



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

# TDT4160 OG IT2201

## DATAMASKINER GRUNNKURS

### EKSAMEN

20. DESEMBER, 2006, 09:00–13:00

**Kontakt under eksamen:**

Gunnar Tufte 73590356

**Tillatte hjelpemidler:**

D.

Ingen trykte eller håndskrevne hjelpemidler tillatt.

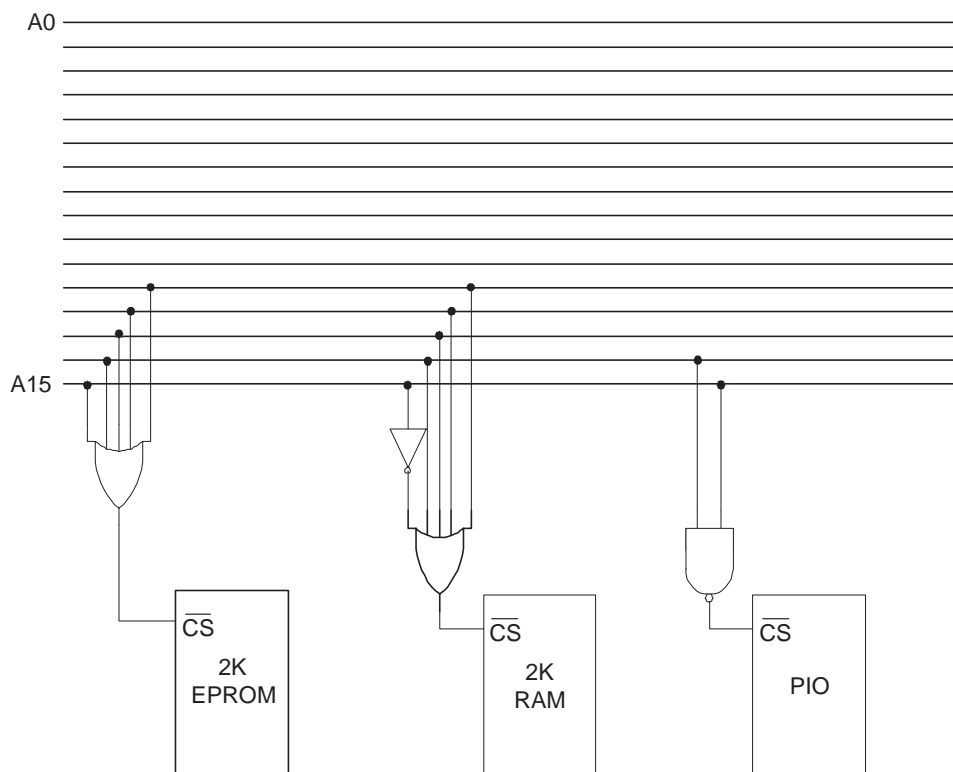
Bestemt, enkel kalkulator tillatt.

**Målform:**

Bokmål

## OPPGAVE 1: BUSS OG ADRESSEDEKODING (20%)

- a. I Figur 1 er EPROM, RAM og PIO koblet til en felles buss. Finn adresseområde for EPROM, RAM og PIO. Alle enhetene har aktivt lavt (logisk "0") CS (Chip Select) signal.



Figur 1: Adressedekoding.

- b. Anta at PIO-kretsen kun har mulighet til å adressere fire enheter. Hvordan kan addressedekodingen endres for å gi maksimal utvidelsesmulighet for RAM-området?

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

Bruk vedlagte diagram og tabeller for IJVM til å løse oppgavene.

- a. Forklar funksjonen til registeret "PC".
- b. Lag mikroinstruksjon(er) for følgende IJVM-operasjon: last register "H" med innholdet i register "TOS".

