

Tietokoneen rakenne

Ohjausyksikkö



Ch 16-17 [Sta06]

- n Mikro-operaatiot
- n Ohjaussignaalit
- n Langoitettu ohjaus
- n Mikro-ohjelmoitu ohjaus

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 1

Mitä ohjaus/kontrolli tarkoittaa?

- n Arkkitehtuuri määrää CPU:n ulkosen, ohjelmoijalle (myös KJ) näkyvän toiminnan
 - u Millainen käskykanta käytössä, mitä käskyt tekevät?
 - u Mikä operaatio, missä operandit?
 - u Miten keskeytykset hoidellaan?

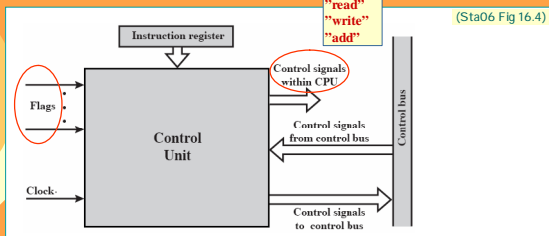
Functional requirements for CPU!

1. Operations
2. Addressing modes
3. Registers
4. I/O module interface
5. Memory module interface
6. Interrupt processing structure

- n Ohjausyksikkö CU määrää kuinka käskyssä kuvatut asiat saadaan tehdyksi laitteistossa (CPU, MEM, väylät, I/O)
 - u Selvittää miten piirien täytyy toimia tietyllä hetkellä
 - u Valitsee millaisia ohjaussignaaleja pitää antaa: mille piireille, milloin, missä järjestyksessä, ...
 - u Fyysiset johtimet välittävät ohjaussignaalit piireille täsmällisesti kellonpulssin ajoittamina

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 2

Ohjaussignaalit



(Sta06 Fig 16.4)

- n Ohjaus = kontrolloidaan data siirtämistä paikasta toiseen
 - u CPU:n sisällä: REG \leftrightarrow REG, ALU \leftrightarrow REG, ALU-operaatiot
 - u CPU \leftrightarrow MEM (I/O-ohjain): osoite, data, ohjaus
- n Ajoitus, järjestys

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 3

Mikro-operaatiot

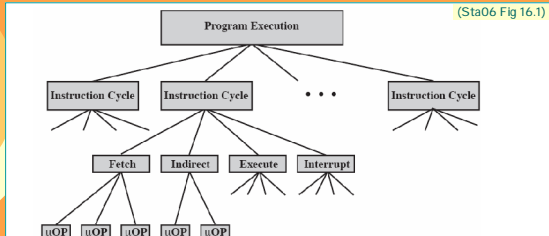
- n Yksinkertaisia ohjaussignaaleja, jotka aiheuttavat yhden pienen toiminnon (operaation)
 - u Esim. bitit siirtyvät rekisteristä r1 väylää pitkin edelleen ALU:un
- n Kellosyklin kesto pisimmän operaation perusteella
- n Kunkin syklin aikana useita mikro-operaatioita
 - u Osa voidaan suorittaa samanaikaisesti, jos tapahtuvat eri osissa piirejä
 - u Vältettävä resurssikonfliktit
 - § WaR tai RaW, ALU, väylä
 - u Osa suoritettava peräkkäin tietyssä järjestyksessä, semantiikan säilyttävä

t1: MAR \leftarrow PC
 t2: MBR \leftarrow MEM[MAR]
 PC \leftarrow PC + 1
 t3: IR \leftarrow (MBP)

Toteutus siten, että ei tarvita ALUa

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 4

Käskysykli

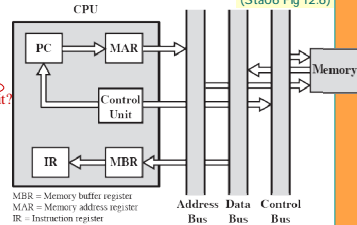


(Sta06 Fig 16.1)

- n Kun mikro-operaatiot kohdistuvat laitteiston eri piireihin, laitteisto voi suorittaa useita samanaikaisesti
- n Ks. luvun 12 käskysykli esimerkit (seur. kalvot)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 5

Käskyn noutosykli



(Sta06 Fig 12.6)

Esim:

- t1: MAR \bullet PC
- t2: MAR \bullet MMU(MAR)
- Control Bus \bullet Reserve
- t3: Control Bus \bullet Read
PC \bullet PC + 1
- t4: MBR \bullet MEM[MAR]
Control Bus \bullet Release
- t5: IR \bullet MBR

MBR = Memory buffer register
 MAR = Memory address register
 IR = Instruction register
 PC = Program counter

Onko suoritusjärjestyksellä merkitystä?
 Mitä voi suorittaa rinnakkain?
 Mitä samaan sykliin, mitkä tarvitsevat oman syklin?

