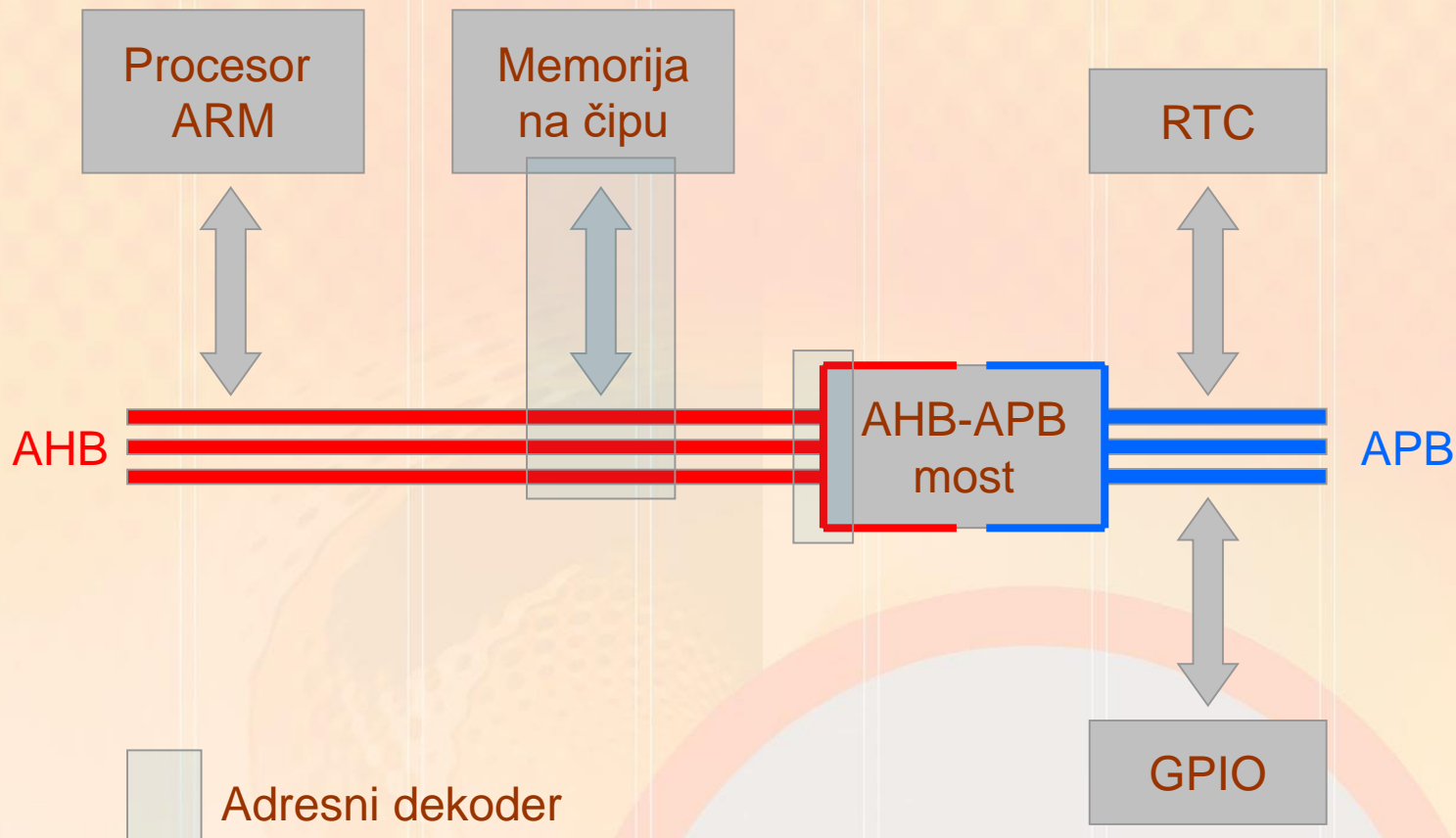


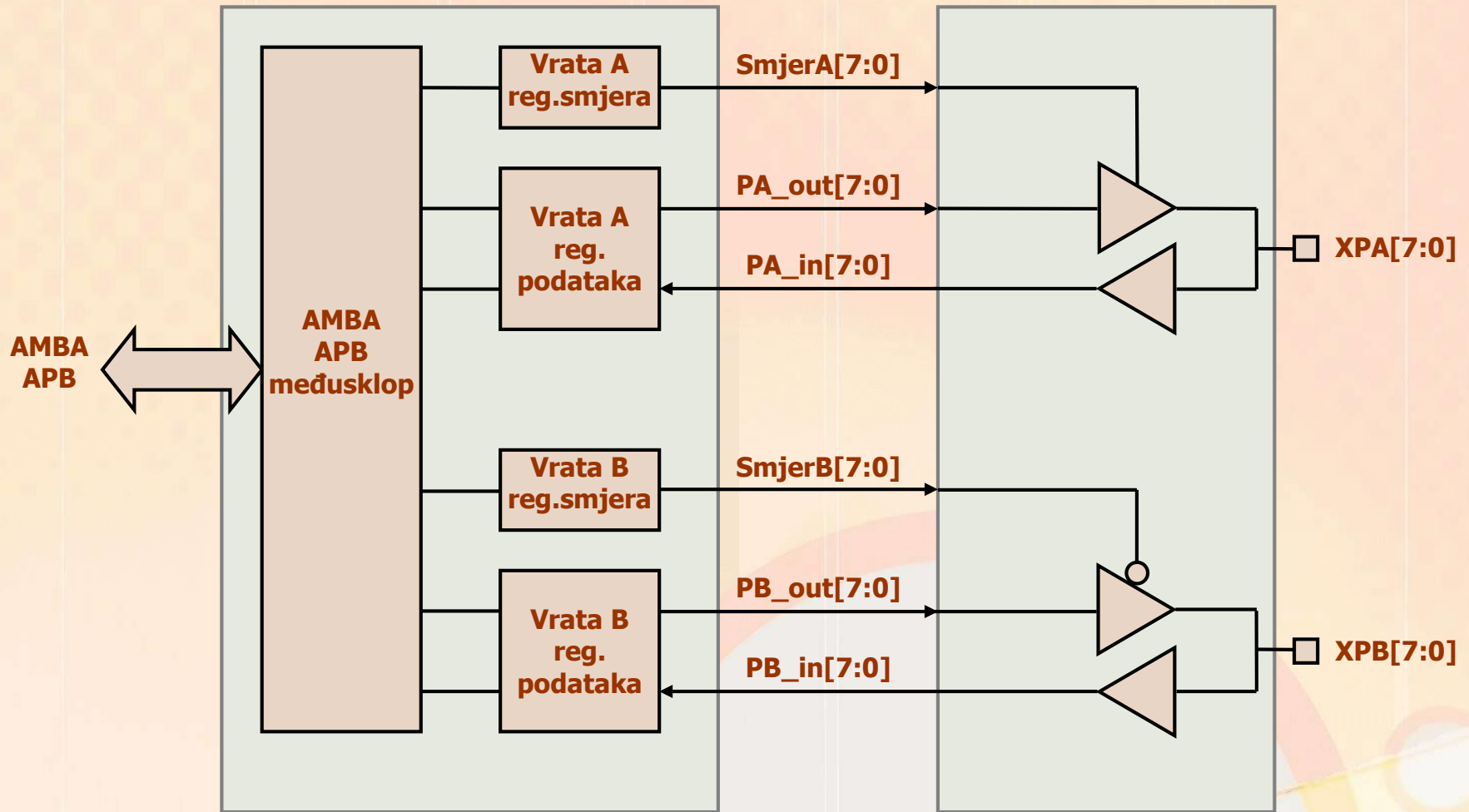
Vanjske jedinice za procesor ARM

GPIO i RTC

Naš ARM-sustav u ATLAS-u



Sklop GPIO (General Purpose Input Output)



Registri sklopa GPIO

Adresa	Naziv registra	Opis
GPIO_bazna_adr	GPIOPADR	8-bitni registar podataka, vrata A
GPIO_bazna_adr + 4	GPIOPBDR	8-bitni registar podataka, vrata B
GPIO_bazna_adr + 8	GPIOPADDR	8-bitni registar smjera podataka za vrata A
GPIO_bazna_adr + C	GPIOPBDDR	8-bitni registar smjera podataka za vrata B

Registri GPIO

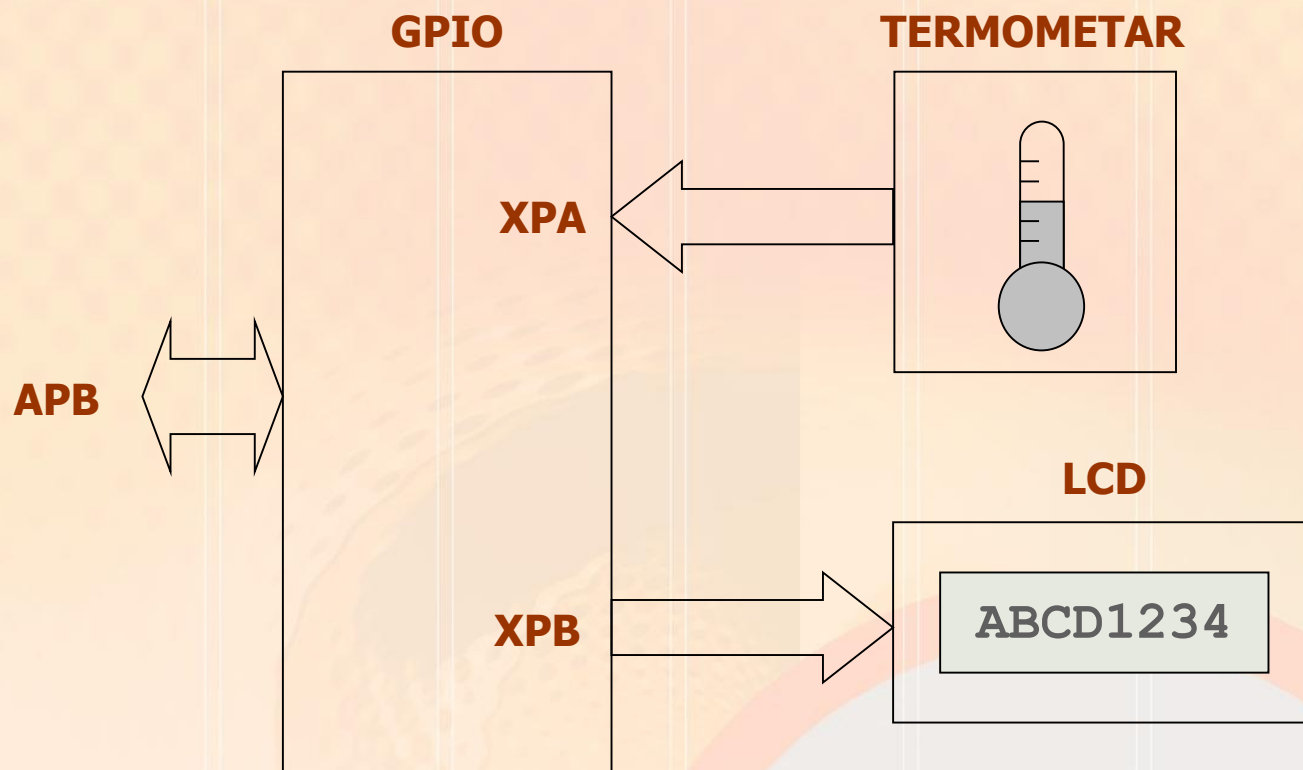
- GPIOPADR (GPIO Port A Data Register)
 - 8-bitni registar podataka za vrata A.
 - Podatci upisani u ovaj registar postavljeni su na izlazne priključke ako je pripadni bit u registru smjera podataka (GPIOPADDR) postavljen u logičku jedinicu.
 - Čitanje ovog registra vraća trenutno stanje priključaka koji su programirani kao ulazni. Za priključke programirane kao izlazne, vraća se vrijednost zadnjeg podatka upisanog u ovaj registar.
- GPIOPBDR (GPIO Port B Data Register)
 - 8-bitni registar podataka za vrata B.
 - Podatci upisani u ovaj registar postavljeni su na izlazne priključke ako je pripadni bit u registru smjera podataka (GPIOPBDDR) postavljen u logiku nulu.
 - Čitanje ovog registra vraća trenutno stanje priključaka koji su programirani kao ulazni. Za priključke programirane kao izlazne, vraća se vrijednost zadnjeg podatka upisanog u ovaj registar.