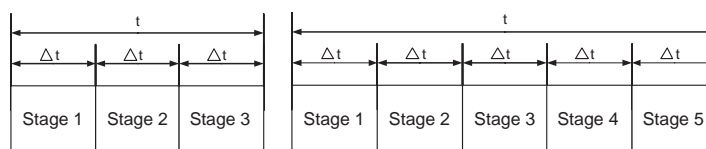
Se bort fra Addr- og J-feltene i mikroinstruksjonsformatet. Angi korrekte bit for ALU, C, Mem og B gitt i Figur 4.

- c. Lag mikroinstruksjon(er) for følgende IJVM-operasjon:  
 $TOS = TOS + OPC$ .

Se bort fra Addr- og J-feltene i mikroinstruksjonsformatet. Angi korrekte bit for ALU, C, Mem og B gitt i Figur 4.

### OPPGAVE 3: FORBEDRE YTELSEN (20%)

- a. Hvilken av disse alternativene er **ikke** en grunn til at man har begynt å utvikle en-brikke multiprosessorer (CMP)?
- 1 Tilgjengelig ILP (Instruction Level Parallelism) i instruksjonsstrømmen er begrenset
  - 2 En-kjerneprosessorer har ofte høyt effektforbruk.
  - 3 Store cacher på brikken gjør at vi ikke har plass til større kjerner.
  - 4 En moderne en-kjerneprosessor er svært kompleks og derfor kostbar å utvikle.
- b. Anta at en prosessor har et samlebånd. Samlebåndet bruker tiden  $t$  på å utføre en instruksjon. Hvert trinn har en forsinkelse på  $\Delta t$ . Samlebåndet med tre trinn er illustrert i Figur 2(a). For å få høyere ytelse erstattes samlebåndet med et femtrinns samlebånd, vist i Figur 2(b). Hva skjer typisk med  $t$  og  $\Delta t$  dersom tretrinns samlebåndet erstattes med et samlebånd med fem trinn som kan øke ytelsen til prosessoren? Begrunn svaret.



(a) Tretrinns samlebånd.

(b) Femtrinns samlebånd

Figur 2: Samlebånd med angitt tidsforsinkelse.

## OPPGAVE 4: SUPERSKALARE PROSESSORER - SCOREBOARD (20%)

Et program som utfører fire operasjonar:

1  $R1 = R2 + R3$

2  $R4 = R1 * R1$

3  $R1 = R5 * R6$

4  $R7 = R0 + R0$

a. Marker følgende typer dataavhengigheter i programmet ovenfor:

- *Sanne dataavhengigheter (RAW-avhengighet).*
- *Utavhengigheter (WAW-avhengighet).*
- *Antiavhengigheter (WAR-avhengighet).*

b. Anta at programmet ovenfor skal utføres på en superskalar prosessor med åtte synlige registre hvor instruksjoner startes og avsluttes i rekkefølge og maksimalt to instruksjoner kan startes per klokkesykel. Anta at for instruksjoner som dekodes i sykel  $n$ , så startes utførselen i sykel  $n+1$ . Anta at for addisjonsoperasjoner (+) som startes i sykel  $n$ , skrives resultatet tilbake til registeret i slutten av sykel  $n+2$ , og for multiplikasjonsinstruksjoner (\*) som startes i sykel  $n$ , skrives resultatet tilbake til registeret i slutten av sykel  $n+3$ .

Fullfør det påbegynte scoreboardet i Tabell 1 for alle fire instruksjoner i programmet ovenfor.

Tabell 1: Scoreboard.

Cy	#	Decoded	Iss	Ret	Reg. being read	Reg. being written
					0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
1	1	$R1=R2+R3$	1			
	2	$R4=R1*R1$	-			

## OPPGAVE 5: DIVERSE (20%)

Ta stilling til om påstandene er korrekte eller ikke. Korrekt svar gir 2% uttelling, feil svar gir tilsvarende straff og vet ikke (ikke svar) gir ingen uttelling.

