NTNU
Kunnskap for en bedre verden

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4160 datamaskiner og digitalteknikk

Faglig kontakt under eksamen: Gunnar Tufte

Tlf.: 97402478

Eksamensdato: 28. november 2016

Eksamenstid (fra-til): 9:00–13:00

Hjelpemiddelkode/Tillatte hjelpemidler: D: Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillat.

Annen informasjon:

Målform/språk: Bokmål

Antall sider (uten forside): 9

Antall sider vedlegg: 2

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig x **2-sidig ☐**

sort/hvit x **farger ☐**

skal ha flervalgskjema ☐

Kontrollert av:

Dato Sign

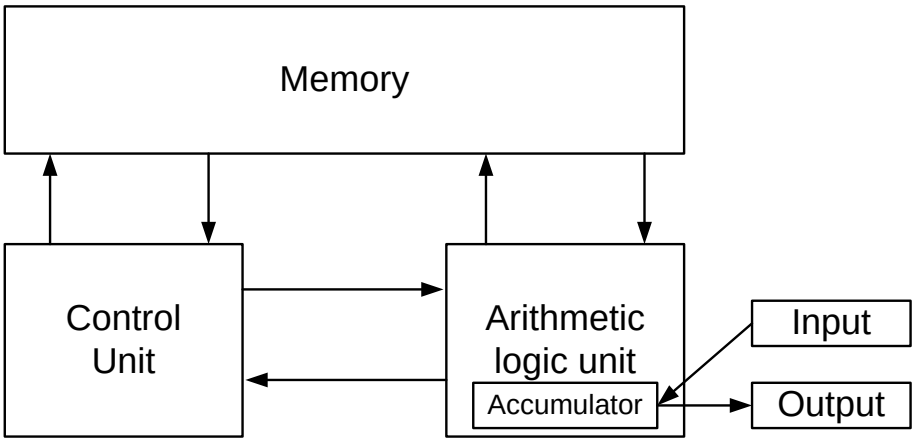
Merk! Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

Oppgave 1 Oppstart, litt av hvert (20%)

a)

Figur 1 viser en prinsippskisse av von Neumann-datamaskiner. Relater enhetene i lista under (i – iii) til minne (memory), kontroll (control unit) eller ALU (arithmetic logic unit) i Figur 1.

i) register
ii) mikroinstruksjonsminne (microprogram control store)
iii) hurtigbuffer (cache)



Figur 1 Prinsippskisse av von Neumann-maskin.

b)
Forklar kort hva som ligger i begrepene:

i) Instruksjonsnivå parallellitet (Instruction-Level Parallelism (ILP))
ii) Prosessornivå parallellitet (Processor-level Parallelism)

c)
Forklar kort forskjellen på programmert I/O med travel venting (programmed I/O with busy waiting) og avbruddsdrevet I/O (interrupt-driven I/O).

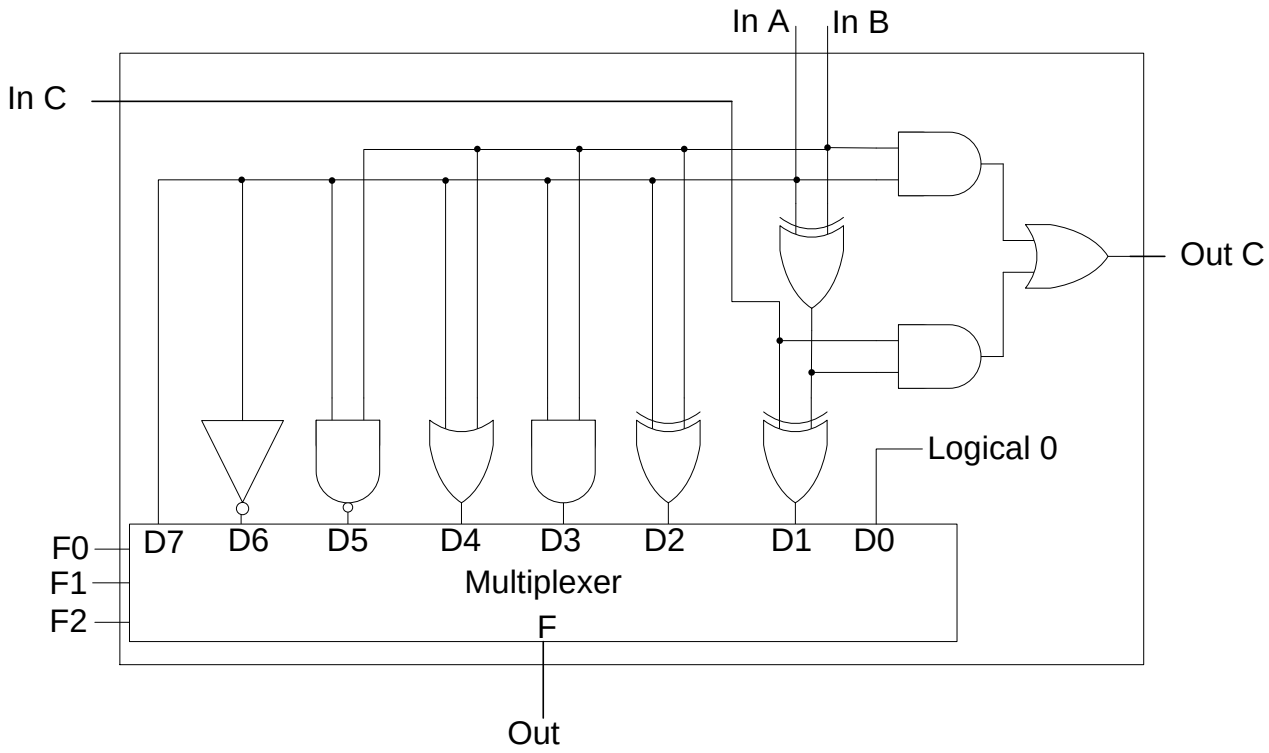
d)
Forklar kort hva som ligger i begrepet lokalitet (locality) i forbindelse med minnesystem, f. eks. hurtigbuffer og virtuelt minne (paging).

