

Pentium: Keskeytykset

Sta06 Table 12.3

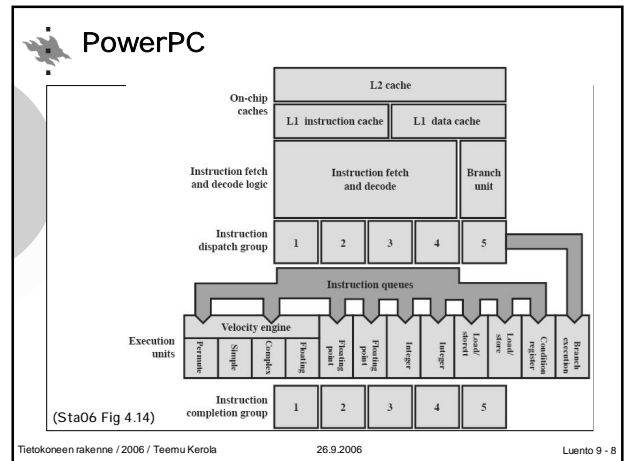
Keskeytykskäsitteellään siirtyminen (atominen laiteolinto)

- Jos ei jo etuoikeutetussa tilassa
 - PUSH(SS) pinosegmentin valitsin pinoon
 - PUSH(ESP) pino-osoitin
 - PUSH(EFLAGS) tilarekisteri vrt. aliohjelmakutsu
- EFLAGS.IOPL • 00 etuoikeutettu tila
- EFLAGS.IF • 0 keskeytyksen esto
- EFLAGS.TP • 0 poikkeusten esto
- PUSH(CS) koodisegmentin valitsin
- PUSH(EIP) käskyosoitin
- PUSH(error code) vain tarvittaessa
- numero • keskeytysohjaimelta / INT-käskystä / tutki tilarekisteri
- CS • keskeytyksen vektori[numero].CS Osoitemuunnos nyt uudella tavalla
- EIP • keskeytyksen vektori[numero].EIP

Paluu

- Etuoikeutettu IRET-käsky (interrupt return)
- Palata pinoon kaikkien entiselleen

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 7



PowerPC: Rekisterit (user visible)

- 32 kokonaislukurekisteriä, a' 64 b, ja Exception Register (XER), 32 b** Sta06 Fig 12.23
- 32 liukulukurekisteriä, a' 64 b, ja FP Status & Control Register (FPSCR), 32 b** Sta06 Tbl 12.4
- 3 rekisteriä hyppyjen käsittelemiseksi**
 - Condition Register, 32b, 8 kenttää, a' 4 b Sta06 Fig 12.24, Tbl 12.5
 - CR0 kokonaisluvuille, CR1 liukuluvuille (> 0, < 0, = 0, Overflow)
 - Jokainen suoritettu käsky asettaa
 - CR0-CR7 vertailukäsky asettaa (op1 > op2, op1 < op2, op1 = op2)
 - Voi pitää vertailujen tuloksia tallessa
 - Link Register, 64 b
 - Esim. aliohjelman paluusoite tälle
 - Count Register, 64 b
 - Esim. iteraatiolaskuri, epäsuora osoitus hyppysä

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 9

PowerPC: Rekisterit (control & status)

- Machine State Register, MSR, 64 b** Sta06 Tbl 12.7
 - 48: ulkoiset keskeytykset estetty/ei
 - 49: etuoikeutettu/käyttäjätila
 - 53: jokaisen käskyn jälkeen KJ:hin/ei
 - 54: hyppykäskyn jälkeen KJ:hin/ei Tracing
 - 52&55: milloin liukuluvun käsittelystä poikkeus
 - 59: MMU:n osoitemuunnos ON/OFF
 - 63: big/little endian
- Save/Restore Registers: SRR0 ja SRR1**
 - Käytetään keskeytykskäsitteilyn yhteydessä
 - Talletuspaikka PC:lle ja MSR:lle