- a. Moore sin lov er oppkalt etter Gordon Moore. Dette er en lov som sier noe om fysiske egenskaper i halvledermaterialet.
- b. hex(0FF0) i "Big endian" representerar et større tall enn hex(0FF0) i "Little endian".
- c. RISC-prosessorer er raske fordi de har få kompliserte instruksjoner.
- d. Cache (hurtigbuffer) er oftest implementert med SRAM-celler.
- e. IJVM er en typisk RISC-arkitektur.
- f. Aritmetiske uttrykk på omvent polsk notasjon (Reverse Polish Notation) er enkelt å utføre på stakkmaskiner.
- g. IJVM-arkitekturen kan maksimalt ha et instruksjonsett bestående av 256 forskjellige instruksjoner..
- h. "Throughput" er et ytelsesmål der man måler hvor mye arbeid man får gjort per tidsenhet.
- i. "Throughput" er et ytelsesmål som måler hvor mange instruksjoner en prosessor gjør ferdig per klokkesykel.
- j. DRAM er raskere enn SRAM (kortere aksesstid).

# IJVM vedlegg







# ANSWER KEY FOR THE EXAM

## OPPGAVE 1: BUSS OG ADRESSEDEKODING (20%)

- a. I Figur 1 er EPROM, RAM og PIO koblet til en felles buss. Finn adresseområde for EPROM, RAM og PIO. Alle enhetene har aktivt lavt (logisk "0") CS (Chip Select) signal.

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

- b. Anta at PIO-kretsen kun har mulighet til å adressere fire enheter. Hvordan kan adressedekodingen endres for å gi maksimal utvidelsesmulighet for RAM-området?

**Answer:** Endre adressedekodingen til PIO til kunn adressere området hex(FFFC) - hex(FFFF).

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

Bruk vedlagte diagram og tabeller for IJVM til å løse oppgavene.

- a. Forklar funksjonen til registeret "PC".

**Answer:** PC Program Counter inneholdt minne adressa til instruksjon som skal utførast.

- b. Lag mikroinstruksjon(er) for følgende IJVM-operasjon: last register "H" med innholdet i register "TOS".

Se bort fra Addr- og J-feltene i mikroinstruksjonsformatet. Angi korrekte bit for ALU, C, Mem og B gitt i Figur 4.

**Answer:** ALU: 010100 (B) C: 10000000 (H) Mem: 000 (ingen mem opprasjon) B: 0111 (7 TOS)

- c. Lag mikroinstruksjon(er) for følgende IJVM-operasjon:  
TOS = TOS + OPC.

Se bort fra Addr- og J-feltene i mikroinstruksjonsformatet. Angi korrekte bit for ALU, C, Mem og B gitt i Figur 4.

**Answer:** Må ha to mikroinstruksjonar for å klare å utføre TOS = TOS + OPC.

1: Laste TOS inn i H ALU: 010100 (B) C: 10000000 (H) Mem: 000 (ingen mem opprasjon) B: 0111 (7 TOS)

2 addere H + OPC skriv til TOS ALU: 111100 (A+B) C: 001000000  
(TOS) Mem: 000 (ingen mem opprasjon) B: 1000 (8 OPC)  
eller OPC kan lastast H i fyrste opperasjon og TOS + H i andre  
operasjon.

### OPPGAVE 3: FORBEDRE YTELSEN (20%)

a. Hvilken av disse alternativene er **ikke** en grunn til at man har begynt å utvikle en-brikke multiprosessorer (CMP)?

- 1 Tilgjengelig ILP (Instruction Level Parallelism) i instruksjonsstrømmen er begrenset
- 2 En-kjerneprosessorer har ofte høyt effektforbruk.
- 3 Store cacher på brikken gjør at vi ikke har plass til større kjerner.
- 4 En moderne en-kjerneprosessor er svært kompleks og derfor kostbar å utvikle.

**Answer:** 3

b. Anta at en prosessor har et samlebånd. Samlebåndet bruker tiden  $t$  på å utføre en instruksjon. Hvert trinn har en forsinkelse på  $\Delta t$ . Samlebåndet med tre trinn er illustrert i Figur 2(a). For å få høyere ytelse erstattes samlebåndet med et femtrinns samlebånd, vist i Figur 2(b).

