

# Oppgave 1

a)

i) Dersom C og D gjør en request samtidig, vil D-enheten få bussen.

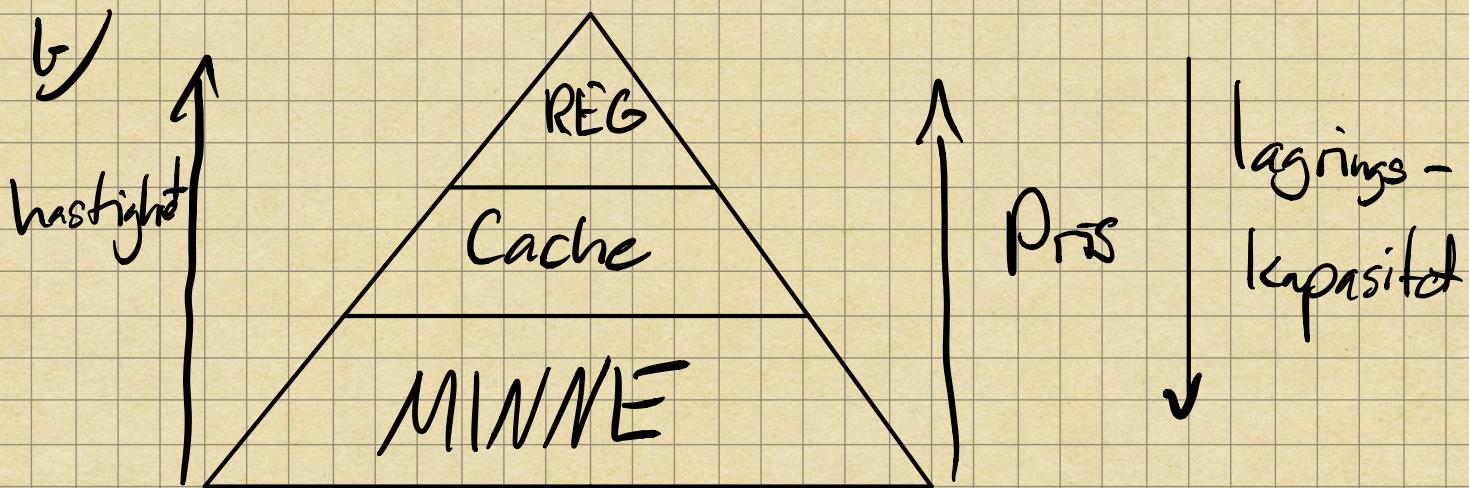
Dette fordi den er fysisk nærmest arbiteren og vil avbryte signaler den sender ut. Den setter også busy-signalet.

ii)

Dette er en desentralisert arbiteringsmekanisme.

Det er ingen logisk sentralitet som enhetene må forhandle om.

Busy-chaining.



c) Latency-hiding er en teknikk  
for å skjule at minne er betraktelig  
tregere enn prosessorer.  
Datakopier, prefetching og multithreading  
er eksempler på teknikker.

d) En interrupt-vektor er en peker  
til en adresse for handling av interrupt.  
Enten en interrupt-code eller en fukstisk  
hanndler.

e)  
L1: Write til R<sub>i</sub>  
L2: Read fra R<sub>i</sub>, må altså vente  
til L1 er ferdig

True dependence.

## Oppgave 2

a)

Burde gi pi én klokkeperiode.

Innater over Bi høyre MUX og tar med 1-er fra venstre MUX før å indikere togr-komplement, ADDER si og gi sum pi resultat.

b)

Figur 3:

RAM 1

1 000 0000 XXXX XXXX  
[8000, FFFF]

RAM 2

0000 0001 XXXX XXXX  
[0100, 7FFF]

ROM:

0000 0000 XXXX XXXX  
[0000, 00FF]

# Figur 4:

## RAM 1

0000 0001 XXXX XXXX  
 [0100, FFFF]

## ROM

0000 0000 XXXX XXXX  
 [0000, 00FF]

De omfatta samme adressrom. Uproblemotisk.

$S_{ij}$

	Current STATE	Next inp = 00	Next inp = 01	Next inp = 10	Next inp = 11	A	MR	R
000	IDLE	IDLE	MW	MR	ALU	0	0	
001	MW	WAIT	WAIT	WAIT	WAIT	0	0	
010	MR	WAIT	WAIT	WAIT	WAIT	0	1	
011	ALU	IDLE	IDLE	IDLE	IDLE	1	0	0
100	WAIT	IDLE	IDLE	IDLE	IDLE	0	0	0

ii) Det er 5 tilstander. Tallet fem representeres med 3 bit. State - reg kan være minimum 3 bit.

