

Tietokoneen rakenne

Luento 7

Käskykannat

Ch 10-11 [Sta06]

- n Operaatioista
- n Operandeista
- n Osoitustavoista
- n Pentium / PowerPC

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 1

Käskysykli

- n CPU suorittaa ohjelmaa konekielinen "käsky kerrallaan"
- n Käskyn suoritus muodostuu vaiheista, joita CPU toistaa jokaiselle käskylle

(Sta06 Fig 10.1)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 2

Konekäskyt

- n Käskykanta =
 - u CPU:n tunnistama konekielisten käskyjen kokoelma
- n **Operaatiokoodi**
 - u Mitä käsky tekee? Pureskeltu TITO kurssilla
- n **Viitteet operandeihin (yksi/useita)**
 - u Mistä data, jolle operaatio tehdään?
 - § Rekistereistä, muistista, I/O laitteelta Nopeus?
 - u Minne tulokset talletetaan?
 - § Rekistereihin, muistiin, I/O laitteelle
- n **Mikä käsky seuraavaksi**
 - u Implisiitisesti? Explisiitisesti?
- n **I/O?**
 - u Muistiinkuvattu I/O z samat tavat kuin muistille

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 3

Käskyt ja Data

		Address					Contents								
käsyt	101	0010	0010	0000	0001	101	2201	data	201	0000	0000	0000	0010	201	0002
	102	0001	0010	0000	0010	102	1202		202	0000	0000	0000	0011	202	0003
	103	0001	0010	0000	0011	103	1203		203	0000	0000	0000	0100	203	0004
	104	0011	0010	0000	0100	104	3204		204	0000	0000	0000	0000	204	0000

(a) Binary program (b) Hexadecimal program

Address	Instruction	Operand	Label	Operation	Operand
101	LDA	201		LDA	I
102	ADD	202		ADD	J
103	ADD	203		ADD	K
104	STA	204		STA	N

(c) Symbolic program

Label	FORMUL	Operation	Operand	symbolinen tunnus
I		DATA	2	}
J		DATA	3	
K		DATA	4	
N		DATA	0	

(d) Assembly program (Sta06 Fig 10.11)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 4

Millaisia käskyjä tarvitaan?

Sta06 Table 10.3

- n **Siirto muistista rekisteriin / rekisteristä muistiin**
 - u LOAD, STORE, MOVE, PUSH, POP, ...
- n **I/O-laitteen ohjaus**
 - u Kuten yllä (muistiinkuvattu I/O)
 - u Omat I/O-ohjauskäskyt (ei muistiinkuvattu I/O)
- n **Aritmeettiset ja loogiset operaatiot**
 - u ADD, MUL, CLR, SET, COMP, AND, SHR, NOP, ...
- n **Estytyspamuunnokset**
 - u TRANS, CONV, 16bTo32b, IntToFloat, ...
- n **Käskyjen suoritusjärjestyksen ohjaus, ehdoton/ehdollinen**
 - u JUMP, BRANCH, JEQU, CALL, EXIT, HALT, ...
- n **Palvelupyynnöt**
 - u SVC, INT, IRET, SYSETER, SYSEXT, ...
- n **Etuolkeutetut käskyt**
 - u DIS, IEN, flush cache, invalidate TLB, ...

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 5

Miltä operaatioissa tapahtuu?

(Sta06 Table 10.4)

Data Transfer	Transfer data from one location to another
	If memory is involved: Determine memory address Perform virtual-to-actual-memory address transformation Check cache Initiate memory read/write
Arithmetic	May involve data transfer, before and/or after
	Perform function in ALU Set condition codes and flags
Logical	Same as arithmetic
Conversion	Similar to arithmetic and logical. May involve special logic to perform conversion
Transfer of Control	Update program counter. For subroutine call/return, manage parameter passing and linkage
	Issue command to I/O module
I/O	If memory-mapped I/O, determine memory-mapped address

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 6

Millaista dataa käsitellään?

