

Tietokoneen rakenne

Ohjausyksikkö



Ch 16-17 [Sta06]

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

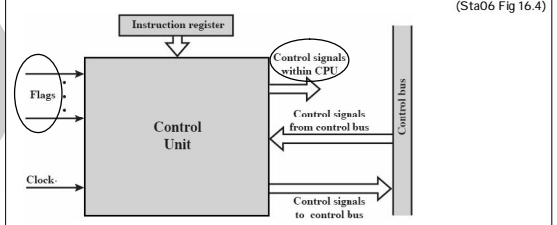
Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 1

Mitä ohjaus/kontrolli tarkoittaa?

- n **Arkkitehtuuri määrää CPU:n ulkosen, 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 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 kellopulssein ajoittamina

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

Ohjaussignaalit



(Sta06 Fig 16.4)

- n **Ohjaus = signaalin / 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

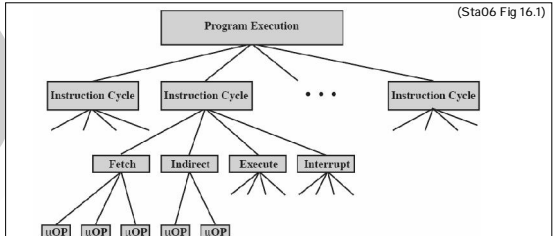
- n **Yksinkertaisia ohjaussignaaleja, jotka aiheuttavat yhden pienen toiminnon (operaation)**
 - u Esim. bitit siirtyvät rekisteristä r1 väylää pitkin edelleen ALU:un
- n **Kellosyklin kesto pisimmän operaation perusteella**
- n **Kunkin syklin aikana useita mikro-operaatioita**
 - u Osa voidaan suorittaa 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

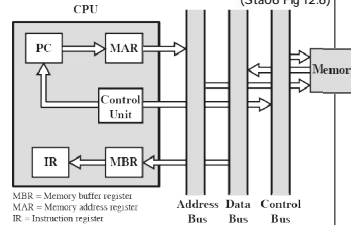


(Sta06 Fig 16.1)

- n Kun mikro-operaatiot kohdistuvat laitteiston eri piireihin, laitteisto voi suorittaa useita samanaikaisesti
- n Ks. luvun 12 käskysykliäesimerkit

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

- Operandien noutosykli**
 - 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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 7

Käskesykli tila-automaattina

ICC: Instruction Cycle Code register

(Sta06 Fig 16.3)

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

Käskesykli kontrolli tila-automaattina

- CU:n toiminnan voi esittää tila-automaattina**
 - Tila: missä käskesyklin vaiheessa CPU menossa
 - Altila: riippuu ajoituksesta, muodostuu ryhmästä mikro-operaatioita, jotka voi suorittaa yhdellä syklillä
- Altilan 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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 9

Ohjaussignaalien tuottaminen

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

(Sta06 Fig 16.5)

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 ← (PC)	C ₂
	t ₂ : MBR ← Memory	C ₅ , C _R
	t ₃ : PC ← (PC) + 1	??
Indirect:	t ₁ : IR ← (MBR)	C ₄
	t ₂ : MAR ← (IR(Address))	C ₈
	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 : MAR ← Save-address PC ← Routine-address	??
	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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 11

Yleinen organisointi

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

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

(Sta06 Fig 16.6)

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

Tietokoneen rakenne

Langoitettu ohjaus

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

Langoitettu ohjausyksikkö (hardwired)

- Kun tiedossa CU:n sisäänmenot ja ulostulot
 - Toiminnallisuus kuvattavissa boolean logiikalla
 - Ohjausyksiköstä 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
 PC = 00 Fetch Cycle
 PC = 01 Indirect Cycle
 PC = 10 Execute Cycle
 PC = 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	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	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	0	0	1	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	1	1	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	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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 15

