

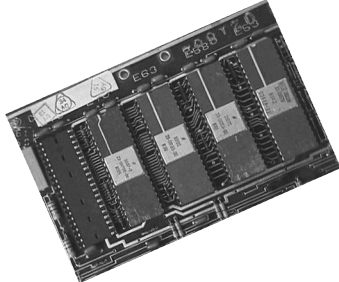
Luento 12

Tietokoneen rakenne

Ohjaus- yksikkö

Ch 16-17 [Sta06]

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



Tietokoneen rakenne / 2006 / Teemu Kerola
5.10.2006
Luento 12 - 1

Mitä ohjaus/kontrolli tarkoittaa?

- n **Arkkitehtuuri määrää CPU:n ulkoisen, ohjelmoijalle (myös KJ) näkyvän toiminnan**
 - u Millainen käskykanta käytössä, mitä käskyt tekevät?
 - u Mikä operaatio, missä operandit?
 - u Miten keskeytykset hoidellaan?
- n **CU määrää kuinka käskyssä kuvatut asiat saadaan tehdyiksi 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 kellopulssin ajoittamina

Tietokoneen rakenne / 2006 / Teemu Kerola
5.10.2006
Luento 12 - 2

Ohjaussignaalit

(Sta06 Fig 16.4)

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

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 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

$$t1: MAR \leftarrow PC$$

$$t2: MBR \leftarrow MEM[MAR]$$

$$PC \leftarrow PC + 1$$

Toteutus siten, että ei tarvita ALUa

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 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

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 5

Käskyn noutosykli

(Sta06 Fig 12.6)

Esim:

t1: MAR • PC
 t2: MAR • MMU(MAR)
 Control Bus • Reserve
 t3: Control Bus • Read
 PC • PC + 1
 t4: MBR • MEM[MAR]
 Control Bus • Release
 t5: IR • 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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 6

Käskesykli

- n **Operandien noutosykli**
 - u Rekistereistä tai muistista
 - u Osoitelaskenta
- n **Suoritusykli**
 - u Suoritus tavallisesti ALUssa
 - u Operandit sisäänmenoihin, ja ohjaus operaatiosta
 - u Tulos ALU:n ulostulosta rekisteriin/muistiin
 - u flags • status
- n **Keskeytyssykli**
 - u ks. luvun 12 esimerkit: Pentium, PowerPC
 - u Mitkä samaan mikrokäskyyn?
 - u 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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 7

Käskesykli tila-automaattina

n ICC: Instruction Cycle Code register

```

    graph TD
        ICC00[ICC = 00] -- "11 (interrupt)" --> Setup[Setup interrupt]
        Setup --> ICC00
        Setup --> ICC11[ICC = 11]
        ICC11 --> ICC00
        
        ICC00 -- "10 (execute)" --> Opc[Opcode]
        Opc --> Exec[Execute instruction]
        Exec --> Int{Interrupt for enabled interrupt?}
        Int -- Yes --> ICC11
        Int -- No --> ICC00
        
        ICC00 -- "11 indirect" --> Read[Read address]
        Read --> ICC10[ICC = 10]
        ICC10 --> ICC00
        
        ICC00 -- "00 (fetch)" --> Fetch[Fetch instruction]
        Fetch --> Ind{Indirect addressing?}
        Ind -- No --> ICC10
        Ind -- Yes --> ICC01[ICC = 01]
        ICC01 --> ICC00
    
```

(Sta06 Fig 16.3)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 8

Käskysykli kontrolli tila-automaattina

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

Tietokoneen rakenne / 2006 / Teemu Kerola

5.10.2006

Luento 12 - 9

Ohjaussignaalien tuottaminen

- n Mikro-operaatio $\bar{\emptyset}$ aktivoitava useita ohjaussignaaleja
- n Esim: Yhden akkurekisterin arkkitehtuuri