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 10

PowerPC: Keskeytykset

- CPU:n sisäiset vs. ulkoiset** Sta06 Tbl 12.6
- Keskeytykskäsitteellään siirtyminen (laitteolinto)**
 - SRR0 • PC
 - SRR1 • MSR + tyyppiin sidottua tietoa
 - MSR • keskeytyksystyyppiin mukaan määräytyvää tietoa
 - Etuoikeutettu ON, keskeytykset OFF, osoitemuunnos OFF
 - Muut bitit määrättyvät keskeytyksen perusteella
 - PC • Keskeytykskäsitteellään osoite
 - Nouda keskeytyksen vektori keskeytyksen "numerolla"
 - Bitti 57: mahdollisuus määrittellä kaksi vektoria
- Paluu**
 - Etuoikeutettu rfi-käsky (return from interrupt)
 - MSR • "reset" ja kopioi SRR1:stä
 - PC • SRR0

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 11

Tietokoneen rakenne

Luento 9

RISC-arkkitehtuuri

Ch 13 [Sta06]

- Käskyanalyysia
- RISC vs. CISC
- Rekisterien käytöstä

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 12

Laitteistotason virstanpylväitä

- Virtuaalimuisti, 1962** (Tom Kilburn)
 - u Helpompi muistinhallinta
- Liukuhlina, 1962** (Tom Kilburn)
 - u Samaa käskykantaan noudattavia koneita
- Arkkitehtuuriperhe, 1964** (Gene Amdahl)
 - u Käsyykantaan helpompi kehittää
- Mikro-ohjelmointi kontrolli, 1964** (Maurice Wilkes)
 - u Käsyykantaan helpompi kehittää
- Moniprosessorijärjestelmä, 1964** (J.P. Eckert, John Mauchly)
 - u test_and_set käsky tarpeen
- Välimuisti, 1965** (Maurice Wilkes)
 - u Huikea suorituskykyparannus
- RISC-arkkitehtuuri, 1980** (John Cocke, 1974; J.L. Hennessy & D.A. Patterson)
 - u Yksinkertainen käskykanta
- Superscalar CPU, 1989** (John Cocke, 1965; IBM, Intel)
 - u Useita käskyjä valmiiksi per sykli
- Hyperthreading CPU, 2001** (CDC, 1964; Intel)
 - u Usea rekisterijoukko ja virtuaalisuoritin lastulla
- Multicore CPU, 2005** (Intel, IBM)
 - u Usea täydellinen suoritin lastulla

"Perinteiset" koneet

- Kantavia ideoita**
 - u Kutista lausekielen ja konekielen semanttista kuilua
 - § Lausekielen ilmaisuvoima kehittynt
 - u Kääntämisen helppous
 - § Lausekielen rakenteilla vastineet konekielessä
 - § Käskykanta valikoitu usein tietty lausekieli mielessä
 - u Paljon erilaisia käskyjä moniin tarkoituksiin
 - u Paljon erilaisia datatyyppäjä
 - u Paljon erilaisia osoitusmuotoja
 - u Tee asiat laiteomintoina, ei ohjelmallisesti
 - § Vähemmän konekielisiä käskyjä suoritettavana
 - § Monimutkaisten operaatioiden suorittaminen tehokasta

= CISC (Complex Instruction Set Computer)

Mitä operaatioita / operandeja?

Vuosi 1982, kohde: VAX, PDP-11, Motorola 68000
Dynaaminen, suoritusaikainen tarkastelu

	Dynamic Occurrence		Machine-Instruction Weighted		Memory-Reference Weighted	
	Pascal	C	Pascal	C	Pascal	C
ASSIGN	45%	38%	13%	13%	14%	15%
LOOP	5%	3%	42%	32%	33%	26%
CALL	15%	12%	31%	33%	44%	45%
IF	29%	43%	11%	21%	7%	13%
GOTO	—	3%	—	—	—	—
OTHER	6%	1%	3%	1%	2%	1%

Weighted Relative Dynamic Frequency of HLL Operations [PAT82a]

	Pascal	C	Average
Integer Constant	16%	23%	20%
Scalar Variable	58%	53%	55%
Array/Structure	26%	24%	25%

80% viittauksista aliohjelmien paikallisiin

(Sta06 Table 13.2, 13.3)

Aliohjelmakutsut?

