Norges teknisk–naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

Faglig kontakt under eksamen:
Magnus Jahre (952 22 309)

EKSAMEN I TDT4160 DATAMASKINER GRUNNKURS

Onsdag 11. august
Tid: 09:00 – 13:00

Hjelpemidler:
D.
Ingen trykte eller håndskrevne hjelpemidler tillatt.
Bestemt, enkel kalkulator tillatt.

Vi anbefaler å svare kort og presist for å få tid til å svare på alle oppgavene. Eksamen teller 100% av karakteren i faget, og de oppgitte prosenttallene angir det maksimale antall poeng man kan oppnå på hver oppgave. Les oppgavene grundig.

Oppgave 1 Digital logikk (20%)

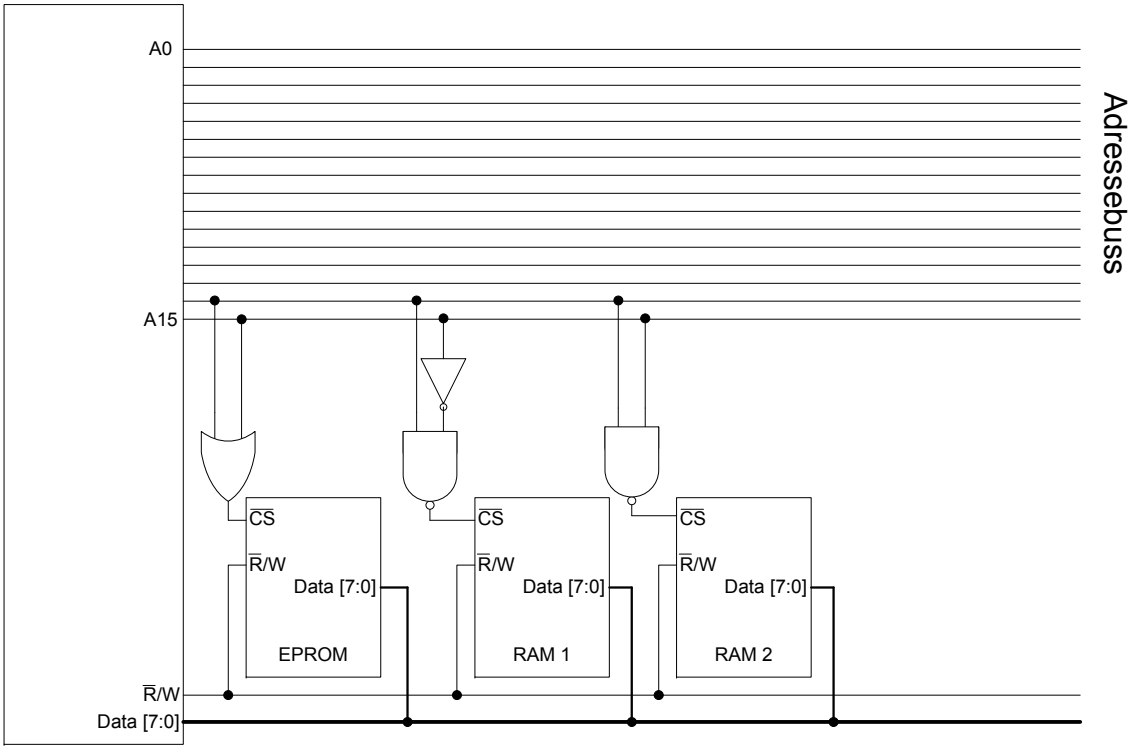
Figur 1 på side 2 viser et blokkdiagram av et system der en EPROM enhet og to RAM enheter er koblet til en felles adresse- og databuss. EPROM og RAM enhetene har et aktivt lavt (logisk "0") CS (Chip Select) signal.

a) (10%) Tegn et minnekart for systemet i figur 1.

Løsning: Figur 2 viser minneområdet for systemet.

b) (5%) Hvor mye RAM kan systemet utvides med hvis alle eksisterende enheter beholdes? Begrunn svaret.

Løsning: Systemet kan utvides med $2^{14} = 16384$ bytes (eller 16KB)



Figur 1: Adressedekoding

c) (5%) Tegn et blokkdiagram som inkluderer den nye RAM-enheten og nødvendig adresseringslogikk.

