Arm načini rada

Načini rada - njihova svrha

- Pri izvedbi računalnih sustava želimo omogućiti kontroliran pristup određenim dijelovima sustava
  - Npr. neki korisnički program ne bi smio moći promijeniti konfiguraciju sustava
- Izvedba OS-a

Procesorski načini rada

- ARM arhitektura koju mi razmatramo u ovim predavanjima podržava sedam procesorskih **načina rada** (engl. modes).
- Postoji jedan korisnički način rada (User) i šest **privilegiranih** načina rada
- U određenom procesorskom načinu omogućen je ili onemogućen pristup određenim resursima sustava
- Onemogućavanje pristupa nekim resursima u korisničkom načinu je od velike važnosti za izvedbu operacijskih sustava
- Procesorski način se definira postavljanjem najnižih 5 bitova u registru CPSR

Procesorski načini rada

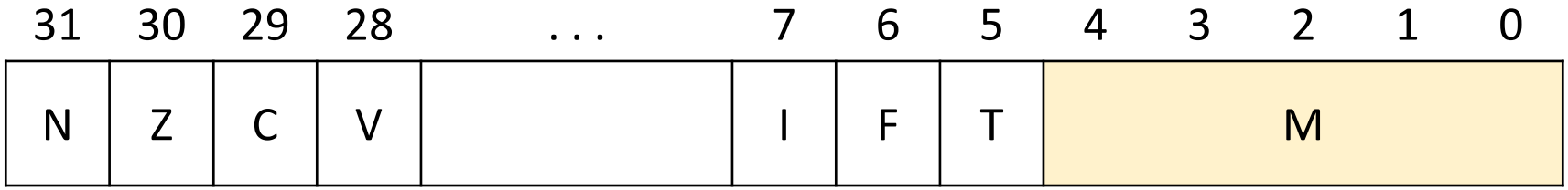
Način	Oznaka	Opis načina
User	usr	Normalno izvođenje korisničkih programa
System	sys	Izvođenje privilegiranih zadataka operacijskog sustava
Supervisor	svc	Zaštićeni način za operacijski sustav
Abort	abt	Za obradu memorijskih grešaka i izvedbu virtualne memorije
Undefined	und	Za obradu nedefiniranih naredaba i za programsku emulaciju koprocesora
Interrupt	irq	Za obradu običnog prekida
Fast Interrupt	fiq	Za obradu brzog prekida

Privilegirani načini rada

Procesorski načini rada - CPSR

- Vrijednosti u bitovima M registra CPSR za sve načine rada su:

Način rada	Vrijednost bitova M u CPSR (binarno)
User	10000
System	11111
Supervisor	10011
Abort	10111
Undefined	11011
Interrupt	10010
Fast Interrupt	10001



Registri

- Do sada smo rekli da su programeru dostupni registri R0-R15 i CPSR
- No Arm interno ima još registara (ukupno 37):
  - 31 registar opće namjene (uključujući i programsko brojilo).
    - Važno je uočiti da (kao što smo inicijalno objasnili) **u jednom trenutku možemo koristiti samo 16 registara** opće namjene (R0-R15) dok ostali nisu u to vrijeme dostupni !!!
  - 6 registara programskog stanja. Registri programskog stanja su također 32-bitni, no samo neki bitovi se koriste te ostali bitovi ne moraju nužno biti izvedeni u sklopovlju.

Registri

- Registri opće namjene (R0-R15)

- Registri opće namjene, R0-R15 mogu se podijeliti u tri grupe:
  - Jednoznačno definirani registri R0-R7
  - Višeznačno definirani registri R8-R14
  - Programsko brojilo R15

- Registri programskog stanja (CPSR, SPSR)

Registri - Procesorski načini rada

user/system	service	abort	undefined	interrupt	fast interr.
R0					
R1					
R2					
R3					
R4					
R5					
R6					
R7					
R8					R8_fiq
R9					R9_fiq
R10					R10_fiq
R11					R11_fiq
R12					R12_fiq
SP=R13	R13_svc	R13_abt	R13_und	R13_irq	R13_fiq
LR=R14	R14_svc	R14_abt	R14_und	R14_irq	R14_fiq
PC=R15					
CPSR					
	SPSR_svc	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

Način rada (mode)

Jednoznačno definirani registri R0-R7

Višeznačno definirani registri R8-R14

Programsko brojilo

Statusni registri

Jednoznačno definirani registri R0-R7

- Procesor ARM sadrži osam fizičkih registara nazvanih R0-R7.
- Registri R0-R7 su jednoznačno definirani, te će program pristupati istim fizičkim registrima bez obzira na to u kojem se načinu rada procesor nalazi.
- Ovi registri su, u punom značenju pojma, registri opće namjene jer za njih nije predviđeno nikakvo posebno značenje te se slobodno mogu koristiti u svim mogućim situacijama.
- Prema tome, za sve načine rada, registri R0- R7 su identični.

