


Luento 9



Tietokoneen rakenne

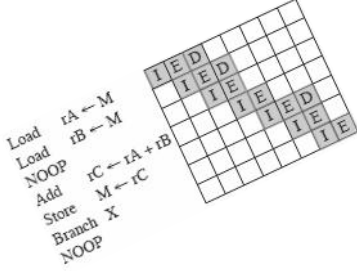
Suoritinesimerkit RISC- arkkitehtuuri

Ch 12.5-6 [Sta06]


- n Pentium/PowerPC

Ch 13 [Sta06]

- n Käskyanalyysia
- n RISC vs. CISC
- n Rekisterien käytöstä



Tietokoneen rakenne / 2006 / Teemu Kerola
26.9.2006
Luento 9 - 1



Tietokoneen rakenne

Pentium

Tietokoneen rakenne / 2006 / Teemu Kerola
26.9.2006
Luento 9 - 2

Pentium: Rekisterit

(a) Integer Unit

Type	Number	Length (bits)	Purpose
General	8	32	General-purpose user registers
Segment	6	16	Contain segment selectors
Flags	1	32	Status and control bits
Instruction Pointer	1	32	Instruction pointer

(b) Floating-Point Unit

Type	Number	Length (bits)	Purpose
Numeric	8	80	Hold floating-point numbers
Control	1	16	Control bits
Status	1	16	Status bits
Tag Word	1	16	Specifies contents of numeric registers
Instruction Pointer	1	48	Points to instruction interrupted by exception
Data Pointer	1	48	Points to operand interrupted by exception

EAX, EBX, EBX, EDX,
ESP, EBP, ESI, EDI

CS, SS, DS, ES, FS, GS

EFLAGS

EIP

käyttö pinona tai
MMX-käskyissä

selector, offset

(Sta06 Table 12.2)

Tietokoneen rakenne / 2006 / Teemu Kerola
26.9.2006
Luento 9 - 3

Pentium: FP / MMX Registers

- n Aliasing
- n FP rekistereitä käytetään pinona
- n MMX multimediakäskyt käyttävät samoja, mutta viittaavat suoraan nimillä
- n Tag kertoo kummasta kyse
- n MMX-käytössä bitit 64-79 ykkösiä g NaN

(Sta06 Fig 12.22)

Tietokoneen rakenne / 2006 / Teemu Kerola
26.9.2006
Luento 9 - 4

Pentium: EFLAGS Register

<p>ID = Identification flag VIP = Virtual interrupt pending VIF = Virtual interrupt flag AC = Alignment check VM = Virtual 8086 mode RF = Resume flag NT = Nested task flag IOPL = I/O privilege level OF = Overflow flag</p>	<p>DF = Direction flag IF = Interrupt enable flag TF = Trap flag SF = Sign flag ZF = Zero flag AF = Auxiliary carry flag PF = Parity flag CF = Carry flag</p>
---	--

(Sta06 Fig 12.20)

Tietokoneen rakenne / 2006 / Teemu Kerola
26.9.2006
Luento 9 - 5

Pentium: Control Registers

<p>PCE = Performance Counter Enable PGE = Page Global Enable MCE = Machine Check Enable PAE = Physical Address Extension PSE = Page Size Extensions DE = Debug Extensions TSD = Time Stamp Disable PVI = Protected Mode Virtual Interrupt VME = Virtual 8086 Mode Extensions PCD = Page-level Cache Disable PWT = Page-level Writes Transparent</p>	<p>PG = Paging CD = Cache Disable NW = Not Write Through AM = Alignment Mask WP = Write Protect NE = Numeric Error ET = Extension Type TS = Task Switched EM = Emulation MP = Monitor Coprocessor PE = Protection Enable</p>
---	--

(Sta06 Fig 12.21)