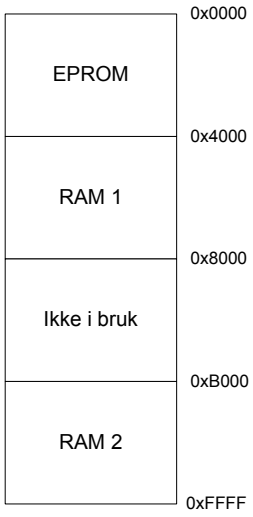
Løsning: Figur 3 viser den nye RAM-enheten og nødvendig adresselogikk. De nye elementene i figuren er vist i grått.

Oppgave 2 Mikroarkitektur og mikroinstruksjoner (20%)

a) (10%) Forklar funksjonen til SP registeret. Hvilke operasjoner brukes SP i?

Løsning: SP peker alltid til toppen av "operand stack". SP brukes i hovedsak når man kaller en metode og når man returnerer fra den. Se side 250–255 i læreboka for flere detaljer.

b) (5%) Lag mikroinstruksjon(er) for følgende IJVM-operasjon: SP = SP + 1



Figur 2: Minnekart for Oppgave 1a

Angi korrekte bit for ALU, C, Mem og B for instruksjonsformatet gitt i figur 6. Du kan se bort fra Addr og J-feltene. Mikroinstruksjon(ene) skal være tilpasset Mic-1 (se figur 7).

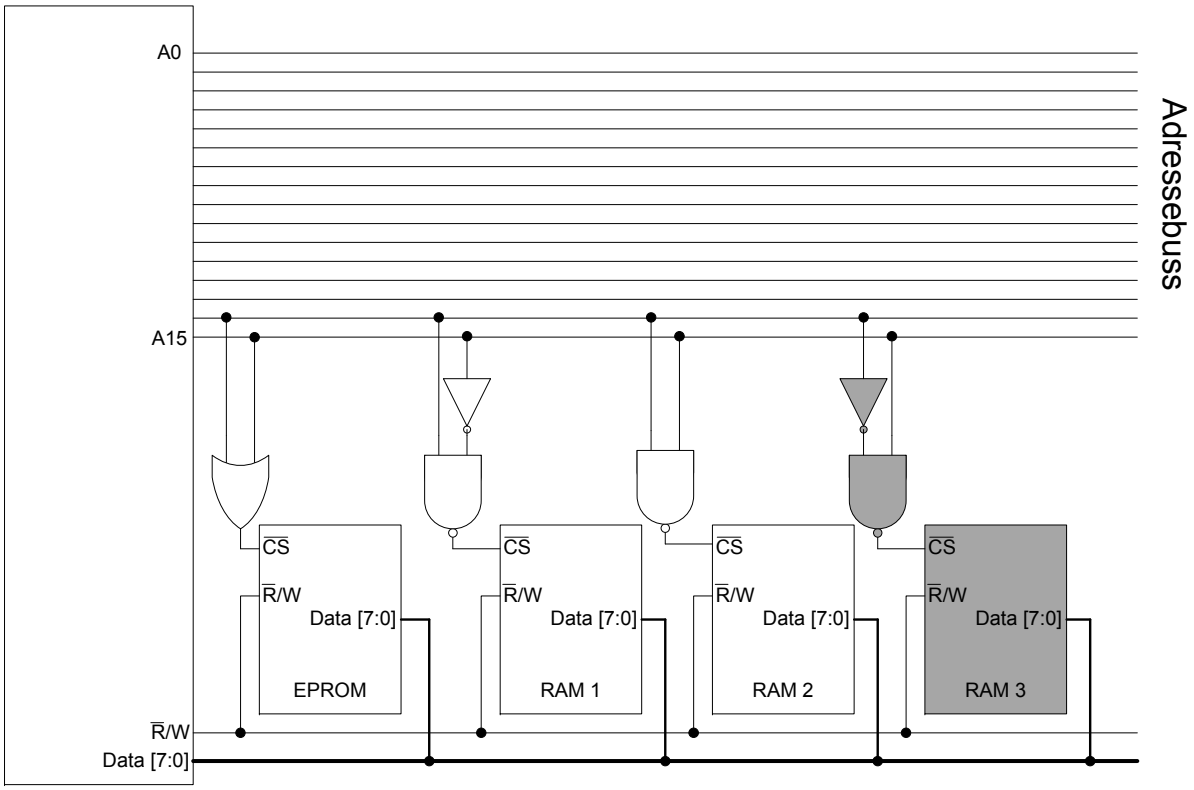
Løsning: ALU: 110101 (B+1), C: 000001000 (Skriv SP), Mem: 000 (ingen minneoperasjon), B: 0100 (Desimalverdi: 4, Les SP)