Hva skjer typisk med  $t$  og  $\Delta t$  dersom tretrinns samlebåndet erstattes med et samlebånd med fem trinn som kan øke ytelsen til prosessoren? Begrunn svaret.

**Answer:**  $\Delta t$  minkar sidan kvart trinn kan vere enklare (mindre tidsforsinkelse). Total tidsforbruk  $t$  samla for trinna må ikkje vere mindre slik. Klokkefrekvensen for prosessoren kan aukast.

## OPPGAVE 4: SUPERSKALARE PROSESSORER - SCOREBOARD (20%)

Et program som utfører fire operasjoner:

1  $R1 = R2 + R3$

2  $R4 = R1 * R1$

3  $R1 = R5 * R6$

4  $R7 = R0 + R0$

a. Marker følgende typer dataavhengigheter i programmet ovenfor:

- *Sanne dataavhengigheter (RAW-avhengighet).*
- *Utavhengigheter (WAW-avhengighet).*
- *Antiavhengigheter (WAR-avhengighet).*

**Answer:** Avhengighet:

RAW: I2 leser R1, I1 skriver R1

WAR: I2 leser R1, I3 skriver R1

WAW: I1 og I3 skriver R1

b. Anta at programmet ovenfor skal utføres på en superskalar prosessor med åtte synlige registre hvor instruksjoner startes og avsluttes i rekkefølge og maksimalt to instruksjoner kan startes per klokkesykel. Anta at for instruksjoner som dekodes i sykel  $n$ , så startes utførselen i sykel  $n+1$ . Anta at for addisjonsoperasjoner (+) som startes i sykel  $n$ , skrives resultatet tilbake til registeret i slutten av sykel  $n+2$ , og for multiplikasjonsinstruksjoner (\*) som startes i sykel  $n$ , skrives resultatet tilbake til registeret i slutten av sykel  $n+3$ .

Fullfør det påbegynte scoreboardet i Tabell 1 for alle fire instruksjoner i programmet ovenfor.

Tabell 2: Scoreboard.

Cy	#	Decoded	Iss	Ret	Reg. being read	Reg. being written
					0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
1	1	$R1=R2+R3$	1			
	2	$R4=R1*R1$	-			

**Answer:** Instr. (4) kan ikke avsluttes før i syklus 11 pga. kravet om i-rekkefølge-avslutning (eng. in-order-completion).

Tabell 3: Scoreboard.

Cy	#	Decoded	Iss	Ret	Reg. being read	Reg. being written
					0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
1	1	R1=R2+R3	1		XX11XXXX	X1XXXXXX
	2	R4=R1*R1	-		XX11XXXX	X1XXXXXX
2					XX11XXXX	X1XXXXXX
3				1	XXXXXXXX	XXXXXXXX
4	3	R1=R5*R6	2		X1XXXXXX	XXXX1XXX
			-		X1XXXXXX	XXXX1XXX
5					X1XXXXXX	XXXX1XXX
6					X1XXXXXX	XXXX1XXX
7				2	XXXXXXXX	XXXXXXXX
8	4	R7=R5+R0	3		XXXXX11X	X1XXXXXX
			4		1XXXX21X	X1XXXXX1
9					1XXXX21X	X1XXXXX1
10					1XXXX21X	X1XXXXX1
11			3		1XXXX1XX	XXXXXXXX1
			4		XXXXXXXX	XXXXXXXX

### OPPGAVE 5: DIVERSE (20%)

Ta stilling til om påstandene er korrekte eller ikke. Korrekt svar gir 2% uttelling, feil svar gir tilsvarende straff og vet ikke (ikke svar) gir ingen uttelling.

- a. Moore sin lov er oppkalt etter Gordon Moore. Dette er en lov som sier noe om fysiske egenskaper i halvledermaterialet.

**Answer:** Nei

- b. hex(0FF0) i "Big endian" representerar et større tall enn hex(0FF0) i "Little endian".