- Kokonaislukuja, liukulukuja totuusarvoja**
- Merkkejä, merkkijonoja**
 - IRA (aka ASCII), EBCDIC
- Vektoreita, taulukoita**
 - N kpl alkioita potkossa
- Muistiosoitteita**
- Erikokoisia operandeja**
 - 8 /16/32/ 64b, ...
 - Kutakin tyyppiä/kokoa varten omat operaatiokoodit

Operation Mnemonic	Name	Number of Bits Transferred
L	Load	32
LH	Load Halfword	16
LR	Load	32
LER	Load (Short)	32
LE	Load (Short)	32
LDR	Load (Long)	64
LD	Load (Long)	64
ST	Store	32
STH	Store Halfword	16
STC	Store Character	8
STE	Store (Short)	32
STD	Store (Long)	64

(Sta06 Table 10.5)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 7

Käskyformaatti

- Paljonko kullekin käskyn osalle bittejä?**
 - Montako erilaista käskykoodia tarvitaan?
 - Montako operandia voi osoittaa yhdessä käskyssä?
 - Onko operandi rekisterissä vai muistissa?
 - Montako rekisteriä osoitettavissa?
- Vakio- vs. vaihtelevanmittaiset käskyt?**

Number of Addresses	Symbolic Representation	Interpretation
3	OP A, B, C	A ← B OP C
2	OP A, B	A ← A OP B
1	OP A	AC ← AC OP A
0	OP	T ← (T - 1) OP T

AC = accumulator T = top of stack
A, B, C = memory or register locations (T - 1) = second element of stack

(Sta06 Table 10.1)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 8

Montako rekisteriä?

- Vähintään 16-32 kpl**
 - Työdata rekistereissä
- Rekisterijoukot omiin tarkoituksiinsa?**
 - Esim. Kokonaisluvuille omansa ja liukuluvuille omansa, indeksoinnille omansa ja datalle omansa
 - Ne voi molemmat numeroida alkaen 0:stä
 - Käskykoodi määrää kumpaa käytettävä
- Enemmän rekistereitä kuin käsky voi viltata?**
 - CPU huolehtii sisäisesti niiden allokoinnista
 - Rekisteri-ikkuna
 - Esim. Aliohjelman parametrit aina rekistereissä
 - Ohjelmoijan mielestä rekistereissä r8-r15, CPU sijoittelee rekisterijoukkoon välille 8-132 (palataan tähän myöhemmin)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 9

Käskyarkkitehtuureja

- Akkukone**
 - Vain yksi rekisteri, implisiittinen osoittaminen
- Pinokone** Ks. Sta06 Appendix 10A
 - Operandit pinossa, implisiittinen osoittaminen
 - PUSH, POP Esim. JVM
- Yleisrekisterikone**
 - Vain yhdenkoon rekistereitä
 - Käskyssä 2 tai 3 operandia
- Load/Store arkkitehtuuri**
 - Vain LOAD/STORE viittaavat muistiin LOAD R3, C
 - ALU-operaatioissa tav. 3 rekisteriä LOAD R2, B
 - ADD R1, R2, R3
 - STORE R1, A

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 10

Tavujärjestys: Big vs. Little Endian

Ks. Sta06 Appendix 10B

- Kuinka usean tavun kokoinen luku talletetaan?**

0x1200:

Sanaosoite (Word) 0x1200 0x1201 0x1202 0x1203

Tavuosoitteet (Byte)

STORE 0x11223344, 0x1200 ???