c) (5%) Lag mikroinstruksjon(er) for følgende IJVM-operasjon: TOS = TOS + OPC

Angi korrekte bit for ALU, C, Mem og B for instruksjonsformatet gitt i Figur 6. Du kan se bort fra Addr og J-feltene. Mikroinstruksjon(ene) skal være tilpasset Mic-1 (se figur 7).

Løsning: Siden TOS og OPC deler B bussen i Mic-1, må vi bruke H som mellomlager. Vi velger å flytte TOS til H først og så legge sammen. En annen riktig løsning er å flytte TOS til H og så legge sammen.

Mikroinstruksjon 1 (H = OPC)
ALU: 010100 (B), C: 100000000 (Skriv H), Mem: 000 (ingen minneoperasjon), B: 1000 (Desimalverdi: 8, les OPC)
Mikroinstruksjon 2 (TOS = H + TOS)
ALU: 111100 (A+B), C: 001000000 (Skriv TOS), Mem: 000 (ingen minneoperasjon), B: 0111 (Desimalverdi: 7, les TOS)



Figur 3: Blokkdiagram med ny RAM enhet (Oppgave 1c)

Oppgave 3 Instruksjonssett (20%)

a) (10%) Du har fått i oppgave å lage et instruksjonsformat til en enkel 8-bitsarkitektur. Formatet skal støtte 4 forskjellige instruksjoner, 8 registre og 2 operander. Foreslå et mulig instruksjonsformat og forklar hvorfor det oppfyller kravene gitt i denne oppgaven.

Løsning: Vi setter av 2 bit til opcode to registerfelt av 3 bit. De to opcodebitene gir $2^2 = 4$ mulige opcoder mens registerfeltene kan adressere $2^3 = 8$ registre.

b) (10%) Forklar forskjellen på register adressering (Engelsk: register addressing) og register-indirekte adressering (Engelsk: register indirect addressing).

Løsning: Med register adressering peker tallet man oppgir i instruksjonen til et register. I register-indirekte adressering peker verdien i instruksjonen også til et register, men her inneholder registeret en minneadresse som man så bruker til å hente de ønskede dataene.

Oppgave 4 Datamaskiner (20%)

a) (10%) Figur 9 viser Mic-3 som er forbedring av mikroarkitekturen Mic-1 vist i figur 7. Nevn de to viktigste forbedringene og forklar på hvilken måte de bidrar til økt ytelse.