**Answer:** Nei

- c. RISC-prosessorer er raske fordi de har få kompliserte instruksjoner.

**Answer:** Nei

- d. Cache (hurtigbuffer) er oftest implementert med SRAM-celler.

**Answer:** Ja

- e. IJVM er en typisk RISC-arkitektur.

**Answer:** Nei

f. Aritmetiske uttrykk på omvent polsk notasjon (Reverse Polish Notation) er enkelt å utføre på stakkmaskiner.

**Answer:** Ja

g. JVM-arkitekturen kan maksimalt ha et instruksjonsett bestående av 256 forskjellige instruksjoner..

**Answer:** Ja

h. "Throughput" er et ytelsesmål der man måler hvor mye arbeid man får gjort per tidsenhet.

**Answer:** Ja

i. "Throughput" er et ytelsesmål som måler hvor mange instruksjoner en prosessor gjør ferdig per klokkesykel.

**Answer:** Nei

j. DRAM er raskere enn SRAM (kortere aksesstid).

**Answer:** Nei

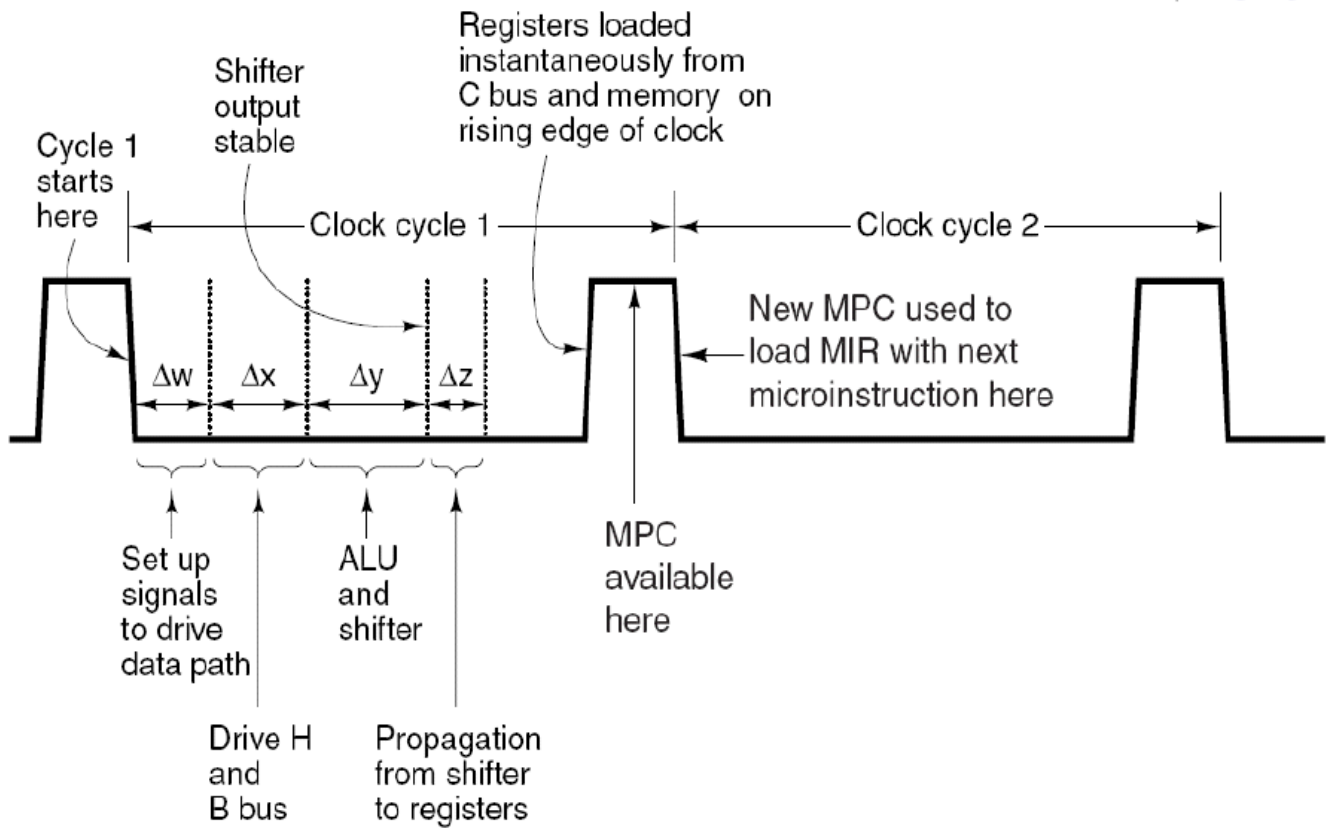
# IJVM vedlegg

$F_0$	$F_1$	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

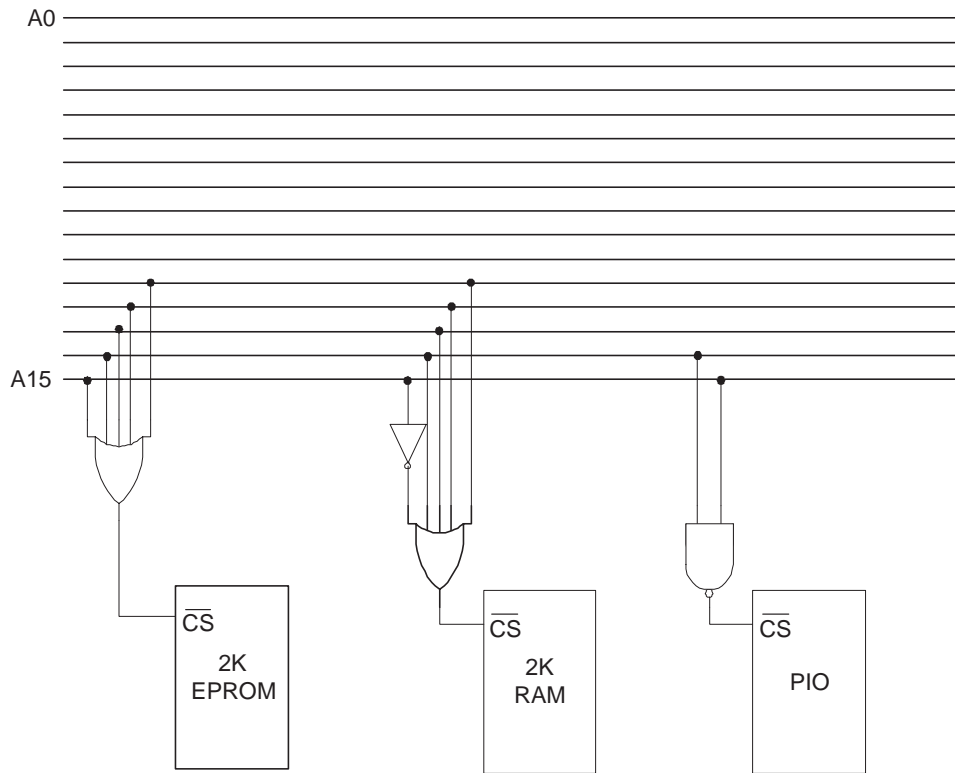
SLR1	SLL8	Function
0	0	No shift
0	1	Shift 8 bit left
1	0	Shift 1 bit right

Figur 5: Funksjonstabell for ALU (IJVM).

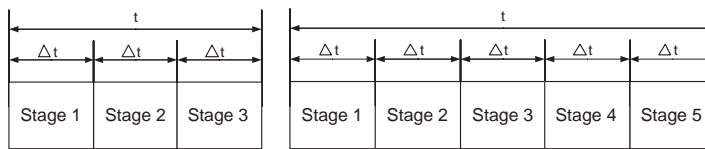




Figur 6: Timingdiagram (IJVM).



