IN1020 uke 4

GRUPPE 6

Plan for dagen

- Menti
- Oppsummere alt det viktigste vi har vært gjennom
- Eksamensoppgaver
- Ukesoppgaver og obligjobbing

Digital representasjon

- Hvordan lagrer vi data?
  - Tall
  - Tekst
  - Bilder
  - Lyd
- Hva er et bit?
  - 0 eller 1
  - False eller True
  - Rød eller grønn
  - Lys av eller på
  - osv
- Binære tall (binary digit, bit)
  - Bruker 0 og 1 for å representere andre tall
  - Telle binært
- Fremgangsmåte for å telle binært:
  - Øk siste siffer i tallet med 1
  - Hvis det ikke er flere sifre, sett siste til 0, og gjenta for sifferet til venstre
  - Hvis på tavlen My!

Notasjon

- Vi bruker notasjon for å vite hvilken base tallet er i
  - 1001 – er det et binærtall eller et desimaltall?
  - 1001_2 og 1001_{10} gjør skillet tydelig
  - $1001_2 = 9_{10}$

OBS: Selve basen skrives alltid i det desimale systemet

- $1001_{10} = 3E9_{16}$
- OBS: 1001_{10} = 0x3E9

25_{10} til binært?

Verdi	Rest
25	1
12	0
6	0
3	1
1	1
0	

Fremgangsmåte:

- Dele tallet på 2 og se hva vi får i rest:
  - Kan få 0 i rest eller 1
  - Ettersom det er et odde eller partall

Svaret leser vi nedenfra: $25_{10} = 11001_2$

11001_2 til desimal?

Posisjon:	4	3	2	1	0
	1	**1**	**0**	**0**	**1**

$1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 0*2^0$

$= 16 + 8 + 0 + 0 + 1$

$= 25$

Heksadesimal notasjon

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Rad 1: Desimal (base 10)

Rad 2: Binær (base 2)

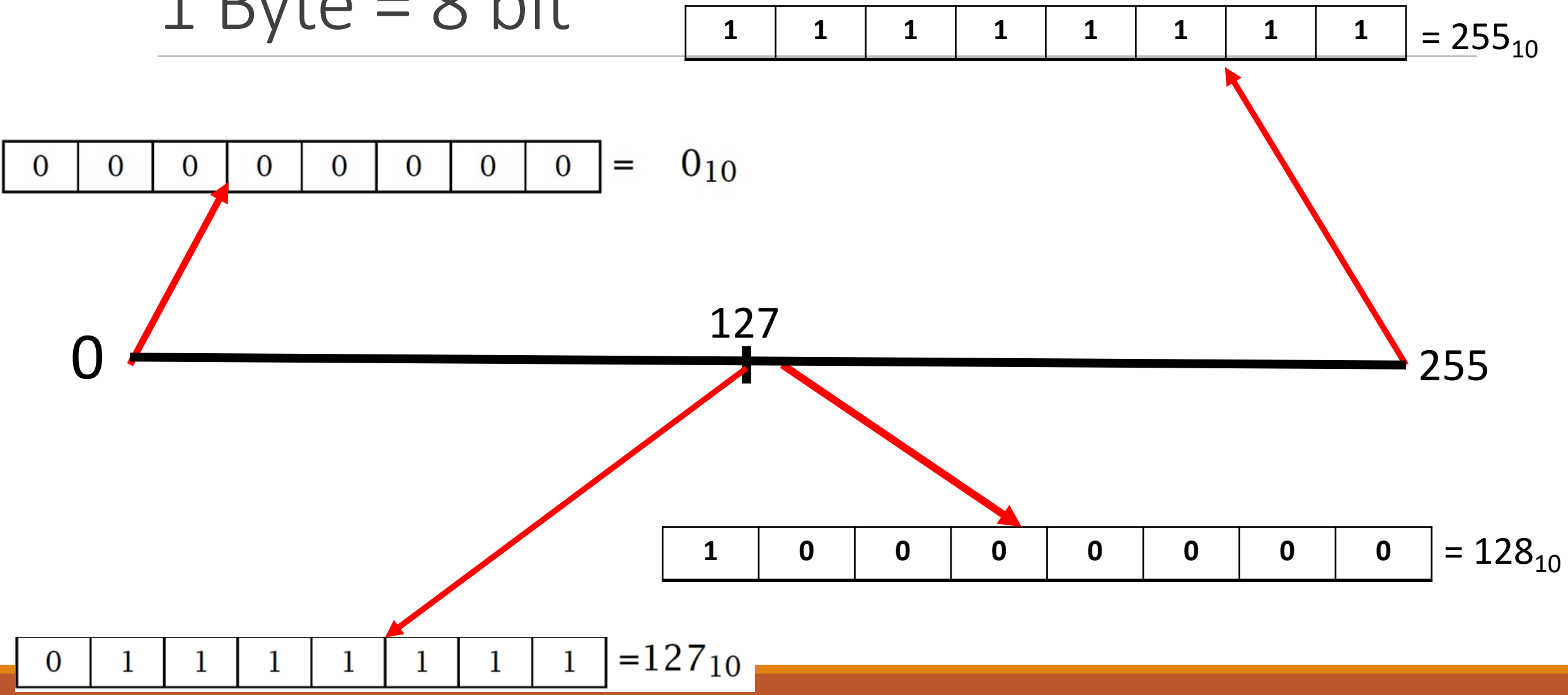
Rad 3: Heksadesimal (base 16)