Big-Endian: eniten merkittävällä tavulla pienin osoite → 0x11 0x22 0x33 0x44
0x1200 0x1201 0x1202 0x1203

Little-Endian: vähiten merkittävällä tavulla pienin osoite → 0x44 0x33 0x22 0x11
0x1200 0x1201 0x1202 0x1203

0x00000044 = 0x44 0x00 0x00 0x00
Sama osoite tavuna ja sanana: 0x1200 0x1201 0x1202 0x1203

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 11

Big vs. Little Endian

- ALU käyttää vain jompaakumpaa**
 - Little-endian: x86, Pentium, VAX
 - Big-endian: IBM 370/390, Motorola 680x0 (Mac), useimmat RI SC-arkkitehtuurin koneet
 - Power-PC kelpuuttaa kummankin
 - Bitti ohjausrekisterissä (MSR, machine status register)
 - Järjestys vaihdetaan tarvittaessa ALUa ennen/jälkeen
- Tavujärjestys huomioitava, kun tietoa siirretään koneesta toiseen**
 - Internet käyttää big-endian muotoa
 - Pistokekirjastossa rutiniit `htonl()` ja `ntohl()` (Host to Internet & Internet to Host)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 12

Datan kohdentaminen (alignment)

- 16b data alkamaan parillisista (tavu)osoitteista **0010...10010**
- 32b data 4:llä jaollisista osoitteista **0010...10100**
- 64b data 8:llä jaollisista osoitteista **0010...11000**
- Kohdennettu data helpompaa käsitellä**
 - esim. 32b data ladattavissa yhdellä muistinoudolla (sanaosoite)

11	22	33	44
----	----	----	----
- Kohdentamaton data ei tuota vällin "hukkatavuja"**
 - Esim. 32b kohdentamaton data tarvitsee kaksi muistinoutoa (sanaosoite) ja yhdistämisen

		11	22
33	44		

```

load r1, 0(r4)
shl r1, =16
load r2, 1(r4)
shr r2, =16
or r1, r2
    
```

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 13

Tietokoneen rakenne

Muistin osoitustavat

Ch 11 [Sta06]

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 14

Missä käskeyn operandit?

- Muistissa**
 - Ohjelman muuttujat, pino, keko
- Rekistereissä**
 - Käsittelyn aikana (nopeus)
- Käskeyn osana**
 - Pienet vakiot
- Miten tuo kerrotaan CPU:lle?**
 - Bitit käskeynformaatisissa
 - Useita osoitusmuotoja

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 15

Tiedon osoitusmuodot

(Sta06 Fig 11.1)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 16

Tiedon osoitusmuodot

Mode	Algorithm	Principal Advantage	Principal Disadvantage
Immediate	Operand = A	No memory reference	Limited operand magnitude
Direct	EA = A	Simple	Limited address space
Indirect	EA = (A)	Large address space	Multiple memory references
Register	Operand = (R)	No memory reference	Limited address space
Register indirect	EA = (R)	Large address space	Extra memory reference
Displacement	EA = A + (R)	Flexibility	Complexity
Stack	EA = top of stack	No memory reference	Limited applicability

- EA = Effective Address
- (A) = Muistipaikan A sisältö
- (R) = Rekisterin R sisältö
- Pinon päällimmäisen alkion osoitteelle oma rekisteri
- Pinon päällimmäinen alkio (tai 2) omassa rekisterissä

(Sta06 Table 11.1)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 17

Siirtymä (Displacement Address)

- Effective address = (R1) + A** Tehollinen muistiosoite
 - rekisterin sisältö + käskeyssä annettu vakio
- Vakio usein pieni (8 b, 16 b?)**
- Käyttötapa**
 - PC:n suhteen viittaus **JUMP *+5**
 - Base rekisterin suhteen **CALL SP, Summation(BX)**
 - Taulukon indeksointi **ADDF F2,F2, Table(R5)**
 - Tietueen kenttään viittaus **MUL F4,F6, Salary(R8)**
 - Pinoon viittaaminen (aktiivintietue) **STORE F2, -4(FP)**