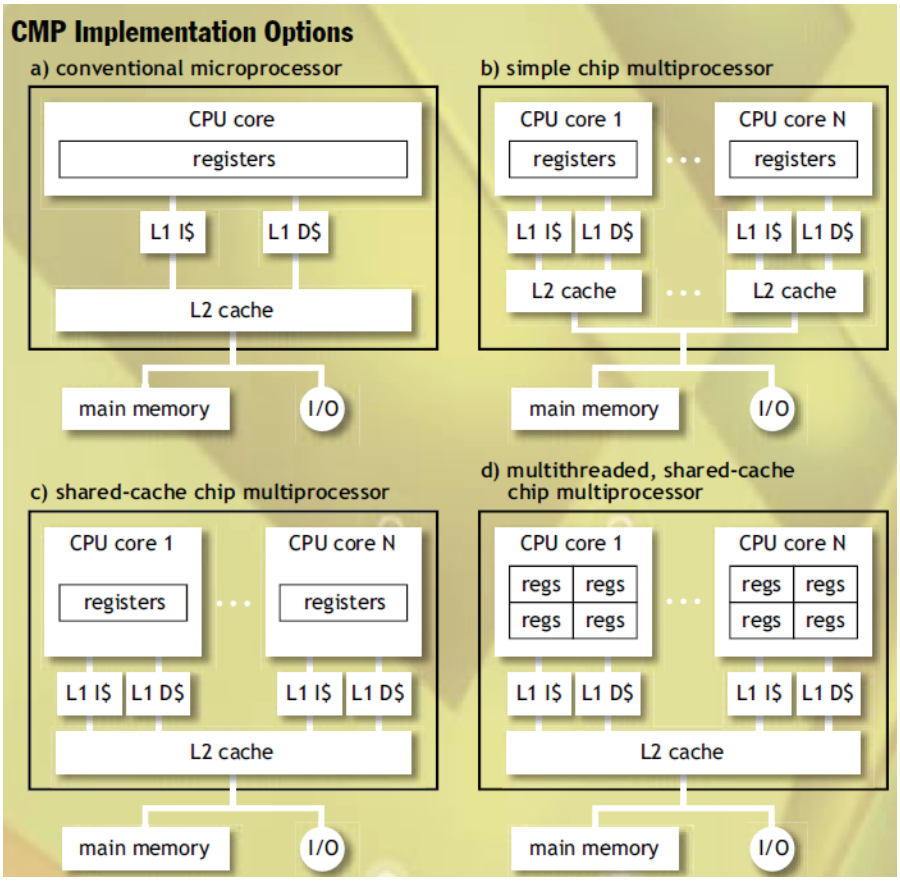
Løsning: *Forbedring 1:* Mikroarkitekturen har tre busser og dette gjør at man slipper å bruke instruksjoner på å mellomlagre verdier hvis man skal legge sammen verdier fra to registere.
Forbedring 2: Det er lagt til latcher på alle bussene. Dette åpner for pipelining som gjør at prosessoren kan jobbe med flere instruksjoner samtidig.

b) (10%) Diskuter kort fordeler og ulemper med å implementere ut-av-rekkefølge instruksjonsutføring (Engelsk: out-of-order execution) i en prosessor.

Løsning: Se læreboka s. 304 – 309. Dette er en typisk "tenke selv" oppgave der man får poeng for alle fornuftige forslag.

Oppgave 5 Diverse (20%)

a) (10%) Forklar forskjellen(e) mellom 1., 2. og 3. generasjons Chip Multiprocessor (CMP). Tegn gjerne en figur.



Figur 4: Chip Multiprocessor generasjoner (Hentet fra Olukotun og Hammond, 2005)

Løsning: Figur 4 viser en konvensjonell prosessor og tre generasjoner med CMPer. En første generasjons CMP er to enkjernearkitekturer plassert på samme brikke. Andregenerasjon har prosessorkjerner designet for bruk i en CMP og deler hurtigbuffer på brikken. Tredjegenerasjon har utvidet kjernene med multi-threading.

b) (5%) Forklar forskjellen på en *write-through* og en *write-back* strategi for hurtigbuffere (Engelsk: caches).

Løsning: Med en write-through strategi blir alle hovedminnet oppdatert på alle skriveoperasjoner til hurtigbufferet. Med write-back skriver man tilbake dataene når hurtigbufferblokken blir kastet ut av hurtigbufferet. (Se læreboka s. 298)

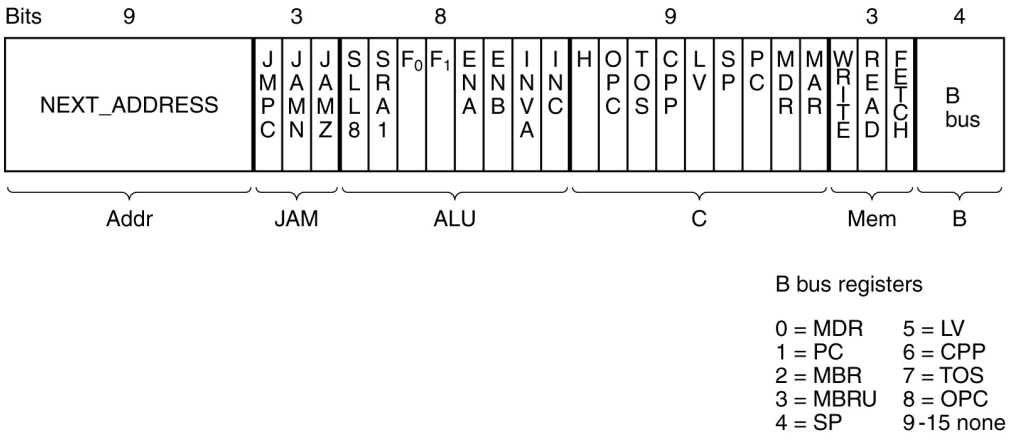
c) (5%) Forklar forskjellen på statisk og dynamisk hopp-prediksjon (Engelsk: branch prediction).

