

ARHITEKTURA RAČUNALA 1

Prikaz podataka i operacije u računalu

čeličnojim podaci

bez predznaka:

NBC - Natural Binary Code \Rightarrow svaki bit predstavlja jednu znamenku

BCD - Binary Coded Decimal

potkodani - po dvoje dekadiske znamenke u bajtu

nepakodani - jedna dekadiska znamenka u bajtu (nisi 4 bita su 0000)

s predznakom

prikaz s bitom za predznak

prikaz jediničnim komplementom

prikaz dvojnimm komplementom

IEEE format - 32-bitni

$$x = (-1)^P \cdot 1.F \cdot 2^{E-127}$$

P(1 bit) - bit predznaka

E(8 bita) - karakteristika $E = E_{\text{eksponent}} + B$ ($B = 127$, predbroj - bias)

F(23 bita) - frakcija mantise

Greške kod aritmetičkih operacija i zastavice

zbrajanje NBC brojeva : greška - pogresnost

oduzimanje NBC brojeva : greška - pogresnost o ili posudba

zbrajanje 2^k brojeva : greška - $ov = C_m \text{ XOR } C_{m-1} = 1$

oduzimanje 2^k brojeva : greška - $ov = 1$ (pojava preljeva)

Cm	Cm-1	ov
0	0	0
0	1	1
1	0	1
1	1	0

Zastavice - usporedba brojeva

NBC brojevi

$x > y$: $c=1$ and $z=0$

$x \leq y$: $c=0$ or $z=1$

$x \geq y$: $c=1$

$x < y$: $c=0$

2^k brojevi

$x > y$: $(s \text{ XOR } ov) = 0$ and $z=0$

$x \leq y$: $(s \text{ XOR } ov) = 1$ or $z=1$

$x \geq y$: $(s \text{ XOR } ov) = 0$

$x < y$: $(s \text{ XOR } ov) = 1$

Prislušuje podataka

- prislušuje ušticama čuva iznos NBC brojeva,

ali ne čuva iznos 2^k brojeva, osim pozitivnih

- predzvukna prislušuje čuva iznos 2^k brojeva,

ali ne čuva iznos NBC brojeva, osim dvoje polovice opsega

Pomaci i rotacije

logički pomak u lijevo - množenje s 2 za NBC brojeve (SHL)

logički pomak u desno - dijeljenje s 2 za NBC brojeve (GHR)

| kao ulazni bit se dovede uštica)

aritmetički pomak u desno - dijeljenje s 2 za 2^k brojeve (ASHR)

| kao ulazni bit se dovede prethodne vrijednost najvišeg bita)

rotacija (ulijesno ili udesno) - kao ulazni bit se dovede vrijednost izlaznog bita

rotacija kroz zastavice - kao ulazni bit se uzme stara vrijednost

zastavice, a izlazni bit postaje novo stanje zastavice

(RDTL, RDTR)

Arhitektura procesora PPS

- ugradljivo računalo:

ulaz - primanje podataka iz vanjskog svijeta

izlaz - slanje podataka iz računala u vanjski svijet

memorija - spremanje programa i podataka za izvođenje procesora

PROCESOR

{ put podataka - dio procesora koji obavlja operacije nad podacima
 upravljačka jedinica - upravlja radom svih ostalih dijelova računala

- procesor (CPU) - osnovne funkcije mu je izvođenje program :

dohvat naredbe

dekodiranje naredbe

izvođenje naredbe

Razine apstrakcije

sistemski razine

arhitektura skupa naredoba - funkcionalnost koju sklopovije treba omogućiti

microarhitektura - sklopovski opis arhitekture procesora (način povezivanja)

razine logičkih vrata

razine rabiopreda

Arhitektura računala s obzirom na memoriski pristup

von Neumannova arhitektura - program i podaci smješteni u jedinstvenom mreži.