Tietokoneen rakenne / 2006 / Teemu Kerola

5.10.2006

Luento 12 - 10



Ohjaussignaalien tuottaminen

Sta06 Fig 16.5

Micro-operations	Timing	Active Control Signals
Fetch:	$t_1: MAR \leftarrow (PC)$	C_2
	$t_2: MBR \leftarrow \text{Memory}$	C_5, C_R
	$PC \leftarrow (PC) + 1$??
	$t_3: IR \leftarrow (MBR)$	C_4
Indirect:	$t_1: MAR \leftarrow (IR(\text{Address}))$	C_8
	$t_2: MBR \leftarrow \text{Memory}$	C_5, C_R
	$t_3: IR(\text{Address}) \leftarrow (MBR(\text{Address}))$	C_4
Interrupt:	$t_1: MBR \leftarrow (PC)$	C_1
	$t_2: MAR \leftarrow \text{Save-address}$??
	$PC \leftarrow \text{Routine-address}$??
	$t_3: \text{Memory} \leftarrow (MBR)$	C_{12}, C_W

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

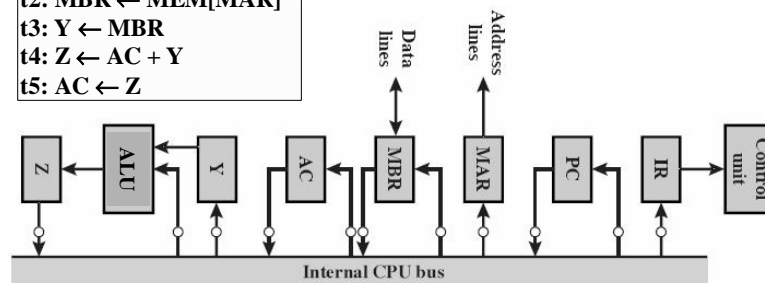
(Sta06 Table 16.1)



Yleinen organisointi

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

ADD I:
t1: MAR \leftarrow IR.address
t2: MBR \leftarrow MEM[MAR]
t3: Y \leftarrow MBR
t4: Z \leftarrow AC + Y
t5: AC \leftarrow Z



(Sta06 Fig 16.6)

Tietokoneen rakenne

Langoitettu ohjaus

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 13

Langoitettu ohjausyksikkö (hardwired)

n Kun tiedossa CU:n sisäänmenot ja ulostulot

- u Toiminnallisuus kuvattavissa boolean logiikalla
- u Ohjausyksiköstä voi muodostaa yhden loogiikkapiirin

n 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

IC: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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 14



Langoitettu ohjausyksikkö

Decoder (4-to-16)

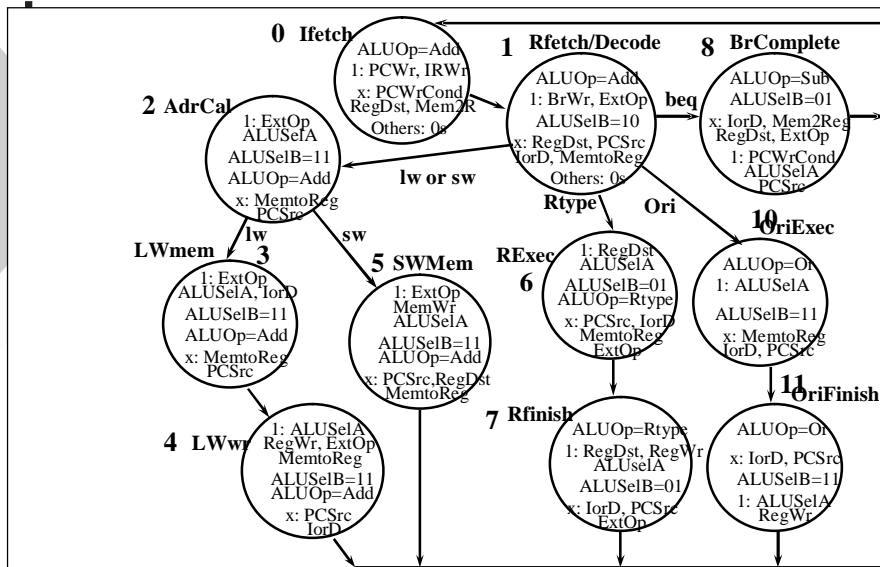
- 4:n bitin käskykoodista yksikäsitteinen ohjaus CU:lle
- Vain yksi signaali kerrallaan aktiivisena

