IN1020 Høst 2023

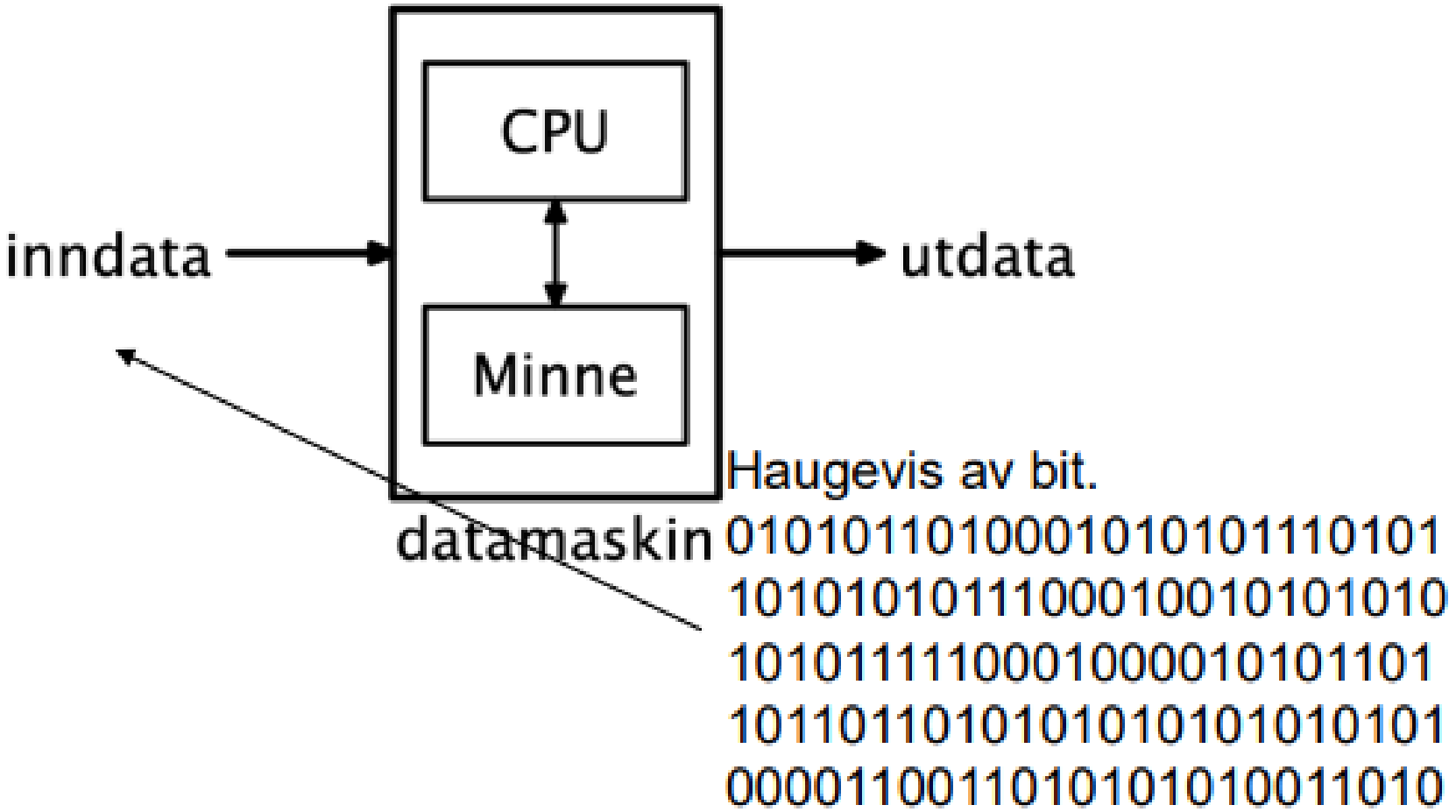
Gruppe 13

shahiaf@uio.no

Shahi Ftah



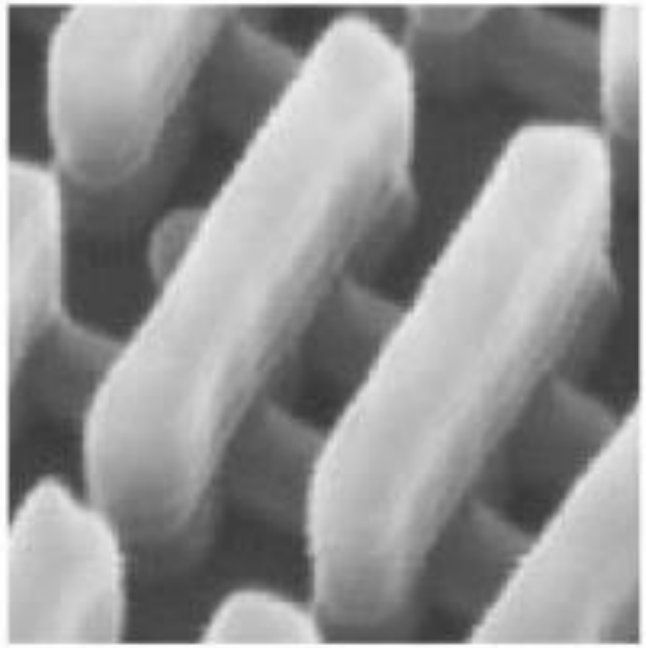
Von Neumann-arkitektur



Hva er en bit?