Äärellinen tila-automaatti

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 16

Tilasiirtymät

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 → State 0	_____ → State4
State 5 → State 0	State2 & op = SW → State5
State 6 → State 7	_____ → State6
State 7 → State 0	State 6 → State7
State 8 → State 0	_____ → State8
State 9 → State 0	State3 & op = JMP → State9
State 10 → State 11	_____ → State10
State 11 → State 0	State 10 → State 11

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 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 / 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

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

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

Mikrokoodi

Kutakin CPU:n käskesyklin vaihetta vastaa "allrutilini", joka suoritetaan käskesyklin aikana

Eslm. ROM -muistissa

(Sta06 Fig 17.2)

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

Horisontaalinen 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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 22

Vertikaalinen mikrokäsky

- Ohjaussignaalit koodattu toimintonumeroiksi
 - Dekoodaa takaisin ohjaussignaaleiksi 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 / 2006 / Teemu Kerola 5.10.2006 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
 - Sequencing Logic**
 - Seuraavaksi suoritettavan mikrokäskyn osoite CAR:iin

(Sta06 Fig 17.4)

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

Mikä mikrokäskey seuraavaksi?

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

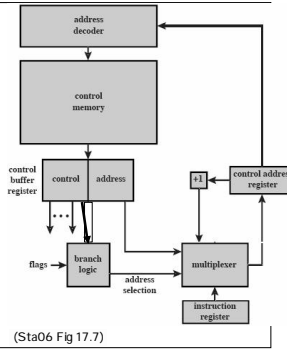


(Sta06 Fig 17.6)

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

Mikä mikrokäskey seuraavaksi?

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

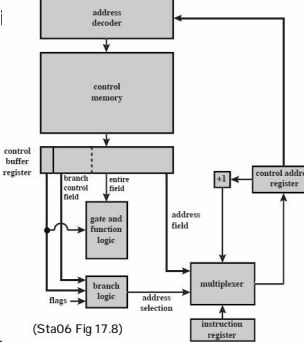


(Sta06 Fig 17.7)

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

Mikä mikrokäskey seuraavaksi?

- c) Vaihteleva formaatti
 - Osa biteistä voidaan tulkita kahdella tavalla
 - 1 b: Käskyssä osoite/ei
 - Vain hyppykäskyssä mukana osoite
 - Hyppykäskyissä ei ole ohjaussignaaleja
 - 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 mikrokäskey seuraavaksi?

- d) Osoitteen generointi suoritusaikana
- Miten päästä oikean ALU-alirutiinin alkuun?
 - Konekäskyn suoritussignaalit riippuvat operaatiosta
- Generoi ensimmäisen mikrokäskyn osoite operaatio-koodista (mapping + combining/adding)
 - Eniten merkitsevät bitit op-koodista
 - Vähiten merkitsevät nolliat tai generoitu nykytilan (status) perusteella
 - Esim: IBM 3033 CAR, 13 b:n osoite
 - Op-koodista 8 bittia -> alirutiini max 32 käskyä
 - lopun 5 bittia statusbittejä tutkimalla



(Sta06 Fig 17.9)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 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 käskyä, 11 b:n osoitteet
 - Operaatiokoodi määrää ens. mikrokäskyn osoitteen
 - Oletus seuraava CAR + CAR+1
 - Jokaisessa mikrokäskyssä bitti: alirutiinin kutsu/ei Alirutiinin kutsukäskey:
 - talleta paluuosoite paluurekisteriin (vain yksi)
 - hyppää alirutiiniin (osoite mikrokäskyssä)
 - Paluukäskey: ota osoite paluurekisteristä

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 29

Mikrokäskyn koodaus

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

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 31

Vertical vs. Horizontal Microcode

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

(Sta06 Fig 17.12)

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 32

Miksi mikro-ohjelmoituna?

vaikka hitaampi suoritus kuin langoitettulla logikalla

- 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 / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 33

Kertauskysymyksiä

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

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 34

Tietokoneen rakenne

Pääotsikoita olivat

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

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 35

-- The End --

STI Cell Power processor element

(a) major units and (b) pipeline

Tietokoneen rakenne / 2006 / Teemu Kerola 5.10.2006 Luento 12 - 36