Registri GPIO

- GPIOPADDR (GPIO Port A Data Direction Register)
 - 8-bitni registar smjera podataka za vrata A. Vrijednost bita u ovom registru zadaje smjer odgovarajućeg priključka na vratima A:
 - 0 – ulazni smjer
 - 1 – izlazni smjer
- GPIOPBDDR (GPIO Port B Data Direction Register)
 - 8-bitni registar smjera podataka za vrata B. Vrijednost bita u ovom registru zadaje smjer odgovarajućeg priključka na vratima B:
 - 0 – izlazni smjer
 - 1 – ulazni smjer
- Početna vrijednost registara
 - Svi registri unutar GPIO-a nakon inicijalizacije (resetiranja) se postavljaju u logičku nulu. Ovime se inicijalno vrata A postavljaju kao ulazna, a vrata B kao izlazna. Sadržaji obaju registara podataka su nula.

← obrnuto od vrata A

Vanjski uređaji



Temperaturni sklop

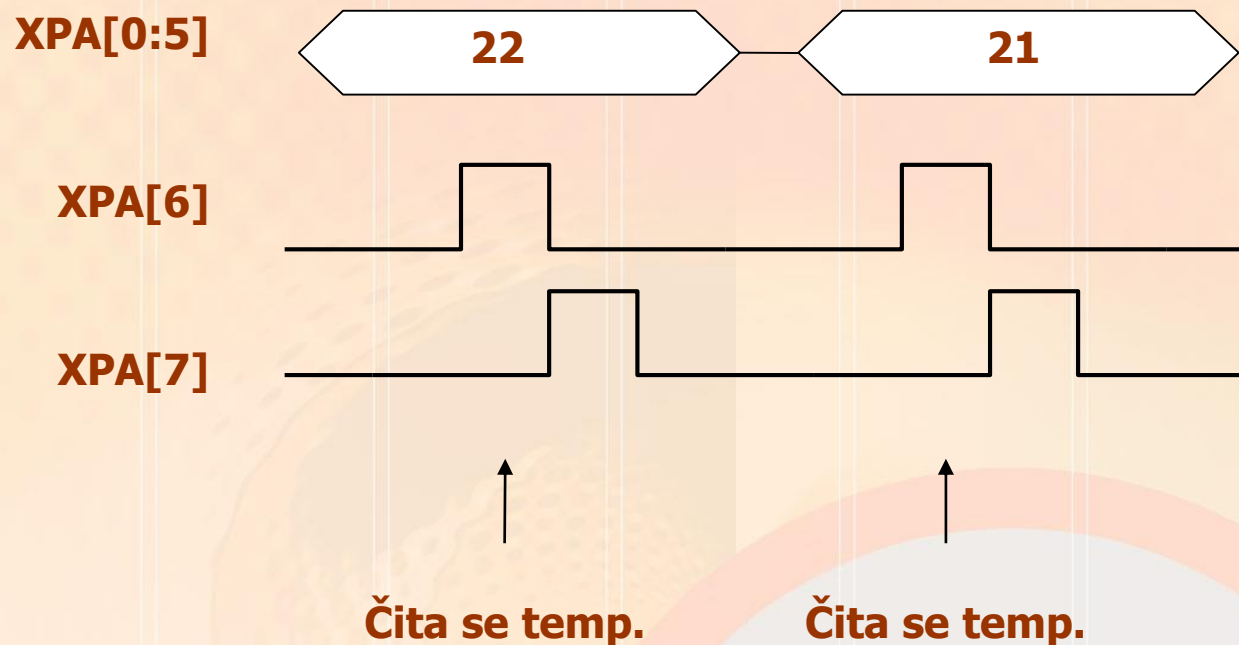
- Temperaturni sklop je jednostavna vanjska jedinica koja omogućuje mjerenje i očitavanje vanjske temperature. Komunikacija s temperaturnim sklopom objašnjena je na primjeru kad je on spojen na vrata A sklopa GPIO.
- Temperaturni sklop periodički mjeri temperaturu i izmjerenu vrijednost u formatu 6-bitnog cijelog broja postavlja na ulaz GPIO i to na priključke XPA[0] do XPA[5].
- Čitanjem vrata A može se pročitati trenutna vrijednost temperature (u nižih 6 bitova).

Temperaturni sklop

- Dva najviša bita (povezana na GPIO-priključke XPA[6] i XPA[7]) služe za sinkronizaciju*. Bitovi 6 i 7 su aktivni u visokoj razini.
- Bit 6 ulazi u GPIO i pomoću njega temperaturni sklop signalizira da je izmjerio temperaturu i postavio njezinu vrijednost na nižih šest bitova. Kad procesor ustanovi da je stanje na bitu 6 u jedinici, može očitati temperaturu s nižih šest bitova.
- Nakon što je pročitao podatke, procesor pomoću izlaznog bita 7 dojavljuje temperaturnom sklopu da je vrijednost temperature pročitana i da treba izbrisati bit 6, sve do trenutka dok temperaturni sklop ne postavi novu vrijednost temperature te ponovo ne aktivira bit 6.

* GPIO nema priključke za sinkronizaciju pa je treba obaviti programski

Vremenski dijagram očitavanja temperature



Primjer: Čitanje temperature

Temperaturni sklop spojen je na vrata B sklopa GPIO. Na najniži bit vrata A spojena je tipka koja daje pozitivni impuls kad je pritisnuta. GPIO je na adresi FFFF1000(16).

Napisati program koji očitava temperaturu svaki puta kad se pritisne tipka i sprema je u memoriju na adresu 1000(16). Nakon 100(16) očitavanja zaustaviti procesor.

ORG 0

GLAVNI

LDR R1, GPIO ; R1 = GPIO bazna adresa

; inicijalizacija GPIO

MOV R0, #0 ; smjer vrata A, bit 0 je ulazni (tipka)

STR R0, [R1, #08]

MOV R0, #1 ; smjer vrata B, bit 7 je izlazni, ostali su ulazni

STR R0, [R1, #0C]

; priprema varijabli za glavnu petlju

MOV R0, #10 ; brojač očitavanja

MOV R4, #1000 ; adresa za spremanje temperatura

Primjer: Čitanje temperature

PETLJA

```
LDR R2, [R1, #0]           ; glavna petlja  
ANDS R2, R2, #0x00000001    ; čitaj stanje tipke (najniži bit na portu A)  
BEQ PETLJA                 ; ako nije pritisnuta, čekaj
```

CEKAJ_TEMP

```
LDR R2, [R1, #4]           ; čekaj spremnost temperaturnog sklopa  
ANDS R2, R2, #0x01000000  
BEQ CAJAJ_TEMP
```

