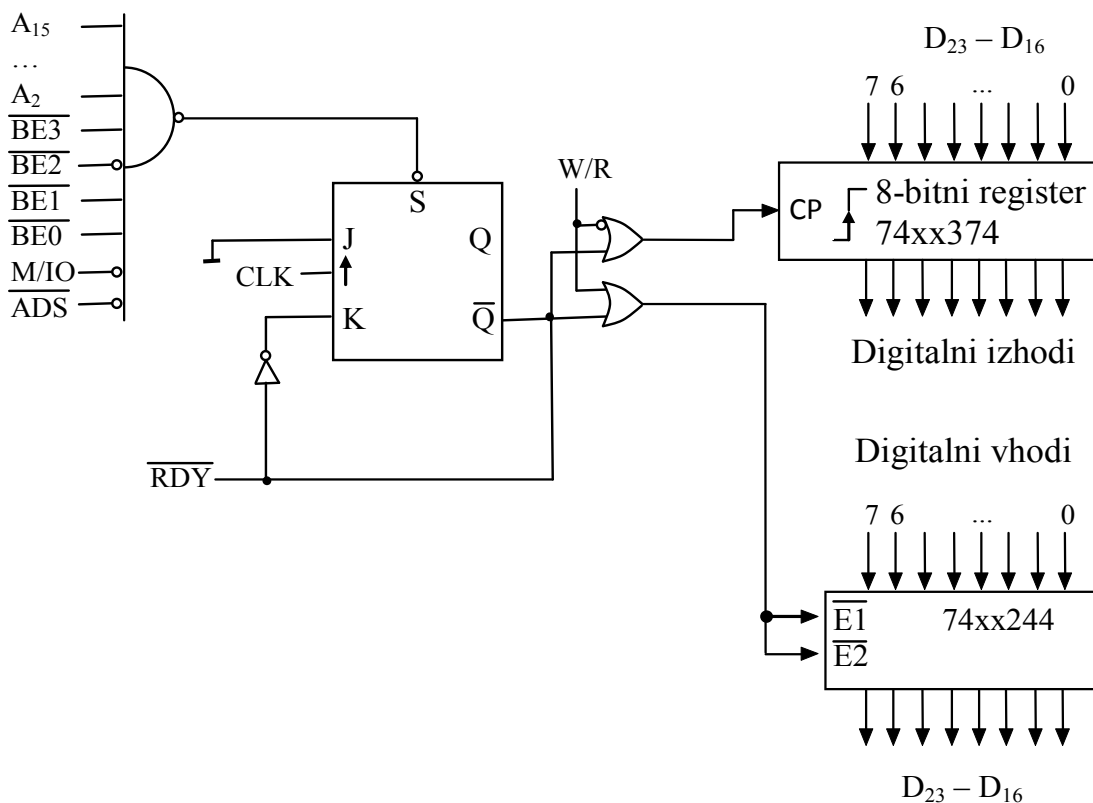


## REŠITVE:

**Naloga 1****Naloga 2**

```

PSP:  sub     lr,lr,#4           /* ustrezno popravi povratni naslov */
      stmfd   sp!, {r0-r2, lr} /* na sklad registri, povratni naslov. */

      ldr     r0, =ADC_BASE
      ldr     r1, [r0, #ADC_DATA] /* beremo podatkovni register */
      mov     r2, #1 << 7
      str     r2, [r0, #ADC_CONTROL] /* nastavimo bit 7 na 1, ostali biti 0 */
      ldr     r0, =BUFFER
      ldr     r2, ODMIK           /* beremo vrednost odmika */
      strh   r1, [r0,r2]         /* shranimo v krožni izravnalnik */
      add    r2, r2, #2
      and    r2, r2, #0x1F       /* popravimo naslov, da je po modulu 32 */

/* bolj splošna rešitev, ki ne zahteva, da je velikost izravnalnika potenca
števila 2, uporablja par ukazov cmp in mov:
cmp     r2, #32
moveq  r2, #0                  */

      str     r2, ODMIK         /* shranimo popravljen odmik */

      ldr     r0, =AIC_BASE
      str     r0, [r0, #AIC_EOICR] /* slepo piši v AIC_EOICR */

      ldmfd   sp!, {r0-r2, pc}^ /* vrni se iz PSP in obnovi CPSR */

```

### Naloga 3

Ker pri tvorjenju signala za izbiro čipa nekateri signali potujejo zaporedno skozi tri vezja, se signal za izbito čipa stabilizira 15 ns ( $3 \cdot 5\text{ns}$ ) za ADS, torej  $18\text{ ns} + 15\text{ ns} = 33\text{ ns}$  po začetku periode, v kateri se je prenos začel. Naprava mora podatke dati na podatkovno vodilo vsaj 5 ns pred zaključkom urine periode, v kateri se prenos zaključuje. Če prenosa ne podaljšamo, ta traja 2 urini periodi in v tem primeru ima naprava na voljo  $2 \cdot 30 - 33\text{ ns} - 5\text{ ns} = 22\text{ ns}$ . Ta čas bi moral presegati dostopni čas naprave, torej je potrebno prenos podaljšati za  $(140\text{ ns} - 22\text{ ns}) / 30\text{ ns} = 4$  urine periode (zaokroženo navzgor na celo število urinih period). Prenos torej traja **6 urinih period oziroma 180 ns**.

### Naloga 4

Pri zaporednih dostopih do naslovov  $00000014_{16}$  in  $00000214_{16}$  imamo dostop do iste vrstice znotraj istega modula, pri zaporednih dostopih do naslovov  $00000014_{16}$  in  $00020014_{16}$  imamo dostop do različnih vrstic v istem modulu, pri zaporednih dostopih do naslovov  $00000014_{16}$  in  $02000014_{16}$  pa imamo dostop do vrstic v različnih modulih.

Zaporedni dostop do naslovov  $00000014_{16}$  in  $00000214_{16}$  je najhitrejši saj potrebujemo le eno odpiranje vrstice in nato lahko po poteku CAS latence v dveh zaporednih urinih periodah beremo podatke.

Malo počasnejši je dostop do naslovov  $00000014_{16}$  in  $02000014_{16}$  in sicer le za eno urino periodo, ker se morata izvesti dva ukaza za odpiranje vrstice in dva ukaza za branje. Ker so moduli neodvisni, si lahko ukaza za odpiranje vrstic in ukaza za branje sledita z zamikom le ene urine periode.

Pri zaporednem dostopu do naslovov  $00000014_{16}$  in  $00020014_{16}$  porabimo največ časa, saj moramo najprej odpreti prvo vrstico, prebrati podatek, vrstico zapreti, odpreti drugo vrstico in prebrati drugi podatek. Zapiranje in ponovno odpiranje vrstice v istem modulu traja več urinih period in zato je zaporedni dostop do naslovov  $00000014_{16}$  in  $00020014_{16}$  najpočasnejši.