Višeznačno definirani registri R8-R14

- Za razliku od R0-R7, prilikom pristupa nekom od registara R8-R14 adresirat će se određeni fizički registar ovisno o tome u kojem načinu rada se procesor nalazi.

- Na primjer, ako se procesor nalazi u privilegiranom načinu Supervisor:
  - pri adresiranju registra R13 procesor će pristupiti posebnom fizičkom registru R13_svc koji je dostupan samo u ovom procesorskom stanju. Isto vrijedi i za registar R14.
  - prilikom pristupanja registrima koji nisu višeznačno definirani (R0-R12 i R15 za način Supervisor), uvijek se pristupa zajedničkim fizičkim registrima.

- VAŽNO: Pri pisanju programa mi NE MOŽEMO napisati npr. R13_svc ili R8_fiq već UVIJEK KORISTIMO SAMO npr R13 ili R8 (znači obična imena registara). PROCESOR INTERNO izabire kojem će fizičkom registru u pojedinom trenutku pristupiti

Višeznačno definirani registri R8-R14

- Razlog za korištenje dodatnih fizičkih registara R13 i R14 je u tome što se registri R13 i R14 koriste za posebne namjene.

- Registar R13 se (po dogovoru) koristi kao pokazivač na vrh stoga (SP, Stack Pointer) te je ovime omogućeno da se u svakom načinu rada može definirati neovisan stog.
  - U programima R13 možemo alterno nazivati i SP

- Registar R14 se koristi kao registar za pohranjivanje povratne adrese za vraćanje iz potprograma ili iznimke.
  - U programima R14 možemo alterno nazivati i LR

- Oba ova registra se mogu koristiti i kao registri opće namjene ukoliko ih sustav ne koristi za ove posebne namjene.

Višeznačno definirani registri R8-R14

- U načinu brzog prekida (fiq) adresira se skup od 7 posebnih fizičkih registara (R8_fiq-R14_fiq).

- To omogućuje da se obrada brzog prekida izvede čim brže - bez potrebe za spremanjem konteksta.

Programsko brojilo R15

- Registar R15 je **programsko brojilo** te se u većini slučajeva korištenje ovog registra za opće namjene ne preporuča, a često i nije dozvoljeno.
  - U programima R15 možemo alterno nazivati i PC

- Čitanjem podatka zapisanog u R15 dobije se vrijednost adrese trenutne naredbe + 8 bajtova (zbog protočne strukture).
- Valja uočiti da su pri dohvatu naredbe najniža dva bita uvijek postavljena u logičku nulu zbog toga jer su naredbe ARM-a uvijek poravnate na širinu riječi.
- Upis vrijednosti u R15 izvodi neku vrstu forsiranog skoka, no ovaj način korištenja se ne preporučuje.

Registri programskog stanja (CPSR, SPSR)

- U registru trenutnog programskog stanja CPSR (Current Program Status Register) spremljeni su bitovi koji definiraju različita stanja procesora i programa.

- Registar CPSR je dostupan u svim procesorskim načinima.

- Registar pohranjenog programskog stanja SPSR (Saved Program Status Register) koristi se kad procesor obrađuje iznimke. U njega se pohranjuje vrijednost registra CPSR

“User” način

- Normalan način u kojem se izvode korisnički programi.
- U ovom načinu program ne može koristiti zaštićene resurse sustava te ne može promijeniti način rada.
- Pokušaj pisanja u CPSR[23:0] u User načinu rada procesor ignorira, tako da programi koji se izvode u User načinu ne mogu promijeniti način u neki od privilegiranih

Privilegirani procesorski načini

- Ostali načini smatraju se privilegiranim i služe pri izvođenju specifičnih operacija.
- Programi koji se izvode u privilegiranim načinima imaju potpuni pristup svim resursima sustava.