e)
Forklar kort forskjellen på homogene (homogeneous) og heterogene (heterogeneous) Multi Prosessorer (CMP).

Oppgave 2 Digitalt Logisk Nivå (20% (a: 6%, b: 6%, c: 6% og 2% på d))

a)

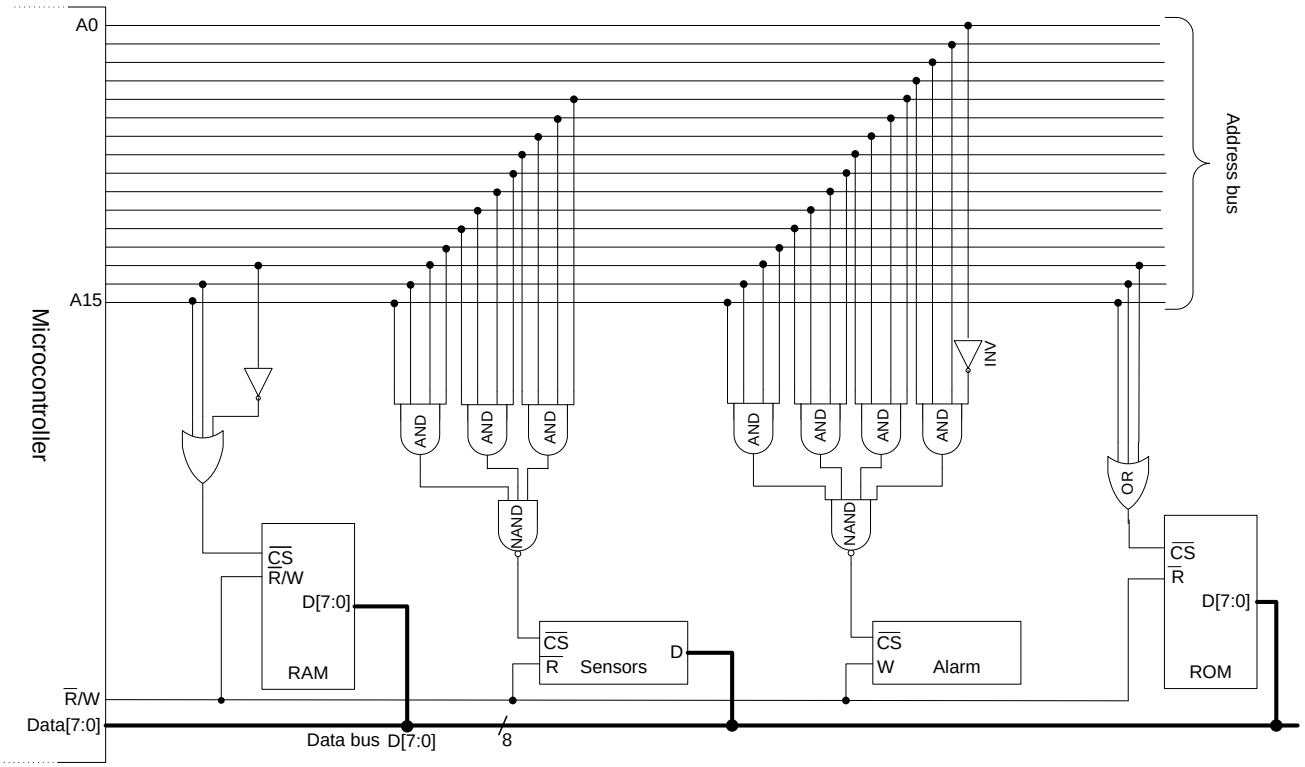
Figur 2 viser en enkel 1-bit ALU. Det er 2 datainnganger (in A og in B), en datautgang (out), 1 C-inngang og 1 C-utgang. ALU-en har 3 kontrollinnganger for å bestemme funksjon (F0, F1 og F2). Finn hva aritmetisk eller logisk funksjon ALU-en gjør for de mulige kombinasjonene av kontrollinngangene (000 - 111). Bruk gjerne en tabell som for IJVM sine ALU-funksjoner.