$1\ 000\ 000_{10} =$

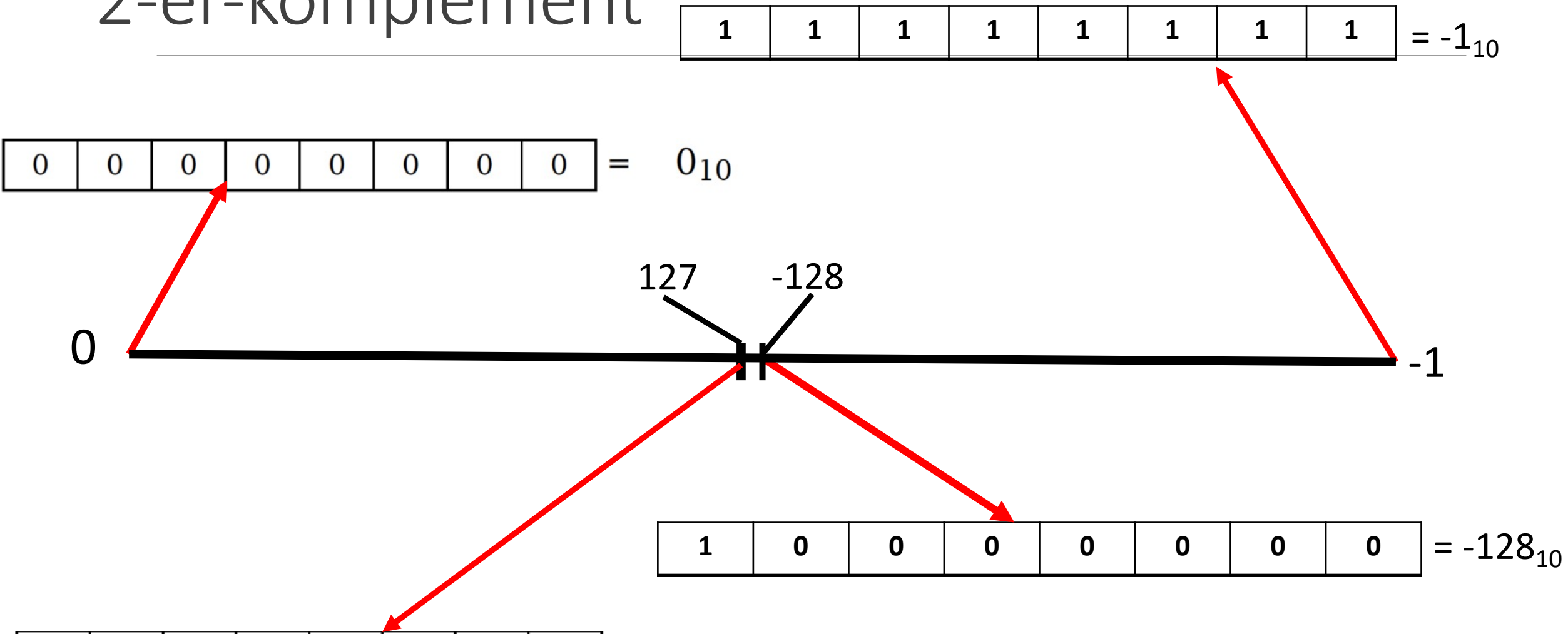
$1111\ 0100\ 0010\ 0100\ 0000_2 =$

0x F 4 2 4 0

1 Byte = 8 bit


1 1 1 1 1 1 1 1 = 255₁₀
0 0 0 0 0 0 0 0 = 0₁₀
0
127
255
1 0 0 0 0 0 0 0 = 128₁₀
0 1 1 1 1 1 1 1 =127₁₀

2-er-komplement



Regning med to-er-komplement

Fra binær til desimal

1	0	1	0	1	0	1	0

$-2^7 + 2^5 + 2^3 + 2^1$

-128 + 32 + 8 + 2 = -86

1	1	1	0	1	0	0	1

$-2^7 + 2^6 + 2^5 + 2^3 + 2^0$

