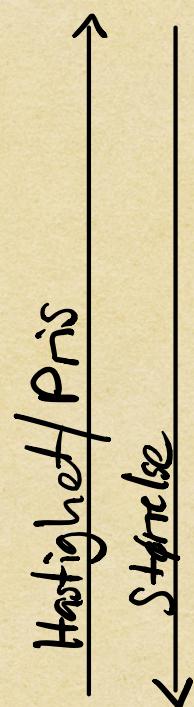
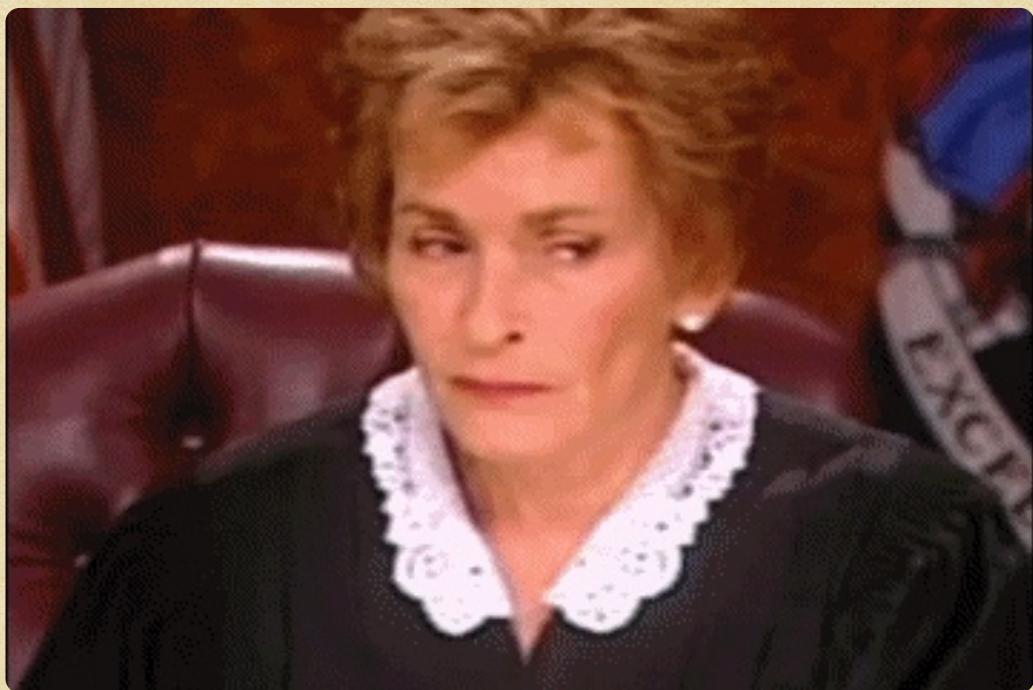


## Oppgave 7

- a) Register  
Cache  
Main Memory  
Magnet Disk  
Optical Disk



b)



c) i) Jo dypere pipelinen er, desto flere steg er det. Jo flere steg man har, desto flere steg kan kjøres i parallelle → graden av ILP øker med dybden.

ii) Dersom man brører og ikke treffer pipelinen, si vi den finnes og fylles på ny. Dette tar tid og "kaster bort" klokke sykler.

d) Minnot myter veggan.

Prosessorthastigheten øker raskere en akcessiden til minnet. Dette sjør at vi ikke får utnyttet prosessoren sin full kraft!