Figur 2 Enkel 1-bit ALU.

b)

I et innvevd system (embedded system) for miljøovervåkning er det en sensorenhet med 16 sensorer tilkoblet. Det er tilkoblet en alarm som går hvis det blir målt verdier over en terskel for en eller flere av sensorene. I systemet er det benyttet en mikrokontroller. Figur 3 viser det eksterne bussgrensesnittet med adressedekodingslogikk for mikrokontrolleren. Det er en ROM-brikke for program, en RAM-brikke, en sensormodul og en alarmmodul. Alle enhetene benytter et aktivt lavt (logisk "0") CS (Chip Select)-signal. Alle enhetene benytter 8-bit data.



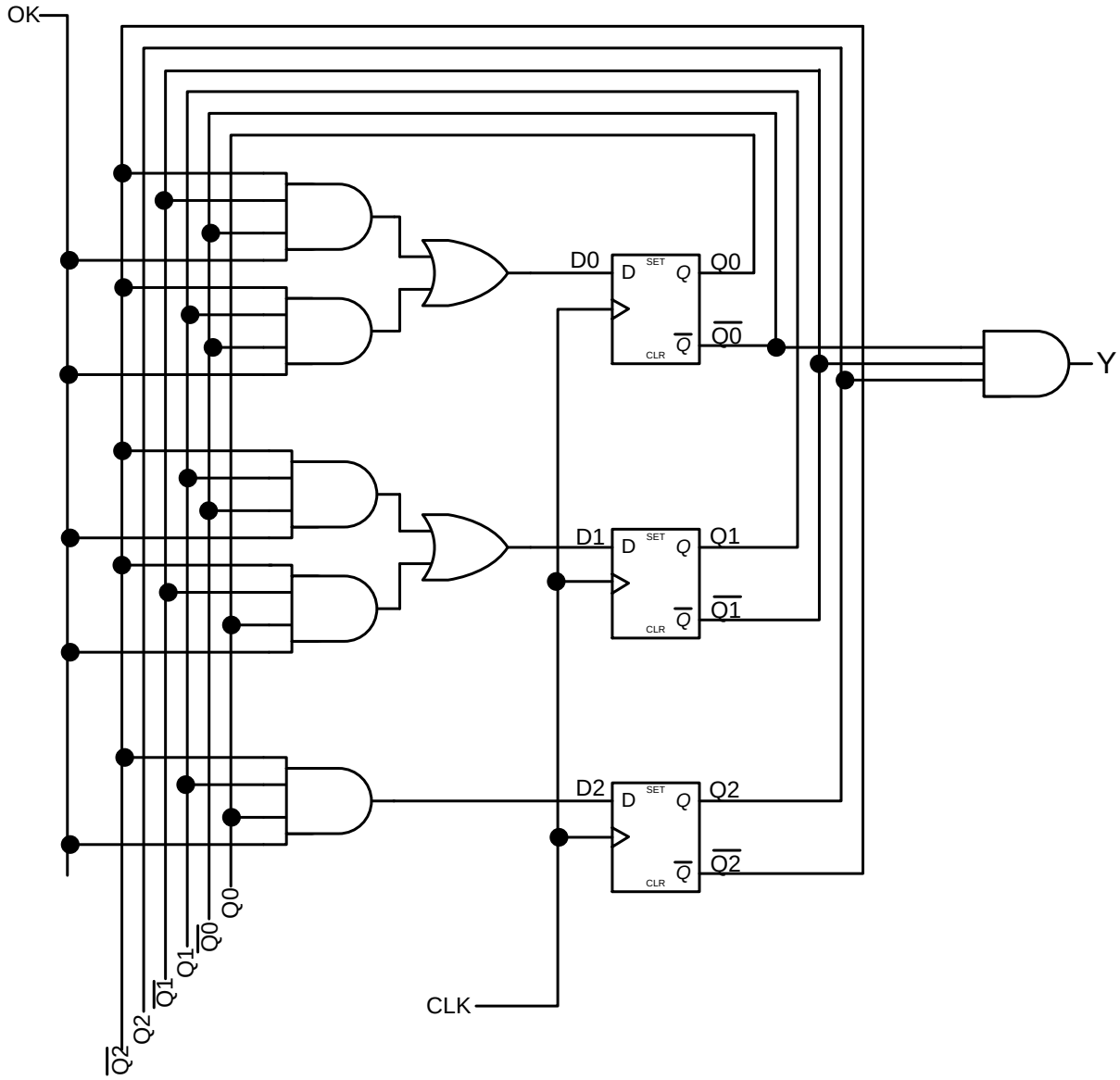
Figur 3 Address decoding.