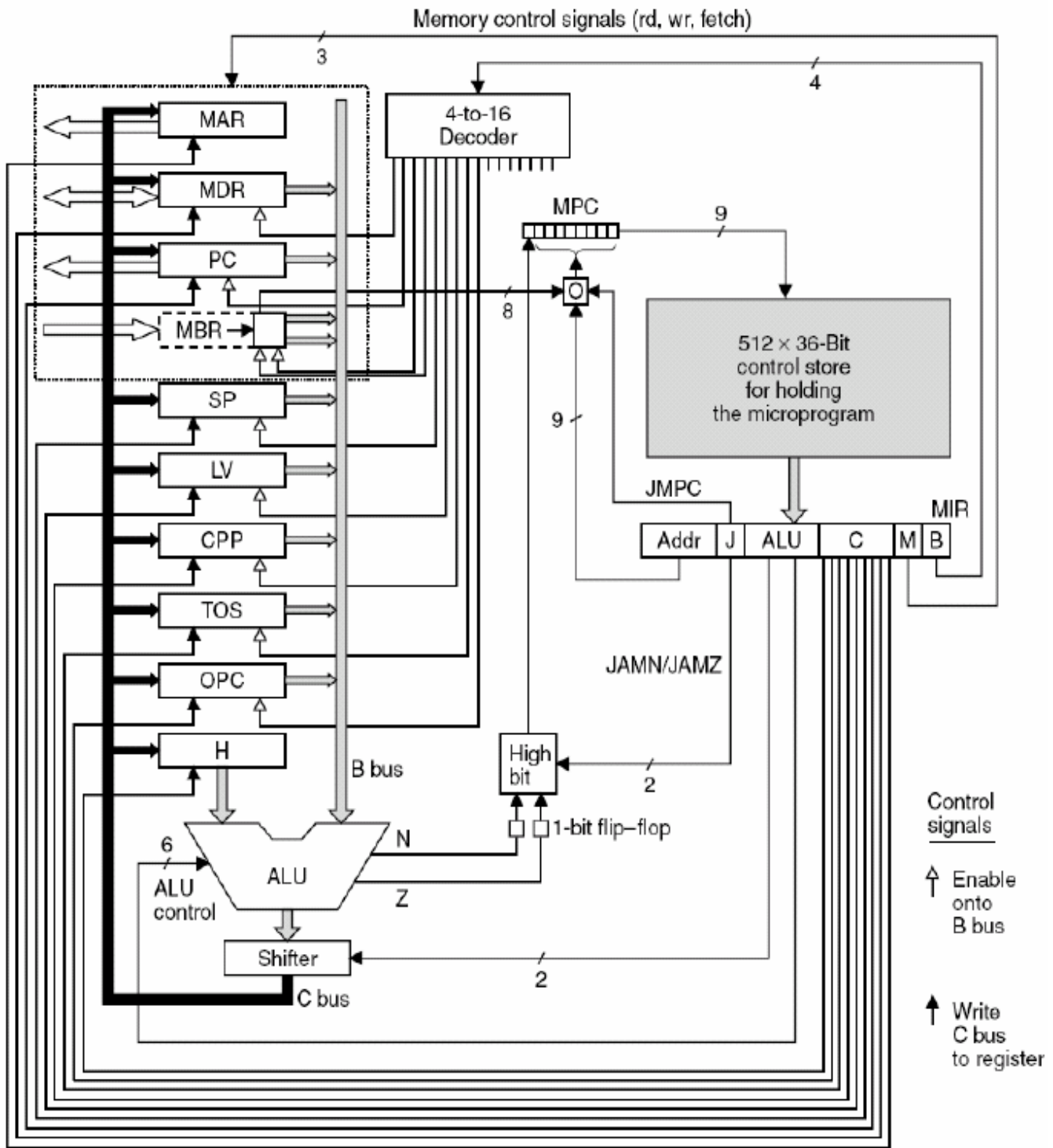
Figur 7: Adressedekoding.



