



Norwegian University of Science and Technology  
Faculty of Information Technology, Mathematics and Electrical Engineering  
The Department of Computer and Information Science

# TDT4160

## DATAMASKINER GRUNNKURS

### EKSAMEN

8. AUGUST, 2007, 09:00–13:00

**Kontakt under eksamen:**

Marius Grannæs 73590670

**Tillatte hjelpemidler:**

D.

Ingen trykte eller håndskrivne hjelpemiddel er tillat.

Enkel godkjent kalkulator er tillate.

**Målform:**

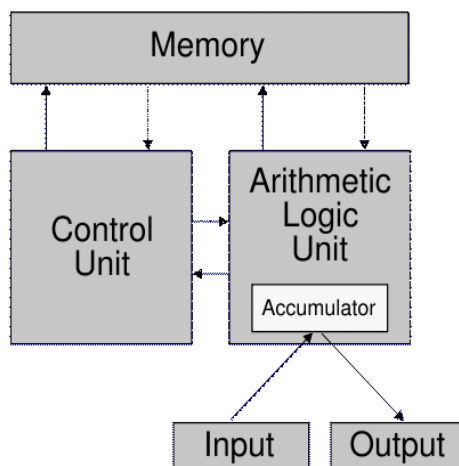
Nynorsk

**Viss ikkje anna er gitt tel deloppgåver likt.**

## OPPGÅVE 1: DATAMASKINER GENERELT (20%)

I Figur 1 er ei maskin med så kalla "Von Neumann architecture" frå rundt 1945 skissert.

- a. Forklar kort kva som var revolusjonerande med denne løysinga.



Figur 1: Von Neumann architecture.

- b. Forklar kort kva oppgåve dei ulike delene i blokkskjemet har.
- c. Kva skil verkemåten og prinsippa som von Neumann skisserte frå moderne prosessorar?

## OPPGÅVE 2: MIKROARKITEKTUR OG MIKROINSTRUKSJONAR (20% (5% PÅ A OG B; 10% PÅ C))

Bruk vedlagte diagram og tabellar for IJVM til å løyse oppgåvene.

- a. Forklar funksjonen til Felta Addr og J (JAM) i MIR (MicroInstruction Register).
- b. IJVM sin styreeining nyttar "microprogram". Ta utgangspunkt i skissa i Figur 4 og forklar kva som er karakteristisk ved denne løysinga. Kva alternativ løysing eksisterar og kva karakteriserar eit mogleg alternativ?

- c. Register H inneheld hex(AA00).  
 Register TOS inneheld hex(5500).  
 Register OPC inneheld hex(A00A).  
 Register SP inneheld hex(7FFF).

Kva ligg i register H, TOS, OPC og SP etter at dei to følgjande mikroinstruksjonane er utført?

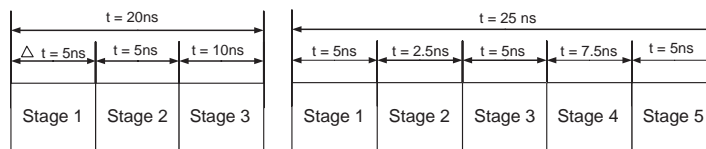
Mikroinstruksjon 1: *ALU: 010100, C: 100000000, Mem: 0, B: 0111*

Mikroinstruksjon 2: *ALU: 111100, C: 001000000, Mem: 0, B: 1000*

Sjå vekk frå felta Addr og J i mikroinstruksjonsformatet.

### OPPGÅVE 3: AUKE YTING (20%)

- a. Kva tyder det at ein prosessor er superskalar?
- b. Kva meinast med omgrepa instruksjonsnivåparallelitet og prosessor-nivåparallelitet?
- c. Anta at ein prosessor har eit samleband med tre trinn som vist i Figur 2(a). Kvar trinn i samlebandet har ein gitt forsinking. I Figur 2(b) er eit fem-trinns samleband vist med gitte trinn forsinkelsar. Kva er maksimal klokkefrekvens for dei to samlebanda?



(a) Samleband med tre steg.

(b) Samleband med fem steg.

Figur 2: Samleband med gitt forsinking.

#### OPPGÅVE 4: SUPERSKALARE PROSESSORAR OG CHIP MULTI-PROSESSORER (CMP) (10%)

a. Eit program som utfører fire operasjonar:

$$1 \quad R3 = R3 * R5$$

$$2 \quad R4 = R3 + 1$$

$$3 \quad R3 = R5 + 1$$

$$4 \quad R7 = R3 * R4$$

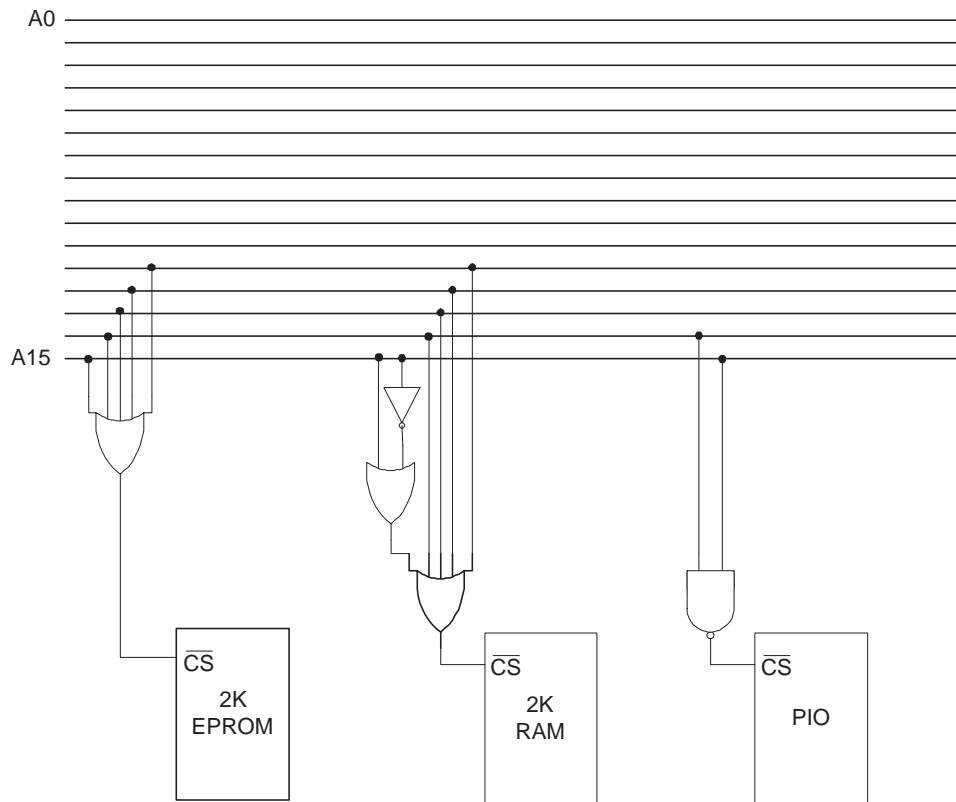
R(nummer) viser til registernummer. Identifiser og forklar dataavhengnadar i programmet skissert over.

b. Ein CMP kan ha heterogene eller homogene kjernar. Kva meinast med dette?

## OPPGÅVE 5: DIVERSE (30% (10% PÅ A; 4% PÅ B — F))

I oppgave b — f skal du ta stilling til om påstandane er korrekte eller ikkje. Korrekte svar gir 4% uttelling, feil svar gir 2% straff. Veit ikkje (ikkje svar) gir inga uttelling (eller straff).

- a. I Figur 3 er EPROM, RAM og PIO kobla til ein felles buss. Alle einingane har aktivt lågt (logisk "0") CS (Chip Select) signal.



Figur 3: Adressedekoding.

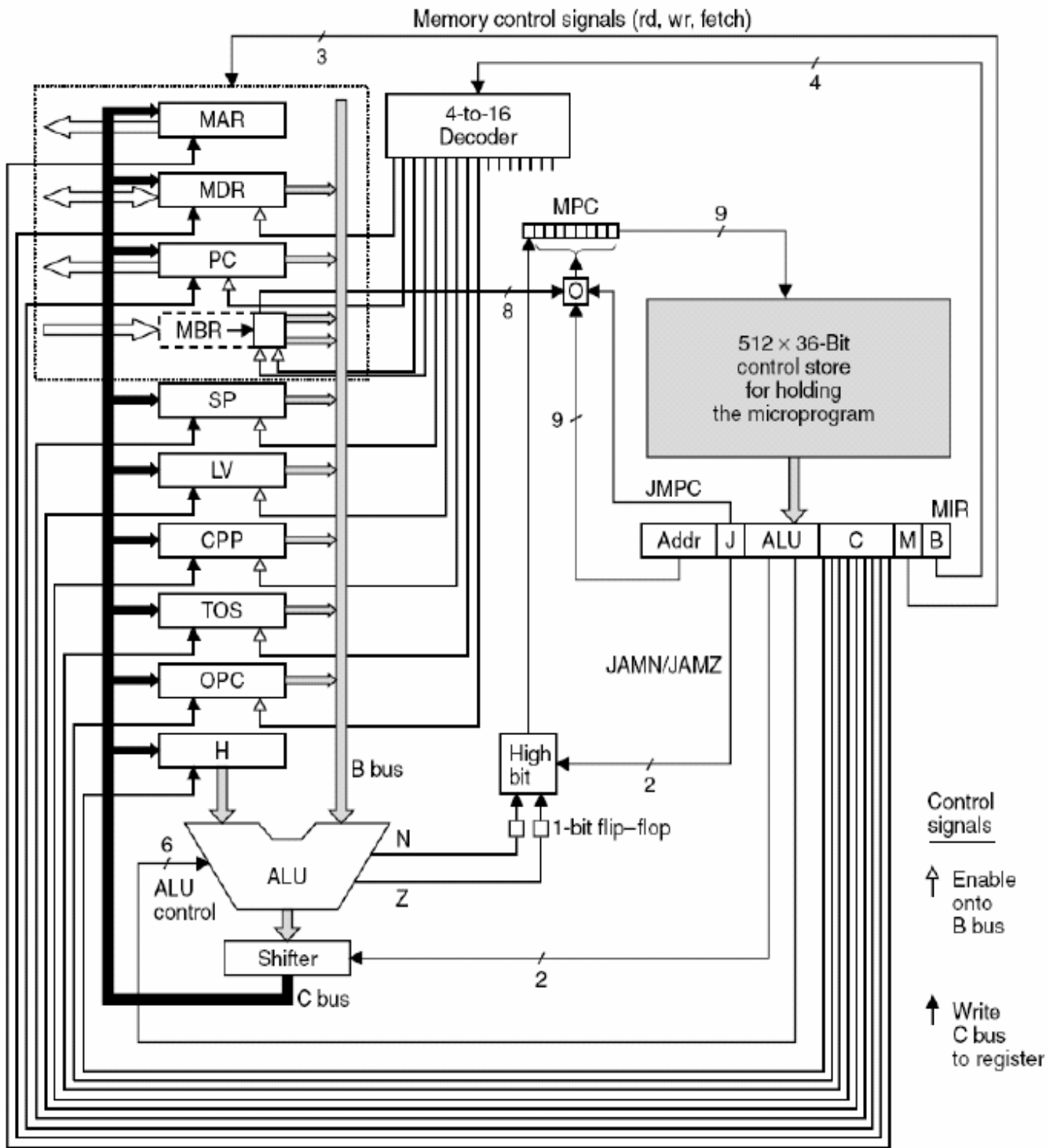
Adresseområde for EPROM, RAM og PIO skal vere:

- EPROM: hex(0000) - hex(07FF)
- RAM: hex(8000) - hex (87FF)
- PIO: hex(C000) - hex(FFFF)

MEN det virkar ikkje som tenkt: Kva er feil i Figur 3 og kva kan gjerast for at verkemåten skal verte korrekt?

- b. JVM har ein multiplikasjonsein.
- c. RISC-prosessorar nyttar ofte "hardwired" styreeining.
- d. I eit "scoreboard" kan ein lese ut antal gonger eit register er brukt som kilde av instruksjonane som er "aktive" (under utføring).
- e. JVM har meir til felles med CISC enn RISC maskiner.
- f. Ein versjon av prosessoren 8051 har multipleksa 8 bit databuss og 16 bit adressebuss. Denne versjonen av 8051 kan adressere 64KB minne.

# IJVM vedlegg



Figur 4: Blokkdiagram (IJVM).





# ANSWER KEY FOR THE EXAM

## OPPGÅVE 1: DATAMASKINER GENERELT (20%)

I Figur 1 er ei maskin med så kalla "Von Neumann architecture" frå rundt 1945 skissert.

- a. Forklar kort kva som var revolusjonerande med denne løysinga.

**Answer:** Viktigast: Programm er lagra i minne på lik linje med data, stored program machine", også utførande og kontroll eining osv.

- b. Forklar kort kva oppgåve dei ulike delene i blokkskjemet har.

**Answer:**

- ALU gir seg sjølv
- Accumulator
- Input Output brukar/omgivelsar
- Control Unit FSM programm og data styrt
- Memory Data og Programm er lagra her

- c. Kva skil verkemåten og prinsippa som von Neumann skisserte frå moderne prosessorar?

**Answer:** Svært lite. Kan eventuelt vere data/programm organisering (Hardward arkitektur)

## OPPGÅVE 2: MIKROARKITEKTUR OG MIKROINSTRUKSJONAR (20% (5% PÅ A OG B; 10% PÅ C))

Bruk vedlagte diagram og tabellar for IJVM til å løyse oppgåvene.

- a. Forklar funksjonen til Felte Addr og J (JAM) i MIR (MicroInstruction Register).

**Answer:**

- Addr: Addressa til mulig neste mikroinstruksjon.
- J: Bestemer korleis neste instruksjon vert velt (betinga hopp).

- b. IJVM sin styreeining nyttar "microprogram". Ta utgangspunkt i skissa i Figur 4 og forklar kva som er karakteristisk ved denne løysinga. Kva alternativ løysing eksisterar og kva karakteriserar eit mogleg alternativ?

**Answer:**

- Microprogram: i prinsippe programmerbar, ein instruksjon bygd opp av mikroinstruksjonar lagra i eit minne. Instruksjonar inneheld kontroll signal og adressa til neste mikroinstruksjon. Ein instruksjon angir adressa til fyrste mikroinstruksjon. OSV. enklare å lage kompliserte lange instruksjonar, ikkje så rask. CISC.

- **Hardwired:** Maskinvare FSM og/eller kombinatorikk, rask, logikk i maskinvare, best for enkle instruksjonar, fast implementasjon. RISC.

- c. Register H inneheld hex(AA00).  
Register TOS inneheld hex(5500).  
Register OPC inneheld hex(A00A).  
Register SP inneheld hex(7FFF).