i) Finn adresseområdene for RAM, ROM, Sensors og Alarm.

ii) Tegn minnekart ut fra adresseområdet.

iii) Er det overlapp i adresseringa. I så fall er dette problematisk? Forklar kort.

c)
Figur 4 viser en FSM som sjekker at et inngangssignal er stabilt over en viss tid målt i klokkepulser. FSM-en vil gi klarsignal om at alt er OK ved å legge Y høyt. Logikk for reset og set for vippene er ikke tegnet inn.



Figur 4 Finite State Machine (FSM)

Finn de logiske uttrykkene for D0, D1 og D2 (excitation equation). Angi de neste tilstandsuttrykkene (next state equations) og lag transisjonstabell (next-state-table) for kretsen, som viser oppførselen til denne FSM-en og utgangssignalet Y.

d)
Er FSM-en i Figur 4 av typen Mealy eller Moore?

Oppgave 3 Mikroarkitektur og mikroinstruksjoner (20% (a: 4%, b: 5%, c: 5%, d: 4% og 2% på e))

Bruk vedlagte diagram i figur 9, figur 10, figur 11 og figur12 for IJVM til å løse oppgavene.

a)
IJVM (og i mange andre arkitekturer) har fire register som grensesnitt mot eksternt minne. I IJVM er det følgende register: MAR, MDR, PC og MBR. Forklar kort hvordan disse registrene brukes og hva som er lagra i dem.

b)
For mikroarkitekturen i Figur 9. Kva er minimum mengde mikroinstruksjoner en må bruke for å kopiere verdien i H-registeret til alle disse registrene: OPC, TOS, CPP. Angi hvordan dette kan gjøres effektivt ved å oppgi mikroinstruksjon(er).

Se vekk fra Addr- og J-feltet i mikroinstruksjonsformatet. Angi korrekte bit for ALU, C, Mem og B gitt i figur 10.

c)
For mikroarkitekturen i Figur 9. Lag mikroinstruksjoner for en funksjon som kan teste om innholdet i register MDR og LV er likt. Angi hvordan en slik likhet kan bli detektert.

Se vekk fra Addr- og J-feltene i mikroinstruksjonsformatet. Angi korrekte bit for ALU, C, Mem og B gitt i figur 10.

d)
OpCode i IJVM er på 8-bit (f.eks. 0x60 IADD og 0x99 IFEQ). MPC er på 9 bit. Hvilken funksjon har det ekstra bit-et i MPC? Når er det i bruk?

e)
For IJVM-arkitekturen i figur 9. Hva er lengden (i bit) på en mikroinstruksjon og hva er maksimal mengde mikroinstruksjoner for denne versjonen av IJVM.