CITAJ_TEMP

```
LDR R2, [R1, #4]           ; čitaj temperaturu (nižih 6 bitova)  
AND R2, R2, #0x00111111  
STRB R2, [R4], #1          ; spremi temperaturu u memoriju
```

```
ORR R2, R2, #0x10000000    ; postavi bit 7 u jedan  
STR R2, [R1, #4]  
AND R2, R2, #0x01111111    ; postavi bit 7 u nulu  
STR R2, [R1, #4]
```

```
SUBS R0, R0, #1            ; smanji brojač za 100 očitavanja  
BHI PETLJA  
SWI 123456
```

```
GPIO    DW    0xFFFF1000    ; GPIO bazna adresa
```

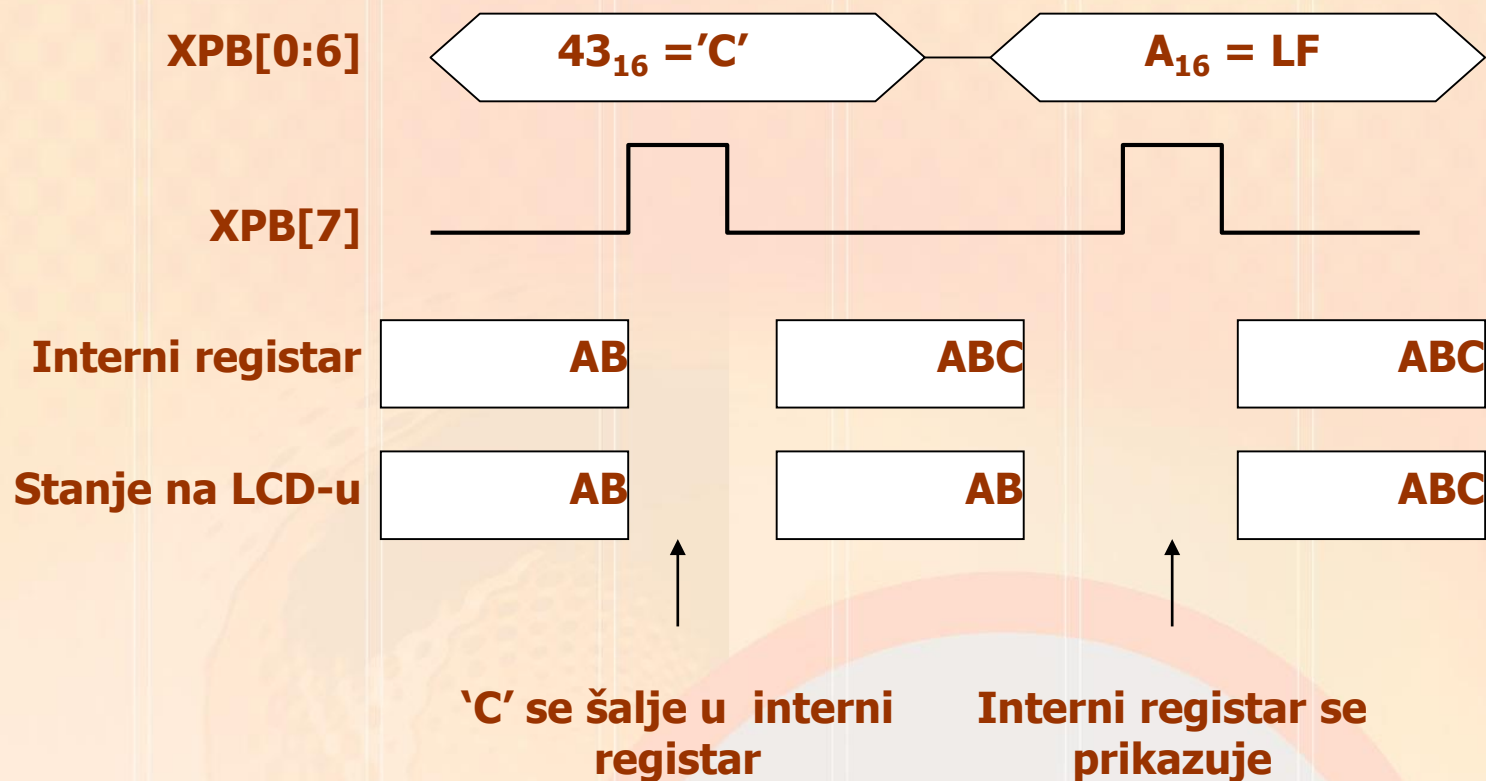
LCD

- LCD-prikaznik je izlazna vanjska jedinica koja omogućuje ispisivanje do osam ASCII-znakova na zaslonu LCD-a. LCD je jednostavna jedinica preko koje se rezultati operacija izvedenih u procesoru mogu prikazati korisniku (ispis znakova tijekom simulacije bit će prikazan u prozoru simulatora ATLAS).
- Komunikacija s LCD-om objašnjena je na primjeru kad je on spojen na vrata B sklopa GPIO.
- Sabirnica LCD-a je 8-bitna i organizirana tako da se nižih sedam bitova (bitovi 0 do 6) koriste za podatke
- Najviši bit (bit 7) je upravljački bit (WR) kojim se podatci upisuju u interni registar LCD-a.
- Registar LCD-a pamti do osam ASCII-znakova.

Način ispisa znakova na LCD

- Dok je bit WR neaktivan (XPB[7] u logičkoj nuli), na bitove XPB[0:6] treba postaviti ASCII-kôd znaka koji se želi ispisati.
- Kad se postavi vrijednost znaka, na bit WR treba poslati pozitivan impuls (stanje mu se iz nule postavi na jedinicu i ponovo vrati u nulu). Impuls uzrokuje upisivanje znaka (postavljenog na XPB[0:6]) na krajnje desno mjesto u interni registar LCD-a.
- Svi ostali do tada ispisani znakovi pomiču se za jedno mjesto ulijevo, a prethodni krajnje lijevi znak se gubi.
- Da bi se znakovi pohranjeni u internom registru prikazali na zaslonu LCD-a, programer treba poslati specijalan ASCII-znak LF (line feed) s kôdom 0A(16).
- Kada LCD primi znak LF, on ne biva spremljen u interni registar, već uzrokuje ispis svih znakova iz internog registra na zaslon LCD-a.
- Postoji još jedan specijalan znak, ASCII-znak CR (carriage return) s kôdom 0D(16), koji briše sadržaj internog registra (na svih osam mjesta upisuje se ASCII kôd 20(16) odnosno znak praznine ili razmaka).

