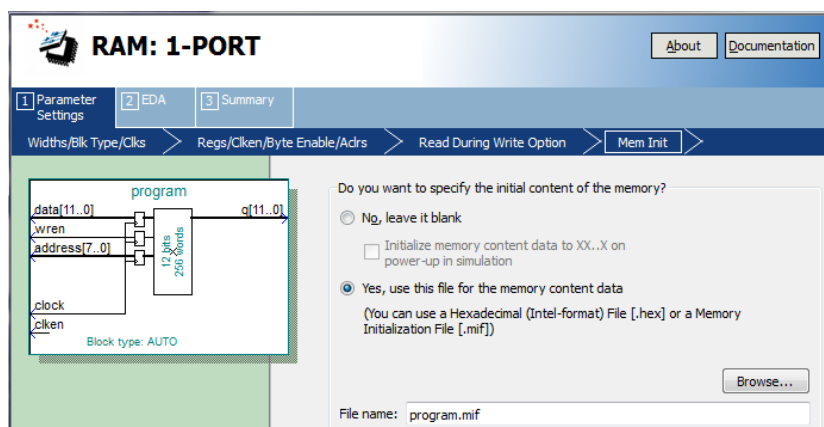
Naredili bomo sistem na čipu, ki je sestavljen iz procesorja in grafičnega krmilnika. S program v procesorju bomo nastavljali prikaz števila na razvojni plošči. Program najprej naloži vrednost x"500" in jo pošlje na izhod z naslovom 0, nato pa v neskončni zanki naloži števec, ga poveča za 1, shrani in pošlje na izhod 1.  
Strojna koda programa bo tokrat zapisana v datoteki vrste Memory Initialization File, ki omogoča spreminjanje vsebine pomnilnika vezja FPGA brez prevajanja celotnega projekta.

## 3.1 Procesorski sistem

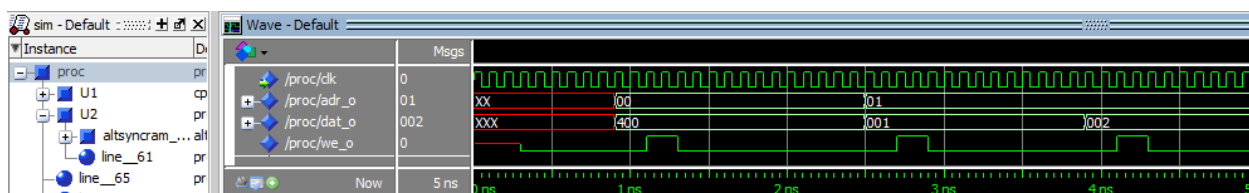
- Prenesi in odpakiraj [datoteke](#) z opisom procesorja v projektno mapo. Pomen datotek:
  - cpu.vhd - opis centralno procesne enote
  - procpak.vhd - knjižnica s kodami ukazov
  - program.mif - vsebina programskega pomnilnika v obliki Memory Initialization File
  - proc.vhd - opis procesorskega sistema s pomnilnikom, ki ga boste dopolnili in vključili v grafični sistem
- Odpri projekt v programu **Quartus**, nato pa odpri vse datoteke s končnico VHD in jih dodaj v projekt (**Project > Add current file to project**).
- V datoteki **proc.vhd** dodaj še sinhroni proces z delilnikom ure za mikroprocesor. Deklariraj dva 24-bitna nepredzačena signala: **del** in **ndel**, ki ju uporabi kot števec ure in modul deljenja ure:

```
if del < ndel then  
    ce <= '0';  
    del <= del + 1;  
else  
    del <= (others => '0');  
    ce <= '1';
```

- Dodaj opis pomnilnika v obliki komponente IP: **Tools > IP Catalog**, pod **Basic Functions**, **On Chip Memory** izberi: RAM: 1-PORT in določi ime: **program**.



- e. Dodaj še en sinhroni proces , ki se proži ob fronti ure in kadar je **ce='1'** in **we\_o='1'** naj vpiše podatek iz **dat\_o** v ustrezni izhodni register:
- **pout0** ob naslovu **adr\_o=0**
  - **pout1** ob naslovu **adr\_o=1**
  - **pout2** ob naslovu **adr\_o=2**
  - **pout3** ob naslovu **adr\_o=3**
- f. Odpre se okno za nastavitve komponente (MegaWizard) in v prvem zavihku nastavi širino pomnilnika 12 bits, v drugem odstani kljukico pri **q** in dodaj kljukico na **Create one clock enable signal...**, tretjega ni potrebno spremeniti, v četrtem pa za vsebino pomnilnika kljukico na: **Yes, use this file...** in z gumbom **Browse...** izberi datoteko **program.mif** ter potrdi vnos IP v projekt.
- g. Za test nastavi `ndel <= x"000001"` in preizkusi delovanje s simulacijo v programu Modelsim, kjer izvedi simulacijo komponente proc in opazuj signale:



- h. Vrednost prvega registra bomo uporabili za nastavljanje delilnika ure, tako da bo 12-bitna vrednost predstavljala zgornje bite, spodnji pa naj bodo na vrednosti `x"001"`:

```
ndel <= pout0 & x"001";
```

## 3.2 Povezava v sistem in testiranje

- a. Odpri datoteko **sistem.vhd** ter deklariraj in poveži komponento **proc**, tako da bodo vezani:
- **clk** na **clk**
  - **rst** na **key(0)**, ki predstavlja tipko S1 na razvojni plošči
  - **pin** na 12-bitni notranji signal **pin** (deklariraj signal !)
  - **pout1** na 12-bitni notranji signal **pout1** (deklariraj signal !)
- b. Poveži 4-bite izhodnih vrat **pout1** na vhod matričnega prikazovalnika (namesto signala key) :
- ```
num <= pout1(3 downto 0);
```
- c. Prevedi program ter preizkusi delovanje na razvojni plošči.
- d. Poskusi spremeniti strojno kodo procesorja brez prevajanja vseh datotek. Odpri datoteko **program.mif** in spremeni vrednost 400, ki je na naslovu 7, v vrednost 100. To število procesor prenese na izhodna vrata **pout0** in predstavlja faktor deljenja ure.

| Addr | +00 | +01 | +02 | +03 | +04 | +05 | +06 | +07 | +08 | +09 | +0a | +0b | +0c |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 000  | 107 | 700 | 108 | 809 | 208 | 701 | 403 | 400 | 000 | 001 | 000 | 000 | 000 |
| 01a  | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 |

Po spremembi programske datoteke izvedi **Processing > Update Memory Initialization file, Processing > Start > Start Assembler** in ponovno naloži kodo na razvojno ploščo.