Kva ligg i register H, TOS, OPC og SP etter at dei to følgjande mikroinstruksjonane er utført?

Mikroinstruksjon 1: ALU: 010100, C: 100000000, Mem: 0, B: 0111

Mikroinstruksjon 2: ALU: 111100, C: 001000000, Mem: 0, B: 1000

Sjå vekk frå felte Addr og J i mikroinstruksjonsformatet.

**Answer:** 1: Laste TOS inn i H ALU: 010100, C: 100000000, Mem: 0, B: 0111

2 addere H + OPC skriv til TOS ALU: 111100 (A+B) C: 001000000 (TOS) Mem: 000 (ingen mem opprasjon) B: 1000 (8 OPC)

- H: hex(5500)
- TOS: hex(F50A)
- OPC: hex(A00A)
- SP: hex(7FFF)

### OPPGÅVE 3: AUKE YTING (20%)

- (i) Kva tyder det at ein prosessor er superskalar?

**Answer:** Duplisering av CPU komponentar

- (ii) Kva meinast med omgrepa instruksjonsnivåparallelitet og prosessornivåparallelitet?

**Answer:** Instruksjonsnivåparallelitet: ein prosessor utfører fleire instruksjonar samstundes. Prosessornivåparallelitet: fleire prosessorar utfører instruksjonar samstundes.

- (iii) Anta at ein prosessor har eit samleband med tre trinn som vist i Figur 2(a). Kvant trinn i samlebandet har ein gitt forsinking. I Figur 2(b) er eit fem-trinns samleband vist med gitte trinn forsinkelsar. Kva er maksimal klokkefrekvens for dei to samlebanda?

**Answer:** 3 trinn: 1/10ns, 5 trinn: 1/7.5ns.

## OPPGÅVE 4: SUPERSKALARE PROSESSORAR OG CHIP MULTIPROSESSORER (CMP) (10%)

(i) Eit program som utfører fire operasjonar:

$$1 \quad R3 = R3 * R5$$

$$2 \quad R4 = R3 + 1$$

$$3 \quad R3 = R5 + 1$$

$$4 \quad R7 = R3 * R4$$

R(nummer) viser til registernummer. Identifiser og forklar data-avhengnader i programmet skissert over.

**Answer:** Avhengighet:

WAR: (Antiavhengighet) Skriv-etter-les, I2 må lese R3 før verdien blir overskrevet av I3

Problem hvis ein instruksjon skriv resultat før ein tidligare får lest operandar

Ein CMP kan ha heterogene eller homogene kjernar. Kva meinast med dette?

(ii) **Answer:** Ulike eller like kjerner på ein chip.

## OPPGÅVE 5: DIVERSE (30% (10% PÅ A; 4% PÅ B — F))

I oppgave b — f skal du ta stilling til om påstandane er korrekte eller ikkje. Korrekte svar gir 4% utteljing, feil svar gir 2% straff. Veit ikkje (ikkje svar) gir inga utteljing (eller straff).

(i) I Figur 3 er EPROM, RAM og PIO kobla til ein felles buss. Alle einingane har aktivt lågt (logisk "0") CS (Chip Select) signal.

Adresseområde for EPROM, RAM og PIO skal vere:

- EPROM: hex(0000) - hex(07FF)
- RAM: hex(8000) - hex (87FF)
- PIO: hex(C000) - hex(FFFF)

MEN det virkar ikkje som tenkt: Kva er feil i Figur 3 og kva kan gjerast for at verkemåten skal verte korrekt?

**Answer:** RAM og ROM har overlappende adresseområde. Fjern 2-inngangs OR krets i RAM adr-dekoding og bruk kun invertert A15. Figur ?? viser korleis rett adressedekoding kan sjå ut.

- (ii) IJVM har ein multiplikasjonseining.  
**Answer:** Nei, dette kan dei finne ut ved å sjå på ALU-tabellen i Figur 6
- (iii) RISC-prosessorar nyttar ofte "hardwired" styreeining.  
**Answer:** Ja, i den forenkla verden her er dette svaret. RISC har like lange enkle instruksjonar som gjer det mogleg å nytte hardwired styreeining.
- (iv) I eit "scoreboard" kan ein lese ut antal gonger eit register er brukt som kilde av instruksjonane som er "aktive" (under utføring).  
**Answer:** Ja, ein del av felta i scoreboard skjema som er brukt som eksempel i buka og forelest.
- (v) IJVM har meir til felles med CISC enn RISC maskiner.  
**Answer:** Ja, forskjellig lengde moglegheit, instruksjonar, med forskjellig antal klokke taks, ikkje typisk RISC adresering.
- (vi) Ein versjon av prosessoren 8051 har multipleksa 8 bit databuss og 16 bit adressebuss. Denne versjonen av 8051 kan adressere 64KB minne.  
**Answer:** Ja, 16 bit adr.buss gir 64kB minne (1 byte i kvar adr).

