ARM

Prekinitve in prekinitveni krmilnik

Prekinitve in izjeme

Prekinitev ali izjema:

- dogodek, ki povzroči, da procesor prekine izvajanje trenutno izvajajočega se programa
- vsaka prekinitev ali izjema ima običajno svoj prekinitveno-servisni program (PSP)'
- procesor ARM ima vektorske prekinitve: vsaki prekinitvi oz. izjemi pripada en vektor
- prekinitveni vektor pri ARM: naslov pomnilniške besede v kateri je shranjen prvi ukaz PSP
- vsaki prekinitvi/izjemi pripada en način delovanja: ob prekinitvi procesor spremeni način delovanja

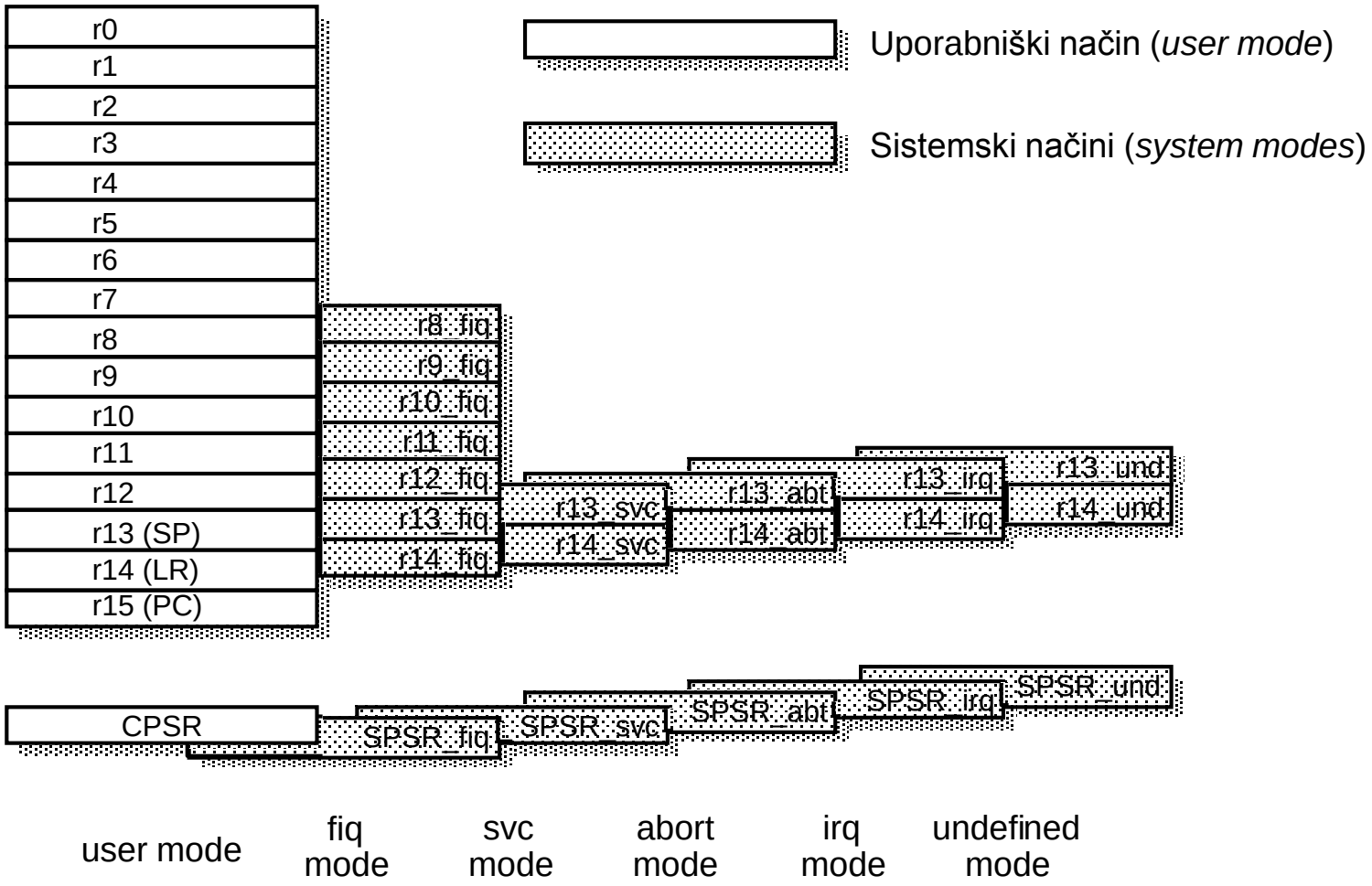
Prekinitve in izjeme pri ARM:

