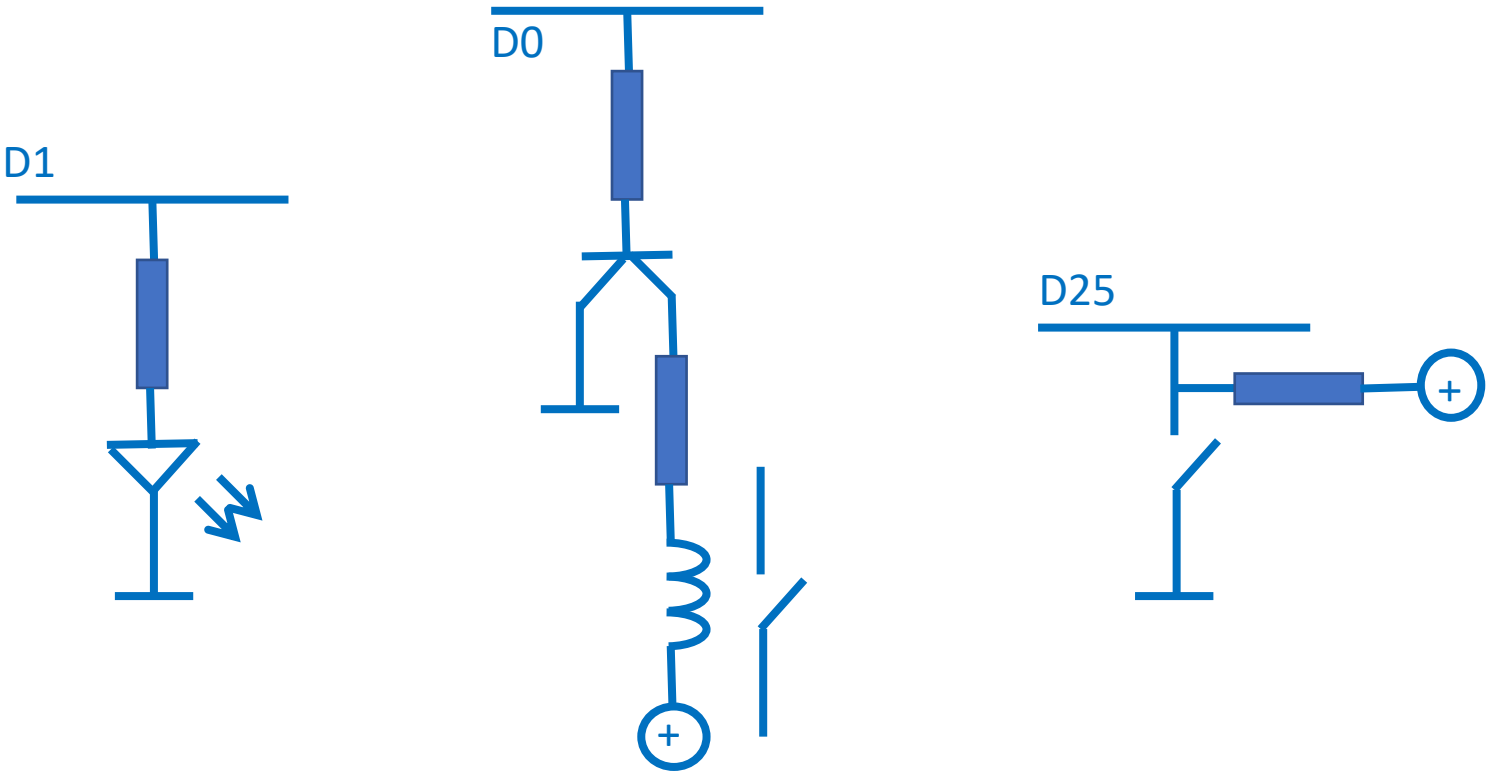
Vanjska jedinica GPIO

Građa
Bezuvjetni prijenos pomoću GPIO
Uvjetni prijenos pomoću GPIO

Razmislite, prodiskutirajmo,…

- Da li možemo spojiti npr LED ili relej direktno na podatkovnu liniju našeg procesora ?
- Da li možemo spojiti neku tipku/prekidač direktno na podatkovnu liniju ?



GPIO - uvod

- **GPIO** (*General Purpose Input Output*) je vrlo jednostavna vanjska jedinica pomoću koje se može čitati ili pisati na tzv vrata (sklop za digitalni ulaz/izlaz)

- GPIO ima dvoja vrata (*port A i B*)
  - Svaka vrata su širine 8 bitova i izvedena su kao 8 jednobitnih priključaka
- Svakom od 8 priključaka može se nezavisno zadati smjer - ili ulaz ili izlaz

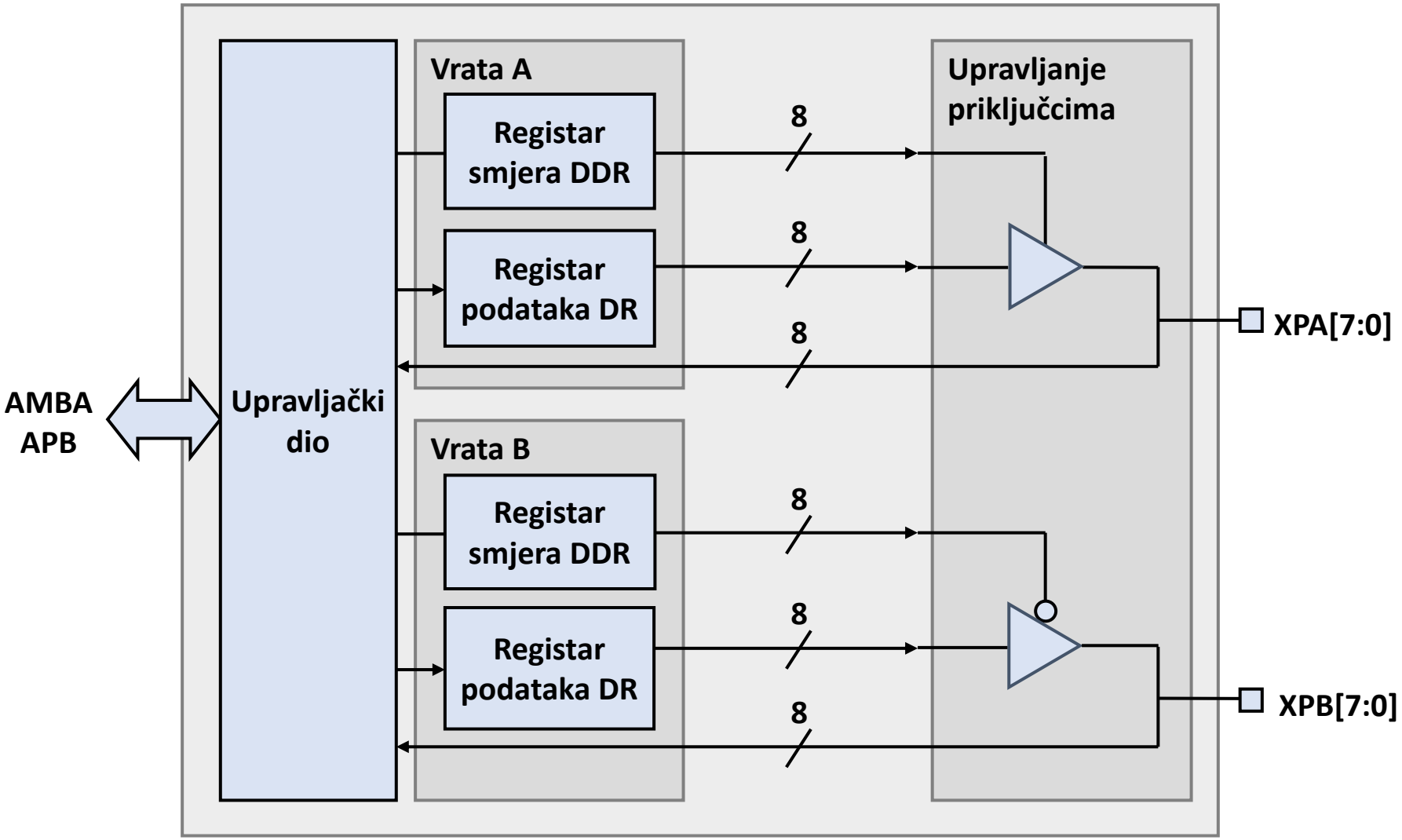
- Čitanja ili pisanja s GPIO-a su bezuvjetna

- GPIO nema sklopovsku sinkronizaciju s periferijom*
  - Ako je potrebna sinkronizacija onda se ona mora koristeći GPIO ostvariti programski, a programski se može ostvariti i uvjetni prijenos (što ćemo vidjeti naknadno)

---

* Neke (starije) jedinice slične namjene imaju sklopovski ugrađene linije za rukovanje pa programer ne treba brinuti za sinkronizaciju UI-jedinice i uređaja

GPIO - građa


AMBA
APB
Upravljački
dio
Vrata A
Registar
smjera DDR
Registar
podataka DR
Vrata B
Registar
smjera DDR
Registar
podataka DR
8
8
8
8
8
8
Upravljanje
priključcima
XPA[7:0]
XPB[7:0]

GPIO - građa

- Dijelovi* GPIO-a su:
  - Upravljački dio (upravljanje, APB sučelje, pristup *vratima*)
  - Vrata **A** i vrata **B** (*port A* i *port B*, skraćeno **PA** i **PB**)
    - Svaka vrata imaju svoj podatkovni registar **DR** (*Data Register*): **PA_DR** i **PB_DR**
    - Svaka vrata imaju svoj registar smjera **DDR** (*Data Direction Register*): **PA_DDR** i **PB_DDR**
  - Dio za upravljanje priključcima **XPA** i **XPB** od vrata A i B

- ARM preko APB sabirnice komunicira s GPIO-om
  - program prvo mora definirati smjer pojedinih priključaka/bitova upisima u DDR
  - program nakon toga čita i šalje podatke preko registara DR

---

* Većina naziva (imena registara, signala, razne kratice itd.) u GPIO-u počinje prefiksom GPIO čime ti nazivi postaju nepregledni i predugački. Gdje god možemo ispuštamo ovaj prefiks i po potrebi dodajemo podcrte da imena budu preglednija.

GPIO - adrese

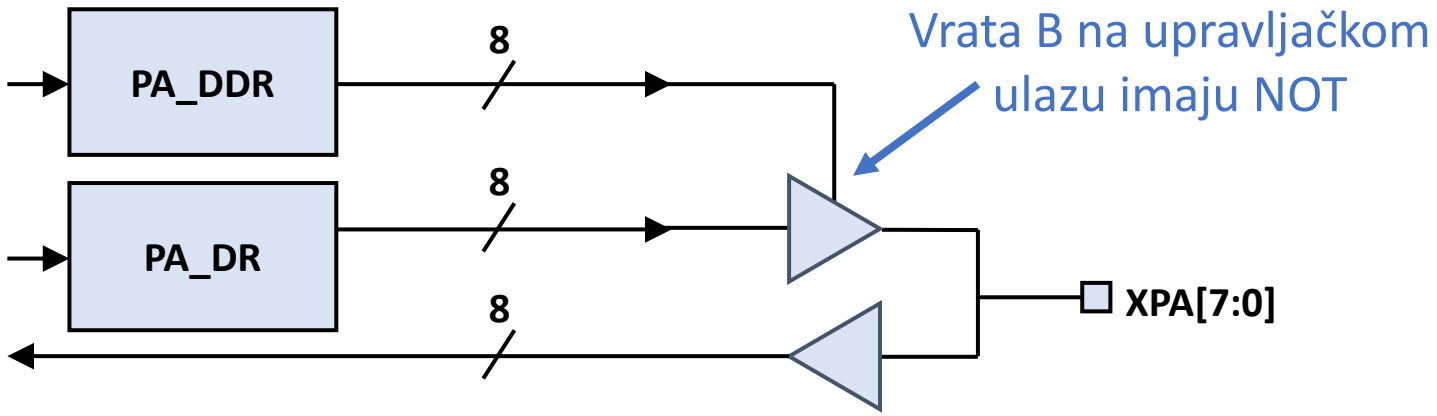
Adresa	Naziv registra	Opis
bazna_adr + 0	PA_DR	8-bitni registar podataka, vrata A
bazna_adr + 4	PB_DR	8-bitni registar podataka, vrata B
bazna_adr + 8	PA_DDR	8-bitni registar smjera podataka, vrata A
bazna_adr + C	PB_DDR	8-bitni registar smjera podataka, vrata B

- Inicijalno, svi registri GPIO-a imaju vrijednost nula
- Prilikom čitanja i pisanja na GPIO možemo, osim naredaba za bajtove, koristiti i naredbe za riječi i poluriječi
  (viši bajtovi se jednostavno zanemaruju)

Registri smjera DDR

- Upisom 0 ili 1 u **pojedini bit** registara smjera DDR definira se smjer odgovarajućeg priključka na vratima
  - PA_DDR: 0=ulaz, 1=izlaz
  - PB_DDR: 1=ulaz, 0=izlaz
  - PAŽNJA: Bitovi u DDR za vrata B imaju **obrnuto** značenje od onih za vrata A !!
- Inicijalno: svi bitovi u registrima smjera su nula: A=ulaz, B=izlaz

Shema za vrata A

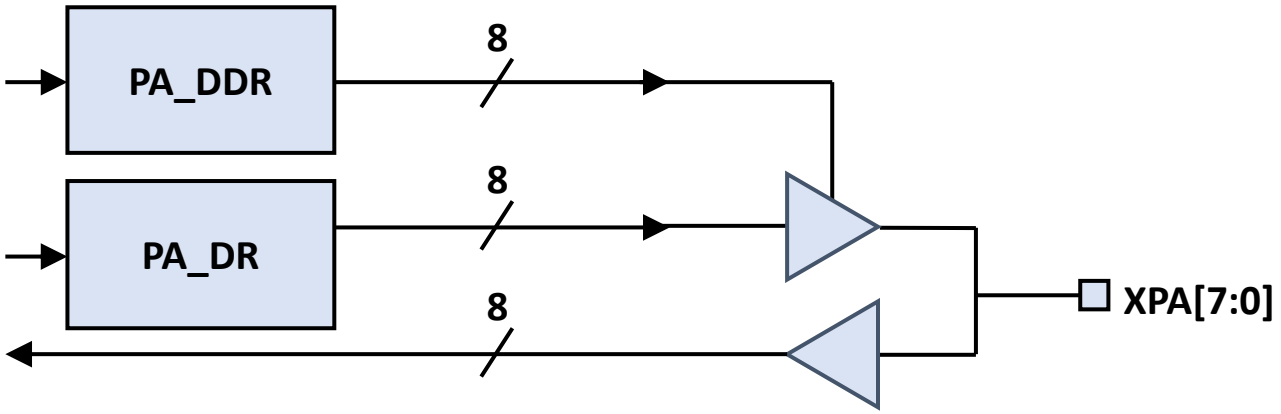


Podatkovni registri DR

- Operacija **pisanja u DR** - upis se obavlja u cijeli registar DR
  - Stanje registra DR će proslijediti samo na izlazne priključke
  - Stanje registra DR ne utječe na ulazne priključke (stanje ulaznih priključaka određuje vanjski uređaj)