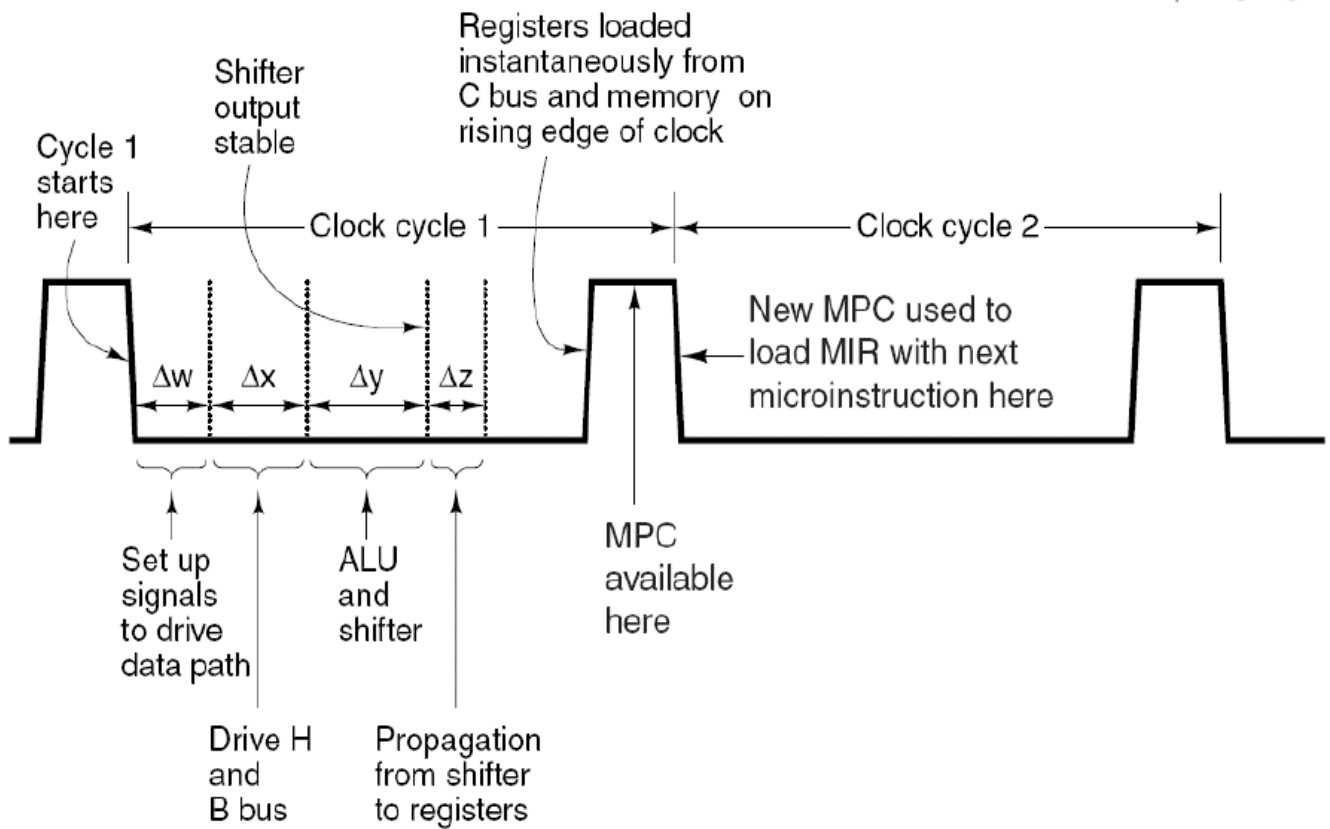


# IJVM vedlegg

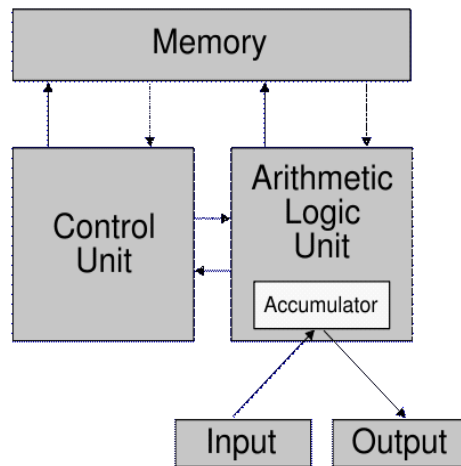
$F_0$	$F_1$	ENA	ENB	INVA	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	$\bar{A}$
1	0	1	1	0	0	$\bar{B}$
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B - A
1	1	0	1	1	0	B - 1
1	1	1	0	1	1	-A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	-1

SLR1	SLL8	Function
0	0	No shift
0	1	Shift 8 bit left
1	0	Shift 1 bit right

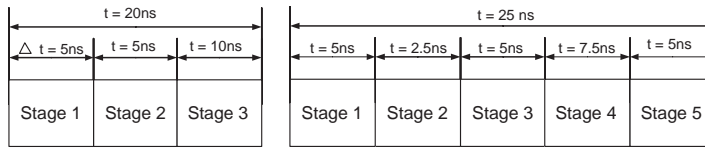
Figur 6: Funksjonstabell for ALU (IJVM).