- FIQ – Fast Interrupt Request - zunanja prekinitev, nastopi ob aktiviranju FIQ# priključka
- IRQ – Interrupt Request - zunanja prekinitev, nastopi ob aktiviranju IRQ# priključka
- SWI – programska prekinitev
- Reset
- Prefetch abort – izjema, ki nastopi ob neuspešnem branju ukaza (npr. zgrešitev v predpomnilniku, napaka strani ipd.)
- Data abort – izjema, ki nastopi ob neuspešnem dostopu do operanda (npr. zgrešitev v predpomnilniku, napaka strani ipd.)
- Undefined instruction – izjema, ki nastopi ob branju nedefiniranega ukaza (sproži tudi ko izvajamo priviligirane ukaze v uporabniškem načinu)

Prekinitve in izjeme

Prekinitev / Izjema	Način	Vektor
Reset	svc	0x00000000
Undefined Inst.	und	0x00000004
SWI	svc	0x00000008
Prefetch Abort	abt	0x0000000C
Data Abort	abt	0x00000010
IRQ	irq	0x00000018
FIQ	fiq	0x0000001C

Prekinitve in izjeme – načini delovanja CPE



Prekinitve in izjeme

Ob prekinitvi / izjemi se avtomatsko zgodi naslednje:

- CPSR se prepiše v SPSR zahtevanega načina delovanja
- PC se shrani v LR zahtevanega načina delovanja
- V CPSR se zapiše zahtevani način delovanja
- V PC se prepiše prekinitveni vektor

Ob prekinitvi / izjemi procesor spremeni način delovanja! Zato ob resetu ne pozabimo nastaviti skladovnega kazalca za posamezne načine delovanja!

Pri prepisu prekinitvenega vektorja v PC se zajame ukaz iz vektorske tabele in s tem se začne izvajanje določenega PSP v zahtevanem načinu delovanja.

Prekinitve in izjeme – vektorska tabela

Vsak prekinitveni vektor vsebuje lahko le en ukaz. To je ukaz s katerim skočimo na PSP za pripadajočo prekinitev oz. izjemo. Ta ukaz je lahko:

- **`B naslov`** – s tem ukazom skačemo relativno glede na PC. PSP se mora začeti blizu vektorske tabele.
- **`ldr pc, [pc, #odmik]`** – s tem ukazom v PC naložimo naslov PSP iz pomnilnika. Naslov je tokrat 32-biten in lahko skočimo na poljuben naslov v pomnilniku, vendar se PSP začne izvajati pozneje zaradi dodatnega dostopa do pomnilnika.
- **`ldr pc, [pc, #-0x0F20]`** – tak ukaz uporabimo pri uporabi prekinitvenega krmilnika. V nadaljevanju bomo spoznali zakaj.

Prekinitve in izjeme

Prekinitev / Izjema	Prioriteta	Postavi I bit v CPSR	Postavi F bit v CPSR
Reset	1	da	da
Undefined Inst.	6	da	ne
SWI	6	da	ne
Prefetch Abort	5	da	ne
Data Abort	2	da	ne
IRQ	4	da	ne
FIQ	3	da	da

Prekinitve in izjeme – Link Register

Pri odzivu CPE na prekinitveno zahtevo CPE najprej dokonča izvajanje trenutno izvajajočega se ukaza in nato PC prepiše v LR. Tako v LR hranimo “povratni” naslov. Vendar se zaradi cevovodnega izvajanja v LR praviloma ne zapiše naslov ukaza pri katerem je bil program prekinjen (v tabeli je to pc), ampak:

Prekinitev / Izjema	Naslov, ki se zapiše v LR	Dejanski povratni naslov
Reset	--	--
Undefined Inst.	pc+4	lr
SWI	pc+4	lr
Prefetch Abort	pc+8	lr-4
Data Abort	pc+12	lr-8
IRQ	pc+8	lr-4
FIQ	pc+8	lr-4

Prekinitev IRQ

Poglejmo si mehanizem odziva na prekinitev IRQ ter primer preprostega PSP za IRQ. Pri odzivu na IRQ prekinitev se v CPE zgodi naslednje:

- CPSR se prepiše v SPSR_irq
- PC+8 se shrani v LR_irq, pri čemer je v PC naslov zadnjega izvedenega ukaza v prekinjenem programu
- V CPSR se zapiše irq način delovanja
- V PC se naloži 0x00000018. Na tem naslovu je skočni ukaz na PSP.