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 18

Lisää tiedon osoitustapoja

- Autoincrement (before/after)** $EA = (R), R \leftarrow (R) + S$
 u Esim. CurrIndex=i++;
- Autodecrement (before/after)** $R \leftarrow (R) - S, EA = (R)$
 u Esim. CurrIndex=-i;
- Autoincrement deferred** $EA = Mem(R), R \leftarrow (R) + S$
 u Esim. Sum = Sum + (*ptrX++);
- Autoscale** $EA = A + (R_j) + (R_i) * S$
 u Esim. Double X;
 X=Tbl[i][j];

operandin koko

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 19

Tietokoneen rakenne

Pentium

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 20

Pentium: Rekisterit

- Yleisrekisterit, 32-b**
 - EAX, EBX, ECX, EDX työreisterit
 accu, base, count, data
 - ESI, EDI source & destination index
 - ESP, EBP stack pointer, base pointer
- Niliden loppuosia voi käyttää 16-bittisinä**
 - AX, BX, CX, DX, SI, DI, SP, BP
- Työreistereiden loppuosia 8-bittisinä**
 - AH, AL, BH, BL, CH, CL, DH, DL
- Segmenttirekisterit, 16b**
 - CS, SS, DS, ES, FS, GS
 - § code, stack, data, stack, extra data
- Käskynosoitin**
 - EIP Extended Instruction Pointer
- Statusrekisterit**
 - EFLAGS
 - § overflow, sign, zero, parity, carry,...

General Registers			
EAX			AX
EBX			BX
ECX			CX
EDX			DX
ESP			SP
EBP			BP
ESI			SI
EDI			DI

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 21

Pentium: Datatyytit

PentiumData Type	Description	Ei kohdennettu Little Endian
General	Byte, word (16 bits), doubleword (32 bits), and quadword (64 bits) locations with arbitrary binary contents.	
Integer	A signed binary value contained in a byte, word, or doubleword, using twos complement representation.	
Ordinal	An unsigned integer contained in a byte, word, or doubleword.	
Unpacked binary coded decimal (BCD)	A representation of a BCD digit in the range 0 through 9, with one digit in each byte.	
Packed BCD	Packed byte representation of two BCD digits; value in the range 0 to 99.	
Near pointer	A 32-bit effective address that represents the offset within a segment. Used for all pointers in a nonsegmented memory and for references within a segment in a segmented memory.	
Bit field	A contiguous sequence of bits in which the position of each bit is considered as an independent unit. A bit string can begin at any bit position of any byte and can contain up to $2^{32} - 1$ bits.	
Byte string	A contiguous sequence of bytes, words, or doublewords, containing from zero to $2^{32} - 1$ bytes.	
Floating point	Single / Double / Extended precision	

(Sta06 Table 10.2)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 22

Pentium: Operaatiot (tässä vain osa)

High-Level Language Support	
ENTER	Creates a stack frame that can be used to implement the rules of a block-structured high-level language.
LEAVE	Reverses the action of the previous ENTER.
BOUND	Check array bounds. Verifies that the value in operand I is within lower and upper bounds.
Segment Register	
LDS	Load pointer into D segment register.
System Control	
HLT	Halt.
LOCK	Asserts a hold on shared memory so that the Pentium has exclusive use of it during the instruction that immediately follows the LOCK.
ESC	Processor extension escape. An escape code that indicates the succeeding instructions are to be executed by a numeric coprocessor that supports high-precision integer and floating-point calculations.
WAIT	Wait until BUSY# negated. Suspends Pentium program execution until the processor detects that the BUSY pin is inactive, indicating that the numeric coprocessor has finished execution.
Protection	
SGDT	Store global descriptor table.
LSL	Load segment limit. Loads a user-specified register with a segment limit.
VERR/VERW	Verify segment for reading/writing.
Cache Management	
INVD	Flushes the internal cache memory.
WBINVD	Flushes the internal cache memory after writing dirty lines to memory.
INVLPG	Invalidate a translation lookaside buffer (TLB) entry.