Vremenski dijagram ispisa na LCD



Primjer: Ispis teksta 'FER' na LCD

LCD je spojen na vrata B sklopa GPIO. GPIO je na adresi FFFFFFF00(16). Treba ispisati tekst "FER". Slanje pojedinog znaka na vrata B treba riješiti potprogramom. Potprogram preko R0 prima ASCII znak, a preko R2 baznu adresu sklopa GPIO.

```
ORG 0
MOVN R2, #0FF          ; u R2 upisujemo baznu adresu GPIO-a

START MOV R0, #0D       ; znak 0xD, briše se interni registar
      BL LCDWR          ; piše se na LCD
      MOV R0, #46       ; ASCII kod znaka 'F'
      BL LCDWR          ; piše se na LCD
      MOV R0, #45       ; ASCII kod znaka 'E'
      BL LCDWR          ; piše se na LCD
      MOV R0, #52       ; ASCII kod znaka 'R'
      BL LCDWR          ; piše se na LCD
      MOV R0, #0A       ; znak 0xA, ispis znakova na zaslon
      BL LCDWR          ; piše se na LCD

SWI 123456
```

Primjer: Ispis teksta 'FER' na LCD

; potprogram za ispis jednog znaka na LCD spojen na port B sklopa GPIO
; R0 = ASCII kôd znaka koji treba ispisati
; R2 = bazna adresa sklopa GPIO

LCDWR AND R0, R0, #7F ; postavi bit 7 u nulu (za svaki slučaj) i pošalji znak
 STRB R0, [R2, #4]

 ORR R0, R0, #80 ; postavi bit 7 u jedan (podigni impuls)
 STRB R0, [R2, #4]

 AND R0, R0, #7F ; postavi bit 7 u nulu (spusti impuls)
 STRB R0, [R2, #4]

 MOV PC, LR ; povratak

© Kovač, Basch, FER, Zagreb

17

; potprogram za ispis jednog znaka na LCD spojen na port B sklopa GPIO

; R0 = ASCII kôd znaka koji treba ispisati

; R2 = bazna adresa sklopa GPIO

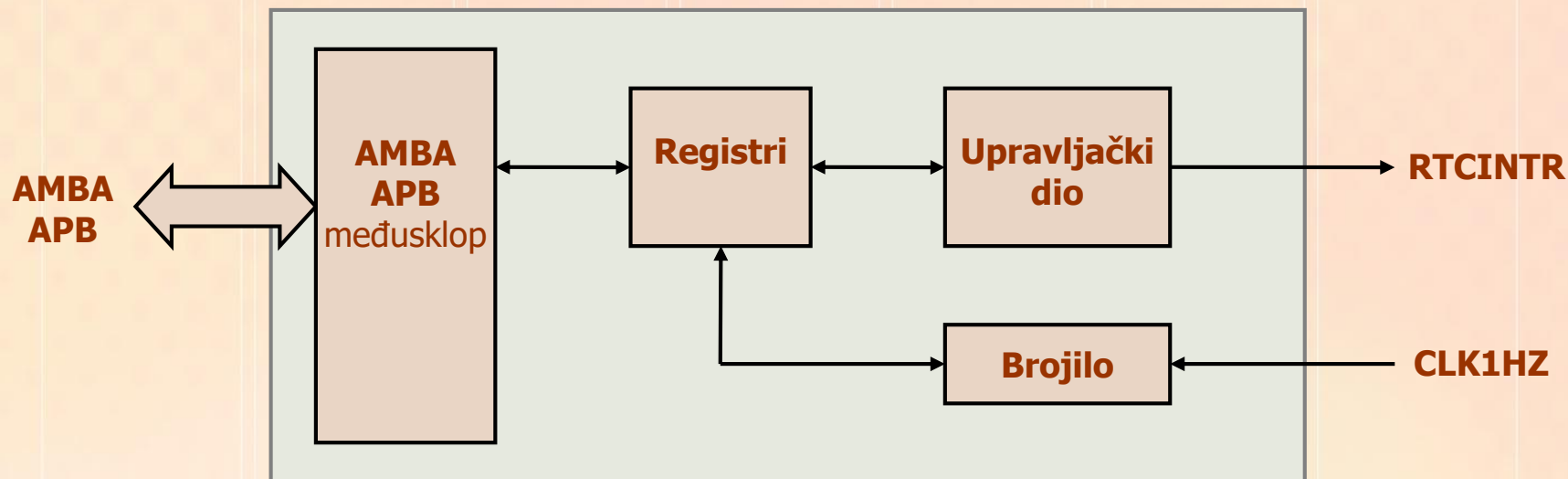
LCDWR	AND R0, R0, #7F	; postavi bit 7 u nulu (za svaki slučaj) i pošalji znak
	STRB R0, [R2, #4]	

```
ORR R0, R0, #80 ; postavi bit 7 u jedan (podigni impuls)
STRB R0, [R2, #4]
```

AND R0, R0, #7F ; postavi bit 7 u nulu (spusti impuls)
STRB R0, [R2, #4]

MOV PC, LR ; povratak

Sklop RTC (Real Time Clock)



Dijelovi arhitekture

- AMBA APB sučelje
 - 32-bitno brojilo
 - 32-bitni registar usporedbe (match register)
 - 32-bitni sklop za usporedbu.
-
- 32-bitno brojilo je glavni dio sklopa RTC. Ovo brojilo **uvećava** svoj sadržaj za jedan na svaki rastući brid signala CLK1HZ. Ako brojilo dođe do vrijednosti FFFFFFFF, tada se na sljedeći brid vrijednost brojila postavlja na 00000000 i nastavlja se dalje normalno brojiti.

Programski model

Adresa	Naziv registra	Opis
RTC_bazna_adr	RTCDR	32-bitni registar podataka (može se samo čitati)
RTC_bazna_adr + 4	RTCMR	32-bitni registar usporedbe
RTC_bazna_adr + 8	RTCSTAT/RTCEOI	1-bitni registar stanja prekida (ako se čita) 0-bitni registar za brisanje prekida (ako se piše)
RTC_bazna_adr + C	RTCLR	32-bitni registar za punjenje brojila
RTC_bazna_adr + 10	RTCCR	1-bitni upravljački registar

RTC: Registri

- RTCDR (Real Time Clock Data Register)
 - RTCDR je 32-bitni registar podataka. Čitanjem ovog registra dobiva se trenutna vrijednost brojila.
 - **Pisanje u ovaj registar nije dozvoljeno! (postavljanje brojila postiže se preko RTCLR registra)**
- RTCMR (Real Time Clock Match Register)
 - RTCMR je 32-bitni registar usporedbe. Upisivanjem podatka u ovaj registar postavlja se nova vrijednost koja služi za usporedbu s brojilom. Čitanjem ovog registra dobiva se zadnja vrijednost upisana u registar usporedbe.