Prekinitveno-servisni program:

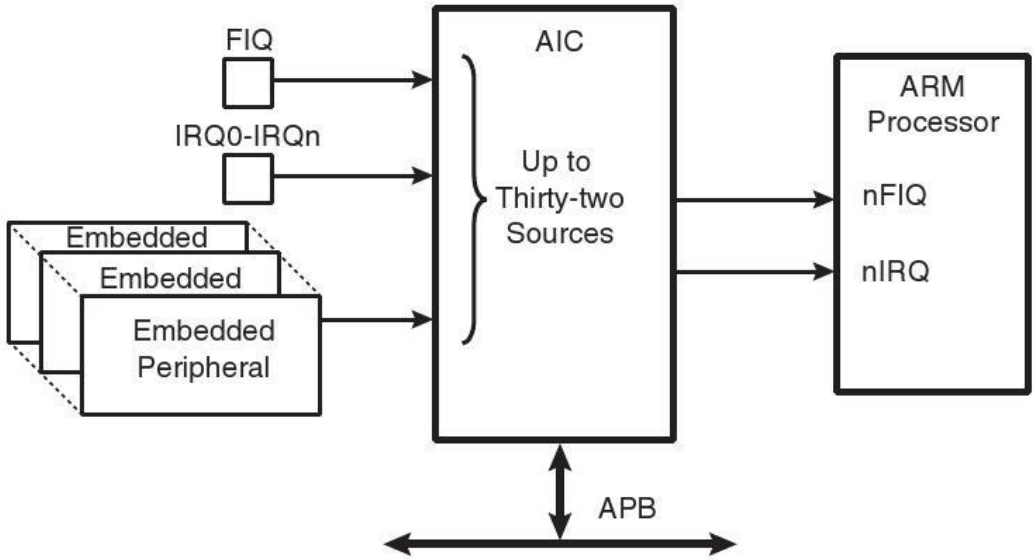
```
IRQ_hand:
        sub r14,r14,#4  /* popraviti moramo povratni naslov */
        stmfd r13!, {r14} /* na sklad shrani povratni naslov.
                             Po potrebi še druge registre! */

        ...               /* servisiraj zahtevo ... */


        ldmfd r13!, {pc}^ /* vrni se na prekinjeni program,
                             pred tem obnovi CPSR iz SPSR_irq*/
```

Prekinitev IRQ

Takoj po resetu ne smemo pozabiti nastaviti skladovnega kazalca za irq način delovanja:

```
_start:
/* izberi irq način */
  mrs r0, cpsr
  bic r0, r0, #0x1F   /* pobriši zastavice načina delovanja */
  orr r0, r0, #0x12   /* nastavi irq način */
  msr cpsr, r0

/* nastavi irq kazalec na sklad */
  ldr sp,_Lirqstack_end

/* izberi uporabniški način način */
  mrs r0, cpsr
  bic r0, r0, #0x1F   /* pobriši zastavice načina delovanja */
  orr r0, r0, #0x10   /* nastavi uporabniški način */
  msr cpsr, r0

/* nastavi uporabniški kazalec na sklad */
  ldr sp,_Lstack_end

/* glavni program */
```

Prekinitveni krmilnik (Advanced Interrupt Controller)



Osnovne lasnosti AIC:

- **Omogoča priključitev 32 prekinitvenih izvorov na procesor ARM**
- **Prekinitvene izvore priključi na prekinitvena vhoda FIQ# in IRQ# na procesorju ARM**
- **AIC sprejema prekinitvene zahteve iz teh 32 virov ter ustrezno aktivira signala FIQ# ali IRQ#**
- **AIC omogoča prioritetno razvrščanje prekinitvenih izvorov**
- **vsakemu izvoru lahko programsko določimo enega izmed 8 prioritetnih nivojev**
- **vsi prekinitveni vhodi so lahko proženi na nivo ali fronto – to določimo programsko**
- **na prekinitvene vhode je lahko povezana poljubna vgrajena V/I ali neka zunanja naprava**