Harvardska arhitektura - razdvajanje memorije za postanak programa od memorije za podatke

Arhitekture s obzirom na subjekat operanda

stogovne arhitekture - Ufo (last in First out) ↓ povećavanje adresa

akumulatorске arhitekture - jedan operand je uvek u posebnom registru (Acc)

Arhitekture registar - memorija

Arhitekture registar - registar (load-store)

Arhitekture s obzirom na stup uvedaba

CISC - Complex Instruction Set Computer (karakteristike):

velik broj uvedaba i višihovih instrukcija

velik broj vrločine adresiranja

kompleksne uvedbe se izvode unutar procesora - uz cikluse

registri imaju posebne funkcije

skoro projektiraju i visoka cijene

- Intel 80x86, Motorola 68000

RISC - Reduced Instruction Set Computer (karakteristike):

relativno malen stup jednostavnih uvedbi, manji broj instrukcija crteže uvedbe

mali broj vrločine adresiranja

pojedine uvedbe se izvode izvan procesora

velik broj ravnopravnih registara opće namjene unutar procesora (R0-R7)

kompleksne uvedbe se izvode izvan procesora ("programski")

koristeće pomoćne strukture za ubrzavanje rada

relativno jeftino projektiraju i viska cijene

- MIPS, ARM, SPARC

- širine registara : 32b

- širine sabitnice podataka : 32b

- širine adresne sabitnice : 32b

- širine interne sabitnice : 32b

- širine PPU jedinice : 32b

(svaka adresa odgovara jednoj memorijskoj riječi, odnosno jednom bajtu)

Definiranje detalja izabrane arhitekture

PC (programsko trajilo) je register u kojem procesor

pamti adresu sljedeće uvedbe koju treba izvesti

Odabir skupov mafedoba - svaka mafedba zauzima 4 memorijске lokacije (32b)

Aritmetičko-logičke mafedbe

zbrajanje - ADD, ADC (add with carry)

oduzimanje - SUB, SBC (subtract with carry)

logički I na maforium - AND

logički IJU ne maforium - OR

logički EKSKLUZIVNI IJU ne maforium - XOR

mafeda operand1, operand2, operand3

(operacija se izvodi između prva dva operanda, i stavlja

(rezultat u treći operand)

+ CMP src1, src2 ; src1 - src2

Memorijске mafedbe

LOAD registar, [adresa)

STORE registar, [adresa)

} 32-bitni podaci

adresa ili adresni registar!

Upravljačke mafedbe

bezvjetni skok : JP adresa (skoci na mafedbu sa zadanim adresom)

vijetni skok : JP-vijet adresa

(registar SR - registar stanja sadrži popis zastanica Z, V, C, N)
(8 mta)

Strojni kod mafedoba procesora FRISC

operacijski kod ima 5 mbitova ($2^5 = 32$ mafedbe)

- Al-mafedbe imaju tri operanda, a tri su registri

- svaki operand kodiramo sa po 3 mbita ($2^3 = 8$ registara R0-R7)

- upravljačke mafedbe - vijet 4 mbita, operand (adresa skoka) je 20 mbita

Poboljšane inicije procesora

DR - podatkovni registar (podaci koji se čitaju i pišu)

- međusklop između unutrašnjosti procesora i podatkovne sabitvice

AR - adresni registar (adresa memorijске lokacije koju procesor želi čitati ili pisati)

- međusklop između unutrašnjosti procesora i adresne sabitvice

IR - naredbeni registar (dohvaćanje naredaba iz memorije)

- čuva strojni kod naredbe za vrijeme uvezivačnog dekodiranja

Uvodljenje registarskih naredoba

MOVE src, dest

src - izvor podatka (SR, R0-R7, ili lob podatak koji je prethodno proučio)

dest - odredište podatka / SR, R0-R7)

Praktisanje memorijskih naredoba - adrese

logički adresni prostor - raspoloživi opseg adresa koje procesor može

generisati u svojim adresnim priključcima

- strukture fizičke veličine memorije i adresnog prostora u računalu

moju je ili jednaka logičkom prostoru

- adresane 0000 0000 do 0007 FFFF adresiramo najviših 2^{19} lokacija
memorije (najviših pola MB)

- adresane FFF8 0000 do FFFF FFFF adresiramo najviših 2^{19} lokacija
memorije (najviših pola MB)

- što ukupno daje 2^{20} lokacija (1 MB)

- na ovaj način NE MOŽEMO adresirati svih 2^{32} lokacija (4 GB = 4096 MB)

