

## Oppgave 1

a) Med stored program computer, mener en maskin der program og data er lagret i samme minne. Det er i det samme minnet maskinen holder styr på tilgangen til data/instruksjoner. Maskinen henter instruksjona inn i en kontrollenhet, data overførs mellom minne og ALU.

- b)
- i) Den er synkron. Det er en klokke.
  - ii) opCode + PC11 = full adresse med all info.  
Port 0 er 8 bit. Port 2 er 8 bit.  
 $16 \text{ bits adressebuss} = 2^{16} \text{ adresser.}$
  - iii)  
Bit-fredde: 8 bit.  
opCode på port 0. er 8-bit.

c)  
i)

### Superskalær:

En superskalær implementerer en form for parallelitet som kallas instruksjonsparallelitet.

Fleire instruksjoner per klokkesyklus.

ii)

Ja, men det vil kreve ekstra legilde.

d)

### Harvard-arkitekter:

Separat minneområde for data og program.

Separate busser også.

e)

Ja, cache er en del av dette.

Cache bruker for å senke gjennomsnittlig  
aksesstid.

## Oppgave 2

a)

Full adder

b)  
i)

ROM :

0000 0000 XXXX XXXX

[0000, 00FF]

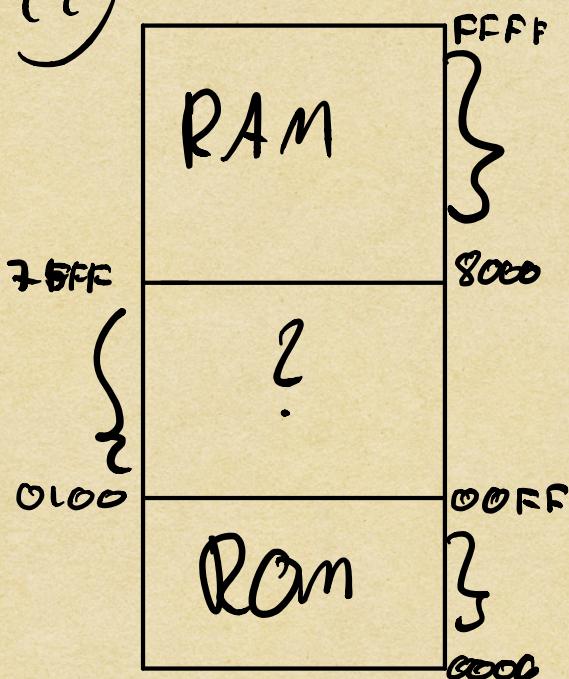
RAM:

1XXX XXXX XXXX XXXX

[8000, FFFF]

Ledig: [0100, 7FFF]

ii)



c)  $D_0 = \overline{Q_0} \cap X$        $Y = Q_0 \cap Q_1$

$$D_1 = (Q_1 \oplus Q_0) \cap X$$

| Y | Current |       | NEXT<br>$x=1$ |       | NEXT<br>$x=0$ |       |
|---|---------|-------|---------------|-------|---------------|-------|
|   | $Q_1$   | $Q_0$ | $Q_1$         | $Q_0$ | $Q_1$         | $Q_0$ |
| 0 | 0       | 0     | 0             | 1     | 0             | 0     |
| 0 | 0       | 1     | 1             | 0     | 0             | 0     |
| 0 | 1       | 0     | 1             | 1     | 0             | 0     |
| 1 | 1       | 1     | 0             | 0     | 0             | 0     |

d)

Det er nå en Mealy-maschin.

Her defineres output av current state og input. Maskinen er ikke lenger synkron med klokken.

Y er nå  $Q_0 \cap Q_1 \cap X$

### Oppgave 3

a) B-lys dekodoren tillater 305 i reduseert antall bit i mikroinstruksjonen, siden kun ett register kan ligge på bussen om gangen.

C-tressen kan dermed ha fleks register  
om gangen, og bruker derfor ett felt  
per registeren den kan skrive til

b)

1. ALU: B, C: H, MEM: DC, B: LV
2. ALU: A OR B, C: H, MEM: 000, B: TOS
3. ALU:  $\bar{A}$ , C: H og MDR, MEM: 600, B: 0

c)

0A3:  
 $J = 0$ , ALU: B, C: LV, MEM: 000, B: MDR, MPC = 0xA4  
LV = 0x4444

0A4:  
 $JZ = 1$ , ALU: A AND B, C: LV, MEM: 000, B: LV, MPC = 0x1A5

i 0A4 sin next address er 0 første 16bit.

Or'er med 2, 1 or 0 = 1.

Hopper til 1A5

LV = 0 ( $H = 0$ ) Z-flag = 1

1A5:

J: 0, ALU: B+1, C = LV, MEM: 000, R<sub>D</sub>: LV, MPC = 0x146

$$LV = 1$$

1A6:

J: 0, ALU: B, C: MDR, MEM: 000, R<sub>D</sub>: LV, MPC = 0x147

$$MDR = LV = 1$$

1A7:

J: 0, ALU: 0, C: 0, MEM: 000, R<sub>D</sub>: 0, MPC: 0x000

### Oppgave 4

a) Ja. Alle instruksjonene er 32 bit.

b) MOV

c)

|                                  |                                  |
|----------------------------------|----------------------------------|
| $R_0 = 0x0000\ 0000\ FFFF$       | $R_{11} = 0x0000\ 0000\ 0000\ 0$ |
| $R_8 = 0xFFFF\ 0000\ 0000\ 0000$ | $R_{12} = 0x0000\ 0000\ 0000\ 3$ |
| $R_{13} = 0x0001\ 0000\ 0000\ 5$ |                                  |

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

$$R_1 = 0x00\ 00\ 00\ 01$$

ADD R<sub>28</sub>, R<sub>1</sub>, R<sub>i</sub>

$$R_{28} = 0x00\ 00\ 00\ 02$$

BZ R<sub>3</sub>

NO BRANCH

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

R<sub>11</sub> = 0x00 00 00 02

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

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

BNZ R<sub>0</sub>

! Brancher til 0x0000 FFFF !

Kjørte 1 gang

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

R<sub>1</sub> = [R<sub>8</sub>] = 0x00 00 00 02

⋮

Koden legges 5 ganger og gir R<sub>11</sub> = 0xA

∴ R<sub>8</sub> = 0xFFFF 0004

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

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

ADD  $R_{28}, R_1, R_8$

$$R_{28} = 0, Z=1$$

B2 n13

Brancher.

Stør R<sub>11</sub>, R<sub>8</sub>

$$R_{11} = \underline{0 \times 0} \quad (\text{lagres på } [n_8])$$

### Oppgave 5

a)

SY begrenser. Ser av tegrivjen at det er latches der. kan innføre pipeline.

$$\text{MAX FART: } \frac{1}{25\text{ns} \cdot 1000} = 40 \text{ MHz}$$

b)

c) Ved innføring av A-bus blir det flere mikro-instruksjoner. Dette fordi det ikke trengs så mellomligge verdier.

ii)

Ja. Instruksjonene må skrives om til å bruke begge busser, kontra å bruke lit av B-bussen.

c) Moores lov viser at det hvert 2. år (ca.) er en dobling av transistorer på en chip

d) klokkefrekvensen kan sinkes, fordi instruksjonene kjøres av flere prosessorer