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 6

Käskysykli

- Operandiennoutosykli**
 - Rekistereistä tai muistista
 - Osoitelaskenta
- Suoritussykli**
 - Suoritus tavallisesti ALUssa
 - Operandit sisäänmenoihin, ja ohjaus operaatioista
 - Tulos ALU:n ulostulosta rekisteriin/muistiin
 - flags • status
- Keskeytussykli**
 - ks. luvun 12 esimerkit: Pentium, PowerPC
 - Mitkä samaan mikrokäskyyn?
 - Mitkä samaan aikaan/peräkkäin?

ADD r1,r2,r3:
 t1: ALUin1 ← r2
 t2: ALUin2 ← r3
 ALUoper ← IR.oper
 t3: r1 ← ALUout
 flags ← xxx

ISZ X, Increment and Skip if zero:
 t1: MAR ← IR.address
 t2: MBR ← MEM[MAR]
 t3: MBR ← MBR+1
 t4: MEM[MAR] ← MBR
 if (MBR=0) then PC ← PC + 1

ehdollisuuskin onnistuu

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 7

Käskysykli tila-automaattina

ICC: Instruction Cycle Code register – suorituksen tila

(Sta06 Fig 16.3)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 8

Käskysykli kontrolli tila-automaattina

- CU:n toiminnan voi esittää tila-automaattina**
 - Tila: missä käskysyklin vaiheessa CPU menossa
 - Alitila: riippuu ajoituksesta, muodostuu ryhmästä mikro-operaatioita, jotka voi suorittaa yhdellä syklillä
- Alitilan tuottamat uudet ohjaukset riippuvat**
 - Tilasta itsestään
 - IR-rekisterin kentistä (operaatio, osoittaminen)
 - Edellisistä tuloksista (flags) = Execution
- Uusi tila edellisen tilan ja lipukkelden perusteella**
 - Myös CPU:n ulkopuoliset keskeytykset vaikuttavat tilaan = Sequencing

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 9

Ohjaussignaalien tuottaminen

- Mikro-operaatio Φ aktivoitava useita ohjaussignaaleja
- Esim: Yhden akkurekisterin arkkitehtuuri

(Sta06 Fig 16.5)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 10

Ohjaussignaalien tuottaminen

Micro-operations	Timing	Active Control Signals
Fetch:	t ₁ : MAR ← (PC)	C ₂
	t ₂ : MBR ← Memory	C ₅ , C _R
	PC ← (PC) + 1	C ₉
Indirect:	t ₁ : IR ← (MBR)	C ₄
	t ₂ : MAR ← (IR(Address))	C ₃
	t ₃ : MBR ← Memory	C ₅ , C _R
Interrupt:	t ₁ : IR(Address) ← (MBR(Address))	C ₄
	t ₂ : MBR ← (PC)	C ₁
	t ₃ : Memory ← (MBR)	C ₁₂ , C _W

C_R = Read control signal to system bus.
 C_W = Write control signal to system bus.

(Sta06 Table 16.1)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 11

Yleinen organisointi

- Kuvan 16.5 organisaatio monimutkainen toteutettavaksi
- Komponentit tavallisesti yhteisen väylän varteen
- ALU:lle apurekisterit: X ja Y

ADD E:
 t1: MAR ← IR.address
 t2: MBR ← MEM[MAR]
 t3: Y ← MBR
 t4: Z ← AC + Y
 t5: AC ← Z

(Sta06 Fig 16.6)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 12

Tietokoneen rakenne

Langoitettu ohjaus

Tietokoneen rakenne / 2007 / Liisa Marttinen
7.12.2007
Luento 12 - 13

Langoitettu ohjausyksikkö (hardwired)

- Kun tiedossa CU:n sisäänmenot ja ulostulot
 - Toiminnallisuus kuvattavissa boolean logiikalla
 - Ohjausyksikköstä voi muodostaa yhden loogiikkapiirin
- Esim. $C_5 = \bar{P} \cdot \bar{Q} \cdot T_2 + P \cdot \bar{Q} \cdot (LDA) \cdot T_2 + \dots$

Fig 16.3, 16.5 ja Tbl 16.1
 ICC:lle bitit P ja Q
 PQ = 00 Fetch Cycle
 PQ = 01 Indirect Cycle
 PQ = 10 Execute Cycle
 PQ = 11 Interrupt Cycle