- Operacija **čitanja iz DR** - čita se stanje priključaka
  - Preko izlaznih priključaka čita se stanje koje na njima „drži” registar DR
  - Preko ulaznih priključaka čita se stanje koje na njima „drži” vanjski uređaj

Shema za vrata A



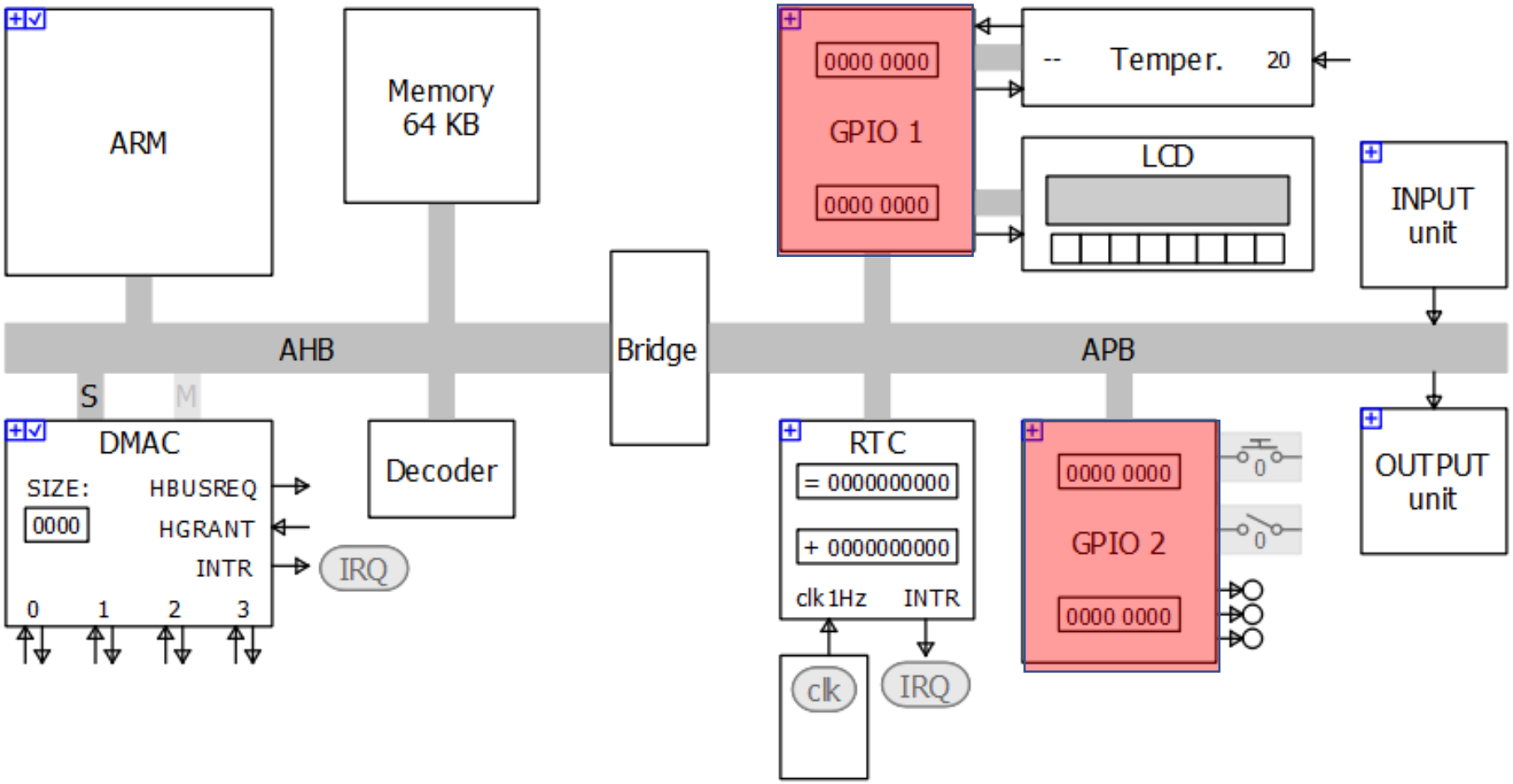
Programiranje GPIO-a

- Prije rada s GPIO-om obavezno treba zadati smjerove svih bitova koji se koriste (izuzetak je slučaj kada svi korišteni bitovi odgovaraju inicijalnim smjerovima)

- Bitovi koji se ne koriste mogu se zadati bilo kako, ali preporuča ih se zadati da budu ulazni

- Nakon toga, mogu se početi slati/primati podatci sa željenih bitova

GPIO u SSPARCSS-u



Bazne adrese vanjskih jedinica:

GPIO1 = FFFF 0F00 RTC = FFFF 0E00 INPUT = FFFF 0D00

GPIO2 = FFFF 0B00 OUTPUT = FFFF 0C00

GPIO2 u SSPARCSS-u

- XPA0 = Tipka (ulaz)
- XPA1 = Sklopka (ulaz)
- XPA2 = ništa
- XPA3 = ništa
- XPA4 = ništa
- XPA5 = Crveni LED (izlaz)
- XPA6 = Žuti LED (izlaz)
- XPA7 = Zeleni LED (izlaz)

- XPB[0..7] = ništa

Bezuvjetni prijenos pomoću GPIO-a

Bezuvjetni prijenos sa tipki, sklopki, LED-dioda i LCD-prikaznika

GPIO - bezuvjetni prijenos - primjer

GPIO2 je na adresi FFFF 0B00.

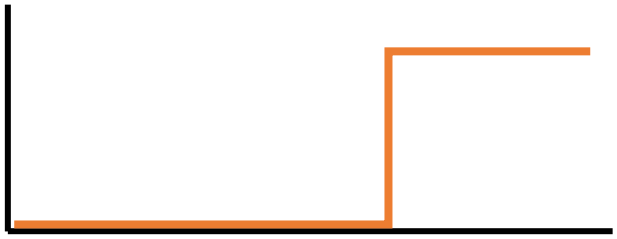
Na bit 1 vrata A spojena je sklopka, a na bit 5 vrata A spojen je crveni LED (tipka, žuti i zeleni led su također spojeni, ali ih nećemo koristiti).

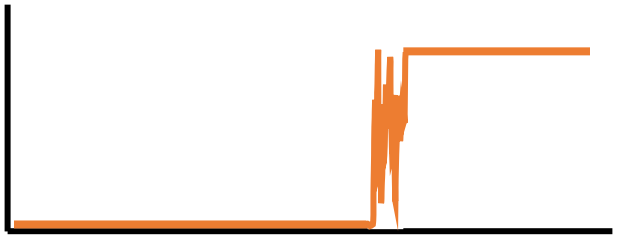
Kad je sklopka uključena, s nje se očitava stanje 1, a inače se očitava stanje 0.

Slanjem broja 0 LED se isključuje, a slanjem 1 se uključuje.

Napisati program koji treba beskonačno ispitivati je li sklopka uključena i samo za to vrijeme držati LED uključen.

Napomena: ovdje se promatra idealna sklopka, a u stvarnosti treba napraviti tzv. *debouncing* (ali ne na ovom predmetu)





GPIO - bezuvjetni prijenos - primjer


```
INIT   LDR    R0, GPIO          ; dohvati adresu GPIO-a
       MOV    R1, #0b11100000   ; smjer vrata A: XPA1=0 ULAZ  - sklopka
                                ;                  XPA5=1 IZLAZ - LED
                                ;                  ostalo: kako je spojeno
SMJER  STR    R1, [R0, #8]      ; pošalji smjer u PA_DDR

CITAJ  LDR    R1, [R0, #0]      ; pročitaj stanje sklopke (tj. PA_DR)
       TST    R1, #0b00000010   ; ispitaj bit sklopke (bit 1)

UKLJ   MOVNE  R1, #0b00100000   ; XPA5 = 1 - uključivanje LED-a
ISKLJ  MOVEQ  R1, #0b00000000   ; XPA5 = 0 - isključivanje LED-a
       STR    R1, [R0, #0]      ; pošalji na LED (tj. PA_DR)
       B      CITAJ

GPIO   DW     0xFFFF0B00        ; adresa GPIO-a
```

GPIO - bezuvjetni prijenos - primjer

(ne može se simulirati na SSPARCSS-u)

GPIO je na adresi FFFF0000 i preko njega ARM upravlja proizvodnim procesom

Na vrata B na bit 0 spojen je relej (električki upravljani prekidač) kojim se može isključiti (0) ili uključiti (1) strojeve u procesu. Na bit 6 spojen je zeleni, a na bit 7 crveni LED (uključuju se stanjem 1, a isključuju stanjem 0).

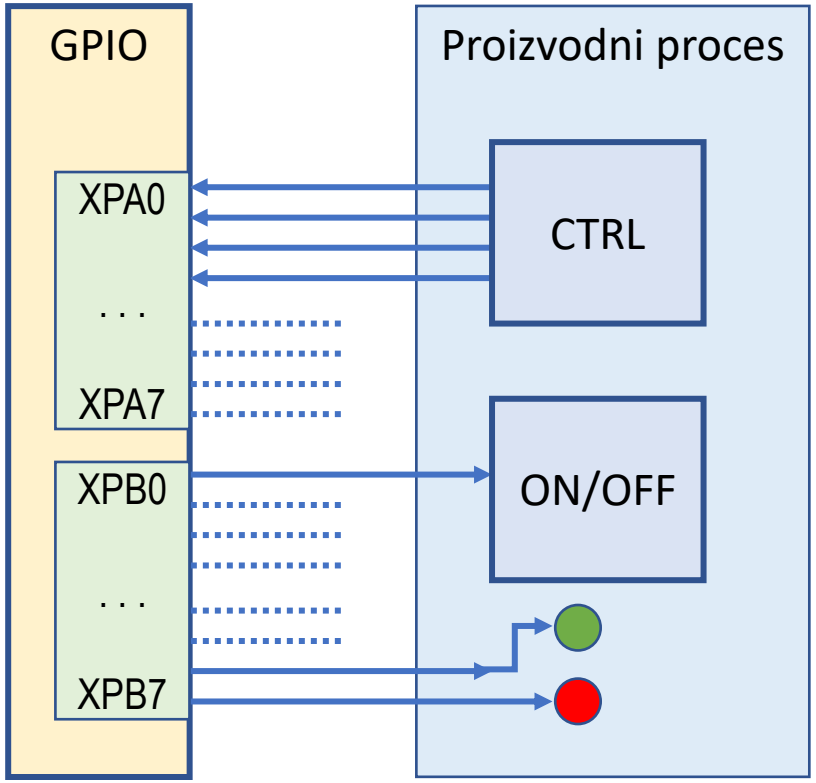
Na vrata A na 4 najniža bita spojen je kontrolni uređaj CTRL koji mjeri 4 temperature unutar procesa. Ako pojedina temperatura prijeđe dozvoljenu granicu, CTRL daje 1 na odgovarajućem bitu, a inače daje 0.

ARM treba isključiti strojeve u procesu ako bilo koja temperatura postane nedozvoljena i treba ih opet uključiti kad sve temperature postanu dozvoljene. Dok su strojevi uključeni treba svijetliti zeleni LED, a kad su isključeni treba svijetliti crveni LED. Upravljanje procesom se ponavlja svake sekunde (clock je 10 MHz) i odvija se beskonačno.

GPIO - bezuvjetni prijenos - primjer

Prijedlog rješenja:

- Inicijalizacija GPIO-a nije potrebna jer inicijalni smjerovi bitova odgovaraju zadatku
- Glavni program će izvoditi beskonačnu petlju
  - Na početku petlje čita se stanje od CTRL
  - Ispituju se bitovi temperature
  - Na temelju toga se proces ili uključi ili isključi te se upali/ugasi odgovarajući LED
  - Čeka se jedna sekunda
- Pristup procesu odvija se bezuvjetno



GPIO - bezuvjetni prijenos - primjer


```
   ORG  0
   MOV  R13, #0x10000 ; stog

   ; dohvat adrese GPIO-a
   LDR  R1, GPIO

   ; inicijalizacija nije nužna
   MOV  R0, #0
   STRB R0, [R1, #0x8]
   STRB R0, [R1, #0xC]

PETLJA
   ; dohvat stanja od CTRL
   ; (port A - 4 najniža bita)
   LDRB R0, [R1]
   AND  R3, R0, #0b1111

   ; ako su sve temperature OK,
   ; onda će R3 biti 0b0000
   CMP  R3, #0
```

```
   ; uključi strojeve i zeleni LED
   MOVEQ  R0, #0b01000001

   ; isključi strojeve i
   ; uključi crveni LED
   MOVNE  R0, #0b10000000

   ; upravljanje procesom
   ; (port B - bitovi 0, 6 i 7)
   STRB   R0, [R1, #4]

   BL     CEKAJ_1_SEK
   B      PETLJA


   ; bazna adresa GPIO-a
GPIO   DW   0xFFFF0000
```

GPIO - bezuvjetni prijenos - primjer


```
CEKAJ_1_SEK                     ; trajanje u ciklusima

      STMFD   R13!, {R0}        ; N+1 = 2

      LDR     R0, KONST         ; 3
LOOP  SUBS    R0, R0, #1        ; KONST x 1
      BNE     LOOP              ; (KONST-1) x 3  + 1 x 1

      LDMFD   R13!, {R0}        ; N+2 = 3
      MOV     PC, LR            ; 3

KONST DW 2500000
```

- 1 sekunda na 10 MHz je 10 000 000 ciklusa.
- Približno trajanje petlje:

  KONST x (1+3) = 4 x KONST = 10 000 000 ciklusa
- KONST mora biti ≈ 2 500 000
- (Precizniji izračun bio bi da umanjimo za 11 ciklusa pa bi KONST bila 2499996 ali je to u ovom primjeru zanemarivo)

GPIO - bezuvjetni prijenos - primjer

Napomena: Iskorištenost procesora u ovom primjeru je vrlo malena, jer veliku većinu vremena (10 milijuna ciklusa) provodi u praznoj petlji (*busy waiting*), a koristan posao traje samo desetak ciklusa. Ovo je prihvatljivo samo ako je to jedina zadaća koji računalo treba obavljati. Ovo je također loše rješenje ako računalo ima baterijsko napajanje.

Zadatak: Promijenite rješenje tako da u procesu postoje 4 releja koji upravljaju sa 4 stroja i da 4 temperature odgovaraju svaka svojem stroju. Treba isključiti samo onaj stroj kojemu temperatura postane nedozvoljena. Ako je bilo koji stroj isključen, onda treba svijetliti crveni LED, a zeleni LED svijetli samo kad sva 4 stroja rade.