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	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	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)



Äärellinen tila-automaatti



Tilasiirtymät


Next state from current state	Alternatively, prior state & condition
u State 0 -> <u>State1</u>	<u>S4, S5, S7, S8, S9, S11</u> -> State0
u State 1 -> S2, S6, S8, S10	_____ -> State1
u State 2 -> S5 or ...	_____ -> State 2
u State 3 -> S9 or ...	_____ -> State 3
u State 4 -> <u>State 0</u>	_____ -> State 4
u State 5 -> <u>State 0</u>	State2 & op = SW -> State 5
u State 6 -> <u>State 7</u>	_____ -> State 6
u State 7 -> <u>State 0</u>	State 6 -> State 7
u State 8 -> <u>State 0</u>	_____ -> State 8
u State 9 -> <u>State 0</u>	State3 & op = JMP -> State 9
u State 10 -> <u>State 11</u>	_____ -> State 10
u State 11 -> <u>State 0</u>	State 10 -> State 11

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 17

Langoitettu ohjaus

- n Ohjaussignaalien generointi suoraan laitteistolla nopeaa
- n Heikkouksia
 - u CU vaikea suunnitella
 - § Piiristä tulee helposti suuri ja monimutkainen
 - u CU vaikea muuttaa
 - § Suunnittelu ja piirin "minimointi" uusiksi
- n RISC-filosofia helpottaa
 - u Yksinkertainen käskykanta ja toteutus

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 18



Tietokoneen rakenne

Mikro-ohjelmoitu ohjaus

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 19



Mikro-ohjelmoitu ohjaus

- n Idea 1951: Wilkes Microprogrammed Control Execution Engine
 - u Suorituta konekäsky yksi mikrokäsky kerrallaan generoimalla suoritusaikana tarvittavat ohjaussignaalit
 - u Tulkitse mikrokäskyt ohjelmallisesti ohjaussignaaleiksi
- n Mikro-operaatiot kuvattu kontrollimuistissa mikrokäskyinä
 - u Laiteohjelmisto (firmware)
- n Kukin mikrokäsky muodostuu kahdesta osasta
 - u Mitä suoritetaan tulevalla syklillä?
 - § Mikrokäskystä käy ilmi tarvittavat ohjaussignaalit
 - § Toimita ohjaussignaalit piireille
 - u Mikä mikrokäsky tulkitaan tämän jälkeen?
 - § Oletus: seuraavasta kontrollimuistin muistipaikasta

Sta06 Table 16.1

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 20

Mikrokoodi

n Kutakin CPU:n käskesyklin vaihetta vastaa "alirutiini", joka suoritetaan käskesyklin aikana
 n Esim. ROM -muistissa

(Sta06 Fig 17.2)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 21

Horisontaalinen mikrokäsky

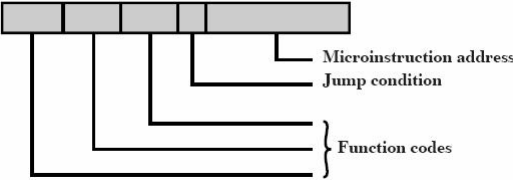
n Kaikki mahdolliset ohjaussignaalit kuvattu bittikarttana jokaisessa mikrokäskyssä
 u Yksi bitti per mahd. ohjaussignaali (1=generoi, 0=älä generoi)
 u Jos paljon erilaisia, mikrokäskystä tulee pitkä
 n Kukin mikrokäsky myös ehdollinen hyppykäsky
 u Mitä statusbittejä tutkittava
 u Seuraavan mikrokäskyn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

(Sta06 Fig 17.1 a)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 22

Vertikaalinen mikrokäskey

- n Ohjaussignaalit koodattu toimintonumeroiksi
- n Dekoodaa takaisin ohjaussignaaleiksi suoritusaikana
- n Lyhyemmät käskyt, mutta dekoodaus vie aikaa
- n Kukin mikrokäskey myös ehdollinen hyppykäskey
 - u Mitä statusbittejä tutkittava
 - u Seuraavan mikrokäskeyn osoite