Løsning: Dynamisk hopp-prediksjon gjøres mens programmet kjører, krever ekstra maskinvare og er usynlig for programmet. Statisk hopp-prediksjon gjøres av kompilatoren ved å sette inn spesielle instruksjoner som forteller prosessoren om kompilatoren tror programmet vil hoppe eller ikke. (Se læreboka s. 301-304)

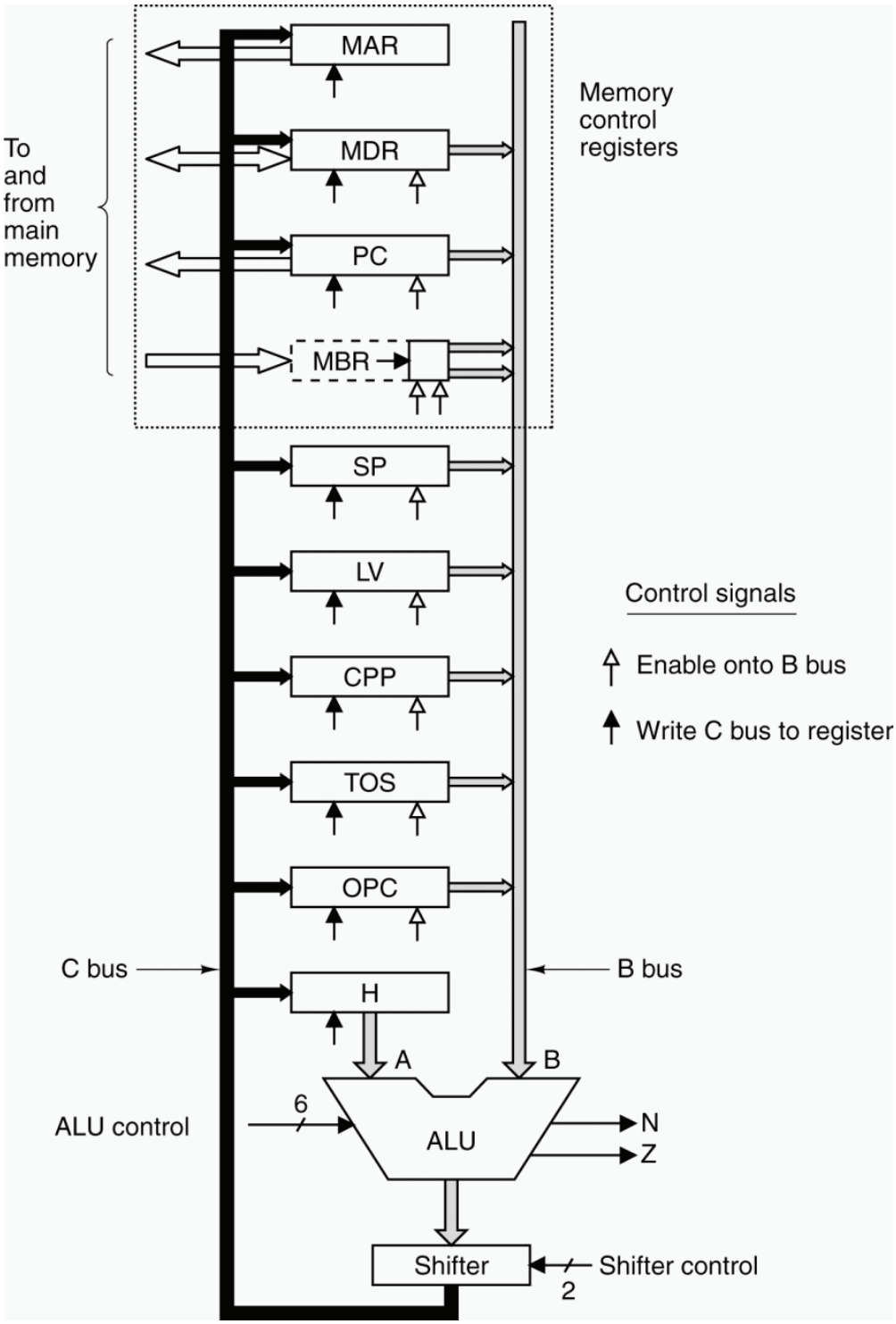
Vedlegg

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	$\overline{A}$
1	0	1	1	0	0	$\overline{B}$
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