Figur 7: Timingdiagram (IJVM).



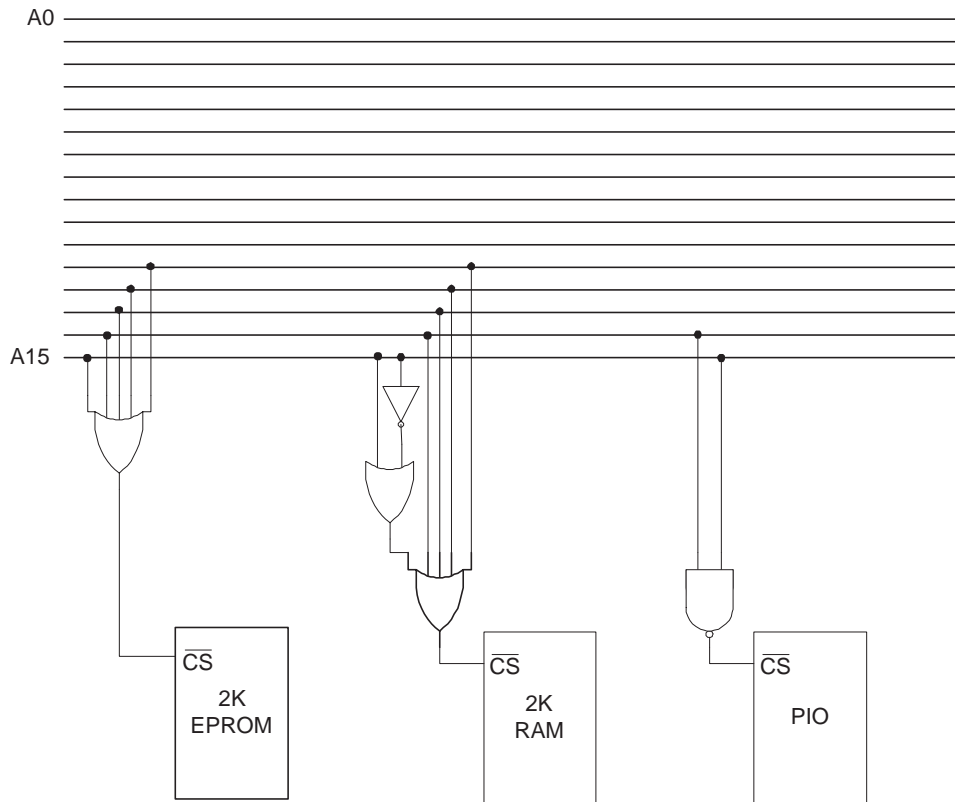
Figur 8: Von Neumann architecture.



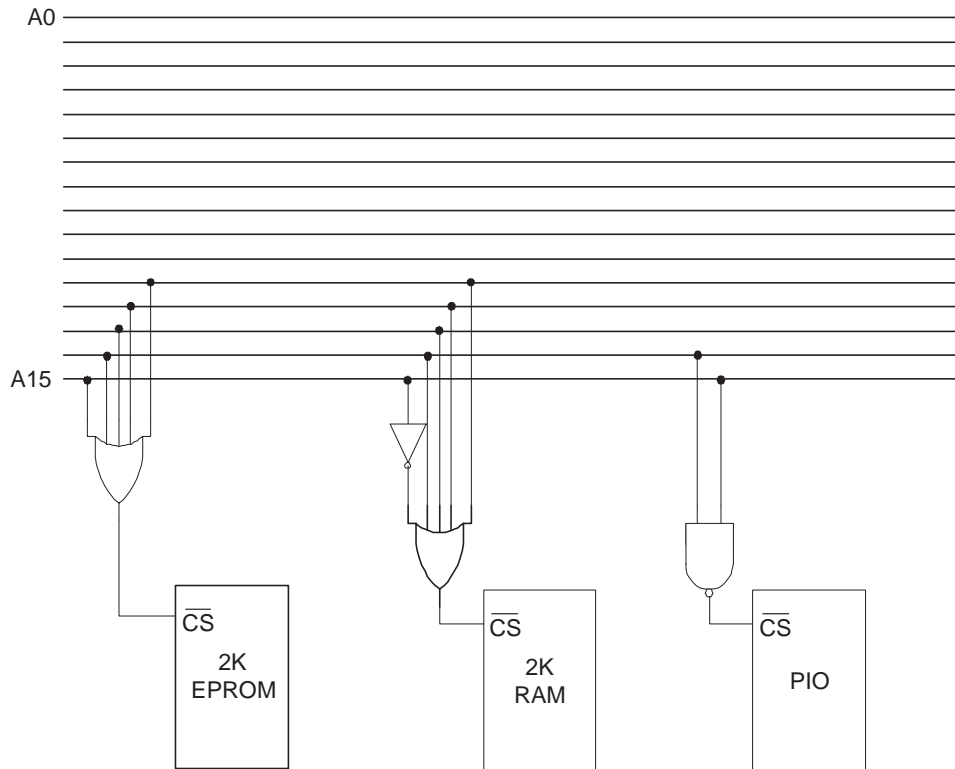
(a) Samleband med tre steg.