Odmak

LOAD dest, | adr)

LOAD dest, | adr reg ± odmak)

STORE src, | adr)

STORE src, | adr reg ± odmak)

opcí registar koji zadaje adresu
(R0-R7)

20 bitni odmak od adrese zadane
sa addreg

Sirine podataka

LOAD, STORE

- rječi (imaju adrese djeliive s 4)

LOADH, STOREH

- poluniječi (imaju adrese djeliive s 2)

LOADB, STOREB

- bajtovi (nemaju ograničenja na adresu)

Endianess

little-endian : mali bajt u manjoj adresi

big-endian : mali bajt u većoj adresi

Stog - vrjek stavlja i uzima 32b podatak

PUSH - stavi ne stog podatak iz jednog od općih registara

POP - uzmi podatak sa stoga, i stavi ga u jedan od općih registara

PUSH src :

POP dest :

R7 - 4 → R7

(R7) → dest

src → (R7)

R7 + 4 → R7

(R7 ili SP)

Relativni skok JR

JR adresu

JR - uvjet adresa

- određiste skoka se ne zadaje absolutnu adresom, već odmakom

od same naredbe JR (odmak je 2ob, i predvukano se proširuje

čime se dobivaju skokovi "unaprijed" i "unatrag")

Potpogrami - inicijalizacija SP : MOVE 1000, SP

CALL - poziv potprograma

REI - povratak iz potprograma (utvrda povratnog adresu sa stoga i skoci u liniju)

(RET, RETN) (PC - adresa sljedeće naredbe koja se treba izvesti, naredba neposredno iza

naredbe CALL, pa znamo gdje se moramo vratići nakon potprograma :

(spremna se povratna adresa na STOG)

HALT - zaustavlja rada procesora (prekida normalan niz red izvođenja)

Nacini adresiranja

Registarsko adresiranje - podatak ili rezultat ulazi se u jednu od registara

Neposredno adresiranje - podatak se zadaje neposredno u naredbi kao broj

Absolutno adresiranje - zadaje se broj u naredbi, koji predstavlja adresu

Relativno adresiranje - koristi se samo u naredbi relativnog skoke JR

Registarsko indirektno adresiranje - zadaje se registrov u zagradama (adresa)

Registarsko indirektno s odmikom - zadaje se register i broj (odmek)

Memorijske naredbe : LOAD, STORE) - STORE RS, (R6) ≡ STORE RS, (R6+0)

Implicitno adresiranje - PUSHT, POP

Asembleriski prevoditelji (asembleri) - pseudonaredbe

'ORG - početna adresa (ako se ispusti, onda se pretpostavlja početna adresa 0)

'END - prekid prevodenja (sve što je iza 'END, zamenjuje se)

'DW - rezervira memoriju lokaciju (ili više linija), te ih puni podacima

'EQU - ne zauzima memoriju, nego samo daje vrijednost labele

'DS - zauzima list memorijskih lokacija, a inicijalne vrijednosti su im uvek 0

'BASE - definira baze branevenog sustava koji želimo konstituti (B, D, H)

Poznavanje podataka

poznavanje pseudonaredbe DW, DH, DB - "poznavanje" na adresu dijijvu s 4

'DW i 'DS - rezerviraju memoriju u bajtovima

Potpogramni - parametri i rezultati

- načini stavlja parametara i vraćanja povratne vrijednosti

Pomoću registara

Prednosti

bit prijenos - rad s registrima je ujedno

jednostavan prijenos - se stajališta programisanju

Nedostaci

mogče se prenijeti ograničen broj argumentata (najviše 7)

registri obično čuvaju metarezultate i nisu slobodni za korištenje kao parametri

nije mogće rekursivno pozivanje potprograma

Pomoću fiksnih memorijskih lokacija

Prednosti

nema ograničenja na broj parametara (već u veličini memorije)

registri su slobodni za druge namjene

Nedostaci

sporiji rad s memorijskim lokacijama, nego s registrima (upis i čitanje)

veliko duži i komplikovaniji program, nego kad se koriste registri

nije mogće rekursivno pozivanje potprograma

Pomoću stoga

- podaci na stogu su uniformno organizirani za svaki potrogram : OBVR STOGA

parametri

povratna adresa

spremljeni registri

lokalne varijable

- dok pišemo potrogram, točno znamo koliko su parametara i koliko registara imam

spremiti na stog, zato lako možemo isračunati odnose pojedinih parametara
u odnosu na SP (SP+0, SP+4, SP+8, SP+C, SP+10, SP+14)