LCD-prikaznik

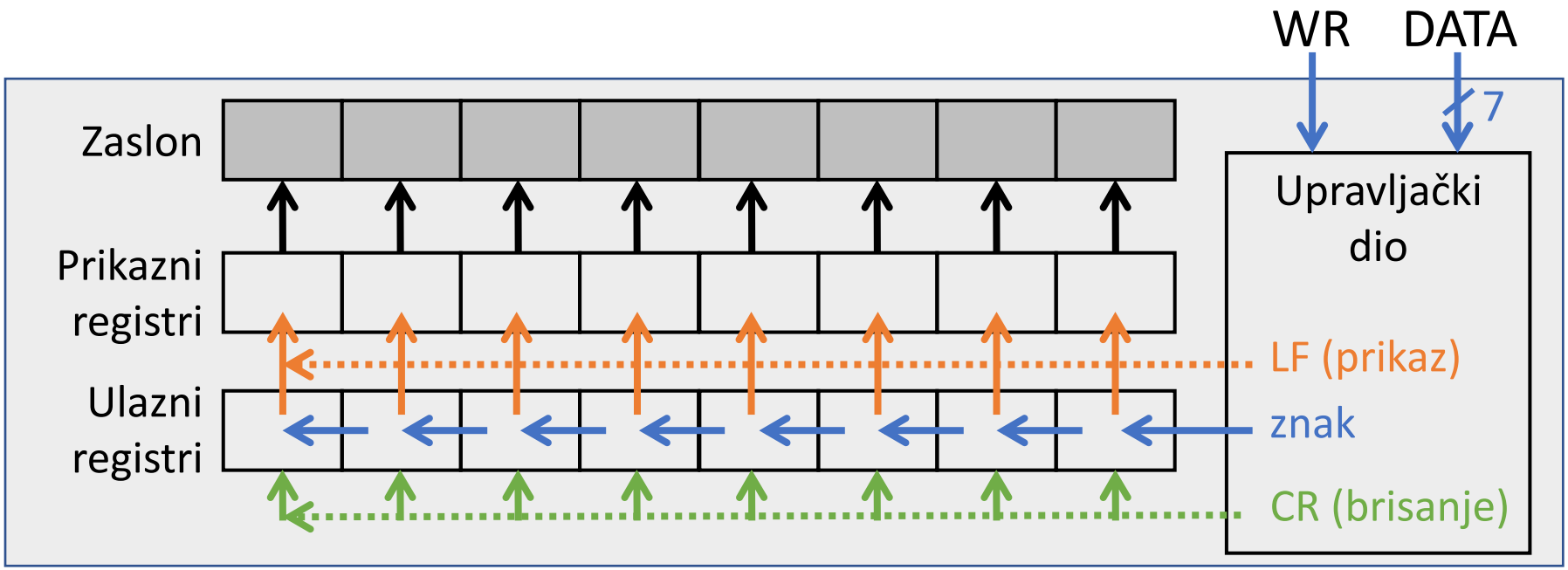
(ili kraće LCD)

Korištenje LCD-a spojenog na GPIO

Bezuvjetni prijenos podataka na LCD

LCD

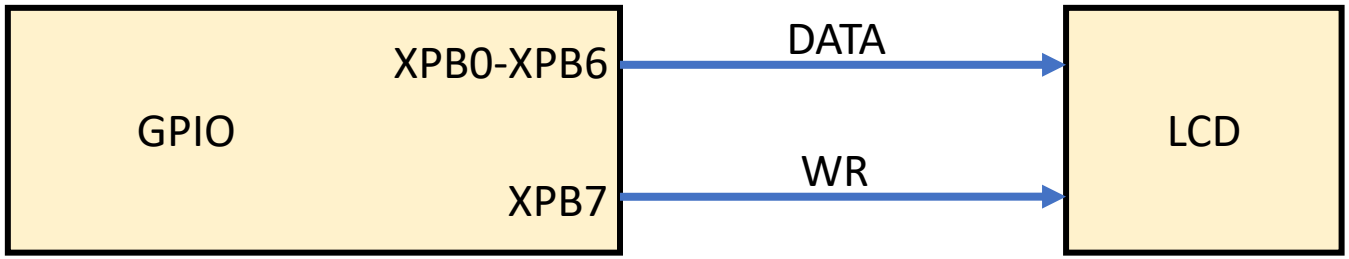
- LCD je hipotetski izlazni uređaj koji prikazuje osam ASCII-znakova na zaslonu
- LCD ima 8 priključaka:
  - **7 ulaznih DATA** priključaka za primanje ASCII-znaka
  - **1 ulazni** priključak **WR** (aktivan visoko) signalizira da je znak na DATA valjan
- LCD ima 8 ulaznih registara preko kojih mu se šalju ASCII-znakovi
- LCD ima 8 prikaznih registara u kojima pamti znakove koje trenutno prikazuje



LCD - spajanje na GPIO

- Priključci LCD-a i GPIO-a se mogu spojiti bilo kako, ali...
  - uobičajeno ćemo ih spajati na jedna vrata od GPIO
  - pogodnija su vrata B jer njihov inicijalni smjer (izlazni) odgovara smjeru priključaka LCD-a
    - Uočite: smjerovi međusobno spojenih priključaka na LCD-u i na GPIO-u su obratni

  - DATA priključci spojeni su na nižih 7 bita XPB0-XPB6 (izlazni)
  - Kontrolni priključak WR spojen je na najviši bit XPB7 (izlazni)



LCD - način rada

- Slanje znaka:
  1. Na bitove DATA treba postaviti ASCII-kôd znaka kojeg se želi poslati LCD-u
  2. Na bit WR treba poslati pozitivan impuls (0→1→0). Impuls uzrokuje da upravljački dio primi znak sa DATA

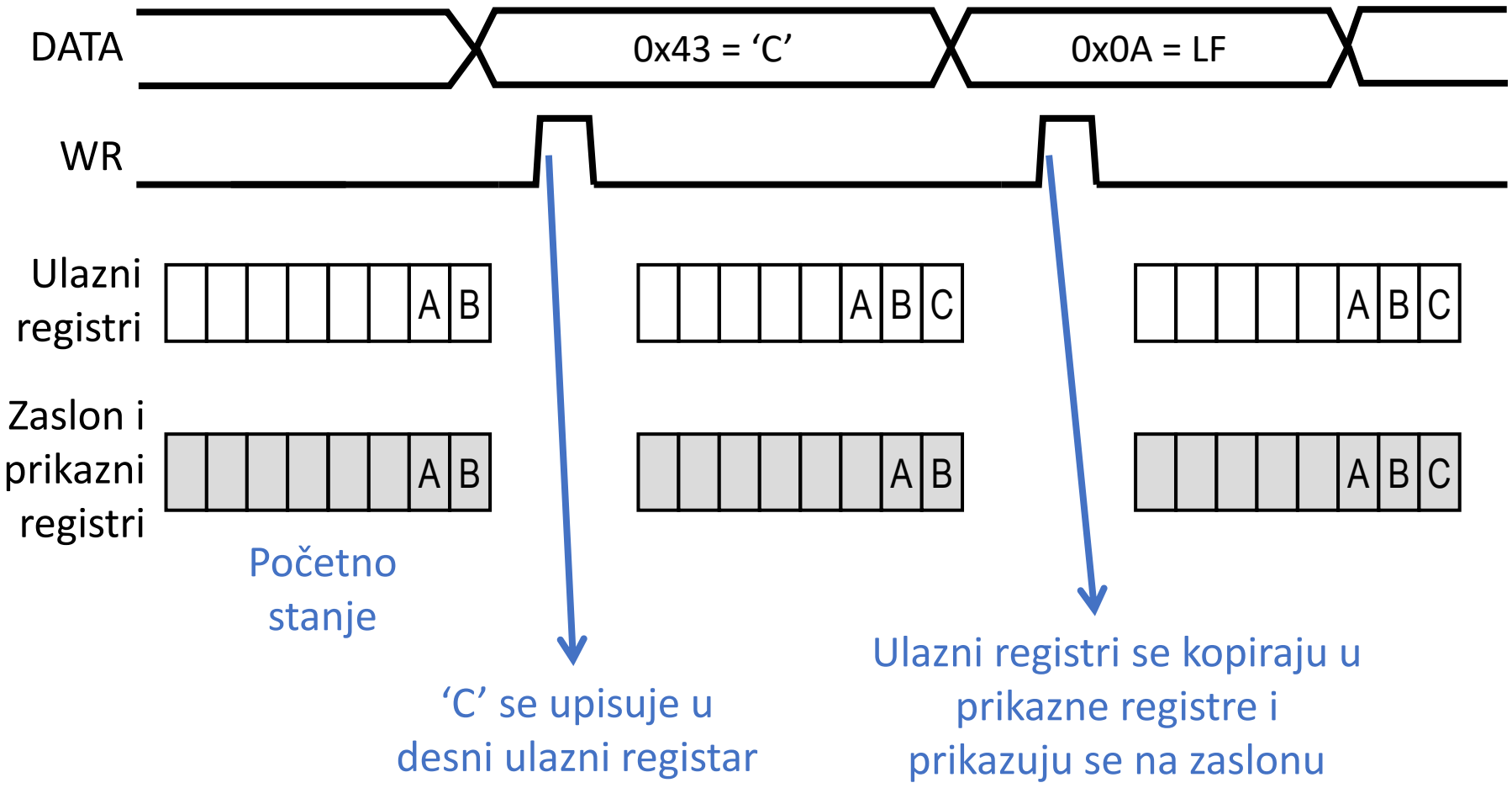
- Pretpostavka je da je **LCD dovoljno brz** da primi znak u bilo kojem trenutku

- Ovakva sinkronizacija između GPIO-a i LCD-a je zapravo jednolinijska sinkronizacija koju inicira pošiljatelj tj. GPIO (samo što se sinkronizacijska linija zove WR, a ne STROBE)

LCD - način rada

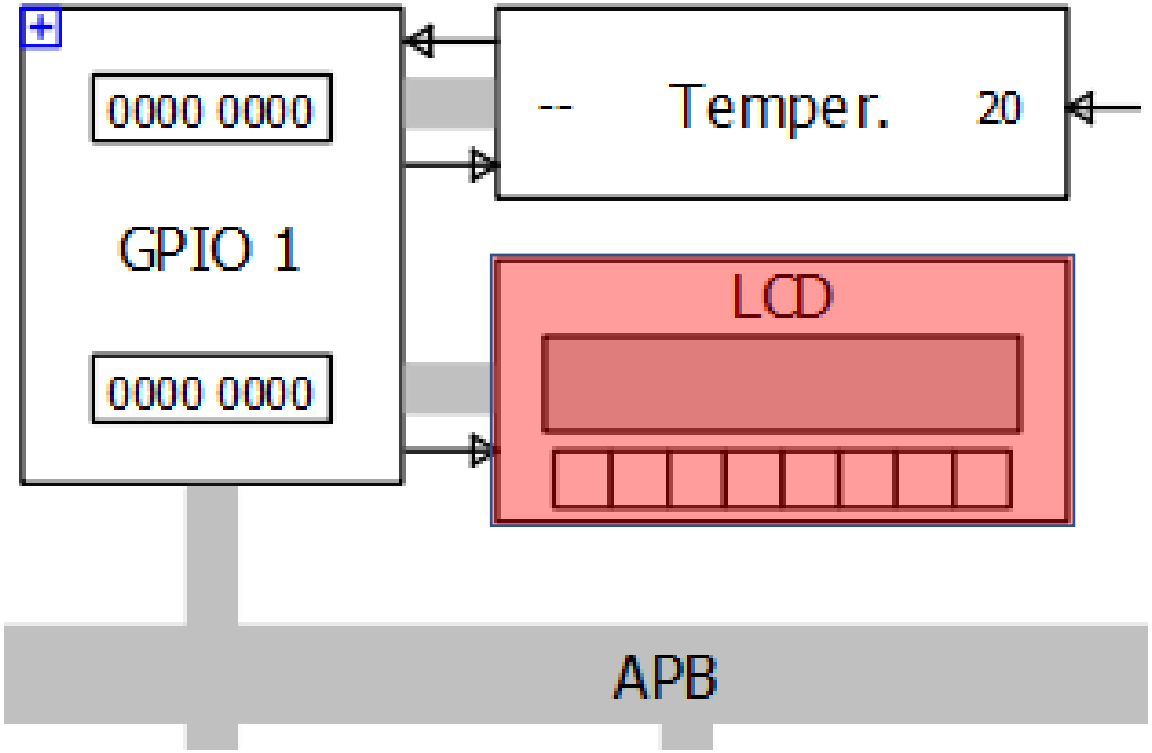
- Ponašanje LCD-a u ovisnosti o poslanom znaku:
  - Ako se pošalje običan **ispisivi** ASCII-znak, onda se on upisuje u desni ulazni registar. Svi ostali znakovi u ulaznom registru pomiču se za jedno mjesto ulijevo, a prethodni krajnje lijevi znak se gubi.
  - Ako se pošalje ASCII-znak **LF** (*line feed*) s kôdom 0x0A, onda se on ne pamti u ulaznom registru, nego uzrokuje da se svi ulazni registri kopiraju u prikazne registre (koji se odmah prikazuju na zaslonu)
  - Ako se pošalje ASCII-znak **CR** (*carriage return*) s kôdom 0x0D, onda se on ne pamti u ulaznom registru, nego uzrokuje da se u svih osam ulaznih registara upiše ASCII kôd 0x20 što je znak praznine/razmaka (to je efektivno „brisanje” ulaznih registara)
  - Ako se pošalje **neispisivi** ASCII-znak, on se zanemaruje i ništa se ne događa.

Vremenski dijagram



LCD u SSPARCSS-u

- U simulatoru je implementiran LCD
- LCD je spojen na vrata B od sklopa GPIO_1



GPIO - bezuvjetni prijenos - primjer

Na vrata B jedinice GPIO spojen je LCD.

Na LCD-u treba prikazati tekst „ARM”. Za slanje pojedinih znakova treba koristiti potprogram LCDWR.

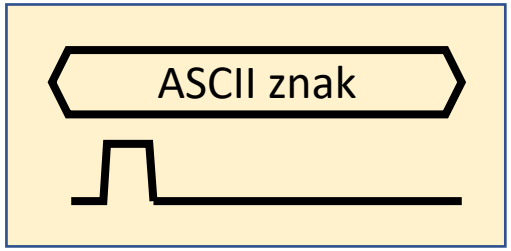
Napisati potprogram LCDWR koji šalje znak na LCD. Parametri potprograma su ASCII-kôd znaka (prenosi se registrom R0) i adresa podatkovnog registra od onih vrata GPIO-a na koja je spojen LCD (prenosi se registrom R1).