-128 + 64 + 32 + 8 + 1 = -23

Fra decimal til binær

F.eks gjør om –50 til binær

1. Bytte fortegn slik at -50 blir til 50
2. Finne binære tallet for 50
3. Inversere det binære tallet
4. Legge til 1 på slutten av binære tallet

Verdi	Rest
50	0
25	1
12	0
6	0
3	1
1	1
0	

= 00110010

inverserer ↓

= 11001101

= 11001101

\+ 1

= 11001110_2 = -50_{10}

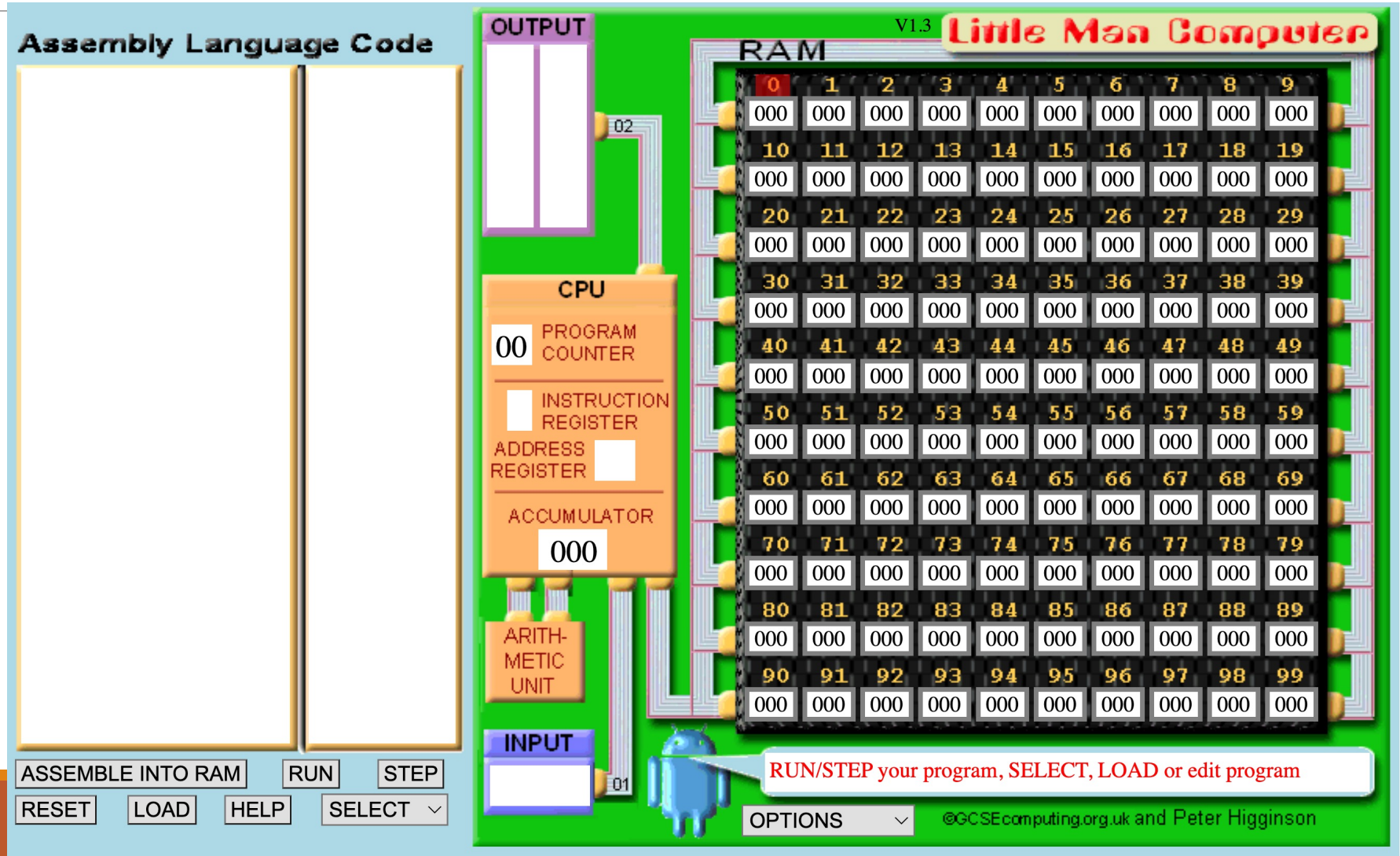
Legger til 2 nuller foran for å få en byte