Oppgave 4 Instruksjonssett arkitektur (ISA) (20% (a: 3%, b: 4%, c: 3% og 10% på d))

BarabbasX42 er en svært enkel prosessor. BarabbasX42 har en ”load”, en ”store”, 8 ALU-instruksjoner og noen spesialinstruksjoner, inkludert NOP-instruksjonen og to flytkontrollinstruksjoner (flow control instructions). Instruksjonsformatet for instruksjonene er vist i figur 5, figur 6 viser instruksjonssettet. Alle register og busser er 32-bit. Det er 32 generelle register tilgjengelig. Prosessoren har en Harvard-arkitektur. Bruk figur 5 og figur 6 til å løyse oppgaven.

a)
Denne prosessoren har mange egenskaper fra RISC-prosessorer. Ut fra tilgjengelig informasjon nevn minst tre av egenskapene til BarabbasX42 som er RISC-inspirert.

b)
Gi eksempel på en BarabbasX42-instruksjon som benytter adresseringsmodi (addressing mode) imidiate adressing.

Gi eksempel på en BarabbasX42-instruksjon med instruksjonsformatet null adresseinstruksjon (zero-address instruction).

c)
Hva er det maksimale adresserommet BarabbasX42 kan adressere?

d)
R8 har følgende verdi: 0xFFFF 0000, R9 har følgende verdi: 0xFFFF 0002. I dataminnet ligg følgende data fra adresse 0xFFFF 0000:

Adresse	Data
0xFFFF 0000:	0x00 00 00 55
0xFFFF 0001:	0x00 AA 00 00
0xFFFF 0002:	0x00 00 00 AA
0xFFFF 0003:	0x00 00 55 00
0xFFFF 0004:	0x00 00 00 00

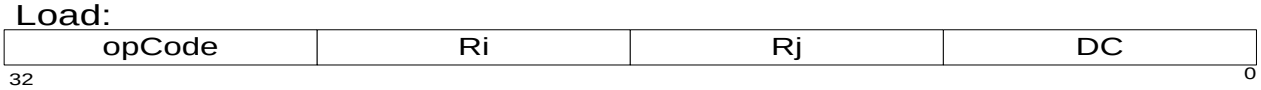
Følgende psaudokode er en del av et større program. Svar på spørsmåla ut fra tilgjengelig informasjon.

```
LOAD R1, R8;
LOAD R2, R9;
STORE R1, R2;
ADD R1, R1, R8;
STORE R1, R8;
LOAD R1, R2
MOVC R16, 0x0055;
ADD R1, R1, R16;
CMP R1, R2;
BZ R1;
```

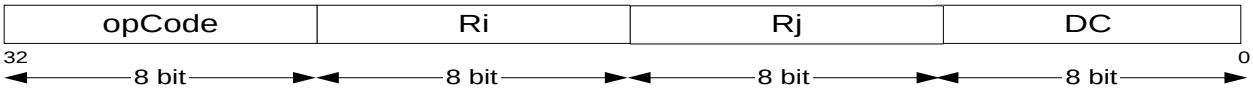
Forklar Hva som skjer i koden. Hvilken verdi vil R1 ha etter at koden har kjørt?

Load/store:

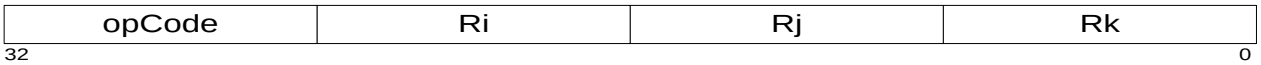
Load:

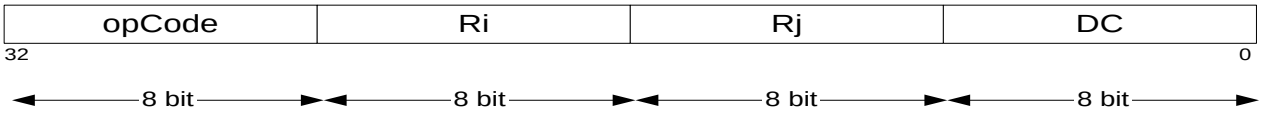


Store:



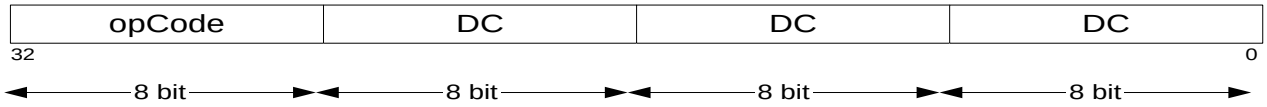
ALU:



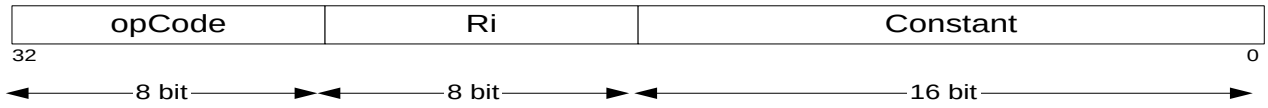


Spesial:

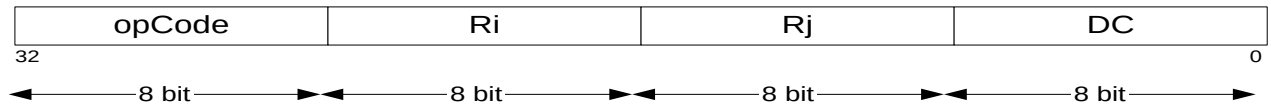
NOP:



MOVC:

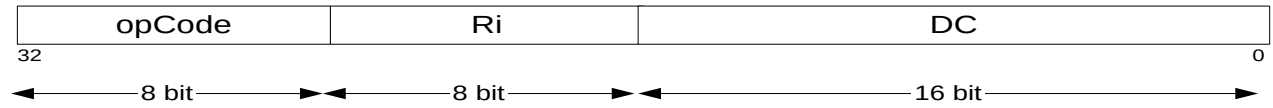


CP:

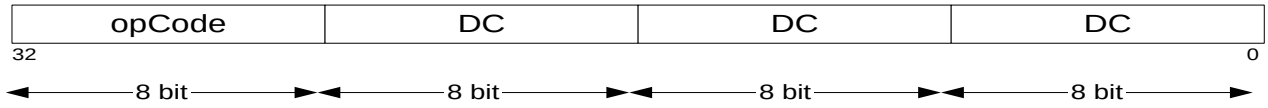


Flow control:

BZ



RT



Ri, Rj and Rk: any user register, R0 - R31
DC: Don’t care: any data memory location

Figur 5 Instruction format BarabbasX42

Instructions set:

LOAD: Load data from memory.

load Ri, Rj Load register Ri from memory location in Rj.

STORE: Store data in memory.

store Ri, Rj Store register Ri in memory location in Rj.

ALU: Data manipulation, register–register operations.

ADD Ri, Rj, Rk ADD, $Ri = Rj + Rk$. Set Z-flag if result =0.
NAND Ri, Rj, Rk Bitwise NAND, $Ri = \overline{Rj \cdot Rk}$. Set Z-flag if result =0.
OR Ri, Rj, Rk Bitwice OR, $Ri = Rj + Rk$. Set Z-flag if result =0.
INV Ri, Rj Bitwice invert, $Ri = \overline{Rj}$. Set Z-flag if result =0.
INC Ri, Rj Increment, $Ri = Rj + 1$. Set Z-flag if result =0.
DEC Ri, Rj Decrement, $Ri = Rj - 1$. Set Z-flag if result =0.
MUL Ri, Rj, Rk Multiplication, $Ri = Rj * Rk$. Set Z-flag if result =0.
CMP, Ri, Rj Compare, Set Z-flag if $Ri = Rj$

Special: Misc.

CP Ri, Rj Copy, $Ri < -Rj$ (copy Rj into Ri)
NOP Waste of time, 1 clk cycle.
MOVC Ri, constant Put a constant in register $Ri = C$.

Flow control: Branch.

BZ, Ri Conditional branch on zero, $PC = Ri$.
RT Return, return from branch.

Ri, Rj and Rk: Any user register.
DC: Don't care.

Figur 6 Instruction set BarabbasX42.

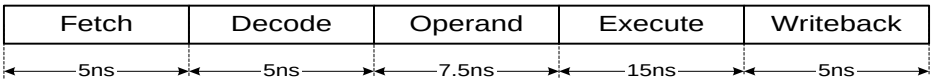
Oppgave 5 Ytelse (20 (20% (a: 5%, b: 10%, og 5% på c))

a)
En prosessor har et minnesystem med RAM og eit nivå med hurtigbuffer (cache). Minnesystemet har følgende egenskaper:

RAM: aksesstid: 100ns
Cache: aksesstid 10 ns
Trefforholdstall (Hit ratio) på 80%

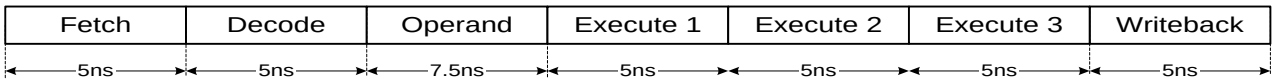
Hva er gjennomsnittlig aksesstid for dette minnesystemet?

b)
1) Figur 7 viser samlebåndet (pipeline) for en prosessor. Det er 5 steg. Estimer maksimal klokkefrekvens for denne prosessoren?