GPIO - bezuvjetni prijenos - primjer


```
      ORG 0
MAIN  MOV SP, #0x10000    ; stog
      MOV R1, #0x100      ; R1=100 jer je tu zapisana adresa GPIO-a
      LDR R1, [R1]        ; u R1 se učitava bazna adresa GPIO-a

      ADD R1,R1,#4        ; u R1 je adresa PB DR (jer ju koristi LCDWR)

PRIKAZ
      MOV R0, #0x0D       ; slanje znaka 0xD => briše se ulazni registar
      BL  LCDWR
      MOV R0, #0x41         ; slanje znaka ‘A'
      BL  LCDWR
      MOV R0, #0x52         ; slanje znaka ‘R'
      BL  LCDWR
      MOV R0, #0x4D         ; slanje znaka ‘M'
      BL  LCDWR
      MOV R0, #0x0A       ; slanje znaka 0xA => ispis znakova na zaslon
      BL  LCDWR

      SWI 0x123456

      ORG 0x100
      DW  0xFFFF0F00      ; adresa GPIO-a
```

GPIO - bezuvjetni prijenos - primjer


```
LCDWR STMFD R13!, {R0}
      AND R0, R0, #0x7F  ; postavi bit 7 u nulu (za svaki slučaj, jer
                         ; u R0 je tu mogla biti 1) i pošalji znak
      STRB R0, [R1]
      ORR R0, R0, #0x80  ; postavi bit 7 u jedan (podigni impuls)
      STRB R0, [R1]
      AND R0, R0, #0x7F  ; postavi bit 7 u nulu (spusti impuls)
      STRB R0, [R1]
      LDMFD R13!, {R0}
      MOV PC, LR         ; povratak
```



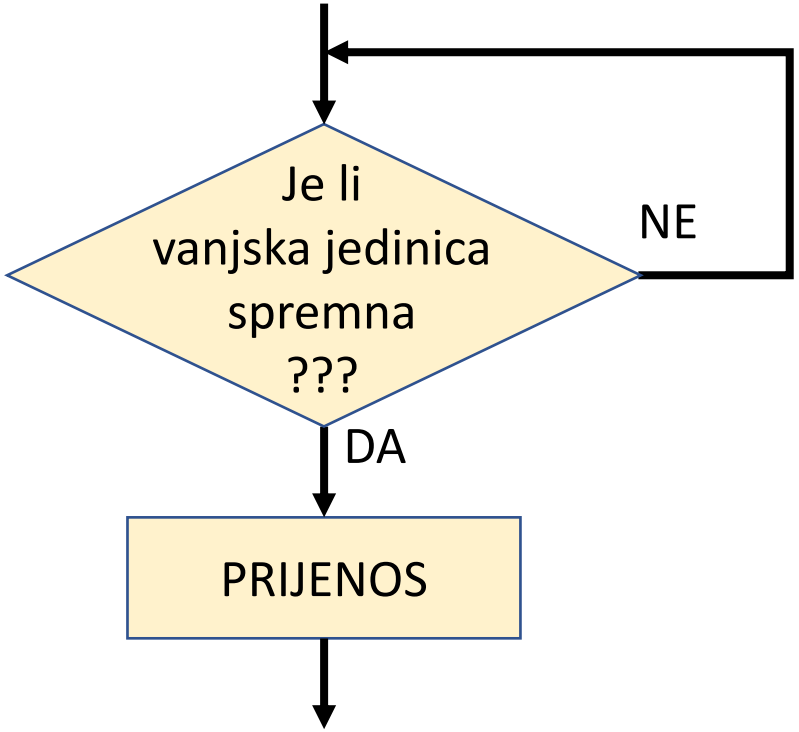
Zadatak: Napišite potprogram PRNSTR koji na LCD ispisuje string terminiran sa \0. Adresa stringa prenosi se registrom R0, a adresa podatkovnog registra od GPIO-a prenosi se registrom R1. PRNSTR mora koristiti LCDWR za slanje pojedinih znakova. Ispisuje se najviše 8 znakova, a ako je string dulji, višak znakova se zanemaruje. Modificirajte glavni program tako da koristi PRNSTR.

Uvjetni prijenos pomoću GPIO-a

Uvjetni prijenos s temperaturnog uređaja, pisača i raznih hipotetskih uređaja

Podsjetnik

- Prije prijenosa se uvijek provjerava je li vanjska jedinica spremna
- Prednost: nema gubitka/uvišestručenja podataka
- Nedostatak: dugotrajna čekanja na spremnost



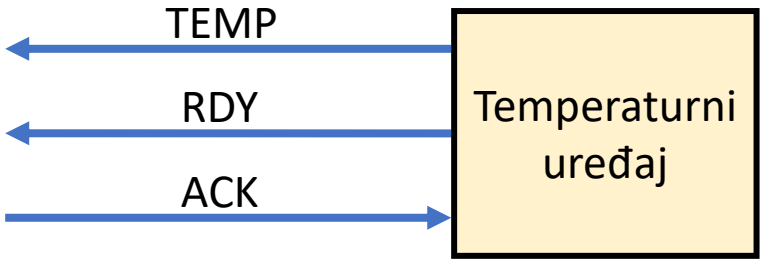
Temperaturni uređaj

Korištenje temperaturnog uređaja spojenog na GPIO

Uvjetni prijenos podataka sa temperaturnog uređaja

Temperaturni uređaj

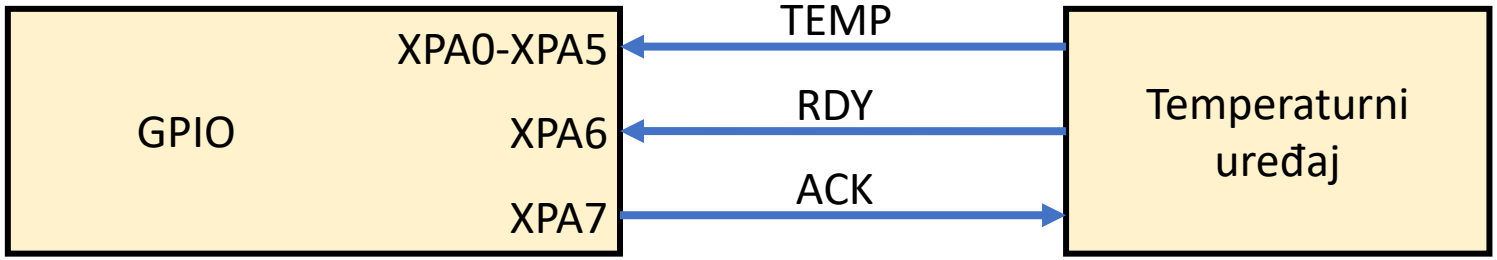
- Temperaturni uređaj je hipotetski uređaj koji mjeri i na izlazu daje temperaturu u digitalnom obliku (digitalni termometar)
- Temperaturni uređaj sadrži temperaturni senzor i AD pretvornik i omogućava očitanje trenutačne temperature u formatu 6-bitnog NBC-a (od 0 do 63 °C)
- Temperaturni uređaj ima jednostavno sučelje od 8 bita (6-podatak, 2-rukovanje):
  - **6** izlaznih priključaka **TEMP** s kojih se može pročitati temperatura
  - **1** izlazni priključak **RDY** (aktivan visoko) kojim temperaturni uređaj javlja da je postavio valjanu temperaturu na TEMP
  - **1** ulazni priključak **ACK** (aktivan visoko) pomoću kojega se temperaturnom uređaju javlja da je temperatura pročitana. Nakon toga temperaturni uređaj deaktivira RDY i ponovno pokreće mjerenje temperature
- Dok traje mjerenje (npr. desetak mikrosekundi), na TEMP se ne nalazi ispravan podatak i ne smije ga se čitati



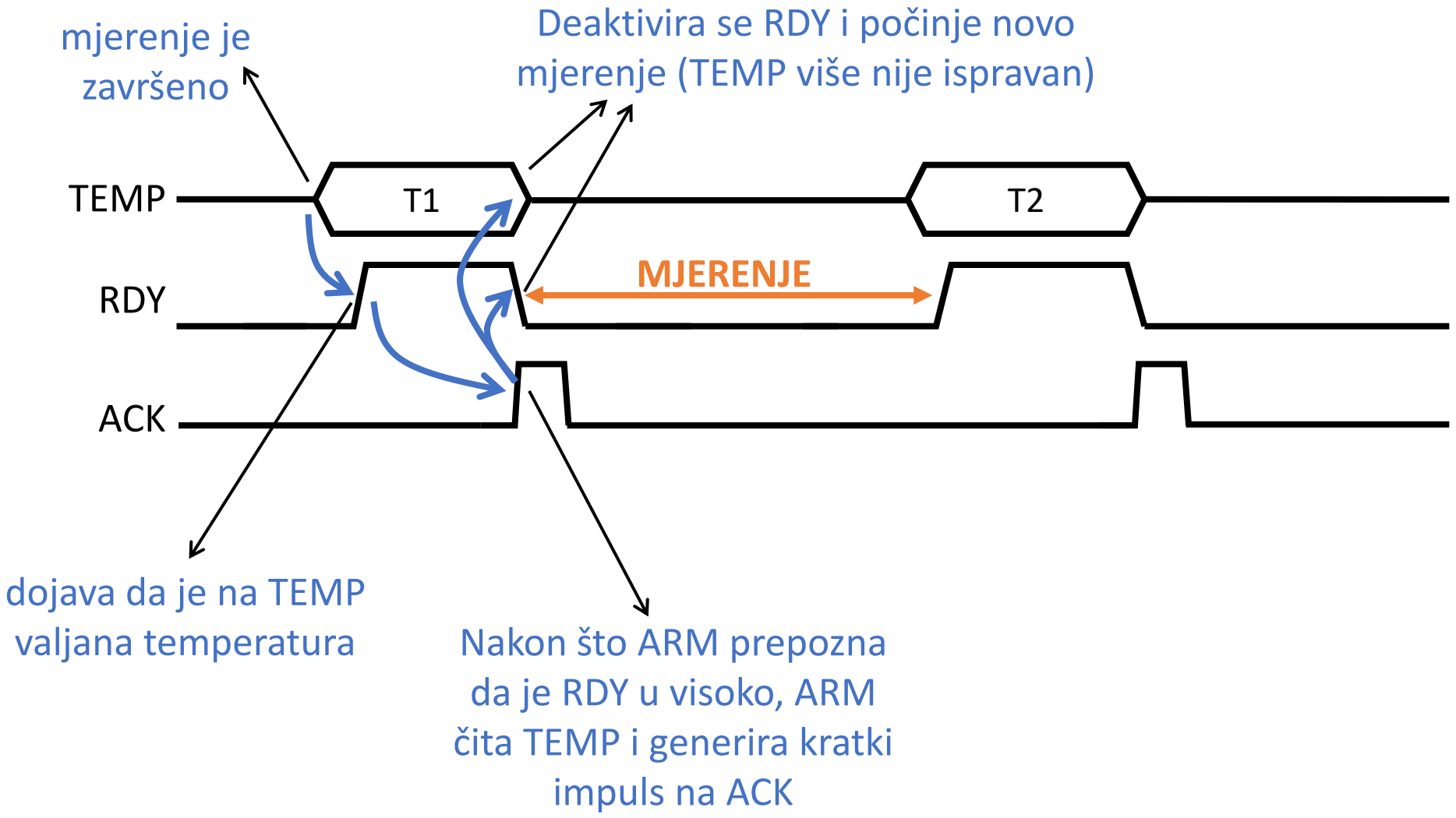
Temperaturni uređaj - spajanje na GPIO

- Priključci temperaturnog uređaja se na GPIO mogu spojiti bilo kako, ali uobičajeno ćemo ih spajati na port A:
  - XPA0-XPA5 (ulazni) - spojeni na TEMP
  - XPA6 (ulazni) - spojen na RDY
  - XPA7 (izlazni) - spojen na ACK

- Uočite: smjerovi međusobno spojenih priključaka na temperaturnom uređaju i na GPIO-u su obratni

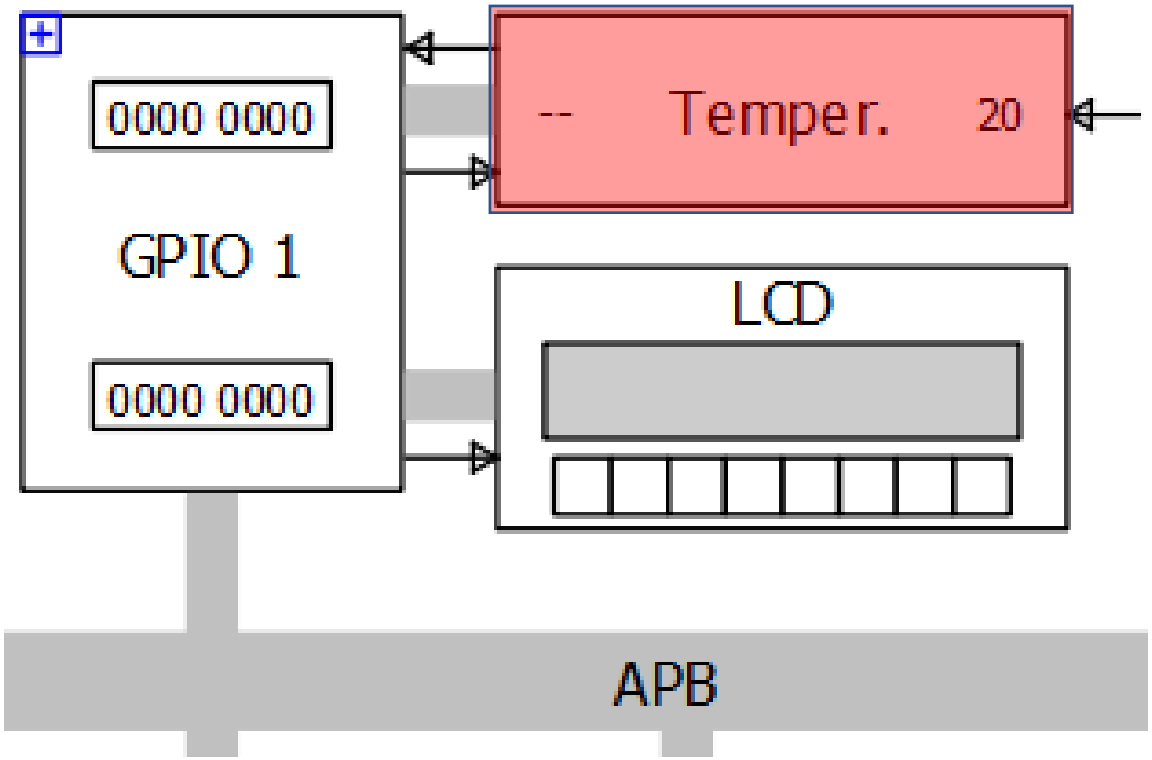


Temperaturni uređaj - vremenski dijagram



Temperaturni uređaj u SSPARCSS-u

- Temperaturni uređaj je implementiran u simulatoru
- Temperaturni uređaj je spojen na vrata A od sklopa GPIO_1



GPIO - uvjetni prijenos - primjer

- GPIO je na adresi FFFF0F00. Temperaturni uređaj spojen je na vrata A a LCD na vrata B jedinice GPIO (kao u simulatoru).
- Napisati program koji očitava 20 temperatura i kao bajtove ih sprema u memoriju od adrese TEMP te ispisuje na LCD-u.

- Komentar rješenja:
  - Budući da temperaturni uređaj ima stanje spremnosti, onda moramo programski ispitati spremnost preko GPIO-a. Prema zadanom načinu komunikacije moramo programski ostvariti rukovanje pomoću linija RDY i ACK.