- Načini fiq, irq, svc, abt i und aktiviraju se kad se u sustavu generira iznimka

- Način svc se aktivira:
  - na početku rada procesora, tj. kada se procesor resetira
    - naši programi na predavanjima i na labosima rade većinom u ovom načinu
  - kada se izvede programski prekid (*software interrupt*) pomoću naredbe SWI
    - U simulatoru na labosima naredba SWI služi za zaustavljanje rada procesora, tj. kao kraj simulacije

Privilegirani procesorski načini

- fiq i irq aktiviraju se kada vanjska jedinica zatraži prekid preko priključaka FIQ odnosno IRQ te kada procesor prihvati taj zahtjev
- *abt* i *und* nećemo koristiti (ne mogu se koristiti ni na labosima)
  - *abt* se aktivira kada dođe do pogreške u pristupu memoriji (aktivira ga obično MMU zbog pristupa nepostojećem ili zaštićenom dijelu memorije, ili zbog pogreške u poravnatosti adrese itd.)
    - *prefetch abort* je pogreška prilikom dohvata strojnog koda
    - *data abort* je pogreška prilikom pristupa običnim podatcima
  - *und* se aktivira kada prilikom dekodiranja ARM prepozna nepostojeći strojni kod (*undefined*) ili ako se prepozna koprocesorska naredba, a koprocesor ne pošalje odgovarajući odgovor

- Način *sys* je privilegiran, ali se u njega ne ulazi uslijed iznimke i nema zasebne registre (služi za poslove vezane za operacijski sustav). Način *sys* također nećemo koristiti.

- Programi pisani u SSPARCSS-u imat će pristup svim resursima sustava.

Iznimke

Obrada iznimaka

- U ARM-u se **iznimkama** nazivaju razne vrste događaja koje ne ulaze u normalno slijedno izvođenje naredaba. Na primjer: pojava prekida, dohvat neispravnog strojnog koda, resetiranje procesora, itd.

- Procesor ARM načelno obrađuje iznimke na sljedeći način:
  - ARM pohranjuje R15 u registar LR (osim za RESET)
  - ARM pohranjuje CPSR u registar SPSR (osim za RESET)
  - ARM prelazi u privilegirani način rada koji ovisi o vrsti iznimke
  - ARM skače u potprogram za obradu iznimke za dotični način rada

* osim za RESET

Obrada iznimaka

- Adrese potprograma za obradu iznimke su fiksno definirane za svaku pojedinu iznimku

- Ako se u isto vrijeme pojave dvije ili više iznimaka, ARM poslužuje onu s većim prioritetom
  - prioriteti su fiksno zadani

- Nakon dovršetka potprograma za obradu iznimke, izvođenje se vraća u „glavni program” na mjesto gdje se iznimka i dogodila (osim za iznimku *Reset*)

Iznimke: adrese potprograma i prioriteti

Tip iznimke	ARM se prebacuje u način rada	Adresa potprograma*	Prioritet
Reset	Supervisor	0x00000000	1
Undefined instruction	Undefined	0x00000004	6
Software interrupt (SWI)	Supervisor	0x00000008	6
Prefetch Abort (instruction fetch memory abort)	Abort	0x0000000C	5
Data Abort (data access memory abort)	Abort	0x00000010	2
IRQ (interrupt)	IRQ	0x00000018	4
FIQ (fast interrupt)	FIQ	0x0000001C	3

* Uočite da je za svaki potprogram na raspolaganju samo 4 bajta, pa se tu u pravilu stavlja naredba skoka na odsječak za obradu iznimke (osim potprograma na adresi 0x1C)

Reset

- Slijed operacija koje se obave pri pojavi pojedine iznimke vrlo je sličan za sve iznimke
- Kada se na ulazu u procesor aktivira signal Reset, procesor odmah prekida izvođenje naredbe. Nakon što se Reset deaktivira, obavi se sljedeći niz operacija:

  - R14_svc = UNPREDICTABLE value
  - SPSR_svc = UNPREDICTABLE value
  - CPSR[4:0] = 0b10011 /* Enter Supervisor mode */
  - CPSR[5] = 0 /* Execute in ARM state */
  - CPSR[6] = 1 /* Disable fast interrupts */
  - CPSR[7] = 1 /* Disable normal interrupts */
  - PC = 0x00000000

Reset

- Povratak iz iznimke Reset nije predviđen. U slučaju inicijalizacije procesora (npr. uključivanje napajanja) očekuje se da će vanjska logika aktivirati signal Reset kako bi procesor normalno započeo s radom.