- Aliohjelmakutsuja paljon
- Kutsussa harvoin monta parametria
- Kutsut harvoin sisäkkäisiä

Percentage of Executed Procedure Calls With	Compiler, Interpreter, and Typesetter	Small Nonnumeric Programs
>3 arguments	0-7%	0-5%
>5 arguments	0-3%	0%
>8 words of arguments and local scalars	1-20%	0-6%
>12 words of arguments and local scalars	1-6%	0-3%

Procedure Arguments and Local Scalar Variables

- Em. tietojen hyödyntäminen?
 - 98% alle 6 parametria
 - 92% alle 6 paikallista muuttujaa

Huomioita

- Pääosa operandeista yksinkertaisia
- Hyppykäskyjä runsaasti
- Kääntäjatkään eivät aina hyödynnä tarjolla olevia monipuolisia konekäskyjä
 - u Käyttävät vain osaa käskykannan käskyistä
- Johtopäätökset?
 - Occamin partaveitsi (Occam's razor)
 - "Entia non sunt multiplicanda praeter necessitatem"
 - ("Entities should not be multiplied more than necessary")
 - William Of Occam (1300-1349)
 - English monk, philosopher
 - "It is vain to do with more that which can be done with less"

Optimointia

- Optimoi asioita, jolssa kuluu eniten aikaa**
 - u Aliohjelmakutsut, silmukat, muistinviihtaukset, ...
- Esimerkki huonosta optimoinnista**
 - u Tuplaa liukulukuaritmetiikan nopeus
 - u 10% käskyissä liukulukuaritmetiikkaa

$$ExTime_{new} = ExTime_{old} * (0.9 * 1.0 + 0.1 * 0.5)$$

$$= 0.95 * ExTime_{old}$$

$$Speedup = ExTime_{old} / ExTime_{new} = 1 / 0.95 = 1.053 \ll 2$$

Amdahlin laki

Speedup due to an enhancement is proportional to the fraction of the time (in the original system) that the enhancement can be used.

Optimointia

- Optimoi suoritusnopeutta, älä kääntämisen helppoutta / suoravilvaisuutta
 - Kääntäjät erinomaisia, koneet tehokkaita
 - Osaavat ja ehtivät optimoida
 - Tee yleisimmät tehtävät laite toimintoina, tehokkaasti
 - Esim. 1-uloitteiseen taulukkoon viittaus
 - Tee loput ohjelmallisesti
 - Esim. merkijonon muunnos
 - Tarjota valmiit kirjastorutiinit

RISC arkkitehtuuri (Reduced Instruction Set Computer)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 19


RISC arkkitehtuuri

- Paljon rekistereitä (väh. 32)
 - Kääntäjät optimoimaan rekistereiden käyttöä
- LOAD / STORE arkkitehtuuri
 - Vain LOAD ja STORE viittaavat muistiin
- Vain vähän ja yksinkertaisia käskyjä
- Yksinkertainen vakio pituinen käskyformaatti (32b)
 - Käskyjen nouto ja dekodaus helppoa
- Vain vähän ja yksinkertaisia osoitusmuotoja
 - Ei epäsuoraa osoitusta
 - Nopea operandin osoitelaskenta
- Vähän erilaisia operandeja
 - 32 b:n kokonaisluvut, liukuluvut
- Kullakin syklillä valmistuu yksi tai useampi käsky

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 20

RISC arkkitehtuuri

- CPU helpompi implementoida
 - Liukuhinaa helpompi hallita ja optimoida
 - Langoitettu toteutus (hardwired)
- Pienempi piirin koko
 - Enemmän per lastu
 - Pienempi hukka%
- Halvemmat kustannukset
- Nopeammin markkinoille



25% yield (OK) 55% yield (OK)
75% hukkaan 45% hukkaan

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 21

RISC vs. CISC

Characteristic	Complex Instruction Set (CISC)/Computer			Reduced Instruction Set (RISC) Computer		Superscalar		
	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000	PowerPC	Ultra SPARC	MIPS R10000
Year developed	1973	1978	1989	1987	1991	1993	1996	1996
Number of instructions	208	303	235	69	94	225		
Instruction size (bytes)	2-6	2-57	1-11	4	4	4	4	4
Addressing modes	4	22	11	1	1	2	1	1
Number of general-purpose registers	16	16	8	40-520	32	32	40-520	32
Control memory size (Kbits)	420	480	246	—	—	—	—	—
Cache size (KBytes)	64	64	8	32	128	16-32	32	64