GPIO - uvjetni prijenos - primjer


```
 ORG 0
MAIN    MOV SP, #0x10000    ; stog
        MOV R5, #TEMP         ; blok za spremanje temperatura
        MOV R2, #20          ; brojac ocitanja za petlju

INIT    LDR R1, GPIO          ; R1 = GPIO bazna adresa
        MOV R0, #0b10000000   ; smjer vrata A, bit 7 je
        STR R0, [R1, #0x8]    ; izlazni, ostali su ulazni

LOOP    ; citaj sa vrata A, ispiši na LCD i spremaj temperaturu 20 puta
CEKAJ   ; cekaj spremnost temperaturnog uredaja na bitu 6
        LDR  R0, [R1, #0]
        ANDS R0, R0, #0b01000000
        BEQ  CEKAJ

CITAJ   LDR  R0, [R1, #0]          ; citaj temperaturu
        AND  R0, R0, #0b00111111   ; izdvoji samo bitove 0-5

SPREMI STRB R0, [R5], #1           ; spremi temperaturu u blok
        BL LCD                     ; ispis temperature na LCD
```

GPIO - uvjetni prijenos - primjer


```
IMPULS ; kratki impuls na bitu 7 – potvrda da je podatak pročitan
       ORR  R0, R0, #0b10000000   ; digni bit 7 u jedan
       STR  R0, [R1, #0]
       BIC  R0, R0, #0b10000000   ; vrati bit 7 u nulu
       STR  R0, [R1, #0]

       SUBS R2, R2, #1            ; smanji brojac i vrti petlju
       BNE  LOOP

KRAJ  SWI 0x123456

LCD    STMFD R13!, {LR}
       MOV R3,R0           ; prebaci u dvije dekadske znameke
       BL DIV10            ; dijeljenje s 10 ulaz u R3, rez u R4, ostatak u R3
       MOV R0, #0x0D       ; slanje znaka 0xD => briše se ulazni registar
       BL  LCDWR
       ADD R0,R4,#48
       BL  LCDWR           ; pošalji znamenku desetice
       ADD R0,R3,#48
       BL  LCDWR           ; pošalji znamenku jedinice
       MOV R0, #0x0A       ; slanje znaka 0xA => ispis znakova na zaslon
       BL  LCDWR
       LDMFD R13!, {LR}
       MOV PC,LR
```

GPIO - uvjetni prijenos - primjer


```
DIV10 MOV R4,#0            ; dijeljenje s 10 ulaz u R3, rez u R4, ostatak u R3
L1    CMP R3, #10
      MOVLO PC, LR
      SUB R3, R3, #10
      ADD R4,R4,#1
      B L1

LCDWR STMFD R13!, {R0}    ; slanje znaka na LCD

      AND R0, R0, #0x7F   ; postavi bit 7 u nulu (za svaki slucaj, jer
                          ; u R0 je tu mogla biti 1) i pošalji znak
      STRB R0, [R1,#4]
      ORR R0, R0, #0x80   ; postavi bit 7 u jedan (podigni impuls)
      STRB R0, [R1,#4]
      AND R0, R0, #0x7F   ; postavi bit 7 u nulu (spusti impuls)
      STRB R0, [R1,#4]
      LDMFD R13!, {R0}
      MOV PC, LR          ; povratak

GPIO  DW  0xFFFF0F00              ; adresa GPIO-a

 ORG 0x300
TEMP  DS  20                     ; blok za pohranu 20 temperatura
```

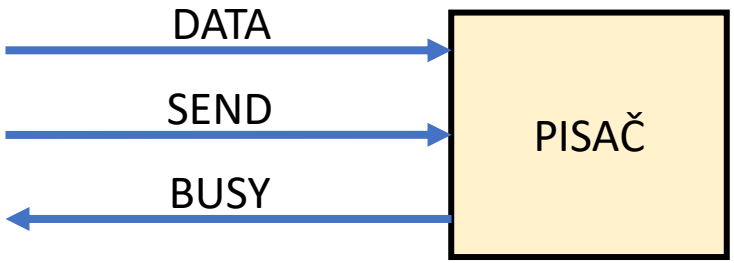
Pisač

Korištenje pisača spojenog na GPIO

Uvjetni prijenos podataka na pisač

Pisač

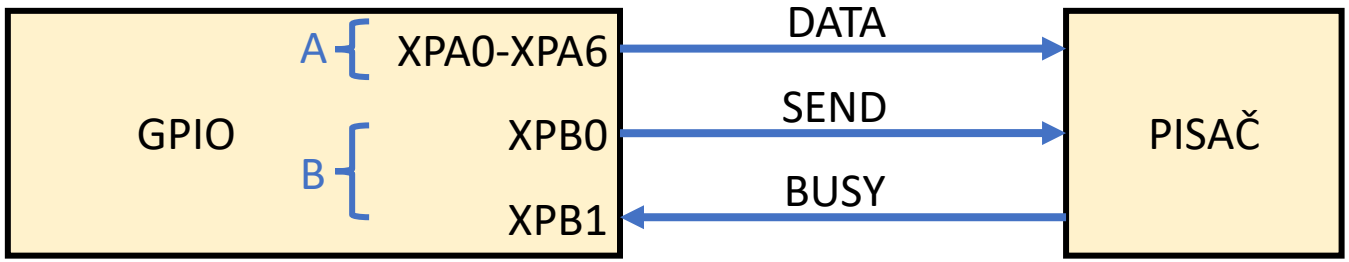
- Hipotetski pisač je uređaj koji prima i ispisuje ASCII-znakove
- Pisač nije implementiran u SSPARCSS-u
- Prije slanja svakog znaka treba provjeriti spremnost pisača jer dok traje jedan ispis nije moguće primanje drugog znaka
- Sučelje pisača ima 9 bitova (7 za podatke + 2 za rukovanje):
  - **7** ulaznih priključaka **DATA** preko kojih se šalje ASCII-znak na ispis
  - **1** ulazni priključak **SEND** preko kojih se pisaču javlja da je na DATA prisutan ASCII-znak (aktivan visoko)
  - **1** izlazni priključak **BUSY** preko kojeg pisač javlja da ispisuje znak i da ne može primati nove znakove (aktivan visoko)



Pisač - spajanje na GPIO

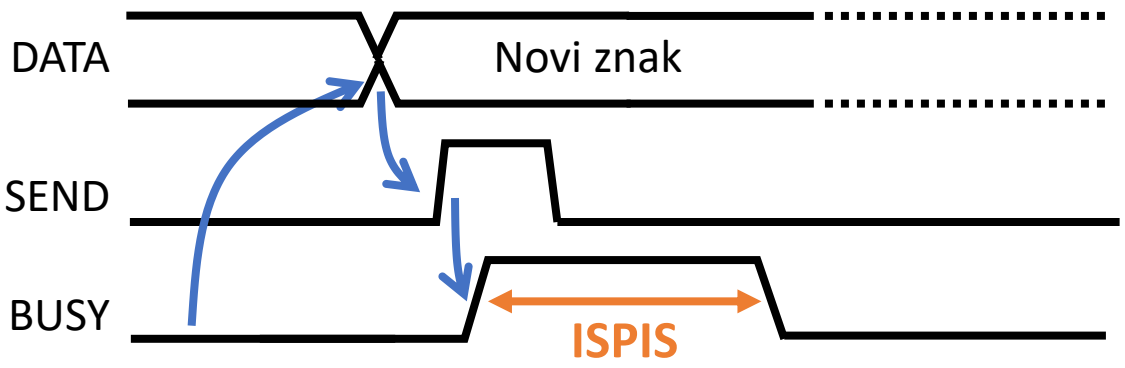
- Priključci pisača se na GPIO mogu spojiti bilo kako, ali uobičajeno ćemo ih spajati ovako:
  - DATA priključci spojeni su na port A:
    - XPA0-XPA6 (izlazni)
  - Kontrolni priključci spojeni na port B:
    - SEND na XPB0 (izlazni)
    - BUSY na XPB1 (ulazni)

- Uočite: smjerovi međusobno spojenih priključaka na pisaču i na GPIO-u su obratni



Pisač – vremenski dijagram

- Uvjetni prijenos ostvaruje se tako da se **prije slanja svakog znaka ispituje spremnost** pisača za primanje znaka. To je ekvivalentno kao da je GPIO spreman za primitak znaka od ARM-a.
- Spremnost se prepoznaje kada je signal BUSY u niskoj razini (neaktivan).
- Zatim treba postaviti ASCII-znak na DATA, i nakon toga pozitivni impuls na SEND
- Pisač na rastući brid od SEND prima znak, podiže BUSY i započinje s ispisom znaka. Nakon ispisa, pisač spušta BUSY.



GPIO - uvjetni prijenos - primjer

Na GPIO (adresa FFFF0100) je spojen pisač na uobičajeni način:

- na izlaze XPA[6:0] spojene su linije DATA za prijenos ASCII znaka na pisač
- na izlaz XPB[0] spojen je signal SEND (aktivan visoko)
- na ulaz XPB[1] spojen je signal BUSY (aktivan visoko)

Treba napisati potprogram PRINT koji ispisuje string (terminiran NUL-znakom, koji se ne ispisuje). Adresa stringa prenosi se preko registra R0.

Glavni program treba pomoću potprograma PRINT ispisati tekst „ARM”.

GPIO - uvjetni prijenos - primjer

Prijedlog rješenja:

Glavni program treba napraviti samo dvije stvari:

1. Prvo treba inicijalizirati smjerove priključaka na GPIO-u
2. Zatim treba pozvati potprogram PRINT

Adresa od GPIO nije zadana pa proizvoljno odabiremo adresu izvan memorijskog opsega FFFF0100.

Potprogram PRINT će u petlji čitati znak po znak iz stringa i slati znakove na GPIO. Budući da je GPIO bezuvjetna jedinica (**nema bistabil stanja!**), a pisač uređaj koji ima stanje spremnosti, onda moramo programski ispitati spremnost pisača preko GPIO-a. Prema zadanom načinu komunikacije pisača moramo programski ostvariti rukovanje pomoću linija SEND i BUSY.

GPIO - uvjetni prijenos - primjer


```
      ORG  0
MAIN  MOV  SP, #0x10000

INIT  LDR  R0, GPIO          ; inicijalizacija GPIO-a

      MOV  R1, #0x7F         ; smjer XPA[6:0] je izlazni
      STR  R1, [R0,#8]       ; osim nekorištenog bita 7 koji je ulazni

      MOV  R1, #0b11111110 ; XPB[0] je izlazni, a XPB[1] ulazni
      STR  R1, [R0,#0xC]     ; nekorišteni bitovi 2-7 su ulazni

SALJI MOV  R0, #STR          ; slanje stringa
      BL   PRINT

      SWI  0x123456

GPIO  DW   0xFFFF0100
STR   DSTR ”ARM”
```

GPIO - uvjetni prijenos - primjer


```
PRINT STMFD SP!, {R0-R3}          ; postindeksiranje mijenja R0
      LDR   R1, GPIO

LOOP  LDRB  R3, [R0], #1          ; dohvati znak iz stringa
      CMP   R3, #0                ; NUL znak se ne ispisuje, nego
      BEQ   VAN                   ; završava s petljom

WAIT  LDR   R2, [R1,#4]           ; čekanje spremnosti prije
      ANDS  R2, R2, #0b00000010   ; slanja znaka:
      BNE   WAIT                  ; BUSY mora biti 0

      STR   R3, [R1,#0]           ; slanje znaka na port A

      MOV  R2, #1                 ; impuls na SEND
      STR  R2, [R1,#4]
      MOV  R2, #0
      STR  R2, [R1,#4]

      B    LOOP                   ; ponovi za sljedeći znak

VAN   LDMFD SP!, {R0-R3}
      MOV   PC, LR
```

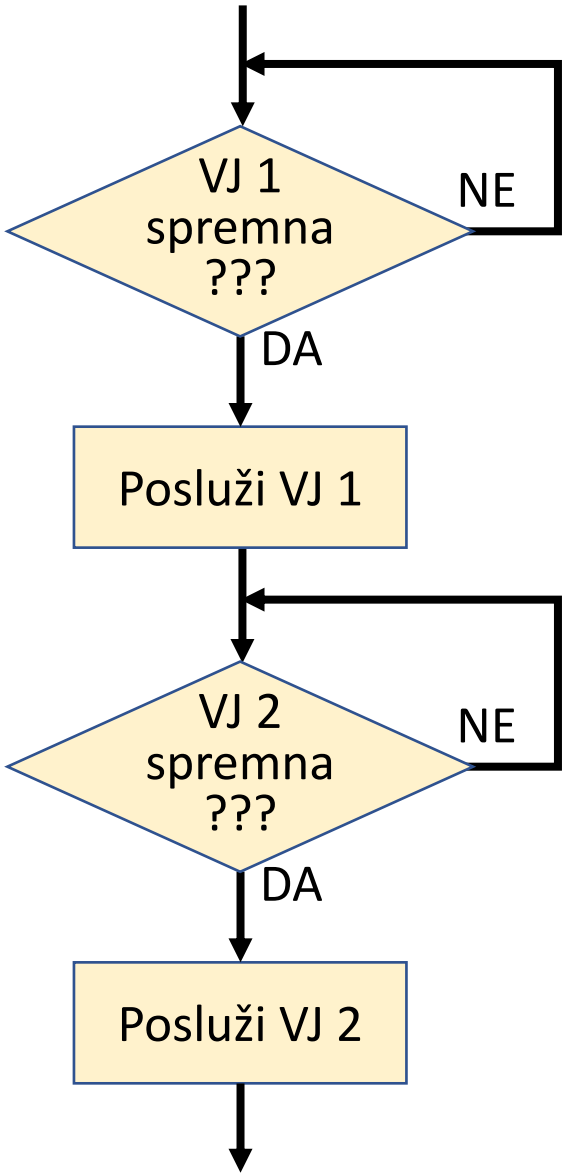
Posluživanje više uvjetnih vanjskih jedinica

Posluživanje zavisnih jedinica

Posluživanje nezavisnih jedinica

Posluživanje više zavisnih uvjetnih jedinica

- U radu sa zavisnim jedinicama bitan je redoslijed njihovog posluživanja
- Ako su jedinice **zavisne**, onda za svaku moramo **čekati** da postane spremna
- Na primjer, na slici je slučaj gdje se podatak primljen od VJ1 prenosi na VJ2



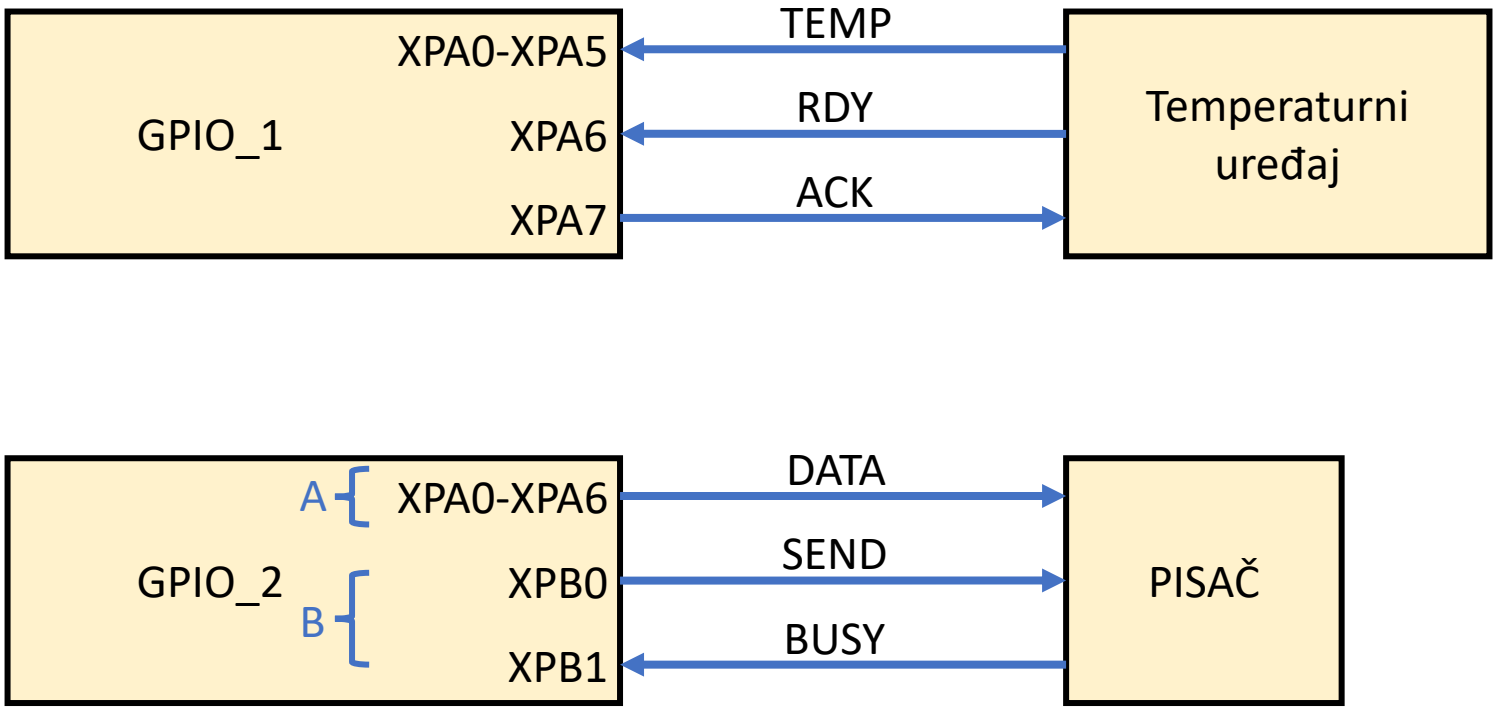
GPIO - uvjetni prijenos - primjer

Temperaturni uređaj spojen je na vrata A jedinice GPIO 1, a pisač je spojen na GPIO 2 kao u prethodnim zadatcima (DATA na vrata A, SEND na XPB0, a BUSY na XPB1).