= 110010

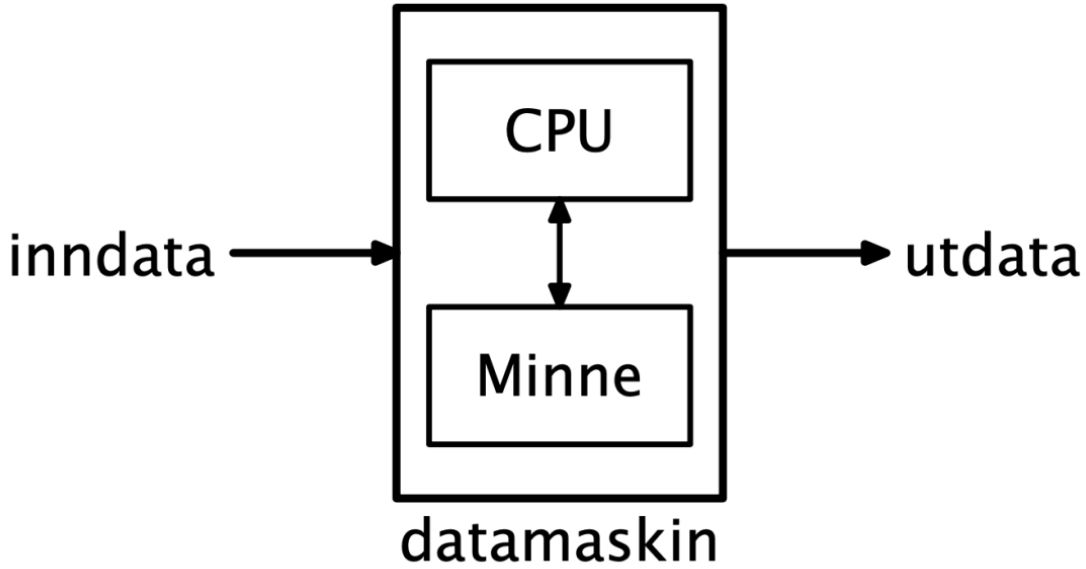
= 00110010

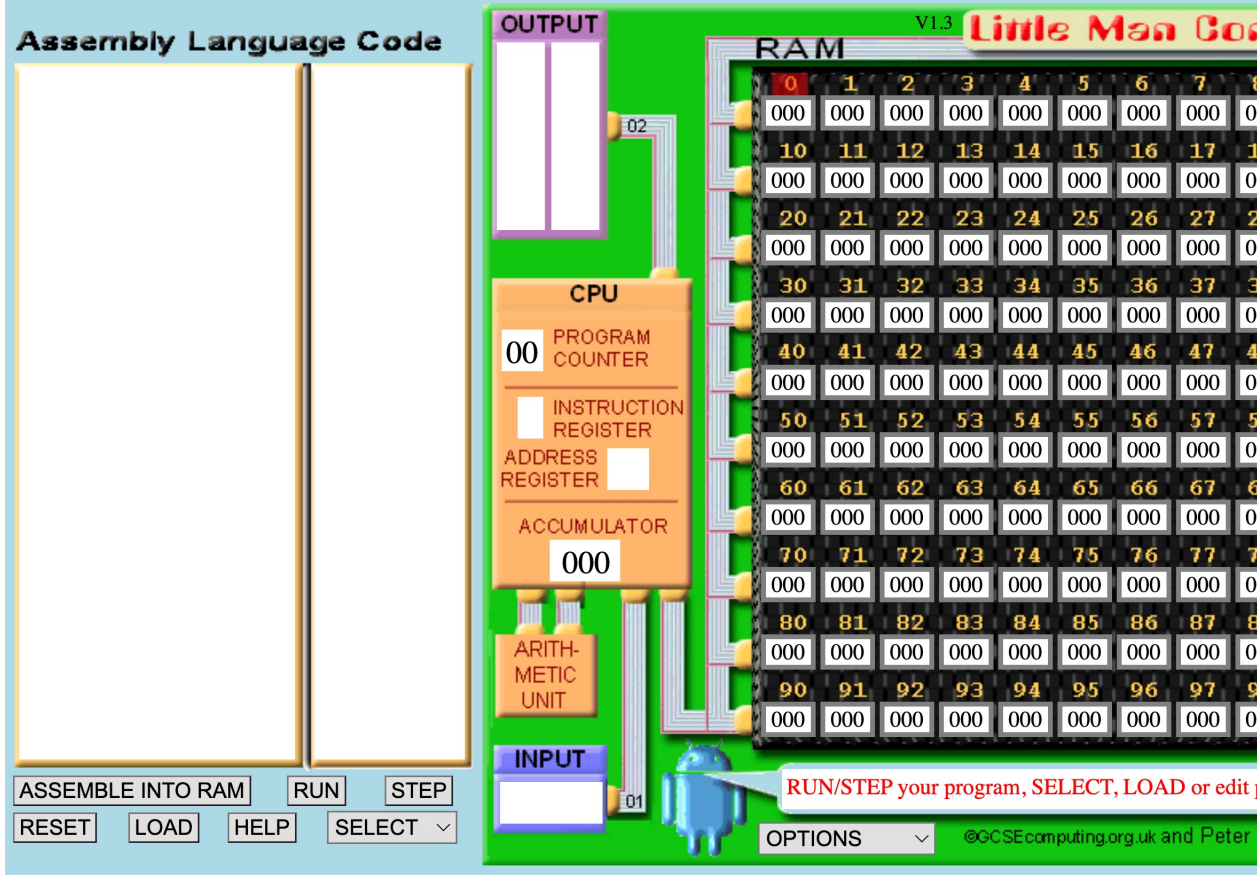
LMC – Little man computer

- Minne (RAM)
- OUTPUT
- INPUT
- CPU
- ALU (Arithmetic Logic Unit)
- Assembly code