Minnot blir en flaskehals.

e) i) Areal  
Energiforbruk  
Akcesstid

SRAM	DRAM
Stort	Litt
Stort	Litt
Kort	Lang



## Oppgave 2

a)

### Synkron:

- Følger klokke
- Instruksjoner er tinet av klokken

### Asynkron:

- Handshake,
- Ingen klokke

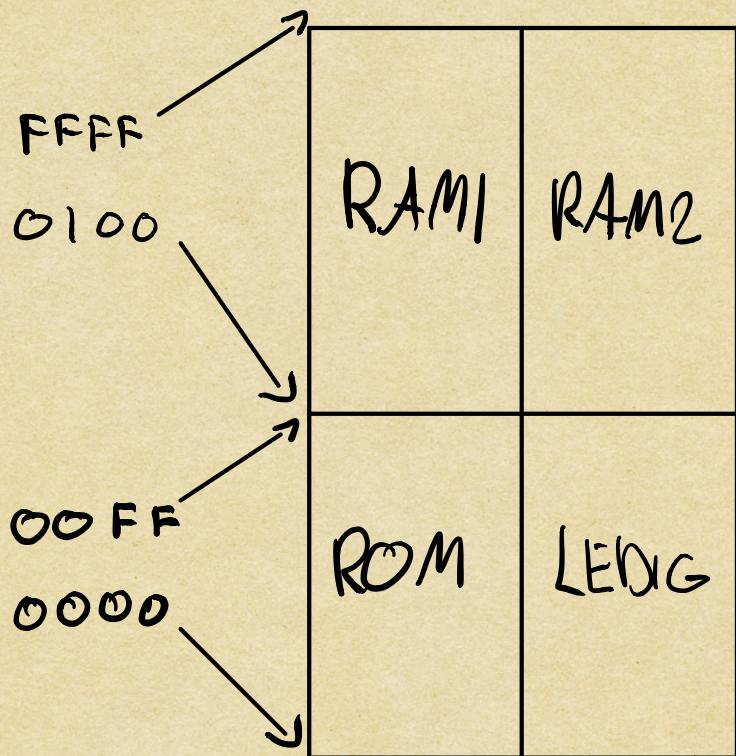
b)  
c) RAM 1 = RAM 2 = RAM

X X X Y    X X X I    X X X X    X X X X

[0100, FFFF]

### ROM:

[0000, 00FF]



ii)

ROM kan adressere  $0x00FF + 1 = 0x0100$  bytes  
 $= 16^4 = 256$  bytes.

RAM kan adressere  $(0xFFFF + 1) - ROM(0x0100)$

$16^4 - 256 = 65280$  bytes. Siden det er to  
 ROM-mikker:  $2 \cdot 65280 = 130560$  bytes.

c)

$$D_0 = \overline{Inn} \cap \overline{Q_0}$$

$$D_1 = (Q_1 \oplus Q_0) \cap \overline{Inn}$$

iii) current

		$Next^1$ $Inn = 1$		$Next^1$ $Inn = 0$	
		$Q_1$	$Q_0$	$Q_1$	$Q_0$
$Q_1$	$Q_0$				
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
1	1	0	0	0	0

Tabellen matcher

tilstansdiagrammet,  
 og siden jeg er best  
 betyr det at det  
 er rigtig !!

## Oppgave 3

- a) Conditional branches. Når det skal hoppes i koden, angir JMPC til ikke programstier
- b) CISC. Få generelle register og instruksjoner. Kompliserte instruksjoner og ingen load/store-arkitektur

c) Se a

- d)
- i) M1 1: ALU: B , C: H MEM: 000 , B: OPC  
M1 2: ALU: + , C: H MEM: 000 , B: TOS  
M1 3: ALU: + , C: H MEM: 000 , B: CPP  
M1 4: ALU: + , C: H MEM: 000 , B: LV  
M1 5: ALU: B-A, C: OPC MEM: 000 , B: SP

ii) Trenger ikke mellomlagre OPC.

iii)  $H = 0X00 \quad 00 \quad 00 \quad FF$  Arkitektur 2:  
 $OPC = 0X00 \quad 00 \quad 00 \quad 01 \quad H = TOS + OPC = 1+2=3$   
 $TOS = 0X00 \quad 00 \quad 00 \quad 02 \quad H = H + CPP = 3+3=6$   
 $CPP = 0X00 \quad 00 \quad 00 \quad 03 \quad H = H + LV = 6+4=10=1A$   
 $LV = 0X00 \quad 00 \quad 00 \quad 04 \quad OPC = SP - H = A - A = 0$   
 $SP = 0X00 \quad 00 \quad 00 \quad 0A$   
 $Z = 1, \quad N = 0$

### Oppgave 9

a)

Anta at man skal subtrahere Register A og B.  
Da har man  $A + (nB + 1)$

b)

Det er avhengigheter.