(Sta06 Table 10.8)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 23

Pentium: MMX Operaatiot (tässä vain osa)

Category	Instruction	Description
Arithmetic	PADD [B, W, D]	Parallel add of packed eight bytes, four 16-bit words, or two 32-bit doublewords, with wraparound.
	PADD [B, W]	Add with saturation. No under/overflow.
	PADD [B, W]	Add unsigned with saturation. Use closest representation.
	PSUB [B, W, D]	Subtract with wraparound.
	PSUB [B, W]	Subtract with saturation.
	PSUB [B, W]	Subtract unsigned with saturation.
	PMULHW	Parallel multiply of four signed 16-bit words, with high-order 16 bits of 32-bit result chosen.
	PMULLW	Parallel multiply of four signed 16-bit words, with low-order 16 bits of 32-bit result chosen.
	PMADDWD	Parallel multiply of four signed 16-bit words; add together adjacent pairs of 32-bit results.
	Conversion	PACKUSWB
PACKSS [WB, DW]		Pack words into bytes, or doublewords into words, with signed saturation.
PUNPCKH [BW, WD, DQ]		Parallel unpack (interleaved merge) high-order bytes, words, or doublewords from MMX register.
PUNPCKL [BW, WD, DQ]		Parallel unpack (interleaved merge) low-order bytes, words, or doublewords from MMX register.

(Sta06 Table 10.11)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 24

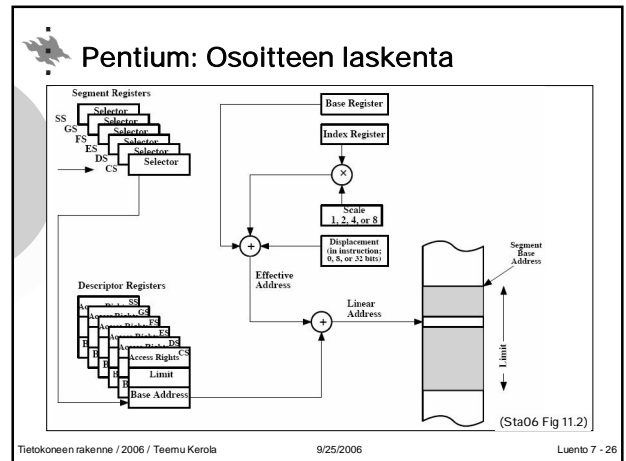
Pentium: Osoitustavat

Pentium Addressing Mode	Algorithm	1, 2, 4, 8B
Immediate	Operand = A	
Register Operand	Operand = (R)	
Displacement	LA = (SR) + A	Rekisterit: 1, 2, 4, 8B
Base	LA = (SR) + (B)	
Base with Displacement	LA = (SR) + (B) + A	
Scaled Index with Displacement	LA = (SR) + (I) × S + A	
Base with Index and Displacement	LA = (SR) + (B) + (I) + A	
Base with Scaled Index and Displacement	LA = (SR) + (I) × S + (B) + A	
Relative	LA = (PC) + A	

LA = linear address R = register
(X) = contents of X B = base register
SR = segment register I = index register
PC = program counter S = scaling factor
A = contents of an address field in the instruction

(Sta06 Table 11.2)

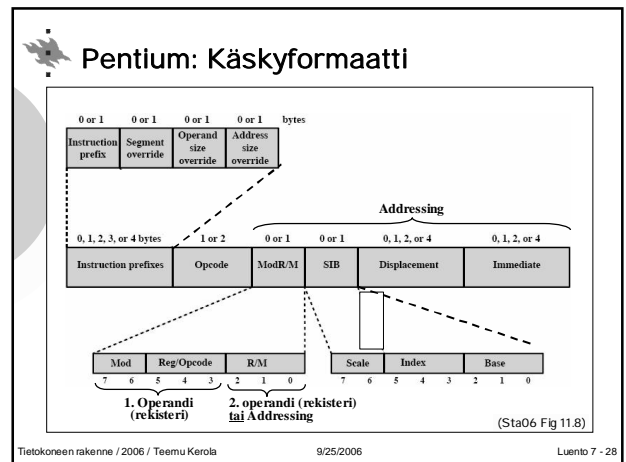
Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 25