(a) Tretrinns samlebånd.

(b) Femtrinns samlebånd

Figur 8: Samlebånd med angitt tidsforsinkelse.



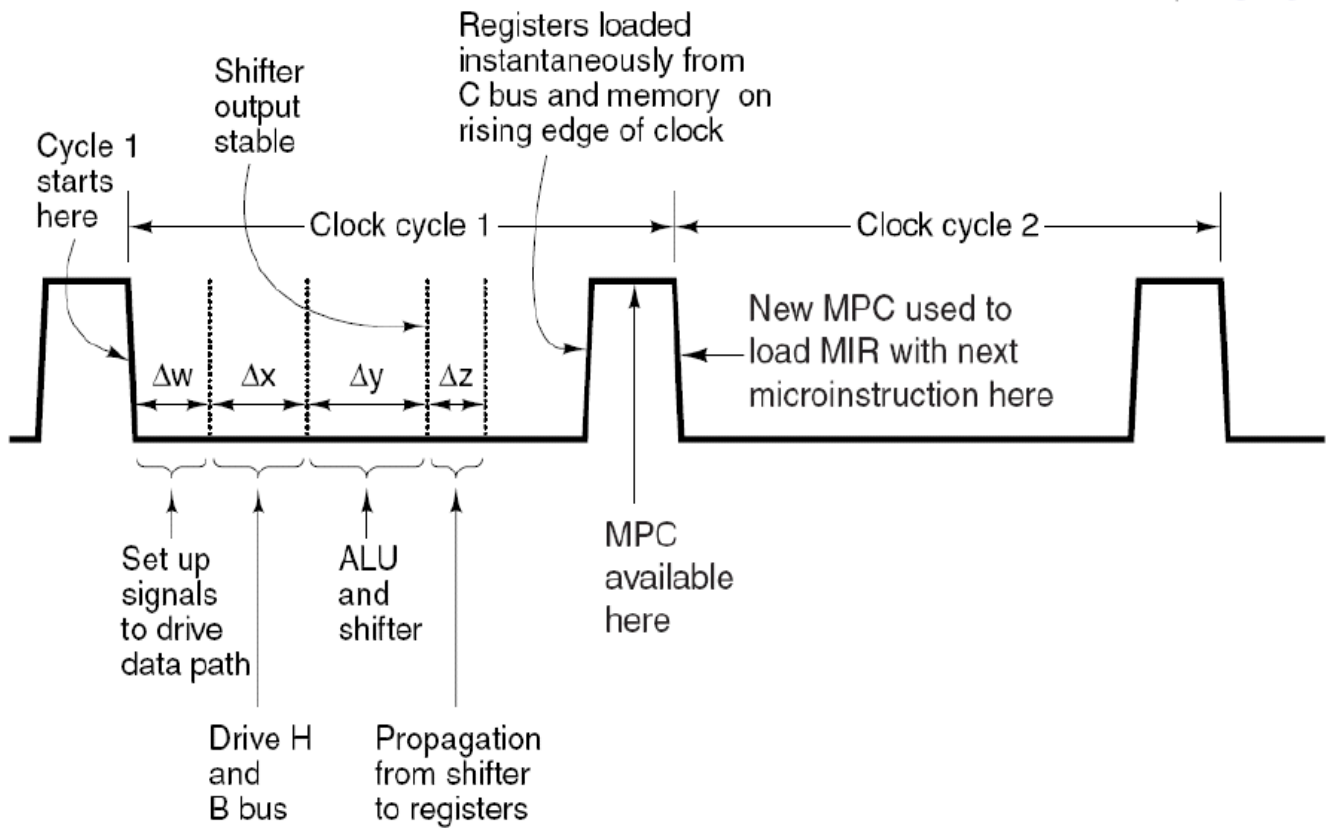
Figur 9: Blokkdiagram (IJVM).



$F_0$	$F_1$	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 11: Funksjonstabell for ALU (IJVM).



Figur 12: Timingdiagram (IJVM).