Figur 7 5 stage pipeline.

2) For å øke ytelsen blir execute-steget delt i tre som vist i figur 8. Hva er nå maksimal klokkefrekvens for prosessoren?



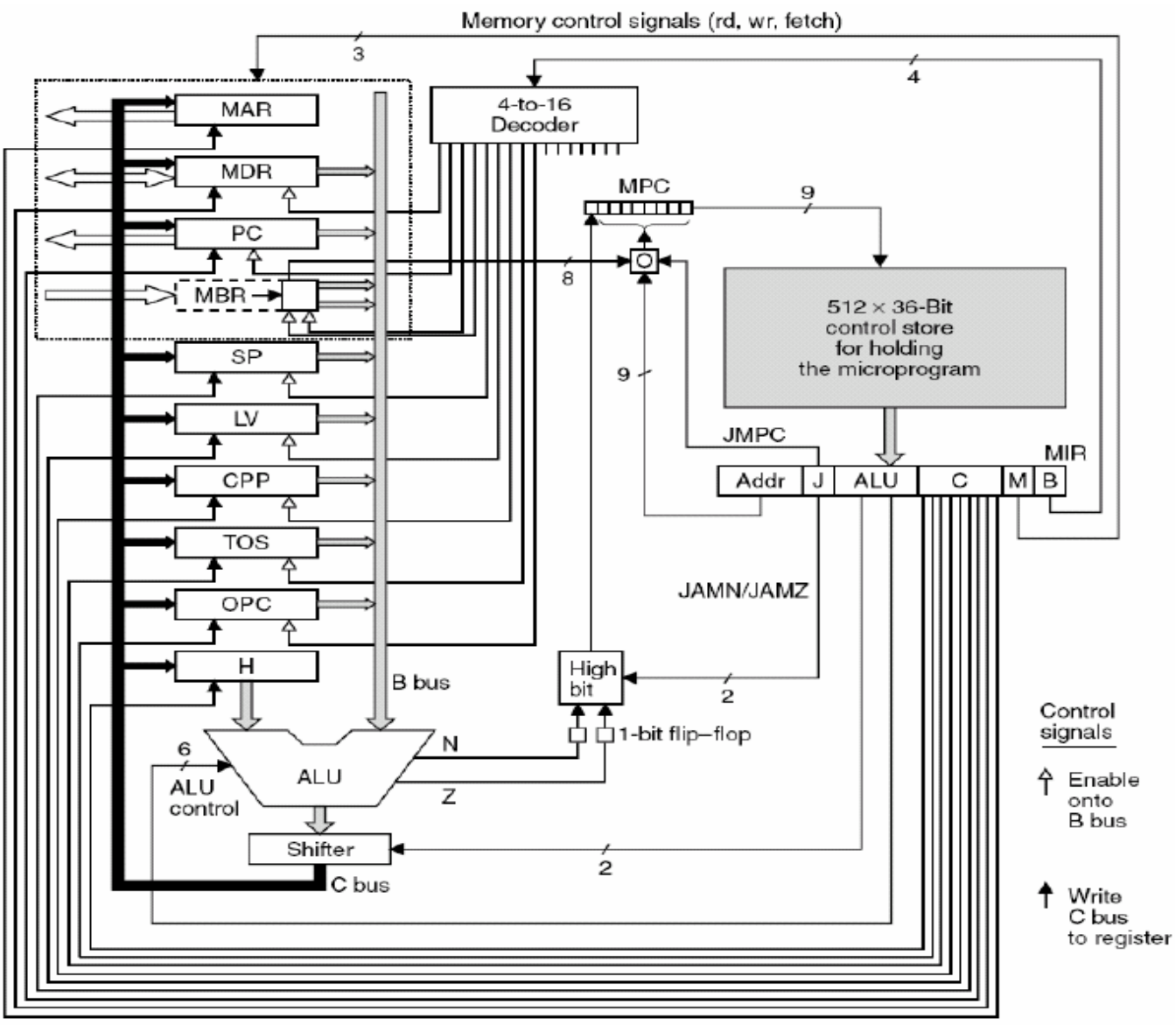
Figur 8 7 stage pipeline.

3) Det eksisterer en variant av prosessoren uten samlebånd. Estimer hva klokkefrekvensen på denne varianten er?