Tietokoneen rakenne / 2006 / Teemu Kerola
26.9.2006
Luento 9 - 6

Pentium: Keskeytykset

Sta06 Table 12.3

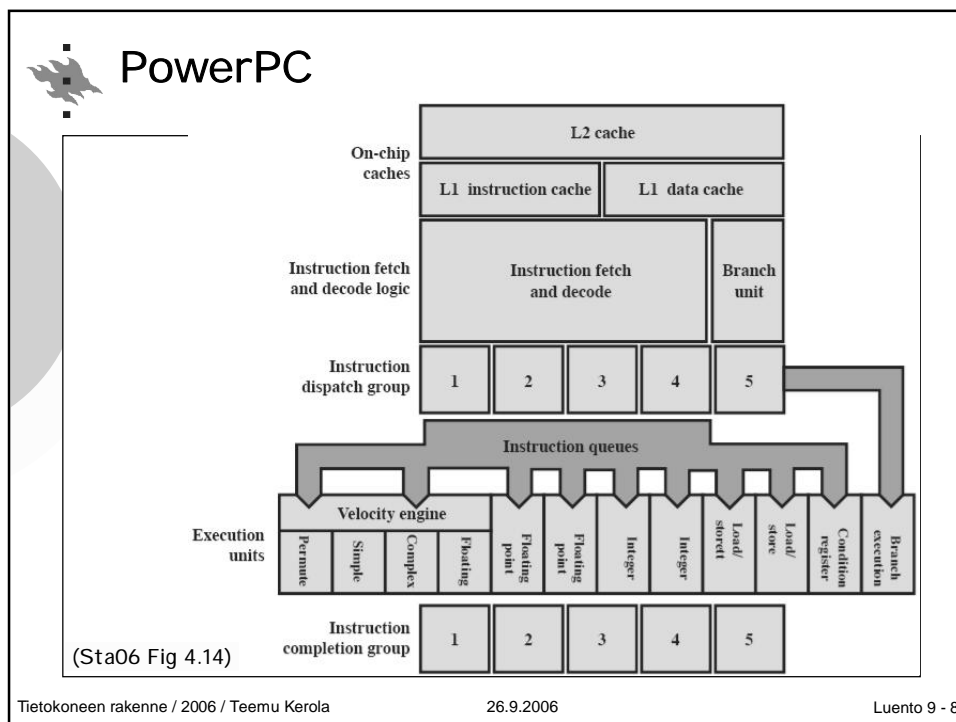
n Keskeytyksikäsitteilyä siirtyminen (atominen laite toiminta)


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

n Paluu

- u Etuoikeutettu IRET-käsky (interrupt return)
- u Palauta pinosta kaikki entiselleen

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






PowerPC: Rekisterit (user visible)

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


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



PowerPC: Rekisterit (control & status)

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

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



PowerPC: Keskeytykset

- n CPU:n sisäiset vs. ulkoiset Sta06 Tbl 12.6
- n Keskeytyksäsittelijään siirtyminen (laitetoiminto)
 - u SRR0 • PC
 - u SRR1 • MSR + tyyppiin sidottua tietoa
 - u MSR • keskeytystyyppin mukaan määräytyvää tietoa
 - § Etuoikeutettu ON, keskeytykset OFF, osoitemuunnos OFF
 - § Muut bitit määräytyvät keskeytyksen perusteella
 - u PC • Keskeytyksäsittelijän osoite
 - § Nouda keskeytysvektorista keskeytyksen "numerolla"
 - § Bitti 57: mahdollisuus määrittellä kaksi vektoria
- n Paluu
 - u Etuoikeutettu rfi-käskey (return from interrupt)
 - u MSR • "reset" ja kopioi SRR1:stä
 - u PC • SRR0

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



Luento 9

Tietokoneen rakenne

RISC- arkkitehtuuri



Ch 13 [Sta06]

- n Käskeyanalyysia
- n RISC vs. CISC
- n Rekisterien käytöstä

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

Laitteistotason virstanpylväitä

Atlas	n	Virtuaalimuisti, 1962	Tom Kilburn
		u Helpompi muistinhallinta	
Atlas	n	Liukuhihna, 1962	Tom Kilburn
IBM S/360, DEC PDP-8	n	Arkkitehtuuriperhe, 1964	Gene Amdahl
		u Samaa käskykantaa noudattavia koneita	
IBM S/360	n	Mikro-ohjelmoitu kontrolli, 1964	Maurice Wilkes
		u Käskykantaa helpompi kehittää	
Univac	n	Moniprosessorijärjestelmä, 1964	J.P. Eckert, John Mauchly
		u test_and_set käsky tarpeen	
IBM S/360	n	Välimuisti, 1965	Maurice Wilkes
		u Huikea suorituskykyparannus	
IBM	n	RISC-arkkitehtuuri, 1980	John Cocke, 1974 J.L. Hennessy & D.A. Patterson
		u Yksinkertainen käskykanta	
IBM, Intel	n	Superscalar CPU, 1989	John Cocke, 1965 IBM, Intel
		u Useita käskyjä valmiiksi per sykli	
Intel	n	Hyperthreading CPU, 2001	CDC, 1964 Intel
		u Usea rekisterijoukko ja virtuaalisuoritin lastulla	
Intel, Sony-Toshiba-IBM	n	Multicore CPU, 2005	Intel, IBM
		u Usea täydellinen suoritin lastulla	



"Perinteiset" koneet

n Kantavia ideoita

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

= **CISC** (Complex Instruction Set Computer)



Mitä operaatioita / operandeja?