Pentium: Käskyformaatti

- n CISC
 - u Complex Instruction Set Computer
- n Paljon valinnaisia osia
 - u Osio esiintyy käskyn bittijonossa tai ei
 - u Alussa 0-4 prefix tavua
 - u Loppuosan tulkinta riippuu aiempien kenttien sisällöistä
- n Monipuoliset osoitustavat
 - u Käskyn operandeista korkeintaan yksi muistissa
 - u 24 erilaista
- n Taaksepäin yhteensopivuus
 - u Icuwanhojen 16-bittisten 8086-ohjelmien pitää toimia
 - § Vanhat käskyt emuloimalla vai simuloimalla?

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 27



Pentium: Käskyformaatti

(Sta06 Fig 11.8)

- n Instruction prefix (optional)
 - u LOCK - jakamaton operaatio moniCPU-järjestelmässä
 - u REP - toista samaa käskyä merkkijonon kaikille merkeille
- n Segment override (optional)
 - u Käytä tässä käskyssä eksplisiittisesti annettua segmenttirekisteriä
 - u Käytä muulloin implisiittistä oletusta
- n Operand size override (optional)
 - u Onko operandi 16 vai 32 bittinen (toinen oletuksena)
- n Address size override (optional)
 - u Onko osoite 16 vai 32 bittinen (toinen oletuksena)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 29

Pentium: Käskyformaatti

(Sta06 Fig 11.8)

- n Opcode
 - u Kullekin käskylle oma bittikombinaatio
 - u Bitit kertovat myös operandin koosta (8/16/32b)
- n ModR/M (optional)
 - u Onko operandi rekisterissä vai muistissa
 - u Mitä osoitusmuotoa käytetään
 - u Joissakin tapauksissa täsmentää operaatiokoodia
- n SIB = Scale/Index/Base (optional)
 - u Joissakin osoitusmuodoissa tarvitaan lisätarkennuksia
 - u Scale: alkion koko indeksointia käytettäessä
 - u Index: indeksirekisterin numero
 - u Base: kantarekisterin numero

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 30

Pentium: Käskyformaatti

Sta06 Fig 11.8

- n **Displacement (optional)**
 - u Tarvitaan eräissä osoitustavoissa
 - u 0, 1, 2 tai 4 tavua
- n **Immediate (optional)**
 - u Tarvitaan eräissä osoitustavoissa
 - u 0, 1, 2 tai 4 tavua

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 31

Tietokoneen rakenne

PowerPC

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 32

PowerPC: Käskykanta

- n **RISC**
 - u Reduced Instruction Set Computer
- n **Kiinteä käskykanta (32b), vähän formaatteja**
 - u Käskyissä tavallisesti 3 operandia
- n **Minimijoukko erilaisia käskyjä**
 - u Helpompi laitteistototeutus, nopeampi suorittaa
 - u Pitemmät ohjelmat?
- n **Vain 2 osoitusmuotoa**
 - u Load/Store-arkkitehtuuri
- n **32 yleisrekisteriä**
- n **Kiinteä datan koko (32/64)**
- n **Ei esim. merkijono-operaatioita**
 - u Kirjastoina

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 33

PowerPC: Osoitustavat

Mode	Algorithm	
Load/Store Addressing		
Indirect	EA = (BR) + D	EA = effective address
Indirect Indexed	EA = (BR) + (IR)	(X) = contents of X
Branch Addressing		
Absolute	EA = I	BR = base register
Relative	EA = (PC) + I	IR = index register
Indirect	EA = (L/CR)	L/CR = link or count register
Fixed-Point Computation		
Register	EA = GPR	GPR = general-purpose register
Immediate	Operand = I	FPR = floating-point register
Floating-Point Computation		
Register	EA = FPR	D = displacement
		I = immediate value
		PC = program counter