Prekinitveni krmilnik – prekinitveni vhodi

Priključitev prekinitvenih izvorov/vhodov:

- **Prekinitveni izvori: IS0 – IS31 (Interrupt Sources)**
- **IS0 – vedno priključen na FIQ vhod vezja AIC**
- **IS1 – to je t.i. sistemska prekinitev – na ta izvor so preko OR vrat priključene naslednje sistemske naprave:**

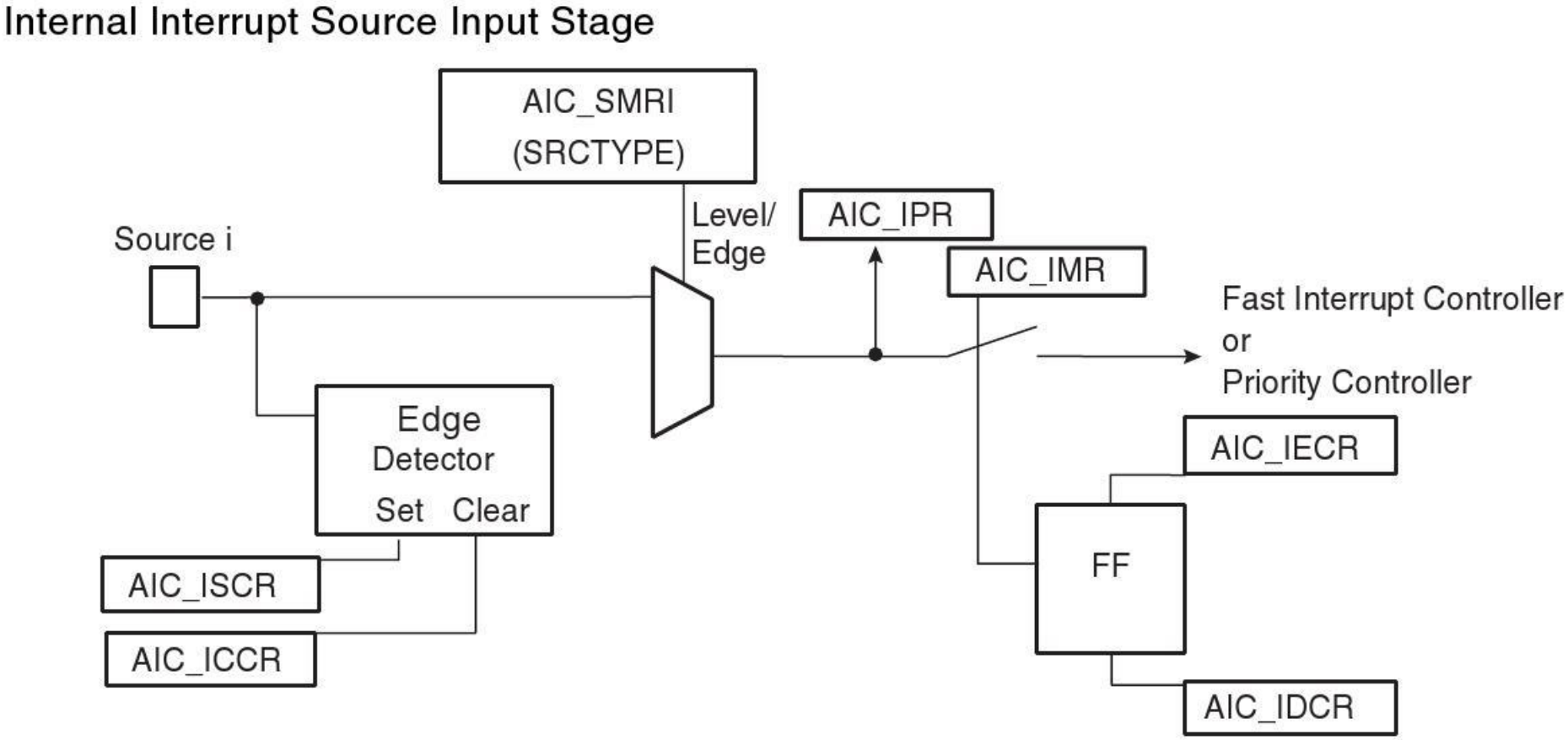
System Timer	(sistemski časovnik)
RTC	(ura realnega časa)
Power Menagement Controller	(krmilnik za upravljanje porabe)
Memory Controller	(pomnilniški krmilnik)
Debug Unit	(enota za razhroščevanje)