Napisati program koji očitava temperaturu, pretvara njen iznos u dekadski broj zapisan kao string i ispisuje temperaturu na pisaču. To se ponavlja 10 puta.

Za pretvorbu temperature u string treba koristiti potprogram PRETVORI koji preko R0 prima temperaturu, a string zapisuje od memorijske lokacije 0x100. Pretpostavimo da potprogram PRETVORI već postoji i da ga ne moramo pisati.

Shema sustava iz zadatka



GPIO - uvjetni prijenos - primjer


```
      ORG  0

MAIN  MOV  SP, #0x10000

      ; inicijalizacija jedinice GPIO_1 (za temperaturni uređaj)
      LDR  R1, GPIO_1
      MOV  R0, #0x80
      STR  R0, [R1,#8]  ; XPA[6:0] ulazno, XPA[7] izlazno

      ; inicijalizacija jedinice GPIO_2 (za pisač)
      LDR  R2, GPIO_2

      MOV  R0, #0x7F
      STR  R0, [R2,#8]  ; XPA[6:0] izlazno, nekorišteni bit ulazni

      MOV  R0, #0b11111110
      STR  R0, [R2,#12] ; XPB[0] izlazno, XPB[1] ulazno, ostalo ulazno
```

(nastavak na sljedećem slajdu)

GPIO - uvjetni prijenos - primjer


```
(nastavak s prethodnog slajda)

        MOV   R10, #10        ; brojač petlje
LOOP    ; čekaj spremnost temperaturnog uređaja pa čitaj temperaturu
        BL    CEKAJ_CITAJ
        BL    PRETVORI        ; pretvori temperaturu iz R0 u
                              ; string zapisan na adresi 0x100
        ; šalji string na pisač sa čekanjima za svaki znak
        BL    CEKAJ_SALJI
        SUBS  R10, R10, #1
        BNE   LOOP
        SWI   0x123456


GPIO_1  DW    0xFFFF0100    ; proizvoljno odabrane adrese GPIO-va
GPIO_2  DW    0xFFFF0200

        ORG   0x100
        DS    32            ; mjesto za spremanje stringa
```

GPIO - uvjetni prijenos - primjer


```
CEKAJ_CITAJ ; potprogram čeka spremnost temperaturnog uređaja pa
            ; čita temperaturu i vraća je preko R0. R1 je adresa GPIO-a

            ; kontekst se ne sprema jer se mijenja samo R0

CEKAJ_T ; čekaj spremnost temperaturnog uređaja na bitu 6
        LDR  R0, [R1, #0]
        TST  R0, #0b01000000
        BEQ  CEKAJ_T

CITAJ_T ; čitaj temperaturu (samo bitove 0-5)
        LDR  R0, [R1, #0]
        AND  R0, R0, #0b00111111

ACK_T   ; kratki impuls na bitu 7
        ORR  R0, R0, #0b10000000   ; digni bit 7 u jedan
        STR  R0, [R1, #0]
        BIC  R0, R0, #0b10000000   ; vrati bit 7 u nulu
        STR  R0, [R1, #0]

        MOV  PC, LR
```

GPIO - uvjetni prijenos - primjer


```
CEKAJ_SALJI ; potprogram ispisuje string: za svaki znak čeka spremnost pisača
            ; i nakon toga mu šalje jedan znak dok ne dođe do NUL-znaka.
            ; R2 je adresa GPIO-a, 0x100 je adresa stringa

          STMFD SP!, {R0, R1, R5}   ; spremi kontekst
          MOV   R5, #0x100          ; adresa stringa

LOOP_P    LDRB  R0, [R5], #1        ; dohvati znak iz stringa
          CMP   R0, #0              ; usporedi znak sa NUL
          BEQ   VAN                 ; ako je NUL => return

CEKAJ_P   LDR   R1, [R2, #4]        ; ispituj i čekaj BUSY prije slanja znaka
          TST   R1, #0b00000010
          BNE   CEKAJ_P

SALJI_P   STR   R0, [R2, #0]        ; slanje znaka na port A

SEND_P    MOV   R1, #1              ; impuls na SEND na XPB0
          STR   R1, [R2, #4]
          MOV   R1, #0
          STR   R1, [R2, #4]

          B     LOOP_P              ; ponovi za sljedeći znak

VAN       LDMFD SP!, {R0, R1, R5}   ; obnovi kontekst
          MOV   PC, LR
```

Posluživanje više nezavisnih uvjetnih jedinica

- U radu s **nezavisnim vanjskim jedinicama** jedna ne ovisi o drugoj, tj. **nije bitan** redoslijed njihovog posluživanja

- Ako su jedinice nezavisne, onda koristimo postupak **prozivanja** (polling)*

- Postupak prozivanja
  - **Kružno** se ispituje spremnost svih nezavisnih jedinica, ali bez čekanja na pojedinu od njih
  - Kada je neka jedinica spremna, onda je poslužimo i nastavimo** s kružnim ispitivanjem od sljedeće jedinice

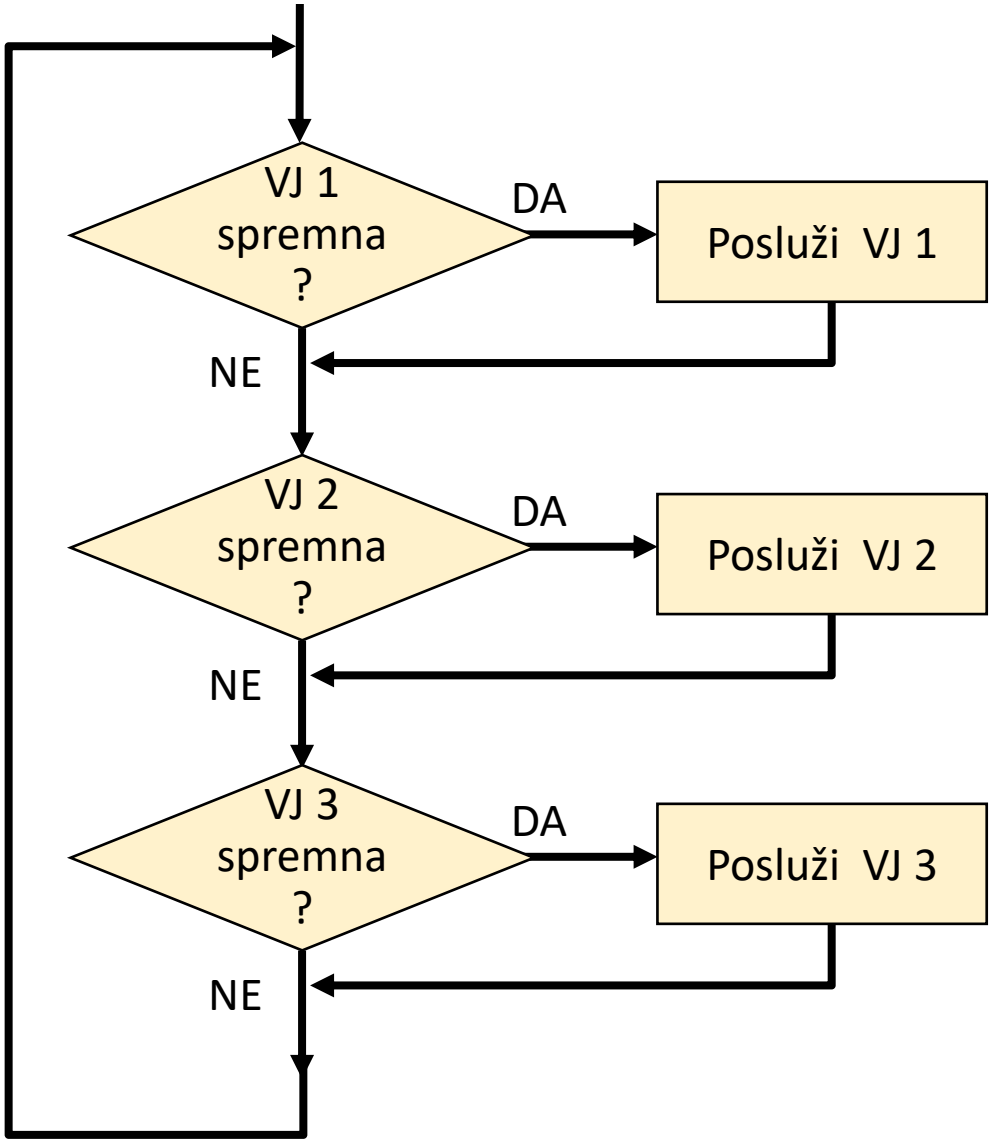
---

* I čekanje je zapravo polling, ali s jednom jedinicom

** Ne bi bilo dobro vratiti se s ispitivanjem na prvu jedinicu zbog mogućnosti „izgladnjivanja" (*starvation*) zadnjih jedinica u lancu ispitivanja

Posluživanje više nezavisnih uvjetnih jedinica

- Prozivanje umanjuje nedostatak uvjetnog prijenosa:
  - Nema čekanja na spremnost pojedine jedinice pa ona ne blokira ispitivanje ostalih
  - Veća je vjerojatnost da će jedna od nekoliko jedinica postati spremna

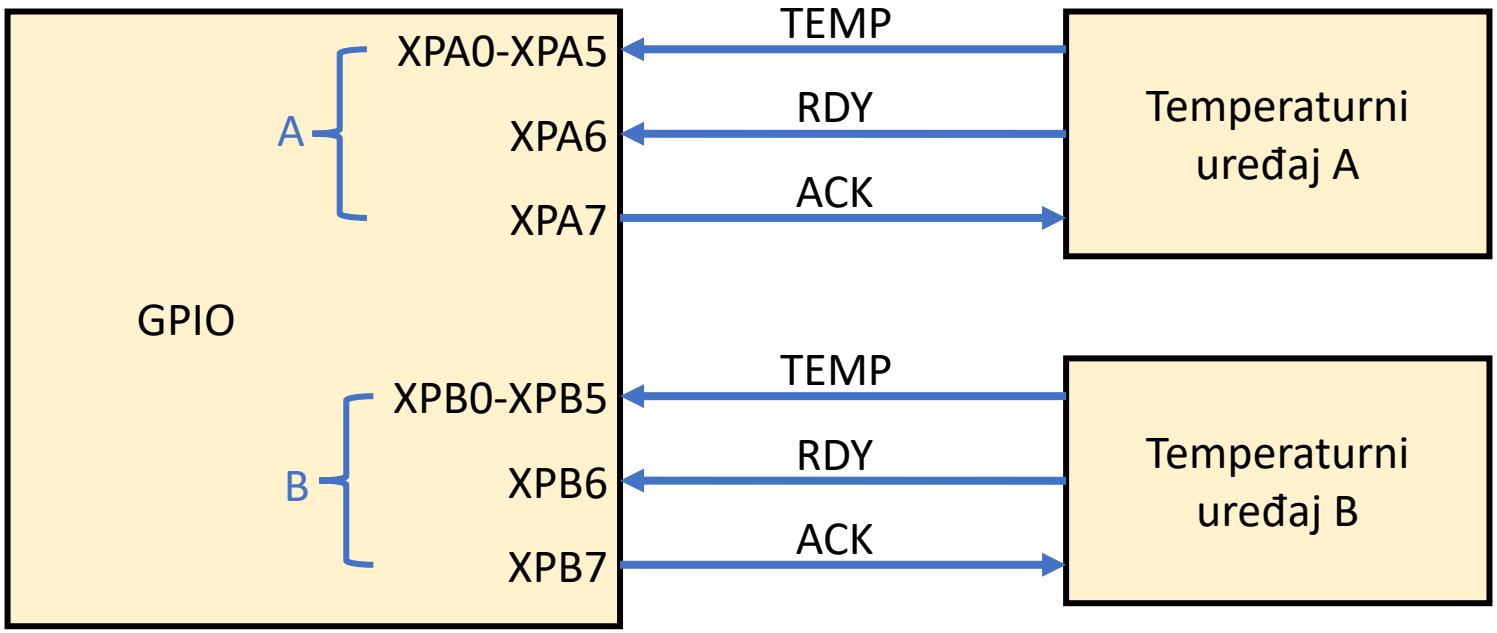


GPIO - uvjetni prijenos - primjer

GPIO je na adresi FFFF1000. Temperaturni uređaj TA spojen je na vrata A, a temperaturni uređaj TB spojen je na vrata B od GPIO-a.

Napisati program koji očitava temperature sa TA i TB i sprema ih u memoriju od adrese 0x1000. Nakon što se spremi ukupno 200 temperatura, treba zaustaviti procesor.