(Sta06 Table 11.3)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 34

PowerPC: Käskyformaatti

XO=Opcode extension, R=Record condition in CRI, O=Record overflow in XER
S = Part of Shift Amount field, * = 64 bitin arkkitehtuureissa

← 6 bits	← 5 bits	← 5 bits	← 16 bits
Ld/St Indirect	Dest Register	Base Register	Displacement
Ld/St Indirect	Dest Register	Base Register	Index Register
Ld/St Indirect	Dest Register	Base Register	Displacement

(c) Load/store instructions

Arithmetic	Dest Register	Src Register	Src Register	O	Add, Sub, etc.	R
Add, Sub, etc.	Dest Register	Src Register	Src Register		Signed Immediate Value	
Logical	Src Register	Dest Register	Src Register		ADD, OR, XOR, etc.	R
AND, OR, etc.	Src Register	Dest Register	Src Register		Unsigned Immediate Value	
Rotate	Src Register	Dest Register	Shift Amt	Mask Begin	Mask End	R
Rotate or Shift	Src Register	Dest Register	Src Register		Shift Type or Mask	R
Rotate	Src Register	Dest Register	Shift Amt	Mask	XO	S R *
Rotate	Src Register	Dest Register	Src Register	Mask	XO	S R *
Shift	Src Register	Dest Register	Shift Type or Mask		S	R *

(d) Integer arithmetic, logical, and shift/rotate instructions

Flt sgl/dbl	Dest Register	Src Register	Src Register	Src Register	Fadd, etc.	R
-------------	---------------	--------------	--------------	--------------	------------	---

(e) Floating-point arithmetic instructions

(Sta06 Fig 11.9)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 35

PowerPC: Käskyformaatti

- n **Yksinkertaiset hyppykäsyt**
 - u CR: mitä CR-rekisterin bittejä tutkitaan
 - u L (Link): onko kyseessä aliohjelman siirtyminen (Aktivoititietue kuntoon!)
 - u A (Absolute): onko annettu hypyn kohdeosoite, vai onko hyppy suhteellinen PC:n suhteen

Branch	Long Immediate			A	L
Br Conditional	Options	CR Bit	Branch Displacement	A	L
Br Conditional	Options	CR Bit	Indirect through Link or Count Register	L	

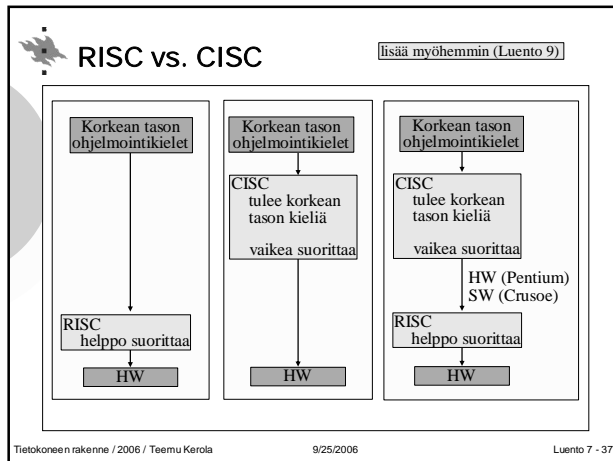
(a) Branch instructions

CR	Dest Bit	Source Bit	Source Bit	Add, OR, XOR, etc.	/
----	----------	------------	------------	--------------------	---

(b) Condition register logical instructions

(Sta06 Fig 11.9)

Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 36



- ### Kertauskysymyksiä
- n Millaisista osista konekielinen käsky muodostuu?
 - n Miten CPU tietää onko sen käsittelemä kokonaisluku 16 bittinen vai 32 bittinen?
 - n Mitä tarkoittaa Big-Endian?
 - n Mitä hyötyä on kiinteästä käskyformaatista verrattuna vaihtelevanpituisen formattiin?
- Tietokoneen rakenne / 2006 / Teemu Kerola 9/25/2006 Luento 7 - 38