## Oppgave 3

a)

MIR:

Micro Instruction Register.

Addr:

Pekar på neste microinstruksjon.

J:

Sier om det kommer et hopp

ALU:

Definer funksjon som skal utføres

C: (Bus)

Sier hvor output fra ALU skal lagres.

M:

Sier om noe skal gjøres med minnet.

B: (Bus)

Sier hva som skal skrives til bussen.

Hva som skal leses fra.

b)

1. ALU: B, C: H, MEM: 000, B: LV

010100 100000000 000 0101

2. ALU: A  $\wedge$  B, C: H, MEM: 000, B: TOS

001100 100000000 000 0111

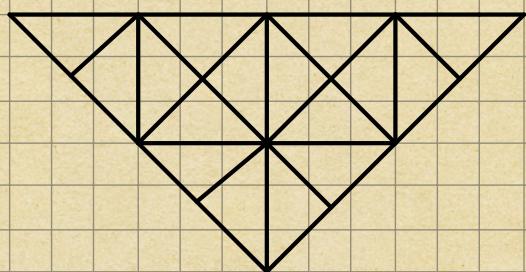
3. ALU:  $\bar{A}$ , C: DC, MEM: 000, B: DC

011010 010000000 000 ????

c)

Klokkefrekvensen er gitt av  
den totale forsinkelsen i dataparten.

$$\Delta w + \Delta x + \Delta y + \Delta z$$



## Oppgave 4

d) Det er separate busser til programminnet og  
data minnet.

b) Instruksjonene har inn 0, 1, 2 og 3 adresser.  
Immediate, direct og register-addressing

E

Load R<sub>1</sub>, R<sub>8</sub>

Legger data fra adressen i R<sub>8</sub> inn i R<sub>1</sub>

R<sub>1</sub>: 0x00 00 00 5F

ADD R<sub>11</sub>, R<sub>1</sub>, R<sub>11</sub>

Adderer innhold i R<sub>1</sub> og R<sub>11</sub>, lagrer i R<sub>11</sub>

R<sub>11</sub>: 0x00 0B 00 55

INC, R<sub>8</sub>, R<sub>8</sub>

Øker verdien i R<sub>8</sub> med R<sub>8</sub> + 1

R<sub>8</sub> = 0xFFFF 0001

DEC R<sub>12</sub>, R<sub>12</sub>

Dekrementer verdien i R<sub>12</sub> med R<sub>12</sub> - 1

R<sub>12</sub> = 0x0000 0602

B/N2, R<sub>0</sub>

Om resultatet i fortige operasjoner ikke blir null, hopp til R<sub>0</sub> = PC

PC = 0x0000 FFFF

Kjører en gang

LOAD R<sub>1</sub>, R<sub>8</sub>

R<sub>1</sub> = 0x00 A4 00 06

ADD R<sub>11</sub>, R<sub>1</sub>, R<sub>11</sub>

R<sub>11</sub> = 0x00 AA 00 55

INC R<sub>8</sub>, R<sub>9</sub>

R<sub>9</sub>: 0xPFFF 0002

DEC R<sub>12</sub>, R<sub>12</sub>

R<sub>12</sub>: 0x0000 00 01

B/N2 R<sub>0</sub>

PC: 0x0000 FFFF

Kjort 2 gynger

Load R<sub>1</sub>, R<sub>8</sub>

R<sub>1</sub> : 0x00 00 00 AA

ADD R<sub>11</sub>, R<sub>1</sub>, R<sub>11</sub>

R<sub>11</sub> : 0x00 AA 00 FF

INC R<sub>8</sub>, R<sub>6</sub>

R<sub>8</sub> : 0x FFFF 00 03

DEC R<sub>12</sub>, R<sub>12</sub>

R<sub>12</sub> : 0

BNZ R<sub>12</sub>

Skjar ikke

R<sub>11</sub> : 0x 00 AA 00 FF

## Oppgave 5

a) Det betyr at det er 80% sannsynlighet for at data/inst er i cache.

b)  
① Det er 5 instruksjoner som skal utføres, disse må settes opp

5 klokkespenner for setup. 99 fil for  
+ 7 kjøring sum 106 instruksjoner.

$$\frac{1}{15 \cdot 10^{-9}} \cdot 104 = 1,56 \mu\text{s}$$

②

$$(7 + 99) \cdot \frac{1}{6 \cdot 10^{-9}} = 0,639 \mu\text{s}$$

c) Nei, det er for i de prosessornivå-parallelitet.