- **IS2-IS31 – na te izvore so priključene vgrajene V/I naprave; mesto priključka je določeno z ID vgrajene naprave (npr. TC0 z ID 17 je priključen na IS17)**

Privzeta nastavitev je, da vsi prekinitveni izvori, razen IS0, povzročijo IRQ prekinitev v CPE. Izvor IS0 povzroči FIQ prekinitev. Programsko sicer lahko določimo FIQ prekinitev tudi za izvore IS1-IS31.

Prekinitveni krmilnik

Vhodna stopnja za vgrajene V/I naprave:



Prekinitveni krmilnik – funkcionalni opis

AIC_SMR[0..31] : Source Mode Register – vsak IS ima pripadajoči AIC_SMR

PRIOR (biti 0-2) : določimo prioriteto prekinitvenega izvora

0 – najnižja, 7 - najvišja

SRCTYPE (bita 5-6) : določimo občutljivost IS

00 – prekinitev se proži ob visokem stanju na IS

01 – prekinitev se proži ob pozitivni fronti na IS

AIC_IECR : Interrupt Enable Comand Register

vpis 1 v bit *i* omogoča prekinitve za IS*i*

AIC_IDCR : Interrupt Disable Comand Register

vpis 1 v bit *i* onemogoča (maskira) prekinitve za IS*i*

AIC_IMR : Interrupt Mask Register

V tem registru lahko preberemo prekinitvene maske za posamezne IS. Če je bit i = 0, je prekinitev za ISi onemogočena (maskirana), sicer je omogočena

Prekinitveni krmilnik – funkcionalni opis

AIC_IPR : Interrupt Pending Register

stanje posameznega bita določa, ali je nek IS zahteval prekinitev
bit i = 0: ISi ni zahteval prekinitev
bit i = 1: ISi je zahteval prekinitev, procesor se še ni odzval

AIC_ISR : Interrupt Status Register

IRQID (biti 0-5) – tu je zapisana številka IS, ki je zahteval in sprožil prekinitev

AIC_SVR[0..31] : Interrupt Source Vector Register

v te registre za vsak IS vpišemo prekinitveni vektor oz. naslov PSP, ki ga sami določimo

AIC_IVR : Interrupt Vector Register

V tem registru lahko preberemo prekinitveni vektor za tisti IS, ki je sprožil trenutno prekinitev. V tem registru dobi CPE dejanski prekinitveni vektor ali naslov PSP za IS, ki je prekinitev prožil.

Prekinitveni krmilnik – delovanje

Predpostavimo, da prekinitveni izvor IS17 zahteva prekinitev. Predpostavimo še, da so kontrolni registri ustrezno nastavljeni (prekinitev omogočena, določena prioriteta) ter da smo PSP za ta izvor napisali na naslovu 0x12345678. Potem moramo v register AIC_SVR17 zapisati 0x12345678. Ob zahtevi za prekinitev iz izvora IS17 se zgodi naslednje:

• Prekinitveni krmilnik mora najprej ugotoviti ali se že servisira kakšna prekinitev.

• Kako AIC ve, ali se je CPE odzvala na kakšno prekinitev in jo servisira?

• CPE mora ob odzivu na prekinitev prebrati prekinitveni vektor oz. naslov PSP iz registra AIC_IVR – branje iz tega registra je za AIC znak, da je CPE začela s servisiranjem prekinitve.

• Če CPE že servisira neko prekinitev in je prioriteta servisirane prekinitve nižja od IS17, potem AIC aktivira prekinitveni vhod na CPE – s tem proži novo prekinitev in CPE prekine servisiranje trenutne (ob predpostavki, da v CPE prekinitve niso maskirane)

Prekinitveni krmilnik – delovanje

• Če CPE že servisira neko prekinitev in je prioriteta servisirane prekinitve višja od prioritete IS17, potem mora AIC počakati da CPE zaključi s servisiranjem prekinitve preden ponovno aktivira prekinitvena vhoda na CPE!