(b) Samleband med fem steg.

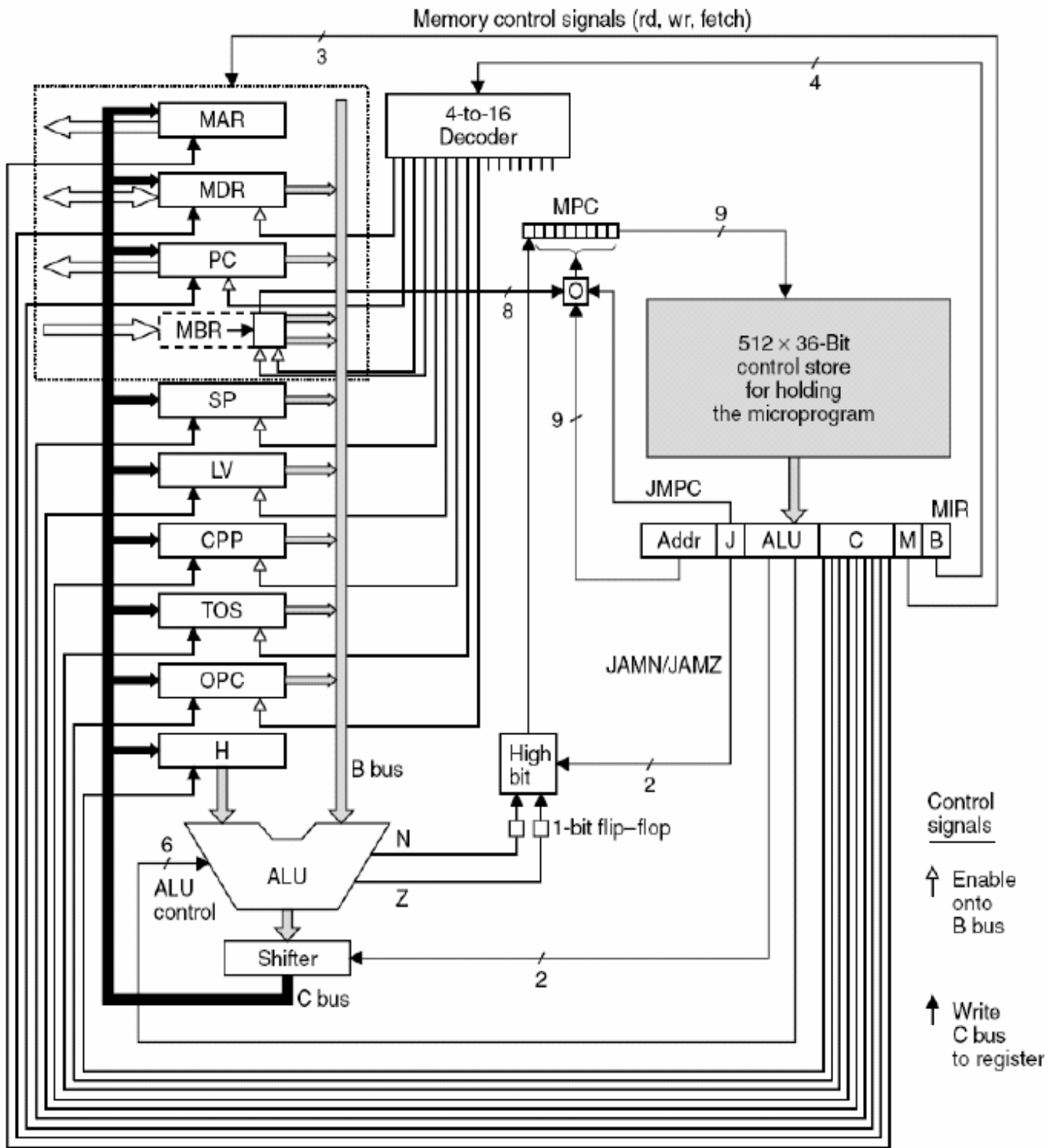
Figur 9: Samleband med gitt forsinkning.



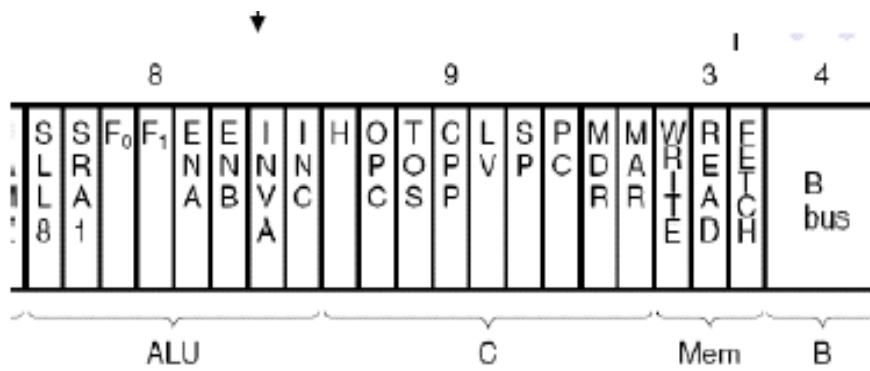
Figur 10: Adressedekoding.