Nacim rada s okvirom

Glavni program

stavlja parametre

stavlja povratnu adresu - CALL

ukljuca parametre

Potprogram

spremna i obavljaja registre

stvara i ukljuca lokalne varijable

ukljuca povratnu adresu - RET

Macrodefdbe

- 'MACRO - 'ENDMACRO

- imaju istu namjenu kao i potprogrami

- potprogrami su podrtkani od strane procesora, pokrenut uaredba CALL i RET,

kojim se obavljaju poziv i povratak iz potprograma

- makrodefdbe su podrtkane od strane assembleckog prevoditelja, koji ih prevede u oznake uaredbe procesora

- makrodefdbe se ne pozivaju, vezo se jednostavno izvode na mjestu u kojem je to potrebno (potprogrami su sponji)

spojci (izvodi, pinovi) - pomocu priključaka se spajaju komponente na sabirnicu

smjer priključaka

ulazni

izlazni

dvosmjerни

Podjela po namjeni

adresna sabirnica

podatkovna sabirnica

upravljačka sabirnica

Prikључci procesora FRISC

Vcc ; GND - za napajanje

CLOCK - takst koji daje za dovođenje pravokutnog signala stabilne frekvencije za rad procesora

ADR (32b) - adrebitanje memorije i vanjskih jedinica

DATA (32b) - pomoću njih se podatak prenosi u procesor ili iz procesora

READ, WRITE - ujedno procesor razvija memoriju ili vanjskoj jedinici da želi pisati ili čitati

SIZE (2b) - širina podatka kojem procesor želi pristupiti

WAIT - za produljenje normalnog ciklusa čitanja ili pisanja podataka

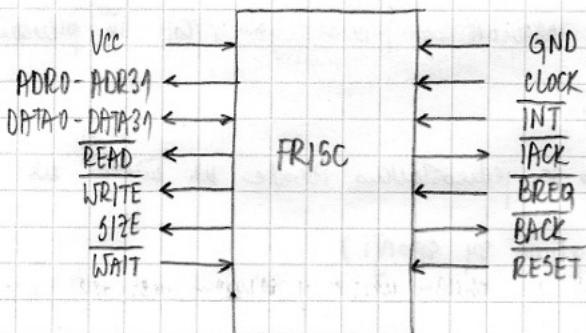
INT (INT0 do INT3) - postavljanje zahtjeva za prekid

IACK - potvrda za prihvatanje zahtjeva za prekid najniže razine

BREQ - DMA-jedinica zahtjevi od procesora da joj prepusti upravljanje nad sabitnicom

BACK - procesor potvrđuje DMA-jedinici da je privratio njen zahtjev za upravljanjem sab

RESET - mikročak polovicu koju se procesor može dovesti u početnu stanju



Sabitnice - podjela

Memorijska sabitnica - povezuje procesor i memoriju, male duljine, brze

Uzimaju - izlazna sabitnica - u-1 jedinice povezuje s procesorom

sabitnice specijalne manjene

Praća načina komunikacije

Sinkrone sabitnice - češće se koriste za memorijske sabitnice

Asinkrone sabitnice - češće se koriste za u-1 sabitnice

Izvođenje naredaba i protične strukture - ciklovi / PRSC

Razine za dohvat (fetch)

dohvat naredbe

dekodificiranje naredbe

dohvat operanada

Razine za izvođenje (execute)

izvođenje ALU operacije

spremanje rezultata

dvije razine traju po jedan period signala vremeninskog vodjenja clock-a

Hazardi

strukturni - procesor u određenom trenutku ne može izvesti sve faze naredba

u protičnoj strukturi (mehurić-bubble omogućuje odgodu izvođenja naredaba)

upravljački - naredba koja se nalazi u protičnoj strukturi, i spremanje je za

izvođenje, nije naredba koja se ustvari ne može izvesti

podatkovni hazard - naredba se ne može izvesti, jer podaci potrebni za njenu

izvođenje još nisu spremani (ARM!!)