Characteristics of Some CISCs, RISCs, and Superscalar Processors

(Sta06 Table 13.1)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 22

RISC vs. CISC

Processor	Number of instruction sizes	Max instruction size in bytes	Number of addressing modes	Indirect addressing	Load/store combined with arithmetic	Max number of memory operands	Unaligned addressing allowed	Max Number of MMU uses	Number of bits for integer register specifier	Number of bits for FP register specifier
AMD29000	1	4	1	no	no	1	no	1	8	3 ^a
MIPS R2000	1	4	1	no	no	1	no	1	5	4
SPARC	1	4	2	no	no	1	no	1	5	4
MC88000	1	4	3	no	no	1	no	1	5	4
HP PA	1	4	10 ^a	no	no	1	no	1	5	4
IBM RT PC	2 ^b	4	1	no	no	1	no	1	4 ^a	3 ^a
IBM RS-6000	1	4	4	no	no	1	yes	1	5	5
Intel i860	1	4	4	no	no	1	no	1	5	4
IBM 3090	4	8	2 ^b	no ^b	yes	2	yes	4	4	2
Intel 80486	12	12	15	no ^b	yes	2	yes	4	3	3
NSC 32016	21	21	23	yes	yes	2	yes	4	3	3
MC88040	11	22	44	yes	yes	2	yes	8	4	3
VAX	56	56	22	yes	yes	6	yes	24	4	0
Clipper	4 ^a	8 ^a	9 ^a	no	no	1	0	2	4 ^a	3 ^a
Intel 80560	2 ^a	8 ^a	9 ^a	no	no	1	yes ^a	—	5	3 ^a

a RISC that does not conform to this characteristic.
b CISC that does not conform to this characteristic.

(Sta06 Table 13.7)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 23

Tietokoneen rakenne

Rekistereiden käytöstä

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 24

Rekisterijoukko

- Enemmän rekistereitä kuin käskyssä voi viitata
 - Esim. SPARCissa rekisterinro 5 b ž suurin nro 32, mutta CPUssa 40-540 rekisteriä
- Kerralla käytössä vain osa rekistereistä, ikkuna
 - Ikkunaan viitataan esim. rekisterinumeroilla r0-r31
 - CPU kuvaa ne tod. rekisterinumeroiksi, esim. r0-r539

(Sta06 Fig 13.3)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 25

Rekisteri-ikkuna

- Aliohjelmakutsu käyttää pino sijasta rekistereitä
 - Kutsussa kiinteä määrä rekistereitä parametreille sekä paikallisille muuttujille
 - Varaukset limittään siten, että parametrit käytettävissä sekä kutsuvassa että kutsutussa osassa

(Sta06 Fig 13.1)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 26

Rekisteri-ikkuna

- Jos sisäkkäisiä kutsuja paljon
 - Rekisterijoukko voi loppua kesken
 - Talleta vanhin muistiin, lataa takaisin, kun paluu hämmöttää
 - Kutsuketju harvoin pitkä, talletus/palautus harvoin
- Globaalit muuttujat?
 - Muistissa tai oma rekisteri-ikkuna
- SPARC
 - r0-r7 globaaleille todelliset rekisterit
 - r8-r15 parametrit (kutsutussa)
 - r16-r23 lokaalit muuttujat
 - r24-r31 parametrit (kutsujassa)
 - virtuaaliset rekisterit

(Sta06 Fig 13.2)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 27

Rekisterijoukko vs. Välimuisti

Large Register File	Cache
All local scalars	Recently-used local scalars
Individual variables	Blocks of memory
Compiler-assigned global variables	Recently-used global variables
Save/Restore based on procedure nesting depth	Save/Restore based on cache replacement algorithm
Register addressing	Memory addressing

(Sta06 Table 13.5)

- Kääntäjän vaikea päätellä etukäteen mitkä globaalit muuttujat pitäisi sijoittaa rekistereihin
- Välimuisti ratkaisee sen dynaamisesti
 - Eniten viitatus pysyvät välimuistissa