RTC: Registri

- RTCSTAT/RTCEOI (Real Time Clock Interrupt STATus Register/Real Time Clock Interrupt Clear Register)
 - RTCSTAT/RTCEOI je virtualni registar bez fizičkog sklopovlja za pohranjivanje podataka. Pisanjem bilo kojeg podatka na ovu adresu čisti se prekidni signal RTCINTR i pripadni registar*. Čitanjem s ove adrese dobiva se podatak koji na bitu 0 (najniži bit) ima trenutачnu vrijednost RTCINTR. Ako je bit 0 postavljen na jedinicu, to znači da je prekidni signal aktivan.
- RTCLR (Real Time Clock Load Register)
 - RTCLR je 32-bitni registar koji služi za upis vrijednosti u brojilo ili čitanje zadnje vrijednosti koja je upisana. Upisana vrijednost se upisuje u brojilo (koje se kasnije mijenja prilikom odbrojavanja), a vrijednost u ovom registru ostaje nepromijenjena.

* interni status-bistabil

RTC: Registri

- RTCCR (Real Time Clock Control Register)
 - RTCCR je 1-bitni upravljački registar kojim programer može omogućiti ili onemogućiti generiranje prekida. Ako se na bit 0 (najniži bit) ovog registra upiše logička nula, tada se RTC-u onemogućuje generiranje prekida. Ako se upiše jedinica, tada se omogućuje generiranje prekida. Čitanjem ovog registra na bitu 0 dobiva se zadnja upisana vrijednost prekidnog bita. Ostali bitovi u ovom registru ne postoje.

RTC: način rada

- Brojilo se povećava na svaki rastući brid na signalu CLK1HZ
- Kad brojilo dosegne vrijednost upisanu u registru usporedbe (match register), onda RTC postaje spreman i postavlja prekid (ako je omogućen u registru RTCCR)
- Brojilo nastavlja s brojanjem i ne vraća se automatski na nulu*
 - Ako se želi ponoviti ciklus brojenja, onda programski treba u brojilo upisati nulu
- Nakon što postane spreman (ili nakon što postavi prekid) RTC-u treba obrisati stanje (tj. prekid ako ga je postavio)

* Za razliku od FRISC-CT-a u kojem ciklus brojenja automatski započinje iznova

Primjer

Treba napisati program za izmjenično ispisivanje znaka 0 i znaka 1 na zaslonu LCD-a i to tako da se znakovi mijenjaju svakih 5 sekundi. Mjerenje perioda valja riješiti korištenjem RTC-a. Program napisati za ATLAS.

LCD je spojen na vrata B sklopa GPIO (GPIO je na adresi FFFFFFF00).

Na ulaz CLK1HZ od RTC-a je spojen signal frekvencije 1 Hz. RTC je na adresi FFFFE00 i spojen je na IRQ

Primjer

```
ORG 0
B GLAVNI
ORG 18           ; adresa za obradu iznimke IRQ
B PREKIDNI
GLAVNI
MOV R13, #1<16   ; stog

; inicijalizacija RTC-a
LDR R3, RTC      ; u R3 učitaj adresu RTC-a
MOV R4, #5
STR R4, [R3, #4] ; napuni RTCMR
MOV R4, #1
STR R4, [R3, #10] ; omogući prekid (RTCCR)
BL PISI_ZNAK      ; inicijalno pisanje znaka (GPIO ne treba inicijalizirati)
MRS R0, CPSR      ; pročitaj CPSR u R0
BIC R0, R0, #80    ; na mjestu bita I se stavlja nula
MSR CPSR_c, R0    ; i to se ponovo zapisuje u CPSR

PETLJA  B PETLJA  ; beskonačna petlja

GPIO    DW  0FFFFFFF00 ; adresa sklopa GPIO
RTC     DW  0FFFFFFE00 ; adresa sklopa RTC
ZNAK    DW  030        ; trenutni znak za ispis (inicijalno ASCII kod znaka '0')
```


Primjer

PREKIDNI

STMFD R13!, {R0, R3, R14}

LDR R3, RTC ; dohvati adresu sklopa RTC

; reinicijaliziraj RTC

STR R0, [R3, #8] ; čisti se bit RTCINTR (dojava prihvata prekida)

MOV R0, #0

STR R0, [R3, #0C] ; vrati brojač na nulu

; MOV R0, #0A

; STR R0, [R3, #4] ; alternativno stavi RTCMR na 10

BL PISI_ZNAK ; ispiši sljedeći znak

LDMFD R13!, {R0, R3, R14}

SUBS PC, R14, #4 ; povratak iz obrade iznimke

Primjer

; potprogram koji dohvaća znak, mijenja ga iz '0' u '1' (ili obratno) te ga ispisuje na LCD

PISI_ZNAK

STMFD R13!, {R0, R2, R14}

LDR R2, GPIO ; u R2 dohvati adresu sklopa GPIO

MOV R0, #0D ; znak 0xD, briše se interni registar

BL LCDWR ; šalje se na LCD

LDR R0, ZNAK ; dohvati znak iz memorije

EOR R0, R0, #1 ; mijenjaj znak '0'<--->'1'

STR R0, ZNAK ; spremi znak natrag u memoriju

BL LCDWR ; šalje se znak '0' ili '1' na LCD

MOV R0, #0A ; znak 0xA, ispis znaka na zaslonu

BL LCDWR ; šalje se na LCD

LDMFD R13!, {R0, R2, R14}

MOV PC, LR

; potprogram za ispis znaka iz R0 na LCD (spojen na vrata B sklopa GPIO na baznoj adresi R2)

LCDWR ... (potprogram kao u primjeru za LCD na slajdu 17)

Razni primjeri programa za GPIO+RTC

Primjer 1

Računalni sustav sastoji se od procesora ARM, GPIO (FFFF0100) i uređaja za ispis računa koji je spojen na port A (8 podatkovnih priključaka) i na port B (izlazni signal SEND (bit 0) i ulazni signal ACK (bit 1)).