• Kako AIC ve, ali je CPE zaključila s servisiranjem neke prekinitve?

• CPE mora ob zaključku servisiranja prekinitve zapisati poljubno vrednost v register AIC_EOICR ! Gre za t.i. slepo pisanje. Pisalni dostop do AIC_EOICR pove AIC, da je CPE zaključila s servisiranjem, in jo lahko spet prekine.

Prekinitveni krmilnik – vektorske prekinitve

AIC omogoča vektorske prekinitve. Ob vsaki prekinitvi tako CPE lahko ugotovi, kateri izvor je prožil prekinitev. To deluje na naslednji način:

• Za vsak prekinitveni izvor moramo v pripadajoči register AIC_SVR zapisati ustrezen vektor ali naslov PSP.

• Ko prekinitveni izvor i zahteva prekinitev, prepiše AIC vsebino iz registra AIC_SVRi v register AIC_IVR, tj:

```
AIC_IVR  <- AIC_SVRi
```

• CPE bo prejela prekinitveni signal preko vhodov IRQ# ali FIQ#. ‘Odzvala’ se bo torej na znanih vektorjih 0x00000018 ali 0x0000001C.

• CPE mora najprej prebrati vsebino AIC_IVR v PC, da bo lahko pričela z izvajanjem ustreznega PSP. Kako preberemo AIC_IVR v PC?

Prekinitveni krmilnik – vektorske prekinitve

• AIC je pomnilniško preslikan na naslov 0xFFFFF000

• register AIC_IVR je na odmiku 0x100, oz. na naslovu 0xFFFFF100

• ob IRQ prekinitvi se v PC zapiše 0x00000018. Na tem naslovu (v vektorski tabeli) moramo zapisati ukaz, s katerim v PC prepišemo AIC_IVR :

```
0x00000018:  ldr pc, [pc, #-0x0F20]
```

saj je 0x00000018 + 8 – 0x0F20 = 0xFFFFF100.

• tako z enim samim ukazom skočimo na PSP in s tem zmanjšamo latenco pri odzivu na prekinitve!

Prekinitveni krmilnik – uporaba

Naslovi AIC registrov:

```
.equ AIC_BASE,    0xFFFFF000        /* Začetni (bazni) naslov AIC */
.equ AIC_SMR17,   0x044             /* odmiki posameznih registrov v AIC */
.equ AIC_SVR17,   0x0C4
.equ AIC_IVR,     0x100
.equ AIC_IECR,    0x120
.equ AIC_EOICR,   0x130
```

Omogočimo prekinitve za IS17, nastavimo prioriteto 4 in odzivnost ter določimo prekinitveni vektor:

```
AIC_IS17_Init:
        ldr r0, =AIC_BASE
        mov r1, #4                  /* high level sensitive, PRIOR=4 */
        str r1, [r0, #AIC_SMR17]
        ldr r1, =IRQ17_Hand
        str r1, [r0, #AIC_SVR17]    /* set interrupt vector for IS17 */
        mov r1, #1 << 17
        str r1, [r0, #AIC_IECR]     /* enable interrupt for IS17 */
        str r0, [r0, #AIC_EOICR]    /* slepo pisanje v EOICR */
```

Prekinitveni krmilnik – zgled

Vektorska tabela:

```
B       _start          /* RESET INTERRUPT */
B       _undef          /* UNDEFINED INSTRUCTION INTERRUPT */
B       _swi            /* SOFTWARE INTERRUPT */
B       _abort_pref     /* ABORT (PREFETCH) INTERRUPT */
B       _abort_data     /* ABORT (DATA) INTERRUPT */
B       _reserved       /* RESERVED */
LDR r15, [r15, #-0x0F20] /* IRQ INTERRUPT */
LDR r15, [r15, #-0x0F20] /* FIQ INTERRUPT */
```

Ne pozabite nastaviti SP za VSE načine delovanja !!!

Prekinitveni krmilnik – zgled

PSP za IS17:

```
IRQ17_Hand :
        sub r14,r14,#4 /* ustrezno popravi povratni naslov */
        stmfd r13!, {…,r14} /* na sklad shrani povratni naslov. */
        /* ter preostale registre, ki jih uporabljaš v PSP*/

        /* … PSP … */
        …

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

        /* vrni se iz PSP in obnovi registre ter CPSR: */
        ldmfd r13!, {…,pc}^
```