Von Neumann-arkitekturen



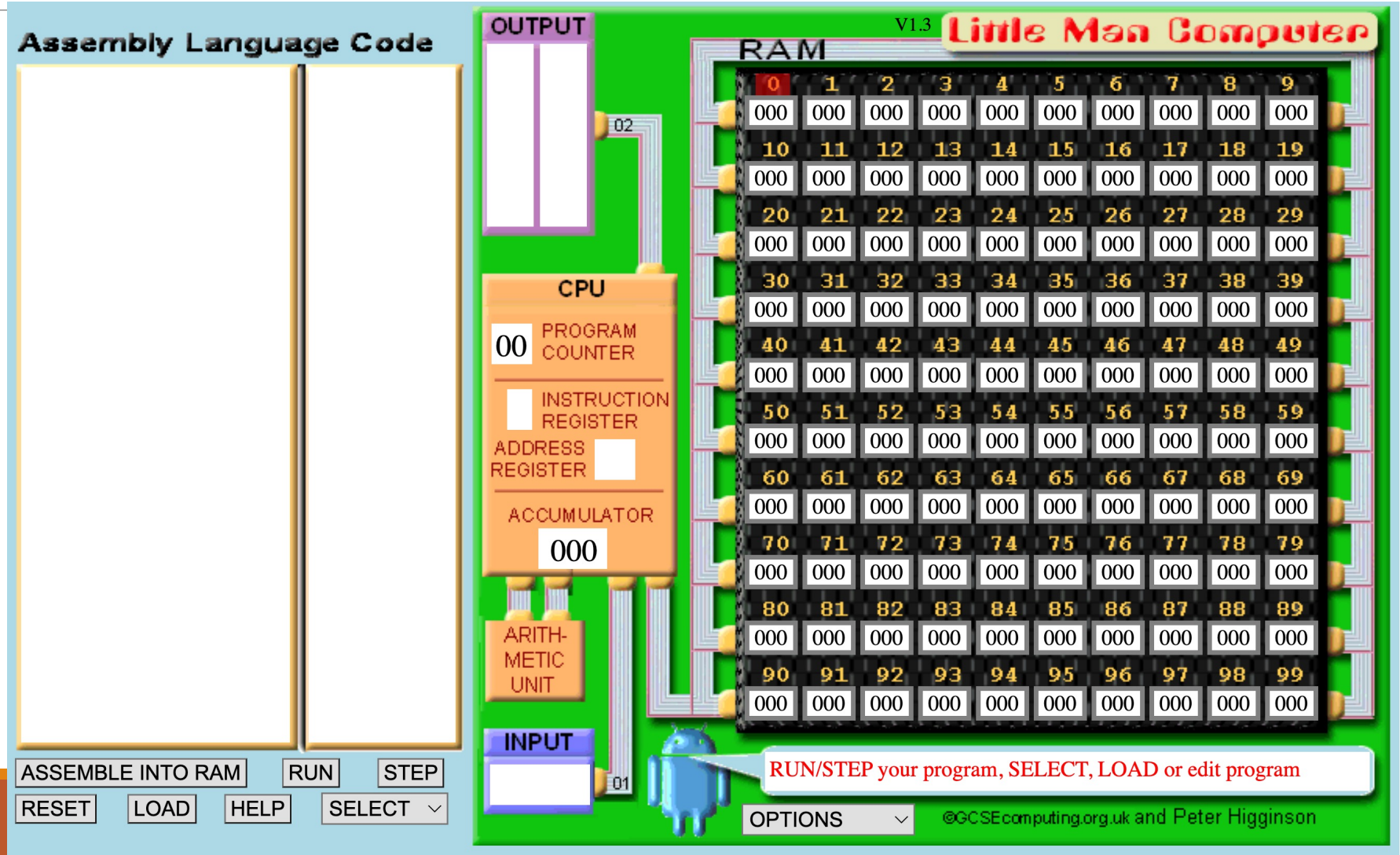


Den store erkjennelsen er at minnet holder på både *data* og *instruksjoner* til CPU-en (altså *programmet*)

Minne

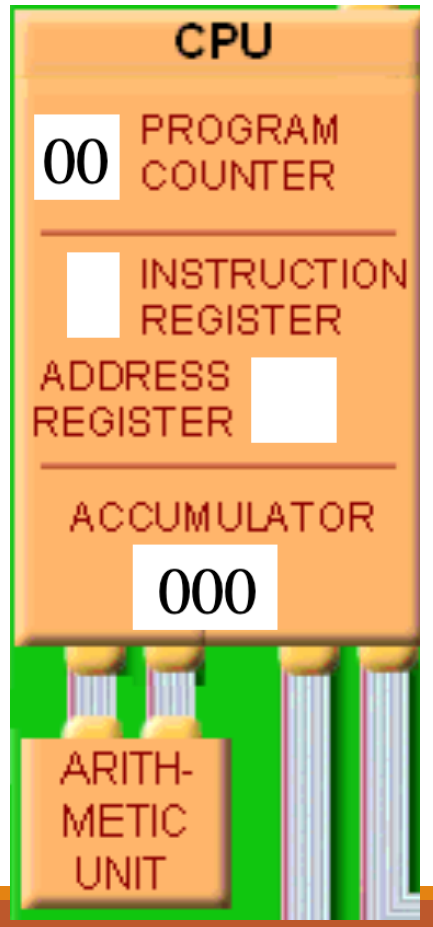
- 99 hyller
- Hylle inneholder:
  - Instruks
  - Data