- n Vuosi 1982, kohde: VAX, PDP-11, Motorola 68000
- n 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%

Dynamic Percentage of Operands

80% viittauksista aliohjelmien paikallisiin

(Sta06 Table 13.2, 13.3)



Aliohjelmakutsut?

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

(Sta06 Table 13.4)

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

- n Em. tietojen hyödyntäminen?

98% alle 6 parametria
92% alle 6 paikallista muuttujaa



Huomioita

- n Pääosa operandeista yksinkertaisia
- n Hyppykäskyjä runsaasti
- n Kääntäjätään eivät aina hyödynnä tarjolla olevia monipuolisia konekäskyjä
 - u Käyttävät vain osaa käskykannan käskyistä
- n 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

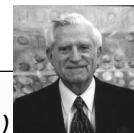
- n Optimoita asioita, joissa kuluu eniten aikaa
 - u Aliohjelmakutsut, silmukat, muistiinviittaukset, ...
- n Esimerkki huonosta optimoinnista
 - u Tuplaa liukulukuaritmetiikan nopeus
 - u 10% käskyissä liukulukuaritmetiikkaa


$$\begin{aligned} \text{ExTime}_{\text{new}} &= \text{ExTime}_{\text{old}} * (0.9 * 1.0 + 0.1 * 0.5) \\ &= 0.95 * \text{ExTime}_{\text{old}} \end{aligned}$$

$$\text{Speedup} = \text{ExTime}_{\text{old}} / \text{ExTime}_{\text{new}} = 1 / 0.95 = 1.053 \ll 2$$

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

- n Optimoi suoritusnopeutta, älä kääntämisen helppoutta / suoraviivaisuutta
 - u Kääntäjät erinomaisia, koneet tehokkaita
 - § Osaavat ja ehtivät optimoida
 - u Tee yleisimmät tehtävät laitetoimintoina, tehokkaasti
 - § Esim. 1-uloitteiseen taulukkoon viittaus
 - u Tee loput ohjelmallisesti
 - § Esim. merkkijonon muunnos
 - § Tarjoa valmiit kirjastorutiinit

Ö RISC arkkitehtuuri (Reduced Instruction Set Computer)

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



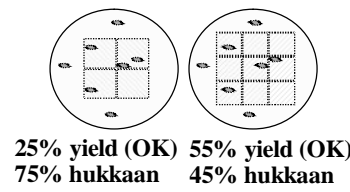
RISC arkkitehtuuri

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

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

RISC arkkitehtuuri

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



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 ^a	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
MC68040	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 80960	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



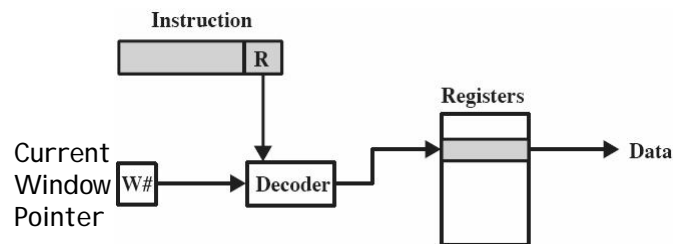
Rekisterijoukko

n Enemmän rekistereitä kuin käskyssä voi viitata

- u Esim. SPARCissa rekisterinrolle 5 b ž suurin nro 32, mutta CPUssa 40-540 rekisteriä

n Kerralla käytössä vain osa rekistereistä, ikkuna

- u Ikkunaan viitataan esim. rekisterinumeroilla r0-r31
- u CPU kuvaa ne tod. rekisterinumeroiksi, esim. r0-r539



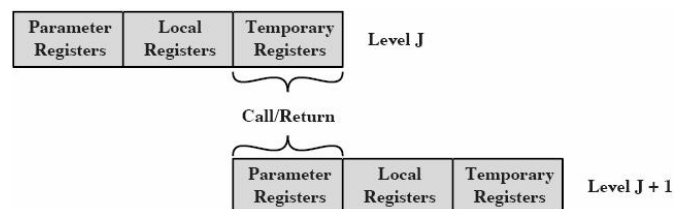
(Sta06 Fig 13.3)