- Prema tome, svaki program mora pretpostaviti obradu iznimke Reset i nakon toga prelazak na izvođenje izabrane aplikacije.

IRQ (prekid)

- Ako je u programu omogućeno prihvaćanje prekida (bit I u CPSR je obrisan), procesor će na kraju izvođenja svake naredbe provjeriti je li ulaz IRQ aktivan. Ako je neki vanjski sklop aktivirao signal prekida (IRQ), procesor će nakon završetka trenutne naredbe obaviti sljedeći niz operacija:

  - R14_irq = address of next instruction to be executed + 4
  - SPSR_irq = CPSR
  - CPSR[4:0] = 0b10010 /* Enter IRQ mode */
  - CPSR[5] = 0 /* Execute in ARM state */
  - /* CPSR[6] is unchanged */
  - CPSR[7] = 1 /* Disable normal interrupts */
  - PC = 0x00000018

IRQ

- Za povratak iz IRQ-potprograma treba izvesti naredbu:

  SUBS PC, R14, #4

- Ova naredba* će obnoviti PC (iz R14_irq) i CPSR (iz SPSR_irq) te nastaviti izvođenje programa na mjestu gdje je prekinut

* SUB napisan s ekstenzijom S i odredišnim registrom PC, znači da treba obnoviti CPSR

FIQ (brzi prekid)

- Brzi prekid namjenjen je primjenama gdje je bitno brzo reagirati i obaviti niz operacija

- FIQ zato ima dovoljan broj 'privatnih' registara tako da ne treba obavljati operacije pohranjivanja i vraćanja konteksta na stog

- Pored toga, adresa prekidnog potprograma FIQ je namjerno zadnja na listi tako da se potprogram može napisati odmah od te adrese bez potrebe za skokom na drugo mjesto u memoriji (izbjegnuto je kašnjenje zbog takvog skoka)

FIQ

- Ako je u programu omogućeno prihvaćanje brzog prekida (bit F u CPSR je obrisan) procesor će na kraju izvođenja svake naredbe provjeriti da li je ulaz FIQ aktivan. Ako je na ulazu u procesor aktiviran signal brzog prekida (FIQ), procesor će nakon završetka trenutne naredbe obaviti sljedeći niz operacija:

  - R14_fiq = address of next instruction to be executed + 4
  - SPSR_fiq = CPSR
  - CPSR[4:0] = 0b10001 /* Enter FIQ mode */
  - CPSR[5] = 0 /* Execute in ARM state */
  - CPSR[6] = 1 /* Disable fast interrupts */
  - CPSR[7] = 1 /* Disable normal interrupts */
  - PC = 0x0000001C

FIQ

- Za povratak iz FIQ-potprograma treba izvesti naredbu (isto kao za IRQ):

```
SUBS PC, R14,#4
```

- Ova naredba će obnoviti PC (iz R14_fiq) i CPSR (iz SPSR_fiq) te nastaviti izvođenje programa na mjestu gdje je prekinut

Naredbe za pristup registrima stanja

- Omogućuju prijenos podataka iz registara stanja (CPSR, SPSR) procesora ARM u neki registar opće namjene i obratno.
- U okviru AR1R mi ćemo koristiti samo prijenos iz/u CPSR
- Pisanjem u CPSR, na primjer, programer može postaviti stanja bitova za omogućavanje prekida kao i procesorski način.
- Naredba **MRS** (Move to Register from Status register) kopira sadržaj registra stanja CPSR u jedan od registara opće namjene koji se može dalje ispitivati ili mijenjati.
- Naredbom **MSR** Move to Status register from Register može se upisati neposredna vrijednost ili sadržaj registra opće namjene u registar stanja CPSR.

Primjer korištenja

- primjer omogućavanja prekida (brisanje bita I u registru CPSR)

```
MRS  R0, CPSR
BIC   R0, R0, #0x80
MSR  CPSR, R0
```