Figur 11: Adressedekoding slik som det skal vere utan overlapping. No er adressedekodingen i samsvar med ønsket område for dei forskjellige einingane.



Figur 12: Blokkdiagram (IJVM).



**B bus registers**

- |          |           |
|----------|-----------|
| 0 = MDR  | 5 = LV    |
| 1 = PC   | 6 = CPP   |
| 2 = MBR  | 7 = TOS   |
| 3 = MBRU | 8 = OPC   |
| 4 = SP   | 9-15 none |

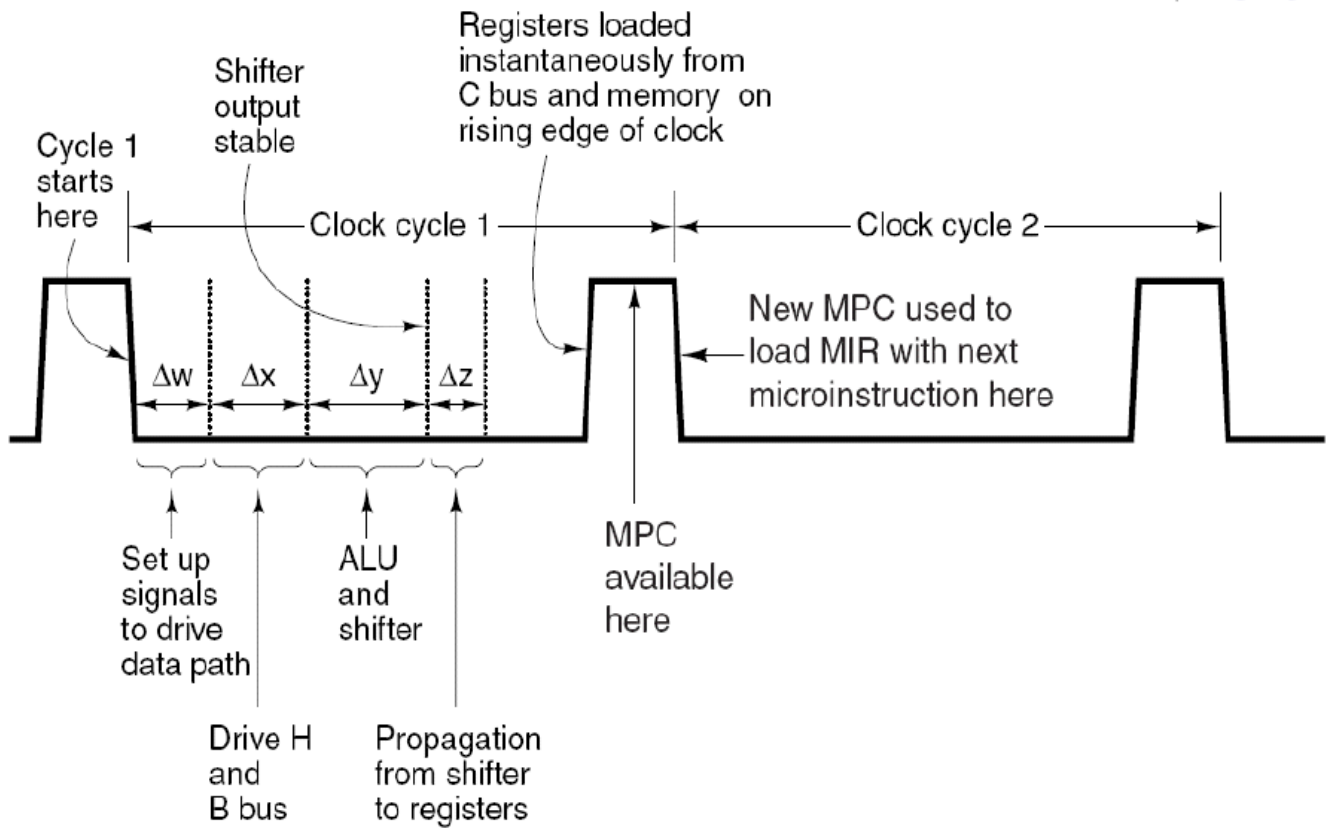
Figur 13: Mikroinstruksjonsformat (IJVM).

$F_0$	$F_1$	ENA	ENB	INVA	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	$\bar{A}$
1	0	1	1	0	0	$\bar{B}$
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B - A
1	1	0	1	1	0	B - 1
1	1	1	0	1	1	-A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	-1

SLR1	SLL8	Function
0	0	No shift
0	1	Shift 8 bit left
1	0	Shift 1 bit right

Figur 14: Funksjonstabell for ALU (IJVM).





Figur 15: Timingdiagram (IJVM).