- Minne = Oppskrift
- CPU = Den som bruker oppskriften



CPU – Central processing Unit

- Inneholder ALU (Arithmetic Logic Unit):
  - ALU kan utføre enkle operasjoner som pluss og minus
- Register:
  - Program Counter:
  - Instruction Register
  - Address Register
  - Accumulator
- Kontrollogikk til å utføre ting i riktig ordre



Alle instruksjonene i LMC

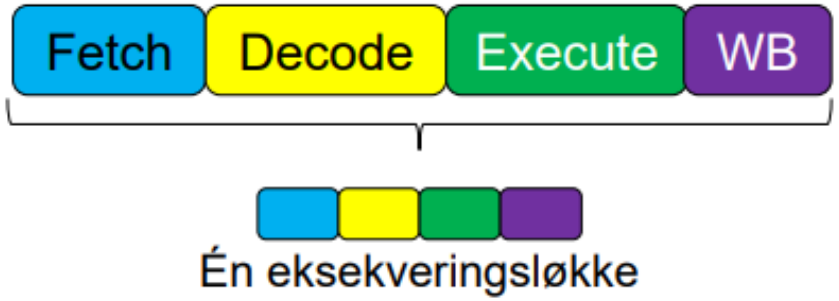
Kode	Navn	Beskrivelse
0xx	HLT	Stopper eksekveringen
1xx	ADD	Adderer verdien i angitt minnelokasjon med akkumulatoren
2xx	SUB	Subtraherer verdien i angitt minnelokasjon med akkumulatoren
3xx	STA	Lagrer akkumulatoren i angitt minnelokasjon
4xx	-	Ikke i bruk
5xx	LDA	Henter verdi fra minnet til akkumulatoren
6xx	BRA	Hopper til angitt adresse
7xx	BRZ	Hopper hvis akkumulatoren er 0
8xx	BRP	Hopper hvis akkumulatoren er ≥ 0
901	INP	Leser verdi fra input, og legger svaret i akkumulatoren
902	OUT	Skriver ut verdien i akkumulatoren
922	OTC	Skriver ut ASCII-tegn (ikke i boka)