Uređaj radi tako da u trenutku aktiviranja impulsa na signalu SEND počne ispis 8-bitnog podatka koji je prisutan na podatkovnim priključcima (port A).

Dovršetak ispisa svakog 8-bitnog podatka uređaj za ispis dojavljuje procesoru generiranjem impulsa na signalu ACK.

Potrebno je napisati potprogram SEND koji inicijalizira sklop GPIO, te dani niz podataka ispisuje na papirnu traku. Adresa niza prenosi se preko registra R0. Duljina niza nije unaprijed poznata, ali se zna da je niz završen podatkom AA. Adresa GPIO-a zapisana je na fiksnoj memorijskoj lokaciji (na adresi 100)

Glavni program treba korištenjem potprograma SEND poslati na ispis podatke od adrese 200.

Rješenje primjera 1

ORG 0

; glavni program
MOV R13,#1<16

; stog

MOV R0, #2<8
BL SEND
SWI 123456

; parametar za SEND (adresa niza)

; bazna adresa GPIO-a
ORG 100
DW 0FFFF0100

; podaci za ispis
ORG 200
DB 0F0, 14, 12, 05, 0C4, 0AA

Rješenje primjera 1

; potprogram SEND

SEND STMFD R13!, {R0, R1, R2}

MOV R1, #1<8

LDR R2, [R1] ; stavi adresu GPIO-a u R2

; inicijalizacija GPIO-a

; port A: izlazni - 8-bitni podatak

MOV R1, #0xB 11111111

STRB R1, [R2, #8]

; port B: bit 0 izlazni, bit 1 ulazni

MOV R1, #0xB 00000010

STRB R1, [R2, #0C]

Rješenje primjera 1

; petlja za ispis podatka na traku

PISI LDRB R1, [R0], #1 ; dohvat podatka iz niza

CMP R1, #0AA ; provjeri kraj

BEQ KRAJ

STRB R1, [R2]; šalji podatak na port A

; slanje impulsa SEND (port B, bit 0)

MOV R1, #1

STRB R1, [R2, #4]

MOV R1, #0

STRB R1, [R2, #4]

; čekanje impulsa ACK (port B, bit 1)

ACK LDRB R1, [R2, #4]

CMP R1, #0

BEQ ACK

B PISI

; ponovi za sljedeći znak

KRAJ LDMFD R13!, {R0, R1, R2}

MOV PC, LR

Primjer 2

Računalni sustav sastoji se od procesora ARM, sklopa GPIO (FFFF0000) i kontrolne jedinice CTRL koja služi za mjerenje udjela triju sirovina u proizvodnom procesu.

Jedinica CTRL spojena je na GPIO na port A i to sa 4 priključka. Prva tri priključka su ulazni i spojeni su na bitove 0, 1 i 2 (XPA[0]-XPA[2]). CTRL na ovim priključcima daje logičku 1 za svaku od 3 sirovine kada je njezin udio u proizvodnom procesu zadovoljavajući, a inače daje logičku 0. Četvrti priključak je izlazni i spojen je na bit 3 (XPA[3]). Pomoću ovog priključka procesor ARM zaustavlja i pokreće proizvodni proces.

Ako udio bilo koje od sirovina nije zadovoljavajući, potrebno je ugasiti proces slanjem logičke 0 na XPA[3]. Kada udio svih sirovina postane zadovoljavajući, treba uključiti proces slanjem logičke 1 na XPA[3]. Upravljanje procesom ponavljati beskonačno.

Rješenje primjera 2

ORG 0

LDR R1, GPIO

; GPIO Port A inicijalizacija

MOV R0, #0xB 1000

STRB R0, [R1, #8]

PETLJA

; dohvat niža 3 bita podatka s porta A

LDR R0, [R1]

AND R3, R0, #0xB 0111

; ako je udio svih sirovina OK, onda će

; u R2 biti broj 0xB 111

CMP R3, #0xB 111

; stavi 1 na bit 4 od R0,

; tj. uključi proces

MOVEQ R0, #0xB 1000

STREQB R0, [R1]

BEQ PETLJA

; stavi 0 na bit 4 od R0,

; tj. isključi proces

MOV R0, #0xB 0000

STRB R0, [R1]

B PETLJA

; bazna adresa GPIO-a

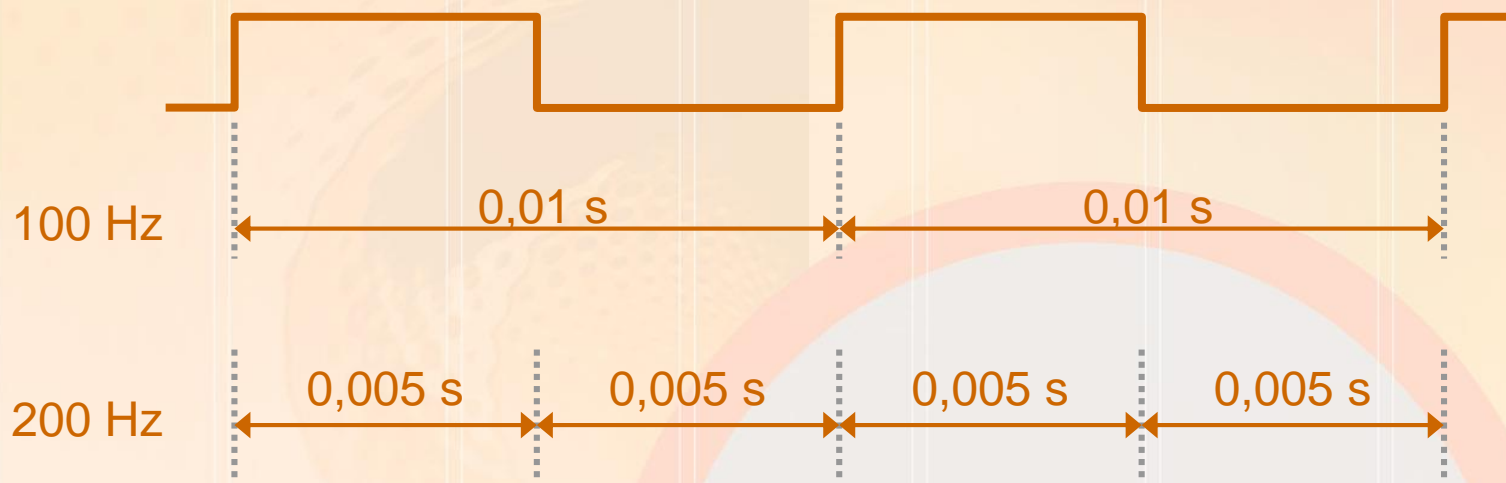
GPIO DW 0xFFFF0000

Komentar: uočite da za komunikaciju sa GPIO-om možemo koristiti naredbe load/store i za riječi i za bajtove

Primjer 3

U računalnom sustavu nalaze se ARM, sklop RTC (na adresi FFFF0000) i sklop GPIO (na adresi FFFF1000). RTC je spojen na FIQ.