I R<sub>i</sub> mi vi passer på at registeret  
er ferdig lest før det skrives til, og føder  
skrevet til før det leses fra.

RAW, WAR

E)

$$R_0 = 0xFFFF\ 0000$$

$$R_{16} = 0x0001\ 0009$$

$$R_{17} = 0xFFFF\ 0010$$

$$R_{18} = 0x0001\ 0003$$

$$R_{19} = 0xFFFF\ EFFF$$

1. Flyttar konstanten 0x00 00 inn i R<sub>10</sub>

$$R_{16} = 0x00\ 00$$

2. Flyttar konstanten 0x00 00 inn i R<sub>9</sub>

$$R_9 = 0x00\ 00$$

3. Legger [R<sub>0</sub>] inn i R<sub>2</sub>

$$R_2 = 0xFF\ FF\ F0\ 00$$

4. Legger [R<sub>2</sub>] inn i R<sub>1</sub>

$$R_1 = 0x00\ 00\ 00\ 03$$

5. Sammenligner innholdet i R<sub>9</sub> og R<sub>1</sub>

$$Z = 0$$

6. Hopper til instruksjon lik [R<sub>16</sub>] om Z=1

Hopper ikke, da Z=0

7. Legger  $[R_{17}]$  inn i  $R_8$

$$R_8 = 0x00\ 00\ 00\ 00\ 01$$

8. Legger resultatet av addisjoner  $R_8 + R_9$  i  $R_9$

$$R_9 = 0x00\ 00\ 00\ 00\ 01$$

9. Dekrementerer  $R_i$  med  $R_i - 1$

$$R_i = 0x00\ 00\ 00\ 00\ 02$$

10. Inkrementerer  $R_{17}$  med  $R_{17} + 1$

$$R_{17} = 0xFFFF\ 00\ 11$$

11. Sjekker  $R_i = R_{10}$

$$Z \neq 0 \quad Z = 0$$

12. Brancher til instruksjon  $[R_{18}]$  om  $Z = 0$

Brancher til punkt 7.

7. Legger  $[R_{17}]$  i  $R_8$

$$R_8 = 0x00\ 00\ 00\ 00\ 02$$

8. Legger resultatet av addisjoner  $R_8 + R_9$  i  $R_9$

$$R_9 = 3$$

9. Dekrementerer  $R_i$  med  $R_i - 1$

$$R_i = 0x00\ 00\ 00\ 00\ 01$$

10. Inkrementerer  $R_{17}$  med  $R_{17} + 1$

$$R_{17} = 0xFFFF\ 00\ 12$$

11. Sjekker  $R_1 = R_{10}$

$$1 \neq 0 \quad Z = 0$$

12. Brancher til instruksjon  $[R_{13}]$  om  $Z = 0$

Brancher til punkt 7.

7. Legger  $[R_{17}]$  i  $R_8$

$$R_8 = 0x80\ 00\ 00\ 00\ 03$$

8. Legger resultatet av addisjonen  $R_8 + R_9$  i  $R_9$

$$R_9 = 6$$

9. Dekrementerer  $R_1$  med  $R_1 - 1$

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

10. Inkrementerer  $R_{17}$  med  $R_{17} + 1$

$$R_{17} = 0xFFFF\ 00\ 13$$

11. Sjekker  $R_1 = R_{10}$

$$0 = 0 \quad Z = 1$$

12. Brancher til instruksjon  $[R_{13}]$  om  $Z = 0$

Brancher ikke.

13. Lagrer  $[R_9]$  i  $R_{19}$

$$0xFFFF\ EFFF = 6 \quad R_9 = \underline{\underline{6}}$$

d)  $0xFFFF\ F000 = 0xFFFF\ 0004$

Løkken vil kjøre én gang mer.