Shema sustava iz primjera



GPIO - uvjetni prijenos - primjer


```
      ORG 0

      MOV SP, #0x10000          ; stog

INIT  LDR R1, GPIO              ; R1 = GPIO bazna adresa

      ; inicijalizacija GPIO
      MOV R0, #0b10000000       ; smjer vrata A, bit 7 je
      STR R0, [R1, #0x8]        ; izlazni, ostali su ulazni

      MOV R0, #0b01111111       ; smjer vrata B, bit 7 je
      STR R0, [R1, #0xC]        ; izlazni, ostali su ulazni

      MOV R4, #0x1000           ; adresa bloka

      MOV R2, #200              ; brojač očitanja za petlju
```

GPIO - uvjetni prijenos - primjer


```
        ; petlja prozivanja

TEST_A  LDR   R0, [R1, #0]      ; ispitaj spremnost od TA
        TST   R0, #0b01000000
        BNE   POSLUZI_TA        ; posluži TA ako je spremna

TEST_B  LDR   R0, [R1, #4]      ; ispitaj spremnost od TB
        TST   R0, #0b01000000
        BNE   POSLUZI_TB        ; posluži TB ako je spremna

        B     TEST_A

KRAJ    SWI   0x123456


GPIO    DW    0xFFFF1000

        ORG   0x1000            ; blok za pohranu temperatura
        DS    200
```

GPIO - uvjetni prijenos - primjer


```
; posluživanje TA (R1=adresa GPIO-a, R4=adresa podatka,
; R2=brojač podataka)

POSLUZI_TA ; čitaj temperaturu (samo bitove 0-5)
           LDR  R0, [R1, #0]
           AND  R0, R0, #0b00111111

SPREMI_TA  STRB R0, [R4], #1          ; spremi temperaturu u blok

IMPULS_TA  ORR  R0, R0, #0b10000000   ; digni bit 7 u jedan
           STR  R0, [R1, #0]
           BIC  R0, R0, #0b10000000   ; vrati bit 7 u nulu
           STR  R0, [R1, #0]

           SUBS R2, R2, #1            ; smanji i provjeri brojač
           BEQ  KRAJ

           B    TEST_B                ; ispitaj sljedeću jedinicu


POSLUZI_TB ...  ; građen analogno kao POSLUZI_TA
```

Zavisne i nezavisne jedinice - primjer

Na vrata A i B jedinice GPIO_1 spojeni su hipotetski uređaji U1 i U2 koji šalju 6-bitne podatke jedinici GPIO_1 i koriste rukovanje isto kao temperaturni uređaji (imaju isti raspored bitova).

Na vrata A i B jedinice GPIO_2 spojen je hipotetski uređaj U3 koji prima 6-bitne podatke preko nižih bitova od vrata A, a na vratima B ima dvije linije za rukovanje SEND i BUSY kao na pisaču (jedina razlika je što pisač prima 7-bitni podatak, a U3 prima 6-bitni podatak).

Treba napisati program koji kontinuirano prima podatke sa U1 i U2 te primljene podatke prenosi na U3 i to redom njihovog dolaska.

Kad se sa U1 ili U2 primi podatak 63, onda treba zaustaviti program.

Zavisne i nezavisne jedinice - primjer

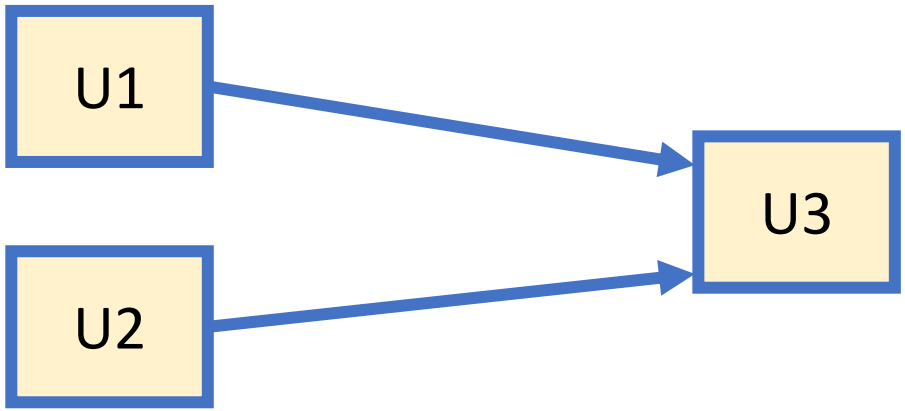
Komentar rješenja:

Sva tri uređaja poslužival ćemo uvjetno jer imaju rukovanje.

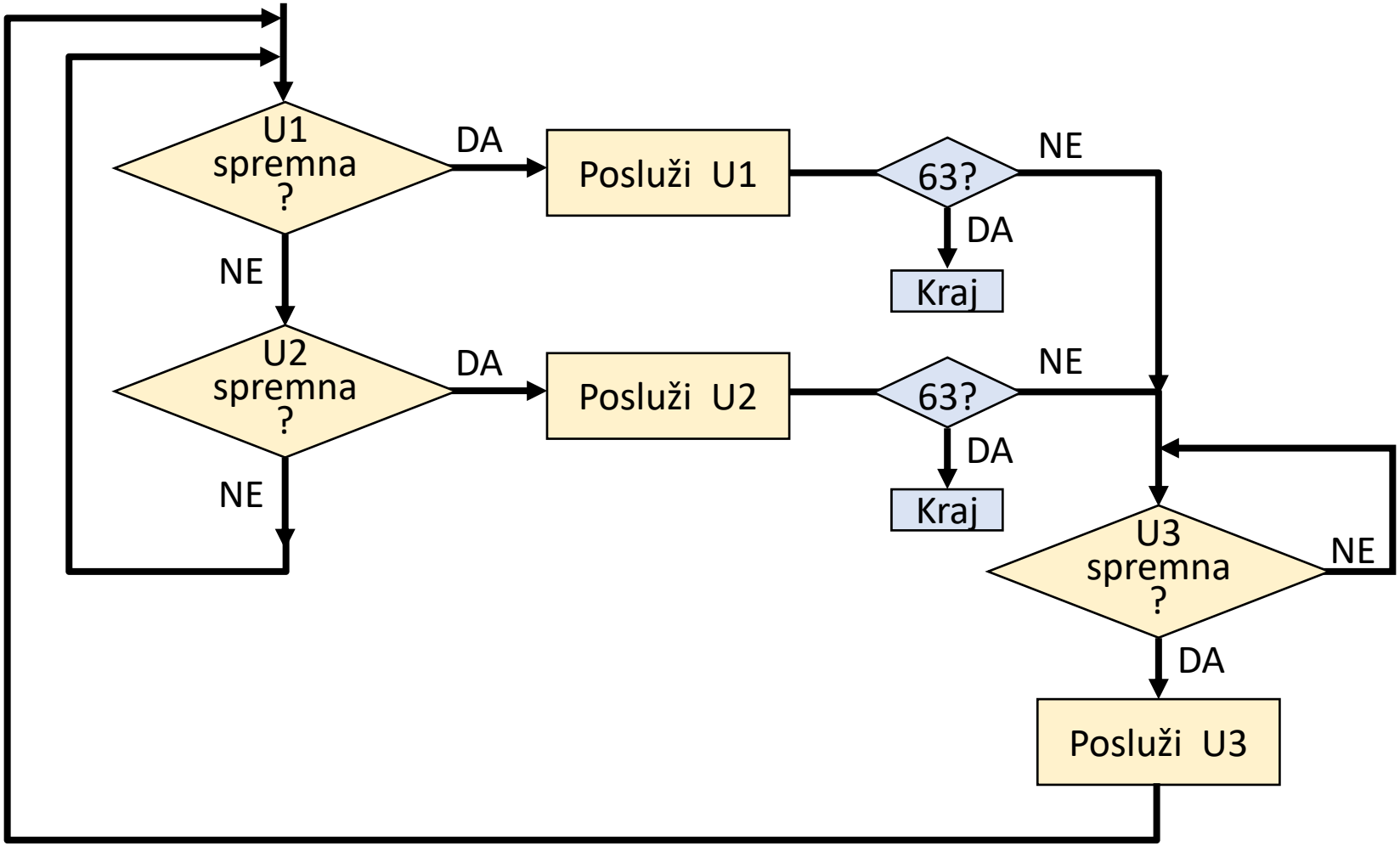
Prema zadanom redoslijedu prijenosa podatka zaključujemo da moramo:

1. prvo prozivati U1 i U2, ali ne beskonačno,
   nego dok s jedne od njih ne primimo podatak
2. zatim čekamo spremnost od U3 pa joj šaljemo podatak
3. nakon toga vraćamo se na korak 1 na prozivanje

Dakle, ovo je kombinacija dva nezavisna uređaja U1 i U2,
ali između U3 i U1/U2 postoji zavisnost.



Zavisne i nezavisne jedinice - primjer



Zavisne i nezavisne jedinice - primjer


```
      MOV SP, #0x10000         ; stog

INIT  LDR R1, GPIO_1            ; R1 = bazna adresa GPIO_1
      LDR R2, GPIO_2            ; R2 = bazna adresa GPIO_2

      ; inicijalizacija GPIO_1
      MOV R0, #0b10000000       ; smjer vrata A, bit 7 je
      STR R0, [R1, #0x8]        ; izlazni, ostali su ulazni

      MOV R0, #0b01111111       ; smjer vrata B, bit 7 je
      STR R0, [R1, #0xC]        ; izlazni, ostali su ulazni


      ; inicijalizacija GPIO_2
      MOV  R0, #0x3F          ; smjer XPA[5:0] je izlazni
      STR  R0, [R2,#8]        ; nekorišteni bitovi 6-7 su ulazni

      MOV  R0, #0b11111110 ; XPB[0] je izlazni, a XPB[1] ulazni
      STR  R0, [R2,#0xC]   ; nekorišteni bitovi 2-7 su ulazni
```

Zavisne i nezavisne jedinice - primjer


```
POOL    ; petlja prozivanja

TEST_U1 LDR  R0, [R1, #0]     ; ispitaj spremnost od U1
        TST  R0, #0b01000000
        BNE  POSLUZI_U1       ; posluži U1 ako je spremna

TEST_U2 LDR  R0, [R1, #4]     ; ispitaj spremnost od U2
        TST  R0, #0b01000000
        BNE  POSLUZI_U2       ; posluži U2 ako je spremna

        B    POOL

U3      B    CEKAJ_POSLUZI_U3
        B    POOL

KRAJ    SWI  0x123456


GPIO_1 DW   0xFFFF1000
GPIO_2 DW   0xFFFF2000
```

Zavisne i nezavisne jedinice - primjer


```
POSLUZI_U1 ; odsječak koji poslužuje prethodno ispitani uređaj U1
           ; program se zaustavlja ako se primi znak 63
           ; skače se na posluživanje U3 ako znak nije 63

           LDR  R3, [R1, #0]          ; čitaj podatak
           AND  R0, R3, #0b00111111

IMPULS_U1  ORR  R3, R3, #0b10000000   ; digni bit 7 u jedan
           STR  R3, [R1, #0]
           BIC  R3, R3, #0b10000000   ; vrati bit 7 u nulu
           STR  R3, [R1, #0]

           CMP  R3, #63               ; provjera kraja programa
           BEQ  KRAJ

           B    U3                    ; pošalji podatak na U3


POSLUZI_U2 ...  ; građen analogno kao POSLUZI_TA
```

Zavisne i nezavisne jedinice - primjer


```
CEKAJ_POSLUZI_U3  ; odsječak koji čeka spremnost od U3
                  ; pa je onda poslužuje i skače natrag na prozivanje

WAIT  LDR   R3, [R2,#4]             ; čekanje spremnosti prije
      ANDS  R3, R3, #0b00000010     ; slanja znaka:
      BNE   WAIT                    ; BUSY mora biti 0

      STR   R0, [R2,#0]             ; slanje znaka na port A

      MOV  R3, #1                   ; impuls na SEND
      STR  R3, [R2,#4]
      MOV  R3, #0
      STR  R3, [R2,#4]

      B    POOL                     ; skok u petlju prozivanja
```

Razni primjeri za samostalno proučavanje

Kombinirani načini prijenosa:
bezuvjetni, uvjetni sa zavisnim i nezavisnim jedinicama

Bezuvjetni i uvjetni prijenos - primjer

GPIO je na adresi FFFF1000. Temperaturni uređaj spojen je na vrata B. Na bitove 0-2 od vrata A spojene su tipke SAVE, RESET i STOP koje daju kratki pozitivni impuls kad su pritisnute.

Napisati program koji očitava temperaturu svaki puta kad se pritisne tipka SAVE. Očitane temperature se kao bajtovi spremaju u memoriju od adrese 0x1000.

Kad se pritisne RESET, onda treba ponovno započeti sa spremanjem od adrese 0x1000.

Kad se pritisne STOP, treba zaustaviti procesor.

Zanemarite mogućnost prepunjenja memorije, kao i mogućnost istodobnog pritiskanja više tipaka.

Bezuvjetni i uvjetni prijenos - primjer


```
    ORG 0

INIT LDR R1, GPIO              ; R1 = GPIO bazna adresa

    ; inicijalizacija GPIO
    MOV R0, #0                 ; smjer vrata A,
    STR R0, [R1, #0x8]         ; bitovi 0-2 su ulazni (tipke)

    MOV R0, #0b01111111        ; smjer vrata B, bit 7 je
    STR R0, [R1, #0xC]         ; izlazni, ostali su ulazni

    ; početna adresa za spremanje temperatura
    MOV R4, #0x1000
```

Bezuvjetni i uvjetni prijenos - primjer


```
LOOP  ; glavna petlja za ispitivanje tipaka (bezuvjetno)

      LDR  R2, [R1, #0]    ; čitaj stanje tipki

      ; ispitivanje tipki, redom: SAVE, RESET, STOP
      MOVS R2, R2, LSR #1  ; bit 0 pomakni u C i ispitaj
      BCS  SAVE

      MOVS R2, R2, LSR #1  ; bit 1 pomakni u C i ispitaj
      BCS  RESET

      MOVS R2, R2, LSR #1  ; bit 2 pomakni u C i ispitaj
      BCS  STOP

      ; nijedna tipka nije pritisnuta => čekaj
      B    LOOP
```

Bezuvjetni i uvjetni prijenos - primjer


```
SAVE  ; pritisnuta je tipka SAVE
      ; => učitaj temperaturu i spremi je u blok

CEKAJ ; čekaj spremnost temperaturnog uređaja na bitu 6
      LDR  R2, [R1, #4]
      ANDS R2, R2, #0b01000000
      BEQ  CEKAJ

CITAJ ; čitaj temperaturu (samo bitove 0-5)
      LDR  R2, [R1, #4]
      AND  R2, R2, #0b00111111
      STRB R2, [R4], #1           ; spremi temperaturu u blok

IMPULS ; kratki impuls na bitu 7
      ORR  R2, R2, #0b10000000    ; digni bit 7 u jedan
      STR  R2, [R1, #4]
      BIC  R2, R2, #0b10000000    ; vrati bit 7 u nulu
      STR  R2, [R1, #4]

     ; povratak u petlju za čekanje tipki
     B    LOOP
```

Bezuvjetni i uvjetni prijenos - primjer


```
RESET ; pritisnuta je tipka RESET
      ;   => „resetiraj” blok za pohranu temperatura
      MOV R4, #0x1000
      B   LOOP

STOP  ; pritisnuta je tipka STOP => zaustavi procesor
      SWI  0x123456


GPIO DW   0xFFFF1000     ; bazna adresa jedinice GPIO

     ORG  0x1000
     DS   5000          ; prostor za pohranu temperatura
```

Bezuvjetni i uvjetni prijenos - primjer

Napomena: Nakon što je tipka pritisnuta moglo bi se čekati na otpuštanje tipke, ali to nije napravljeno budući da je impuls kratkotrajan.