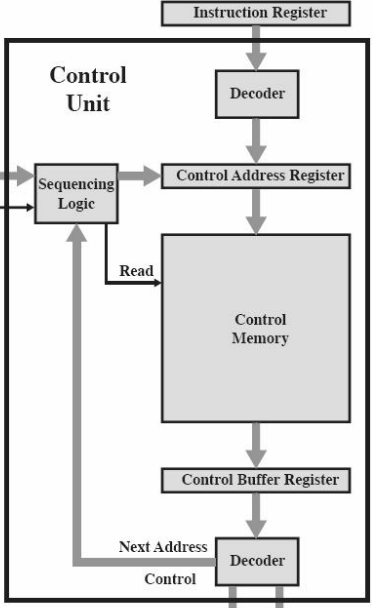


(Sta06 Fig 17.1 b)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 23

Ohjausyksikkö = Execution Engine

- n Control Address Register, CAR
 - u Mikä mikrokäskey seuraavaksi?
 - ~ käskeysoitin, "MiPC"
- n Kontrollimuisti
 - u Mikrokäskeyt
 - u Nouto-suoritus-keskeytys
- n Control Buffer Register, CBR
 - u Rekisteri mikrokäskeyn tulkintaa varten
 - ~ käskeyrekisteri, "MiR"
 - u Tulkitse ja lähetä ohjaussignaalit piireille
- n Sequencing Logic
 - u Seuraavaksi suoritettavan mikrokäskeyn osoite CAR:iin



(Sta06 Fig 17.4)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 24

Mikä mikrokäsky seuraavaksi?

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

(Sta06 Fig 17.6)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 25

Mikä mikrokäsky seuraavaksi?

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

(Sta06 Fig 17.7)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 26

Mikä mikrokäsky seuraavaksi?

c) Vaihteleva formaatti

- u Osa biteistä voidaan tulkita kahdella tavalla
- u 1 b: Käskyssä osoite/ei
- u Vain hyppykäskyssä mukana osoite
- u Hyppykäskyissä ei ole ohjaussignaaleja
- u Jos hyppy, pitääkin suorittaa kaksi käskyä yhden sijasta
 - § Hukkaa aikaa?
 - § Säästää tilaa?

(Sta06 Fig 17.8)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 27

Mikä mikrokäsky seuraavaksi?

d) Osoitteen generointi suoritusaikana

Miten päästä oikean ALU-alirutiinin alkuun?

- u Konekäskyn suoritussignaalit riippuvat operaatiosta

Generoi ensimmäisen mikrokäskyn osoite operaatio-koodista (mapping + combining/adding)

- u Eniten merkitsevät bitit op-koodista
- u Vähiten merkitsevät nollia tai generoitu nykytilan (status) perusteella
- u Esim: IBM 3033 CAR, 13 b:n osoite
 - § Op-koodista 8 bittiä -> alirutiini max 32 käskyä
 - § loput 5 bittiä statusbittejä tutkimalla

(Sta06 Fig 17.9)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 28



Mikä mikrokäsky seuraavaksi?

- n e) Aliohjelmakutsu ja paluu (Residual control)
- n Alirutiinin kutsu mikrotasolla
 - u Ei ympäristön vaihtoa, vain yksi taso
 - u Sisäkkäiset kutsut eivät sallittuja
 - u Esim: LSI -11, 22b:n mikrokäsky
 - § Kontrollimuisti 2048 käskyä, 11 b:n osoitteet
 - § Operaatiokoodi määrää ens. mikrokäskyn osoitteen
 - § Oletus seuraava CAR • CAR+1
 - § Jokaisessa mikrokäskyssä bitti: alirutiinin kutsu/ei Alirutiinin kutsukäsky:
 - talleta paluuosoite paluurekisteriin (vain yksi)
 - hyppää alirutiiniin (osoite mikrokäskyssä)
 - § Paluukäsky: ota osoite paluurekisteristä