Komunikacija procesora s U-I jedinicama

memorijsko UI preslikava - procesor ne jednako načinom pristupa memoriji i U jedinicama

izdvajeno UI adfresira - razlikiti načinom pristupa memoriji i U jedinicama

Viste UI prijenosa podataka

Programski prijenos

bezvjetni prijenos - prije prijenosa se ne projekcira spremnost U jedinice

vijetni prijenos - prije prijenosa se projekcira je li U spremna za prijenos

prekidni prijenos - U jedinica samostalno dojavljuje svoju spremnost

Složovski prijenos

jeftinski pristup memoriji - DMA-jedinica

Prekidni sustav procesora 8086

Prekidni priključci i prioriteti

- INT0, INT1, INT2 - maskirajući prekidi
- INT3 - nemaskirajući prekid
- NACK - potvrda prihvata nemaskirajućeg prekida

- maskirajući prekidi se mogu zatruditi ili omogućiti (maskirati)
 - (dozvoljavaju ili zahtijevaju prihvatanje prekida od strane procesora)
- oni su inicijalno zatrudjeni
- nemaskirajući prekid se ne može zatruditi

Prekidne zastanice u registru stanja SR

EINTi - zabranjuje (0) ili dozvoljava (1) prihvatanje prekida s prekidnog priključka INTi

GIE - zabranjuje (0) sve maskirajuće prekide ili dozvoljava (1) njihovo prihvatanje ne temeljnih zastanica INTi

IIF - nije u registru stanja SR!

(dok se obrađuje INT3, onda je IIF u stanju 0, a inače je u stanju 1)

- naredbe za povratak iz potporograma:

RETI (Return from maskable interrupt) obnavlja stanje GIE = 1

RETN (Return from Nonmaskable interrupt) obnavlja stanje IIF = 1

Spremanje konteksta : R0 i SR

PUST R0

MOVE SR,R0

PUST R0

Sklop PRISC - CT

- brojanje impulsa i mijenjanje vremena
- CT ne prenosi podatke (Counter Timer)
- oslobađa procesor od nepotrebnih čekanja ili čestih posluživanja prekida

CR (Control Register) - upravljački registar

LR (Limit Register) - 16b registar vremenske konstante (grанице)

DC (Down Counter) - 16b brojilo ne dolje

Sklop PRISC - PIO (parallel Input-Output)

- paralelni prijenos podataka (8 bitnih)
- programabilni sklop koji može raditi u 4 načina rada:

Wazni

ispitivanje bitova

izlazni

postavljanje bitova

DR (Data Register) - 8b podatkovni registar

ICR (Input Control Register) - wazni upravljački registar

OCR (Output Control Register) - izlazni upravljački registar

MR (Mask Register) - registar maseke koji služi za odabir bitova za ispitivanje

Stupovski UI prijenos - DMA (Direct Memory Access)

- posebna jedinica koja obavlja istavanj ponistup memoriji
- velika količina prijenosa (jer podaci ne prolaze kroz procesor)

Viste DMA prijenosa - prijenos 32b podataka

Zauzimanje procesora - DMA preuzeće upravljanje i prenese sve podatke

krata ciklusa - DMA preuzeće upravljanje, prenese jedan podatak, vratи proc. upravljanje

blokovski - DMA preuzeće upravljanje, prenese nekoliko (blok) podataka

multiplexirani - kad procesor ne koristi sabirnicu

Arhitektura računala zasnovanog na procesoru ARM

- Acorn RISC Machine postaje Advanced RISC Machine

Programski model ARM - 32b procesor

tipovi podataka - djet (1B), polu-rječ (4), rječ (16)

procesorski naciini rada - usr, sys, svc, abt, und, irq, fiq

registri - 37 registara

registri opće namjene - 31 registar (uključujući i PC) širine 32b

registri programskih stanja - također 32 bitni registri (CPSR, SPSR) - 6 registra