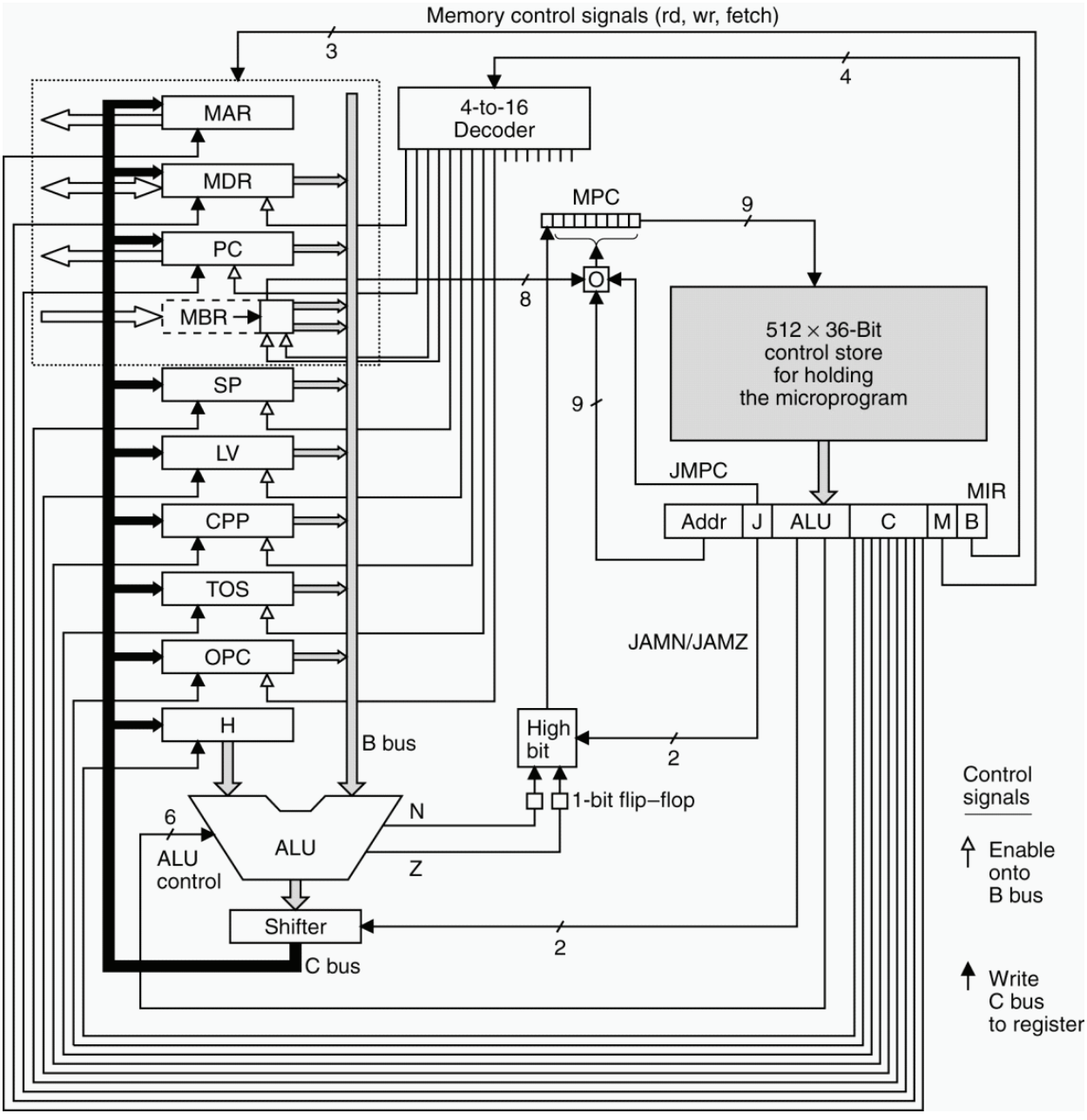
Figur 5: Funksjonstabell for ALU (Mic-1)