(Sta06 Fig 16.10)

Tietokoneen rakenne / 2007 / Liisa Marttinen
7.12.2007
Luento 12 - 14

Langoitettu ohjausyksikkö

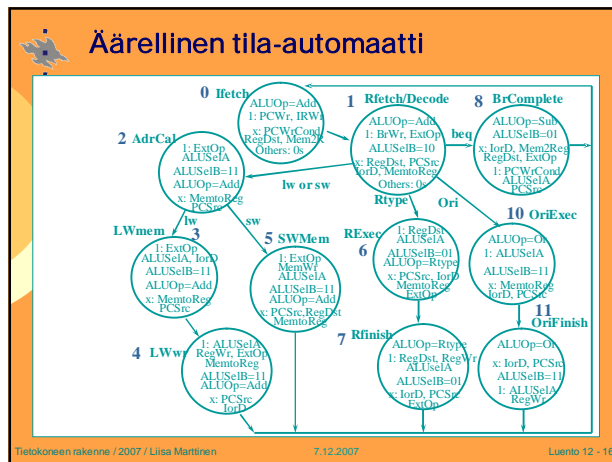
Decoder (4-to-16)

- 4:n bitin käskykoodista yksikäsitteinen ohjaus CU:lle
- Vain yksi signaali kerrallaan aktiivisena

I1	I2	I3	I4	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14	O15	O16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Esim: opcode = 5 (bitit I1, I2, I3, I4) → signaali O11 on tosi (1) (Sta06 Table 16.3)

Tietokoneen rakenne / 2007 / Liisa Marttinen
7.12.2007
Luento 12 - 15



Tilasiirtymät (2)

Next state from current state	Alternatively, prior state & condition
State 0 → State1	S4, S5, S7, S8, S9, S11 → State0
State 1 → S2, S6, S8, S10	_____ → State1
State 2 → S5 or ...	_____ → State2
State 3 → S9 or ...	_____ → State3
State 4 → State0	_____ → State4
State 5 → State0	State2 & op = SW → State5
State 6 → State7	_____ → State6
State 7 → State0	State6 → State7
State 8 → State0	_____ → State8
State 9 → State0	State3 & op = JMP → State9
State 10 → State11	_____ → State10
State 11 → State0	State10 → State11

Tietokoneen rakenne / 2007 / Liisa Marttinen
7.12.2007
Luento 12 - 17

Langoitettu ohjaus

- Ohjaussignaalin generointi suoraan laitteistolla nopeaa
- Helkkouksla
 - CU vaikea suunnitella
 - § Piiristä tulee helposti suuri ja monimutkainen
 - CU vaikea muuttaa
 - § Suunnittelu ja piiriin "minimointi" uusiksi
- RISC-filosofia helpottaa
 - Yksinkertainen käskykanta ja toteutus

Tietokoneen rakenne / 2007 / Liisa Marttinen
7.12.2007
Luento 12 - 18

Tietokoneen rakenne

Mikro-ohjelmoitu ohjaus

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 19

Mikro-ohjelmoitu ohjaus

- Idea 1951: Wilkes Microprogrammed Control Execution Engine
 - Suorituta konekäsky yksi mikrokäsky kerrallaan generoimalla suoritusaikana tarvittavat ohjaussignaalit
 - Tulkitse mikrokäskyt ohjelmallisesti ohjaussignaleiksi
- Mikro-operaatiot kuvattu kontrollimuistissa mikrokäskynä
 - Laiteohjelmisto (firmware)
- Kukin mikrokäsky muodostuu kahdesta osasta
 - Mitä suoritetaan tulevalla syklillä?
 - Mikrokäskystä käy ilmi tarvittavat ohjaussignaalit
 - Toimita ohjaussignaalit piireille Sta06 Table 16.1
 - Mikä mikrokäsky tulkitaan tämän jälkeen?
 - Oletus: seuraavasta kontrollimuistin muistipaikasta
 - Paikanlaskuri, ei tilakone!