Eksekveringsløkken

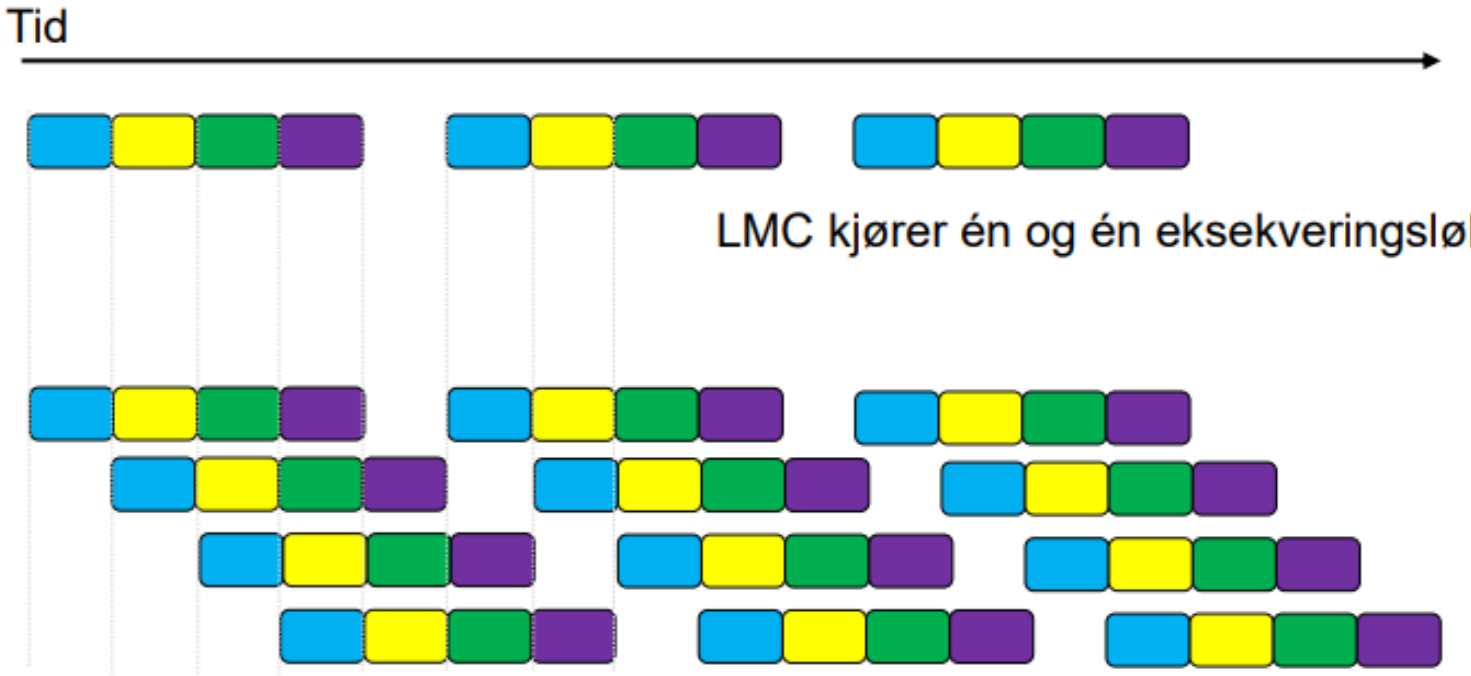
Både LMC og x86-64 har samme eksekveringsløkke:

1. Hent instruksjonen i minnet; programtelleren angir hvor.
2. Dekod instruksjonen i operasjon og adresse; de legges i instruksjonsregisteret og adresseregisteret.
3. Utfør instruksjonen.
4. Om aktuelt, skriv svaret i minnet/register.

x86-64 har i tillegg pipeline for å kunne jobbe raskere.



Pipelining



LMC kjører én og én eksekveringsløkke

x86-64 kjører flere eksekveringsløkker i parallell «pipelining»

3(a) Binære tall 1

Verdien $\mathbf{27_{10}}$ (dvs 27 i 10-tallsystemet) kan også representeres i andre tallsystemer. Hvilke av disse verdiene er lik 27_{10}?

Velg ett eller flere alternativer:

- 102_5
- 123_4
- 1000_3
- 11001_2

Maks poeng: 3

3(c) Bit og byte

En byte inneholder disse bit-ene:

0	1	1	1	0	0	0	1

Hvilke verdier kan representeres av disse bit-ene?

Velg ett eller flere alternativer

- 97
- 113
- -143
- 65

Maks poeng: 3

2(d) Maskinkode 2

```
          start   LDA     x
                  BRZ     slutt
                  SUB     y
                  STA     x
                  INP
                  STA     w
                  LDA     z
                  SUB     w
                  STA     z
                  BRA     start
          slutt   LDA     z
                  OUT
                  HLT

          w       DAT     0
          y       DAT     1
          x       DAT     3
          z       DAT     100
```

Hva skrives ut når denne koden kjøres og brukeren oppgir de tre verdiene **1**, **2** og **3** som inndata?

Velg ett eller flere alternativer (men bare ett er korrekt):

- [ ] 6
- [ ] Noe annet enn de andre alternativene
- [ ] 100
- [ ] 106
- [ ] 94

Maks poeng: 4

Takk for i dag!

- Jobbe videre med oblig eller ukesoppgaver
- Send meg mail eller teamsmelding hvis dere lurer på noe eller trenger hjelp til oblig