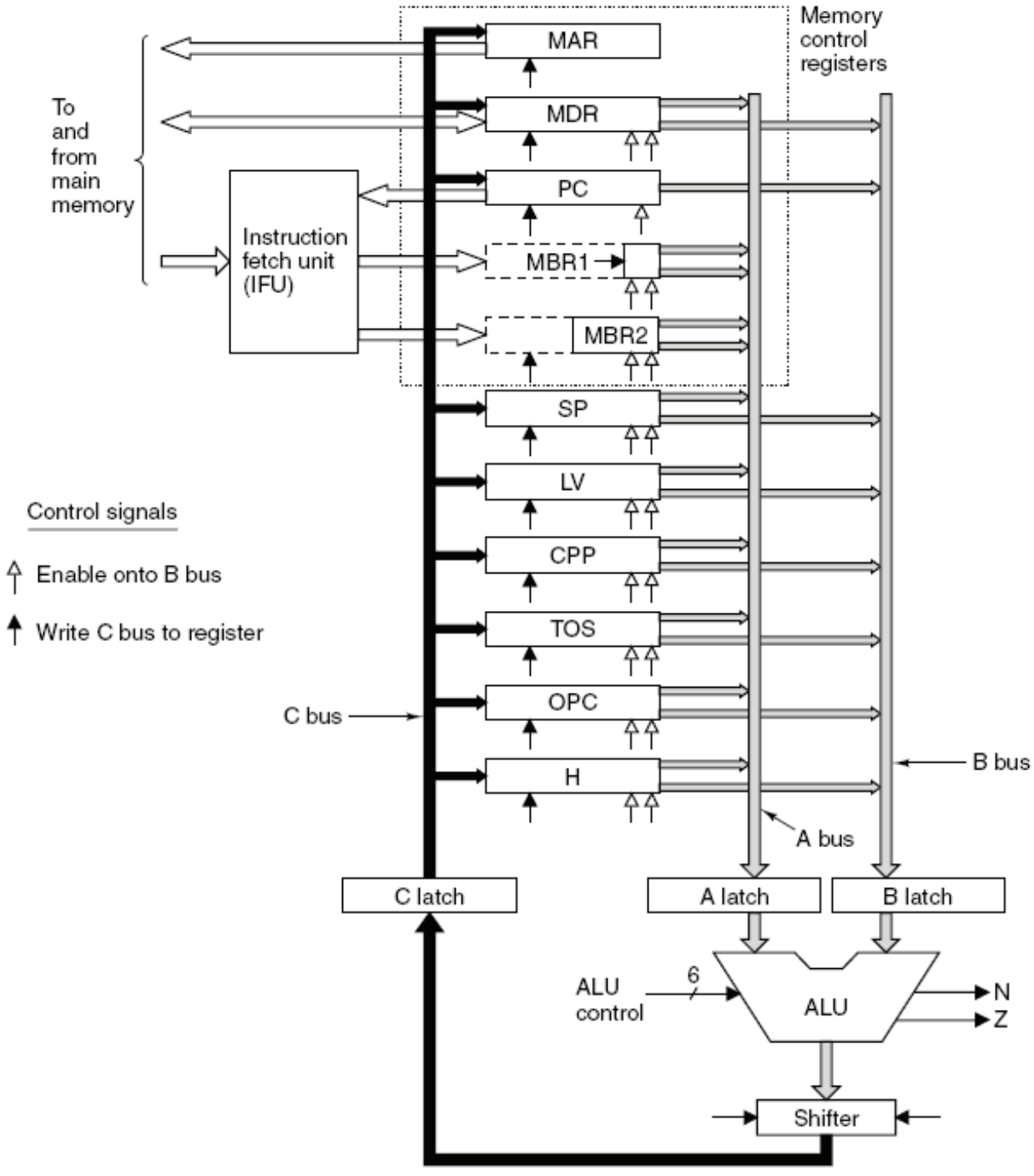
Figur 6: Mikroinstruksjonsformat (Mic-1)



Figur 7: IJVM mikroarkitektur (Mic-1)



Figur 8: Blokkdiagram (Mic-1)



Figur 9: Alternativ IJVM mikroarkitektur (Mic-3)