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 20

Mikrokoodi

- Kutakin CPU:n käskesyklin vaihetta vastaa "allrutini", joka suoritetaan käskesyklin aikana
- Eslm. ROM-muistissa
 - Mikro-ohjelma-muisti
 - Firmware

(Sta06 Fig 17.2)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 21

Horizontaalinen mikrokäsky

- Kaikki mahdolliset ohjaussignaalit kuvattu bittikarttana jokaisessa mikrokäskyssä
 - Yksi bitti per mahdoll. ohjaussignaali (1=generoi, 0=älä generoi)
 - Jos paljon erilaisia, mikrokäskystä tulee pitkä
- Kukin mikrokäsky myös ehdollinen hyppykäsky
 - Mitä statusbittejä tutkittava
 - Seuraavan mikrokäskyn osoite

(Sta06 Fig 17.1 a)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 22

Vertikaalinen mikrokäsky

- Ohjaussignaalit koodattu toimintonumeroiksi
- Dekoodaa takaisin ohjaussignaleiksi suoritusaikana
- Lyhyemmät käskyt, mutta dekoodaus vie aikaa
- Kukin mikrokäsky myös ehdollinen hyppykäsky
 - Mitä statusbittejä tutkittava
 - Seuraavan mikrokäskyn osoite

(Sta06 Fig 17.1 b)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 23

Ohjausyksikkö = Execution Engine

- Control Address Register, CAR**
 - Mikä mikrokäsky seuraavaksi? ~ käskyosoitin, "MIPC"
- Kontrollimuisti**
 - Mikrokäskyt
 - Nouto-suoritus-keskeytys
- Control Buffer Register, CBR**
 - Rekisteri mikrokäskyn tulkintaa varten ~ käskyrekisteri, "MII R"
 - Tulkitse ja lähetä ohjaussignaalit piireille
 - Dekooderin läpi
- Sequencing Logic**
 - Seuraavaksi suoritettavan mikrokäskyn osoite CAR:iin

(Sta06 Fig 17.4)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 24

Mikä mikrokäskey seuraavaksi?

- a) Kerrottu eksplisiittisesti
- Käskeyssä 2 osoitetta
 - Lisäksi kerrottu lipuke, jota tutkittava
 - Hae uusi jommasta kummasta paikasta
 - Usein heti seuraavasta kontrollimuistin muistipaikasta
 - Miksi tallettaa?
 - Ei aikaa laskemiseen!

(Sta06 Fig 17.6)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 25

Mikä mikrokäskey seuraavaksi?

- b) Implisiittinen oletus: seuraavasta kontrollimuistin osoitteesta
- Käskeyssä 1 osoite
 - Lisäksi kerrottu lipuke, jota tutkittava
 - Jos ehto=1, käytä käskeyn osoiteosaa
- Osoiteosaa ei taaskaan käytetä aina
 - Useimmiten hukkatilaa

(Sta06 Fig 17.7)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 26

Mikä mikrokäskey seuraavaksi?

- c) Vaihteleva mikrokäskeyn formaatti
 - Osa biteistä voidaan tulkita kahdella tavalla
 - 1 b: Käskeyssä osoite/ei ohjaussignaaleja
 - Vain hyppykäskeyssä mukana osoite
 - Hyppykäskeyissä ei ole ohjaussignaaleja
 - Jos hyppy, pitääkin suorittaa kaksi käskeyä yhden sijasta
 - Hukkaa aikaa?
 - Säästää tilaa?

(Sta06 Fig 17.8)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 27

Mikä mikrokäskey seuraavaksi?

- d) Osoitteen generointi suoritusaikana
 - Miten päästä oikean ALU-alirutiiniin alkuun?
 - Konekäskeyn suoritus-signaalit riippuvat operaatiosta
 - Generoi ensimmäisen mikrokäskeyn osoite operaatiokoodista (mapping + combining/adding)
 - Eniten merkitsevät bitit op-koodista
 - Vähiten merkitsevät nollia tai generoitu nykytilan (status) perusteella
 - Esim: IBM 3033 CAR, 13 b:n osoite
 - Op-koodista 8 bittiä -> alirutiini max 32 käskeyä
 - lopun 5 bittiä statusbittejä tutkimalla

(Sta06 Fig 17.9)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 28

Mikä mikrokäskey seuraavaksi?

- e) Allohjelmakutsu ja paluu (Residual control)
 - Alirutiinin kutsu mikrotasolla
 - Ei ympäristön vaihtoa, vain yksi taso
 - Sisäkkäiset kutsut eivät sallittuja
 - Esim: LSI-11, 22b:n mikrokäskey
 - Kontrollimuisti 2048 käskeyä, 11 b:n osoitteet
 - Operaatiokoodi määrää ens. mikrokäskeyn osoitteen
 - Oletus seuraava CAR • CAR+1
 - Jokaisessa mikrokäskeyssä bitti: alirutiinin kutsu/ei Alirutiinin kutsukäskey:
 - talleta paluusoite paluurekisteriin (vain yksi)
 - hyppää alirutiiniin (osoite mikrokäskeyssä)
 - Paluukäskey: ota osoite paluurekisteristä