Zadatak: Preuredite rješenje ako tipka daje stanje 1 dok god je pritisnuta i treba čekati na otpuštanje tipke da bi se napravila zadana akcija. Što treba promijeniti ako se želi čekati otpuštanje tipke, ali tako da se akcija napravi odmah čim se tipka pritisne?

Zadatak: Promijenite zadatak tako da se smije pritisnuti više tipki odjednom i to tako da tipka STOP ima prioritet nad RESET, a RESET ima prioritet nad SAVE.

Prozivanje i bezuvjetni prijenos - primjer

GPIO 1 je na adresi FFFF1000. Temperaturni uređaji TA i TB spojeni su na vrata A odnosno B. GPIO 2 je na adresi FFFF2000 i na vratima A ima spojen LCD.

Napisati program koji očitava temperature sa vrata A i B i svaki puta kad se očita nova temperatura, ona se usporedi sa zadnjom temperaturom sa drugih vrata. Ako je veća temperatura na vratima A, onda na LCD-u treba ispisati slovo „A”. Ako je veća temperatura na vratima B, onda na LCD-u treba ispisati slovo „B”. Ako su obje temperature jednake, onda treba ispisati „=”.

Ova zadaća odvija se beskonačno.

Komentar rješenja: TA i TB su neovisni pa treba koristiti prozivanje. Za posluživanje TA i TB ćemo upotrijebiti potprogram. Početno ćemo uzeti temperaturu 20 za oba očitanja. Za prikaz na LCD koristit ćemo već objašnjenu funkciju LCDWR.

Prozivanje i bezuvjetni prijenos - primjer


```
      MOV SP, #0x10000          ; stog

INIT  LDR R1, GPIO_1             ; R1 = bazna adresa GPIO_1
      LDR R2, GPIO_2             ; R2 = bazna adresa GPIO_2

      ; inicijalizacija GPIO_1
      MOV R0, #0b10000000       ; smjer vrata A, bit 7 je
      STR R0, [R1, #0x8]        ; izlazni, ostali su ulazni

      MOV R0, #0b01111111       ; smjer vrata B, bit 7 je
      STR R0, [R1, #0xC]        ; izlazni, ostali su ulazni

      ; inicijalizacija GPIO_2
      MOV R0, #0b11111111       ; smjer vrata A,
      STR R0, [R2, #0x8]        ; svi bitovi su izlazni
```

Prozivanje i bezuvjetni prijenos - primjer


```
        ; petlja prozivanja

TEST_A  MOV   R0, #0
        LDR   R2, [R1, R0, LSL #2]  ; ispitaj spremnost od TA
        TST   R2, #0b01000000
        BLNE  POSLUZI               ; posluži TA ako je spremna

TEST_B  MOV   R0, #1
        LDR   R2, [R1, R0, LSL #2]  ; ispitaj spremnost od TB
        TST   R2, #0b01000000
        BLNE  POSLUZI               ; posluži TB ako je spremna

        B     TEST_A                ; prozivaj beskonačno

GPIO_1  DW    0xFFFF1000
GPIO_2  DW    0xFFFF2000

LAST    DB    20, 20          ; zadnje temperature sa TA i TB
```

Prozivanje i bezuvjetni prijenos - primjer


```
; posluživanje temperaturnog uređaja
; R0=parametar koji zadaje da li poslužujemo TA (0) ili TB (1)
; R1= parametar koji zadaje adresu GPIO-a

POSLUZI     STMFD SP!, {R2, R4, LR}
            LDR  R2, [R1, R0, LSL #2]
            AND  R0, R0, #0b00111111

SPREMI      MOV  R4, #LAST
            STRB R0, [R4, R0]          ; spremi temperaturu

IMPULS_TA   ORR  R0, R0, #0b10000000   ; digni bit 7 u jedan
            STR  R0, [R1, R0, LSL #2]
            BIC  R0, R0, #0b10000000   ; vrati bit 7 u nulu
            STR  R0, [R1, R0, LSL #2]

            BL   PRIKAZ

            LDMFD SP!, {R2, R4, LR}
            MOV  PC, LR
```

Prozivanje i bezuvjetni prijenos - primjer


```
; prikaz temperature na LCD-u
; LAST+0=fiksna lokacija s parametrom - zadnja temperatura TA
; LAST+1=fiksna lokacija s parametrom - zadnja temperatura TB

PRIKAZ   STMFD SP!, {R0, R1, R4, LR}
         LDR   R4, #LAST       ; dohvat zadnjih temperatura
         LDRB  R0, [R4, #0]
         LDRB  R1, [R4, #1]

         CMP   R0, R1          ; usporedba zadnjih temperatura
         MOVLO R0, #0x41       ; slovo A
         MOVHI R0, #0x42       ; slovo B
         MOVEQ R0, #0x3D       ; znak  =

         LDR   R1, GPIO_2      ; adresa DR na kojem je spojen LCD
         BL    LCDWR

         LDMFD SP!, {R0, R1, R4, LR}
         MOV   PC, LR
```

Zavisni uvjetni i bezuvjetni prijenos - primjer

Na vrata A i B jedinice GPIO_1 spojeni su hipotetski uređaji U1 i U2 koji šalju 6-bitne podatke jedinici GPIO_1 i koriste rukovanje isto kao temperaturni uređaji (imaju isti raspored bitova).

Na vrata A i B jedinice GPIO_2 spojeni su hipotetski uređaji B1 i B2 koji primaju 6-bitne podatke preko nižih bitova, a na najvišem bitu vrata je signal za sinkronizaciju kao kod LCD-a.

Treba napisati program koji prvo pročita početnu graničnu vrijednost sa U2, a zatim kontinuirano prima podatke sa U1 i šalje ih na B1 ako su manji od granične vrijednosti ili na B2 ako su veći ili jednaki graničnoj vrijednosti.

Svaki puta nakon što se pročita i prenese 10 podataka za U1, treba pročitati novu graničnu vrijednost sa U2. Ako se pročita granična vrijednost 0, onda treba zaustaviti program.

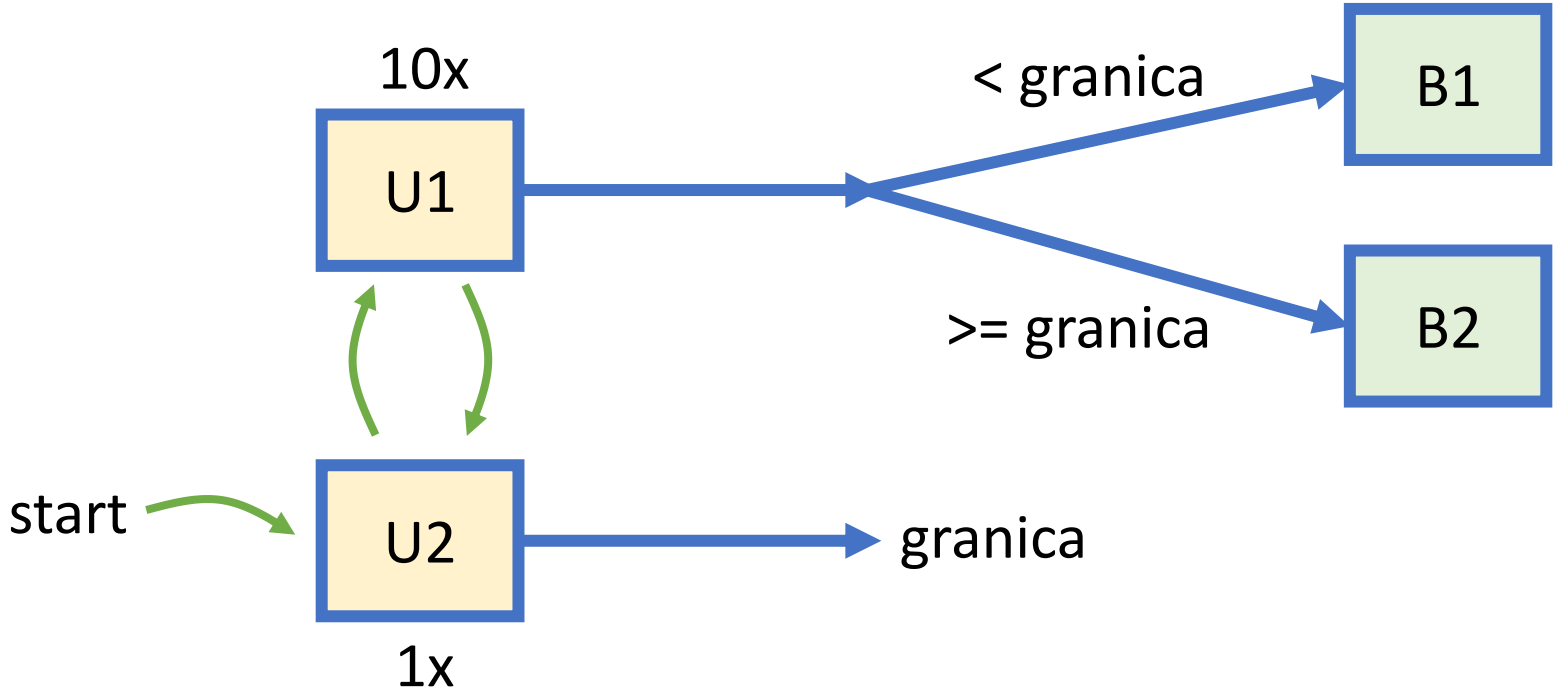
Zavisni uvjetni i bezuvjetni prijenos - primjer

Komentar rješenja:

U1 i U2 posluživat ćemo kao uvjetne uređaje jer imaju rukovanje.

Budući da je zadan redoslijed čitanja podatka sa U1 i U2, onda nećemo koristiti prozivanje.

B1 i B2 posluživat ćemo kao bezuvjetne uređaje jer kod njih nema provjere spremnosti uređaja.



Zavisni uvjetni i bezuvjetni prijenos - primjer

Komentar rješenja:

Kako bi skratili (relativno dugački) program, za posluživanje U1 i U2 koristit ćemo funkciju POSLU, a za B1 i B2 funkciju POSLB. Obje funkcije će preko R0 primati adresu vrata na kojima je odgovarajući uređaj.

Funkcija POSLU će preko R0 kao rezultat vraćati primljeni podatak.

Funkcija POSLB će preko R1 kao parametar primati podatak koji treba poslati na B1 ili B2.

Zavisni uvjetni i bezuvjetni prijenos - primjer


```
      MOV SP, #0x10000          ; stog

INIT  LDR R1, GPIO_1             ; R1 = bazna adresa GPIO_1
      LDR R2, GPIO_2             ; R2 = bazna adresa GPIO_2

      ; inicijalizacija GPIO_1
      MOV R0, #0b10000000       ; smjer vrata A, bit 7 je
      STR R0, [R1, #0x8]        ; izlazni, ostali su ulazni

      MOV R0, #0b01111111       ; smjer vrata B, bit 7 je
      STR R0, [R1, #0xC]        ; izlazni, ostali su ulazni

      ; inicijalizacija GPIO_2
      MOV R0, #0b10111111       ; smjer vrata A, svi bitovi su
      STR R0, [R2, #0x8]        ; izlazni, osim nekorištenog 6

      MOV R0, #0b01000000       ; smjer vrata B, svi bitovi su
      STR R0, [R2, #0x8]        ; izlazni, osim nekorištenog 6
```

Zavisni uvjetni i bezuvjetni prijenos - primjer


```
LOOP    ; čitanje granične vrijednosti
        ADD  R0, R1, #4      ; u R0 stavi adresu od PB_DR of GPIO_1
        BL   POSLU           ; čitaj sa GPIO_1
        STR  R0, LIMIT       ; spremi graničnu vrijednost

        CMP  R0, #0          ; provjera kraja programa kada je...
        BEQ  KRAJ            ; granična vrijednost jednaka nuli

        BL   POSLU1_10       ; posluži 10 puta uređaj U1 i B1/B2

        B    LOOP            ; kontinuirano poslužuj U1 + U2

KRAJ    SWI  0x123456        ; kraj programa

GPIO_1 DW   0xFFFF1000
GPIO_2 DW   0xFFFF2000

LIMIT  DB   0                ; granična vrijednost
```

Zavisni uvjetni i bezuvjetni prijenos - primjer


```
; funkcija koja 10 puta poslužuje uređaj U1 pa zatim B1 ili B2
; nema parametara ni povratne vrijednosti

POSL1_10 STMFD SP!, {R0-R2, R10, LR}
         MOV   R10, #10          ; brojač za 10 posluživanja
         LDR   R2, LIMIT         ; dohvati trenutnu graničnu vrijednost

LOOP10   LDR   R0, GPIO_1        ; PA_DR od GPIO_1
         BL    POSLU
         MOV   R1, R0            ; stavi novi podatak u R1
         LDR   R0, GPIO_2        ; stavi adresu od PA_DR GPIO_2 u R0

ODABIR   CMP   R1, R2            ; usporedi podatak (R1) i granicu (R2)
         ADDHS R0, R0, #4        ; adresa PB_DR od GPIO_2 u R0

         BL    POSL_B            ; pošalji podatak na B1 ili B2

         SUBS  R10, R10, #1
         BNE   LOOP10

         LDMFD SP!, {R0-R2, R10, LR}
         MOV   PC, LR
```

Zavisni uvjetni i bezuvjetni prijenos - primjer


```
; funkcija za posluživanje uređaja U1 ili U2
; R0 je parametar: adresa DR-a na kojemu je uređaj U1/U2
; R0 je povratna vrijednost: podatak pročitan sa U1/U2

POSL_U  STMFD SP!, {R2}

CEKAJ   LDR   R2, [R0]              ; čekaj spremnost uređaja U1/U2
        TST   R2, #0b01000000
        BEQ   CEKAJ

CITAJ   LDR   R2, [R0]              ; čitaj podatak s uređaja U1/U2
        AND   R2, R2, #0b00111111

IMPULS  ORR   R2, R2, #0b10000000   ; digni bit 7 u jedan
        STR   R2, [R0]
        BIC   R2, R2, #0b10000000   ; vrati bit 7 u nulu
        STR   R2, [R0]

        MOV   R0, R2
        LDMFD SP!, {R2}
        MOV   PC, LR
```

Zavisni uvjetni i bezuvjetni prijenos - primjer


```
; funkcija za posluživanje uređaja B1 ili B2 (slična LCDWR-u)
; R0 je parametar: adresa DR-a na kojemu je uređaj B1/B2
; R1 je parametar: podatak koji treba upisati na B1/B2

POSL_B STMFD SP!, {R1}      ; zapravo nepotrebno*

       STR   R1, [R0]       ; pošalji podatak, bit 7 je sigurno 0

       ORR   R1, R1, #0x80  ; postavi bit 7 u jedan (podigni impuls)
       STR   R1, [R0]

       BIC   R1, R1, #0x80  ; postavi bit 7 u nulu (spusti impuls)
       STR   R1, [R0]

       LDMFD SP!, {R1}
       MOV   PC, LR         ; povratak
```

---

* iako se R1 mijenja, uvijek se vrati u početno stanje