Potrebno je na sklopu GPIO, na priključku 0 porta A (XPA[0]) generirati pravokutni signal frekvencije 100 Hz. Na ulaz sklopa RTC doveden je signal frekvencije 10 kHz.



Rješenje primjera 3

ORG 0
B GLAVNI

ORG 1C ; Prekidni potprogram za FIQ

; ne spremamo kontekst jer koristimo registre od načina rada FIQ

MOV R8, #1<8
LDR R9, [R8], #4 ; R9 = RTC bazna adresa
LDR R10, [R8] ; R10 = GPIO bazna adresa

LDRB R8, [R10] ; promijeni stanje
EOR R8, R8, #1 ; pravokutnog signala na
STRB R8, [R10] ; GPIO-u

STR R8, [R9, #8] ; prihvati prekida RTC-a

MOV R8, #0 ; reinicijalizacija RTC-a
STR R8, [R9, #0C]

SUBS PC, LR, #4

Rješenje primjera 3

GLAVNI

MOV R13, #81<8 ; inicijalizacija SP-a

MOV R0, #1<8

LDR R1, [R0], #4 ; R1 = RTC bazna adresa

LDR R2, [R0] ; R2 = GPIO bazna adresa

; inicijalizacija GPIO port A - XPA[0] je izlazni

MOV R0, #00000001

STRB R0, [R2, #8]

; inicijalizacija RTC-a

MOV R0, #50 ; konstanta brojenja

STR R0, [R1, #4] ; RTCMR = 50 = 10 kHz / 200Hz = 10000 / 200

; enable interrupt u RTC-u

MOV R0, #1

STR R0, [R1, #10]

MOV R0, #0

; briši brojilo u RTC-u

STR R0, [R1, #0C] ; RTCLR = 0

Rješenje primjera 3

; omogućavanje prekida FIQ

MRS R0, CPSR

BIC R0, R0, #40

MSR CPSR_c, R0

PETLJA B PETLJA

; bazne adrese RTC-a i GPIO-a

ORG 100

DW 0FFF0000 ; RTC

DW 0FFF1000 ; GPIO

Rješenje primjera 3

ORG 0
B GLAVNI

ORG 1C ; Prekidni potprogram za FIQ

; ne spremamo kontekst jer koristimo registre od načina rada FIQ

MOV R8, #1<8
LDR R9, [R8], #4 ; R9 = RTC bazna adresa
LDR R10, [R8] ; R10 = GPIO bazna adresa

LDRB R8, [R10] ; promijeni stanje
EOR R8, R8, #1 ; pravokutnog signala na
STRB R8, [R10] ; GPIO-u

STR R8, [R9, #8] ; prihvati prekida RTC-a

MOV R8, #0 ; reinicijalizacija RTC-a
STR R8, [R9, #0C]

SUBS PC, LR, #4

Primjer 4

U računalnom sustavu nalaze se ARM, GPIO (FFFF1000) i RTC (FFFF0000).

GPIO preko vrata A prima 8-bitni NBC podatak koji predstavlja temperaturu nekog procesa (u Celsijevim stupnjevima). Pinovi 0, 1 i 2 od vrata B služe za signalizaciju - pomoću njih se pale i gase lampice: crvena (XPB[0]), žuta (XPB[1]) i zelena (XPB[2]). Lampice se pale slanjem logičke 1, a gase slanjem logičke 0 na odgovarajući priključak.

Ako je temperatura veća od 128°C, treba upaliti crvenu lampicu, a ugasiti žutu i zelenu. Isto treba učiniti sa žutom lampicom, ako je temperatura u rasponu od 55°C do 128°C, odnosno sa zelenom ako je temperatura manja od 55°C.

Svake sekunde potrebno je mjeriti temperaturu i obaviti navedenu signalizaciju. Na RTC je spojen signal frekvencije 100 Hz.

Rješenje primjera 4

ORG 0
B GLAVNI

ORG 18 ; adresa prekidnog potprograma
B PREKIDNI

GLAVNI

MOV R13, # 1<16 ; inicijalizacija SP-a

; GPIO je inicijalno dobro postavljen

; inicijalizacija RTC-a

LDR R2, RTC ; R2 = bazna adresa RTC-a

MOV R3, #0

STR R3, [R2, #0C] ; RTCLR - brojilo

MOV R3, #0xD 100

STR R3, [R2, #4] ; RTCMR - konstanta

MOV R3, #1

STR R3, [R2, #10] ; RTCCR - prekidanje

; omogućiti prekid IRQ

MRS R0, CPSR

BIC R0, R0, #80

MSR CPSR_c, R0

PETLJA B PETLJA

GPIO DW 0FFFF1000 ; bazna adresa GPIO-a

RTC DW 0FFFF0000 ; bazna adresa RTC-a

Rješenje primjera 4

PREKIDNI

STMFD R13!, {R2, R4, R7}

LDR R2, RTC ; R2 = RTC bazna adresa
STR R7, [R2, #8] ; brisanje int-zastavice u RTC-u

MOV R7, #0
STR R7, [R2, #0C] ; RTCLR - brisanje brojila

; dohvati temperaturu sa porta A

LDR R2, GPIO ; R2 = bazna adresa GPIO-a
LDR R4, [R2]

; ispitaj u kojem je opsegu temperatura

CMP R4, #0xD 128
MOVHI R7, #0xB 001 ; CRVENA = bit 0
BHI SIG

CMP R4, #0xD 55
MOVLO R7, #0xB 100 ; ZELENA = bit 2
BLO SIG

MOV R7, #0xB 010 ; ŽUTA = bit 1

SIG STR R7, [R2, #4] ; signalizacija lampicama

LDMFD R13!, {R2, R4, R7}
SUBS PC, LR, #4