Mikrokäskyn koodaus

- n Horisontaalinen? Vertikaalinen?
 - u Horisontaalinen: nopea tulkinta
 - u Vertikaalinen: vähemmän bittejä
- n Usein kompromissi eli sekamuoto
 - u Mikrokäsky jaettu kenttiin, kukin kenttä määrää tiettyjen toimintojen ohjaussignaalit
 - u Keskenään toisensa poissulkevat toiminnot koodattavissa samaan kenttään
 - u Koodaus purettava suoritusajana ohjaussignaaleiksi
- n Kun useita koodauksia yhden sijasta, toteutus helpompaa
 - u Useita yksinkertaisia dekodeereita



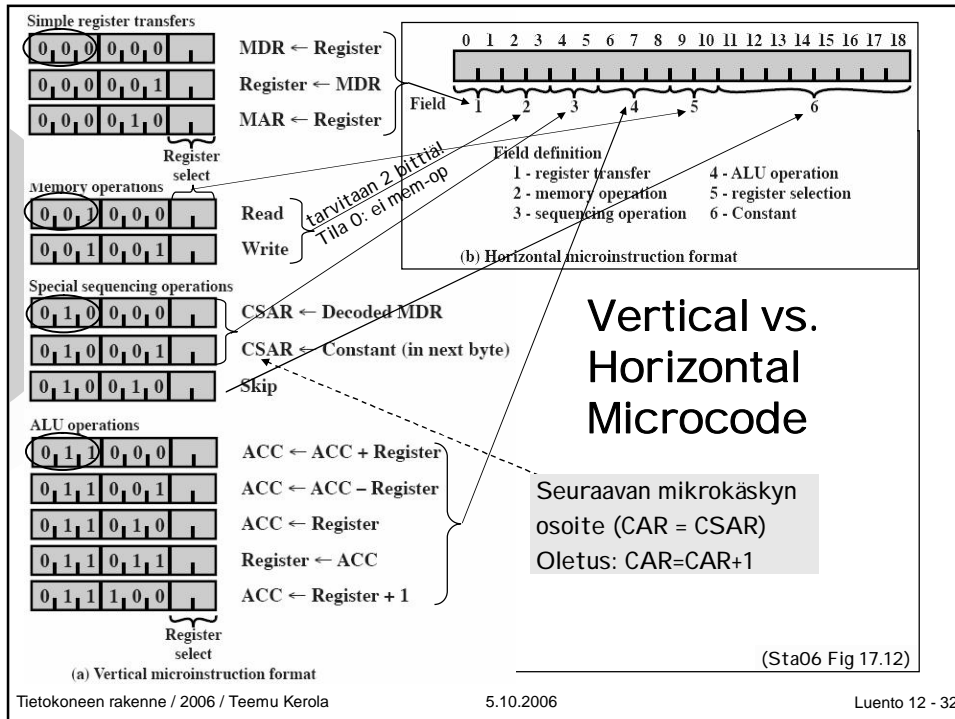
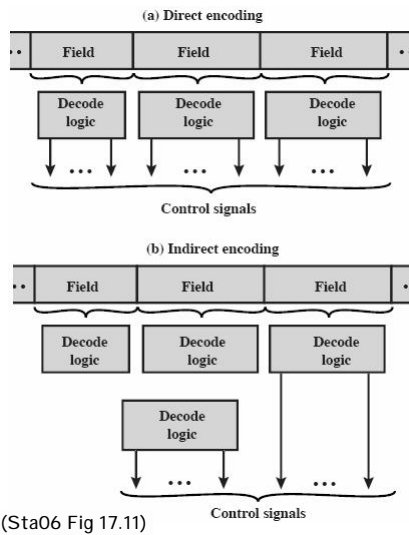
Mikrokäskyn koodaus

Toimintojen mukaan (functional encoding)

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

Resurssien mukaan (resource encoding)

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





Miksi mikro-ohjelmoituna?

vaikka hitaampi suoritus kuin langoitetulla logiikalla

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



Kertauskysymyksiä

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



Tietokoneen rakenne

Pääotsikoita olivat

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



-- The End --

STI Cell Power processor element
(a) major units
and
(b) pipeline