(Sta06 Fig 17.8)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 29

Mikrokäskeyn koodaus

- Horisontaalinen? Vertikaalinen?
 - Horisontaalinen: nopea tulkinta
 - Vertikaalinen: vähemmän bittejä
- Usein kompromissi eil sekamuoto
 - Mikrokäskey jaettu kenttiin, kukin kenttä määrää tiettyjen toimintojen ohjaussignaaleit
 - Keskenään toisensa poissulkevat toiminnot koodattavissa samaan kenttään
 - Koodaus purettava suoritusaikana ohjaussignaaleiksi
 - Jokin kenttä voi ohjata muiden kenttien purkua!
- Kun useita koodauksia yhden sijasta, toteutus helpompaa
 - Useita yksinkertaisia dekodeereita

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 30

Mikrokäskyn koodaus

Toimintojen mukaan (Functional encoding)

- Kukin kenttä kontrolloi tiettyä toimintaa
 - Lataa akkurekisteristä
 - Lataa muistista
 - Lataa ...

Resurssien mukaan (resource encoding)

- Kukin kenttä kontrolloi tiettyä resurssia
 - Lataa akkurekisteristä
 - Talleta akkurekisteriin
 - Lisää akkurekisteriin

(Sta06 Fig 17.11)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 31

Vertical vs. Horizontal Microcode (3)

Seuraavan mikrokäskyn osoite (CAR = CSAR)
Oletus: CAR=CAR+1

(Sta06 Fig 17.12)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 32

Miksi mikro-ohjelmoituna?

vaikka hitaampi suoritus kuin langoitetulla logiikalla

- Suunnittelun helppous ja joustavuus**
 - Muutokset/laajennukset mukaan vaikka loppumetreillä
 - Vanhaan laitteistoon vain kontrollimuistin päivitys
 - Omalla lastullaan vähän vanhemmissa koneissa
 - Tätä varten omat kehitysympäristönsä
- Taaksepäin yhteensopivuus**
 - Vanha käskykanta säilytettävissä vaivatta
 - Lisää vain uudet mikrokäskyt uusille käskyille
- Yleisyys**
 - Yksi laitteisto, useita erilaisia käskykantoja
 - Yksi käskykanta, useita erilaisia organisoitajeja

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 33

Kertauskysymyksiä

- Langoitettu vs. mikro-ohjelmoitu toteutus?
- Kuinka mikrokäskyn osoite määräytyy?
- Mihin tarvitaan kontrollimuistia?
- Horisontaalinen vs. vertikaalinen mikrokäsky?
- Miksi ei mikro-ohjelmoitua?
- IA-64 kontrolli vs. mikro-ohjelmoitu vs. langoitettu kontrolli?

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 34

Tietokoneen rakenne

Pääotsikoita olivat

- Digitaalilogiikka
- Väylät, välimuisti, keskusmuisti
- Virtuaalimuistin osoitemuunnos, TLB
- ALU: kokonais- ja liukulukuaritmetiikka
- Käskykannoista: operaatiot ja osoittaminen
- CPU:n rakenne ja liukuhihna
- Hyppyjen ennustus, datariippuvuudet
- RI SC & superskalaari CPU, nimiriippuvuudet
- IA-64: Explicit Parallel Instruction Computing
- Langoitettu vs. mikro-ohjelmoitu ohjaus

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 35

-- The End --

STI Cell Power processor element

(a) major units and (b) pipeline

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 36

Problem

- Moore's Law will not give us faster processors (any more)
 - But it gives us now more processors on one chip
 - § Multicore CPU
 - § Chip-level multiprocessor (CMP)

Herb Sutter, "A Fundamental Turn Toward Concurrency in SW", Dr. Dobbs's Journal, 2005. ([click](#))

http://www.dj.com/web-development/184406990jaseessionid-BW05DMMMAOT3ZG5GNLDP0KH0CJUNN2JVN?_requestid=1416784

Nämä kalvot esitetty Kerolan Rinnakkaisohjelmointikurssilla

● Clock Speed (MHz)
■ Transistors (000)

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 37

Increasing HW Threads Per Socket
 HT Multi-core Era (Scalar and parallel applications) Many-core Era (Massively parallel applications)

Borkar, Dubey, Kahn, et al. "Platform 2015." Intel White Paper, 2005. ([click](#))
http://download.intel.com/technology/computing/archinnov/platform2015/download/Platform_2015.pdf

Tietokoneen rakenne / 2007 / Liisa Marttinen 7.12.2007 Luento 12 - 38