Rekisteri-ikkuna

n Aliohjelmakutsu käyttää pinon sijasta rekistereitä

- u Kutsussa kiinteä määrä rekistereitä parametreille sekä paikallisille muuttujille
- u Varaukset limittäin siten, että parametrit käytettävissä sekä kutsuvassa että kutsutussa osassa



(Sta06 Fig 13.1)

Rekisteri-ikkuna

Jos sisäkkäisiä kutsuja paljon

- u Rekisterijoukko voi loppua kesken
- u Talleta vanhin muistiin, lataa takaisin, kun paluu hämmöttää
- u Kutsuketju harvoin pitkä, talletus/palautus harvoin

Globaalit muuttujat?

- u Muistissa tai oma rekisteri-ikkuna

SPARC

- u r0-r7 globaaleille todelliset rekisterit
- u r8-r15 parametrit (kutsutussa)
- u r16-r23 lokaalit muuttujat
- u 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	(Sta06 Table 13.5)
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	

n Kääntäjän vaikea päätellä etukäteen mitkä globaalit muuttujat pitäisi sijoittaa rekistereihin

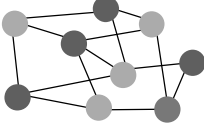
n Välimuisti ratkaisee sen dynaamisesti

- u Eniten viitatut pysyvät välimuistissa

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

Kääntäjä: Rekistereiden allokointi

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

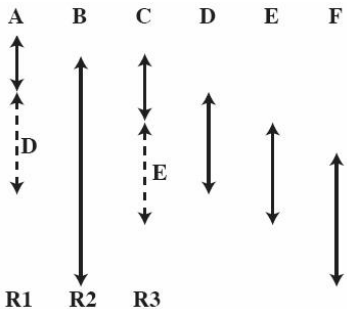


Laskennan teoria
-kurssin asioita

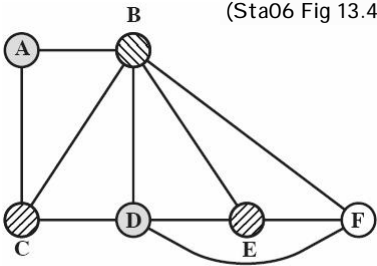
Tietokoneen rakenne / 2006 / Teemu Kerola 26.9.2006 Luento 9 - 29

Kääntäjä: Rekistereiden allokointi

- u solmu = symbolinen rekisteri
- u särmä = symbolisten rekistereiden yhtäaikainen käyttö
- u 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					
				I	E	D			
					I	E			

(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						
		I		E	D				
			I		E				
				I		E			
					I		E		
						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						
		I		E	D				
			I		E				
				I		E			
					I		E		
						I		E	

(c) Three-stage pipelined timing
Two port MEM

Load rA ← M
Load rB ← M
NOOP
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						
		I	E ₁	E ₂					
			I	E ₁	E ₂	D			
				I	E ₁	E ₂	D		
					I	E ₁	E ₂		
						I	E ₁	E ₂	

(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						Traditional
101 ADD l, rA		I		E					
102 JUMP 105				I	E				
103 ADD rA, rB					I				
105 STORE rA, Z						I	E	D	
100 LOAD X, rA	I	E	D						RISC with inserted NOOP Two port MEM
101 ADD l, rA		I	E						
102 JUMP 106			I	E					
103 NOOP				I	E				
106 STORE rA, Z					I	E	D		
100 LOAD X, rA	I	E	D						RISC with reversed instructions (Sta06 Fig 13.7)
101 JUMP 105		I	E						
102 ADD l, rA			I	E					
105 STORE rA, Z				I	E	D			

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



RISC & CISC United?

- n Pentium, CISC
 - u Laitetoiminto kääntää 1 - 11 tavun pituisen CISC-käskyn yhdeksi tai useammaksi 118 bittiseksi mikro-operaatioksi (L1 tason käskyvälimuistiin) Käännös joka käskyn suorituskerralla
 - u Alemmat tasot kuten RISC
 - u Paljon työkistereitä: laitteisto ottaa käyttöön
- n Crusoe (Transmeta)
 - u Ulospäin CISC-arkkitehtuuri Just in time (JIT) compilation
 - u 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
 - u Alemmat tasot kuten RISC

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



Kertauskysymyksiä

- n Mitkä ovat RISC arkitehtuurin tunnuspiirteet?
- n Miten rekisteri-ikkunoita käytetään?

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