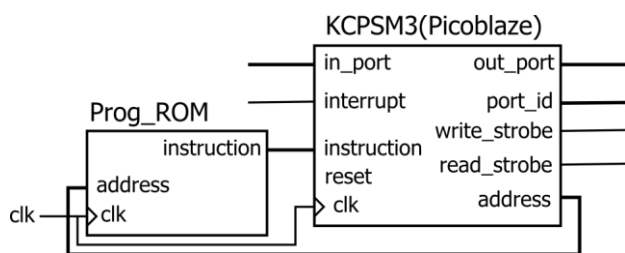
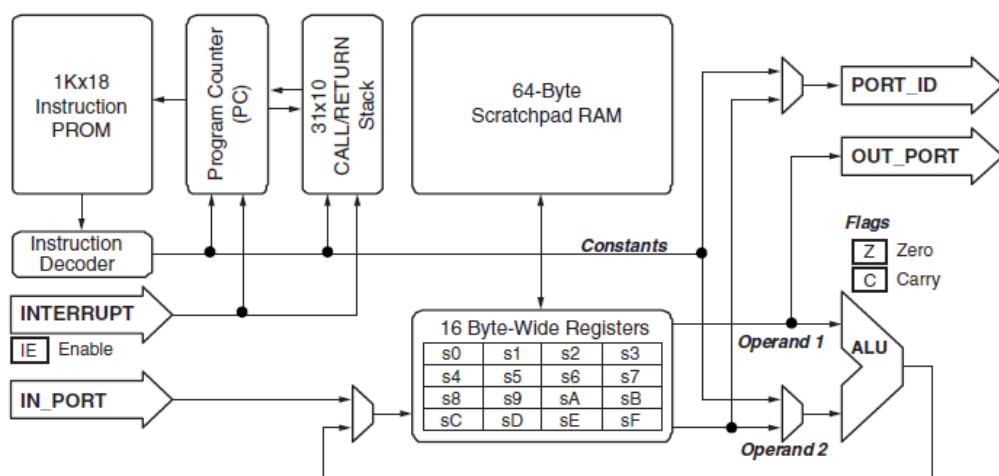


4. Mikroprocesorski sistem



4.1 Mikrokrmilnik PicoBlaze

Uporabili bomo 8-bitno mikrokrmilniško jedro KCPSM3, ki je zgrajeno iz gradnikov vezij FPGA in optimizirano za programirljiva vezja iz družine Xilinx Spartan-3. Vezje je zelo majhno, saj npr. zasede 5% celic v čipu XC3S200. Programska koda, ki ima do 1024 strojnih ukazov, se nahaja v blokovnem pomnilniku (PROM).



UG129_e1_01_051204

Lokalne podatke shranjujemo v 16 registrih z oznakami s0, s1,... sF, za komunikacijo z zunanjimi enotami pa uporabljamo vhodna in izhodna vrata z do 256 naslovi (PORT_ID). Kratek opis nekaterih strojnih ukazov:

LOAD sX, kk	Naloži konstanto kk v register Sx
IN sX, nn	Naloži v register sX vrednost iz vhodnih vrat, PORT_ID=nn
OUT sX, nn	Shrani vrednost registra sX v izhodna vrata, PORT_ID=nn
ADD sX, kk	Prištej registru sX konstanto kk in shrani v Sx, nastavi zastavice C in Z
SUB sX, kk	Odštej konstanto kk od registra sX in shrani v Sx, nastavi C in Z
COMP sX, kk	Odštej konstanto kk od registra sX in nastavi zastavice C in Z
JUMP oznaka	Skok na vrstico z oznako
JUMP Z, oznaka	Skok na vrstico z oznako, če je postavljena zastavica Z (rezultat operacije =0)
JUMP NZ, oznaka	Skok na vrstico z oznako, če ni postavljena zastavica Z (rezultat operacije <>0)

4.2 Prevajalnik in simulator

Program bomo razvili v orodju pBlazIDE.exe, ki vsebuje urejevalnik, prevajalnik in simulator strojnih ukazov. V orodju najprej nastavimo vrsto procesorskega jedra: **Settings > Picoblaze 3**. Za začetek naložimo kratek vzorčni program:

```
; Primer programa, ki prišteva 2 in pošilja vrednost na IZHA
VHDL      "ROM_blank_s3.vhd", "prog_rom.vhd", "prog_rom"

IZHA      DSOUT      0          ; naslov PORT_ID=0

          DINT        ; onemogoči prekinitve
          LOAD        s0, 10    ; naloži začetno vrednost
loop:     OUT         s0, IZHA   ; pošlji na zihod
          ADD         s0, $02    ; prištej 2
          JUMP        loop      ; ... in ponovi
```

Prevajanje programa sprožimo z modro ikono (Assemble), če ni napak lahko izvajamo simulacijo. Delovanje najlažje opazujemo s simulacijo strojne kode po korakih (F7).



4.3 Naloga

1. Ustvari mapo za projekt mikroprocesorski sistem in prenesi datoteke: ROM_blank_s3.vhd, KCPSM3.vhd, tesp.psm in pBlazIDE.exe.
2. Napiši program, v katerem definiraj ena vhodna vrata z imenom EN (naslov 0) in dvoje izhodnih vrat IZHA (naslov 0) in IZHB (naslov 1), preko katerih bomo nastavljali parametre elipse. Vrednosti shrani v registrih (npr. s0 in s1). Preizkusi delovanje programa v simulatorju pBlazIDE.
 - a. Program na začetku nastavi vrednosti parametrov na 10 in 5 ter ju prenese na IZHA in IZHB,
 - b. nato naj v zanki bere vhod (EN), dokler se ne postavi na vrednost različno od 0;
 - c. povečaj prvi parameter za 2 in drugega za 5 ter prenesi vrednosti na izhodna vrata;
 - d. če je prvi parameter enak 100 naj gre na korak a., sicer pa na korak b.
3. Naredi strukturni projekt v orodju ISE v katerega vključi obstoječe datoteke kcpsm3.vhd, prog_rom.vhd in ProcSist.vhd. Dopolni datoteko s strukturnim opisom procesorskega sistema, tako da bo na izhodu dekodirnik signalov *port_id* in *write_strobe*, ki bo nastavljal kontrolne izhode *ena* in *enb*, kadar bo program pisal na izhodna vrata. Preizkusi delovanje sistema na simulatorju.