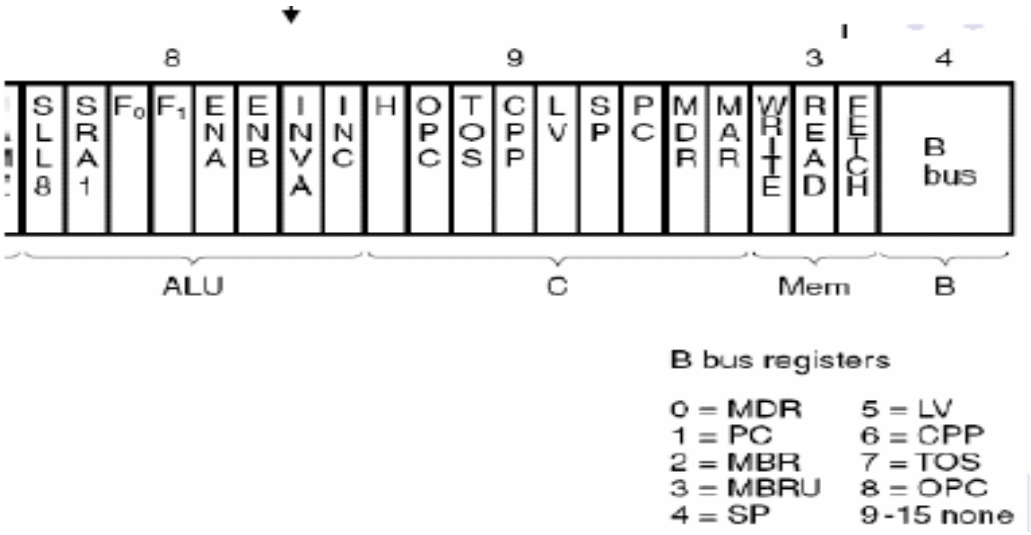
4) Hvilke ulemper medfører det å ha dype (mange steg) i samlebånd? Forklar kort.

c)
Figur 9 viser mikroarkitekturen til IJVM. Foreslå to tiltak som øker ytelsen.

Vedlegg IJVM



Figur 9 IJVM

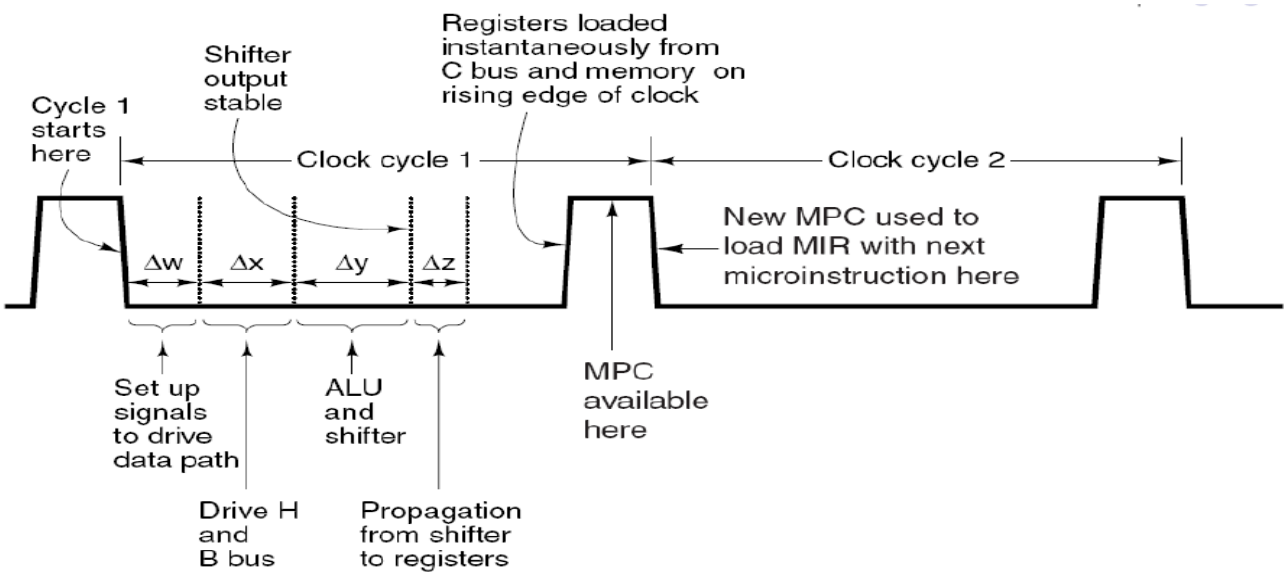


Figur 10 Microinstruction format.

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

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

Figur 11 ALU functions.



Figur 12 Timing diagram.