R13 - pokazivač na vrh stoga (SP, Stack Pointer)

R14 - registar za pohranjivanje povratne adrese (po potrebi)

R15 - programsko brojilo (PC)

CPSR - Current Program Status Register (registar trenutnog prog. stanja)

SPSR - Saved Program Status Register (reg. pohranjenog programskog stanja)

izvijuke

memorija - jedinstveni adresni prostor od 2^{32} 8b podataka

rječi (32b) - pohranjivate tako da počinju na adresi djeljivoj s 4

poluriječi (16b) - na parnoj adresi

bajtovi (8b) - tako gdje u memorijском prostoru

Skup naredoba procesora ARM

naredbe load i store (+ naredbe load i store multiple, te swap)

naredbe za obradu podataka - AL naredbe, naredbe za množenje, naredba div

naredbe grananja - B (Branch), BL (Branch and Link)

naredbe za mijenjanje registara stanja

koprocesorske naredbe - komunikacija procesora ARM i koprocesora

naredbe za generisanje izvijuke

Naredbe load i store

- izvršice adresiranja memorije:

DuoVii odmak Reg \leftrightarrow mem [BazniReg + Odmak]

(izračuna se adresa, te se obavi prijenos podatka, a vrijednost bazuog registra ostaje nepratljivljena)

Pred-indeksiranje BazniReg = BazniReg + Odmak;

Reg \leftrightarrow mem [BazniReg]

(odmak se zbroji ili oduzme od bazuog registra, rezultirajuća vrijednost se spremi u bazu register, i tek tada se pristupi memoriji)

Post-indeksiranje Reg \leftrightarrow mem [BazniReg];

BazniReg = BazniReg + Odmak

(memoriji se pristupi samo ne temelju vrijednosti bazuog registra, a tek nakon izvršenja prijenosa, odmak se zbroji ili oduzme od bazuog registra, te se izračunata vrijednost spremi u bazu register)

LDMD STM : Macjini adresiranja

Increment Before (IB)

Increment After (IA)

Decrement Before (DB)

Decrement After (DA)

Rad sa stogom

zapisivanje STMFD

čitanje LDMFD

Aritmetičko - logičke naredbe

- koriste dva operanda, i spremaju rezultat u zadani register

- ako se naredbi doda nastavak "S" (Save condition codes), tada će se ukloniti izvršene naredbe osim u zastanci stanja (ARM u zastanci c vijk ima prijenos)

računalni sustav s procesorom ARM

- AMBA (Advanced Microcontroller Bus Architecture)

definira tri tipa sabitnica koje se mogu ugraditi u unutar sustava:

AHB (Advanced High-performance Bus) - brza središnja sabitница

ASB (Advanced System Bus) - također središnja sabitница

APB (Advanced Peripheral Bus) - povezivanje vanjskih uređaja u sustav

(mост AHB-APB je sklop koji povezuje ove dve sabitnice i
uređaje ne ujme u cijeli sustav, te omogućuje prijenos
podataka između uređaja na sabitnicama AHB i APB)

Vanjski sklopovi

GPIO (General Purpose Input Output) - registri:

GPIOADR - 8b registar podataka za vrat A

GPIOBDR - 8b registar podataka za vrat B

GPIOADDR (GPIO Port A Data Direction Register) - 8b

GPIOBDDR - 8b registar smjera podataka za vrat B

Početne vrijednosti registara

- svi registri unutar GPIO mukou inicijalizacije (resetiranja) postavljaju se u logičku muku

- inicijalno vrata A postavljaju kao ulazne, a vrata B kao izlazne

- sadržaji oba registra podataka su muka

(LCD prikaznik - ispisivalje do osam ASCII-znakova u zastoru sklopa, 8b)

RTC (Real Time Clock) - registri

RTCDR (Data Register) - 32b registar podataka

RTCMR (Match Register) - 32b registar usporedbe s brojkom

RTCSTAT | RTCE0 | Interrupt STATUS Register / Interrupt Clear Register) - potraže podatka

RTCLR (load Register) - 32b registar za upis ili čitaće vrijednosti brojila

RTCCR (Control Register) - 1b upravljački reg. za (ne) omogućavanje generisanja prekida