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 28

Kääntäjä: Rekistereiden allokointi

- Verkon väritys
 - Etsi pieniin värimäärä s.e. verkon kahdella vierekkäisellä solmulla ei ole sama väri!
- Ärsyttävän vaikea ongelma (NP-täydellinen)
 - Laskennan teoria -kurssin asioita
- Analysoi koodia, ja muodosta verkko symbolisten rekistereiden käytöstä
 - Symb. rekisteri - mikä tahansa, mikä voisi olla rekisterissä
- Allokoi sitten oikeat rekisterit
 - Jos kahta symbolista rekisteriä ei käytetä samanaikaisesti, niille voi alokoida saman todellisen rekisterin
 - Jos rekistereitä ei vapaana, käytä muistia

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 29

Kääntäjä: Rekistereiden allokointi

- solmu = symbolinen rekisteri
- särmä = symbolisten rekistereiden yhtäaikainen käyttö
- n väriä = n rekisteriä

(a) Time sequence of active use of registers (b) Register interference graph

(Sta06 Fig 13.4)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 30

RISC-liukuhihna

Load rA ← M
Load rB ← M
Add rC ← rA + rB
Store M ← rC
Branch X

I	E	D							
	I	E	D						
		I	E	D					
			I	E	D				
				I	E	D			
					I	E	D		
						I	E	D	
							I	E	D

(a) Sequential execution

Load rA ← M
Load rB ← M
Add rC ← rA + rB
Store M ← rC
Branch X
NOOP

I	E	D							
	I	E	D						
		I	E	D					
			I	E	D				
				I	E	D			
					I	E	D		
						I	E	D	
							I	E	D
								I	E

(b) Two-stage pipelined timing
Single port MEM

Load rA ← M
Load rB ← M
NOOP
Add rC ← rA + rB
Store M ← rC
Branch X
NOOP

I	E	D							
	I	E	D						
		I	E	D					
			I	E	D				
				I	E	D			
					I	E	D		
						I	E	D	
							I	E	D
								I	E

(c) Three-stage pipelined timing
Two port MEM

Load rA ← M
Load rB ← M
NOOP
Add rC ← rA + rB
Store M ← rC
Branch X
NOOP
NOOP

I	E ₁	E ₂	D						
	I	E ₁	E ₂	D					
		I	E ₁	E ₂	D				
			I	E ₁	E ₂	D			
				I	E ₁	E ₂	D		
					I	E ₁	E ₂	D	
						I	E ₁	E ₂	D
							I	E ₁	E ₂
								I	E ₁

(d) Four-stage pipelined timing

(Sta06 Fig 13.6)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 31

RISC-liukuhihna, Delayed Branch

	1	2	3	4	5	6	7	8
100 LOAD X, rA	I	E	D					
101 ADD L, rA		I		E				
102 JUMP 105				I	E			
103 ADD rA, rB					I			
105 STORE rA, Z						I	E	D

Traditional

	1	2	3	4	5	6	7	8
100 LOAD X, rA	I	E	D					
101 ADD L, rA		I	E					
102 JUMP 106			I	E				
103 NOOP				I	E			
106 STORE rA, Z					I	E	D	

RISC with inserted NOOP
Two port MEM

	1	2	3	4	5	6	7	8
100 LOAD X, rA	I	E	D					
101 JUMP 105		I	E					
102 ADD L, rA			I	E				
105 STORE rA, Z				I	E	D		

RISC with reversed instructions
(Sta06 Fig 13.7)

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 32

RISC & CISC United?

- Pentium, CISC**
 - Laite toiminto kääntää 1 - 11 tavun pituisen CISC-käskyn yhdeksi tai useammaksi 118 bittiseksi mikro-operaatioksi (L1 tason käskyvälimuistiin)
 - Alemmat tasot kuten RISC Käännös joka käskyn suorituskerralla
 - Paljon työrekestereitä: laitteisto ottaa käyttöön
- Crusoe (Transmeta)**
 - Ulospäin CISC-arkkitehtuuri Just in time (JIT) compilation
 - Käskyjoukot käännetään ohjelmallisesti juuri ennen suoritusta kiinteän pituisiksi mikro-operaatioksi, operaatioiden optimointia per käskyjoukko
 - VLIW (very long instruction word, 128 bits)
 - 4 uops/VLIW-käsky Käännös vain kerran per käskyjoukko
 - Alemmat tasot kuten RISC

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 33

Kertauskysymyksiä

- Mitkä ovat RISC arkkitehtuurin tunnuspiirteet?
- Miten rekisteri-ikkunoita käytetään?

Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 34