- Representerer «noe» som har to tilstander

0	1
False	True
Rød	Grønn
Lys av	Lys på
0 volt	5 volt



Binærtall eller desimaltall?

1001 – er det et binærtall eller et desimaltall?

1001_2 og 1001_{10} gjør skillet tydelig

$1001_2 = 9_{10}$

Sifrene har vekt i henhold til posisjon

I vårt vanlige desimale tallsystem har posisjonene vekt
1 10 100 1000 osv.

- I det binære tallsystemet har posisjonen tilsvarende vekt
  1 2 4 8 osv.
  Dette tilsvarer
  2^0 2^1 2^2 2^3 osv.

Fra binært til Desimal

Dette er posisjonen til sifferne:	4	3	2	1	0
La oss si at vi har følgende binære tall:	1	1	0	0	1
Dette er vekten til posisjonene:	2^4	2^3	2^2	2^1	2^0
Gang hvert siffer med sin vekt:	$1*2^4$	$1*2^3$	$0*2^2$	$0*2^1$	$1*2^0$
Det blir:	16	8	0	0	1

$$16 + 8 + 0 + 0 + 1 = 25$$

Fra desimal til binært

Verdi	Rest
25	
12	(25-2*12) = 1
6	(12 - 2*6) = 0
3	(6 – 2*3) = 0
1	(3 - 2*1) = 1
0	(1 - 2*0) = 1

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

Heksadesimal

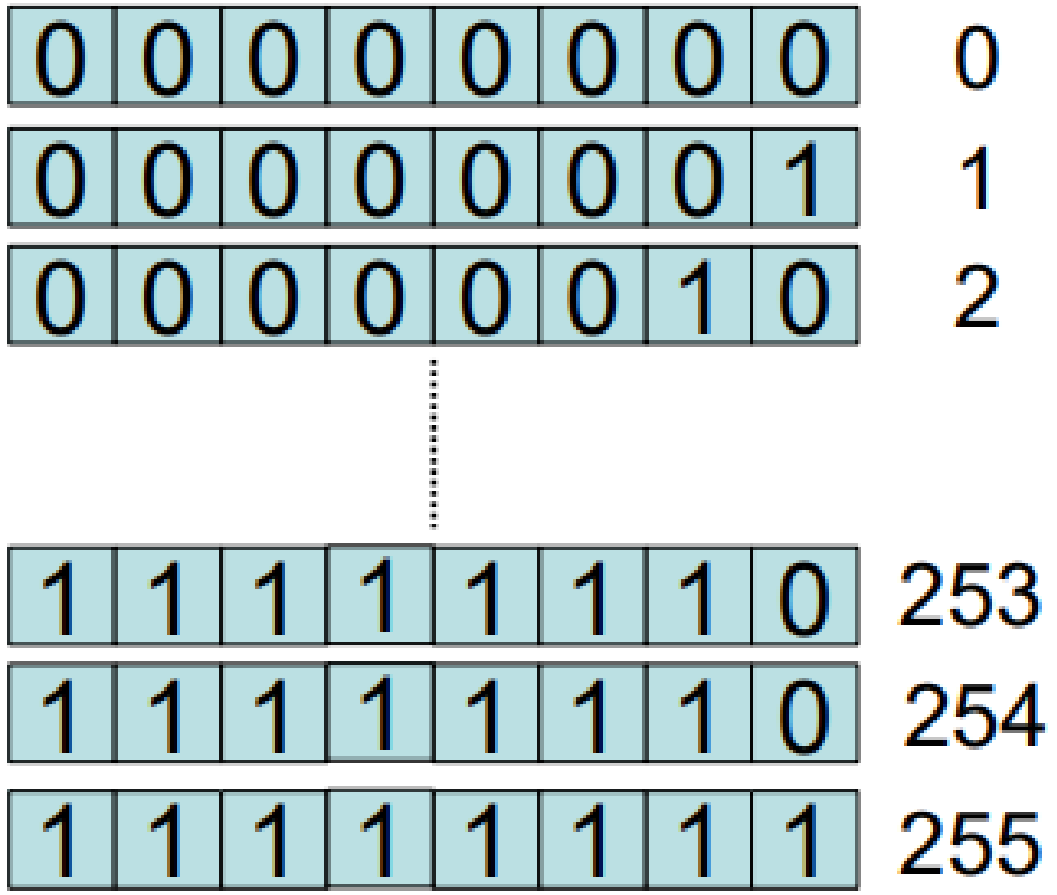
- Samler 4 og 4 bit
- Ox = hex

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

$$11110100001001000000_2$$

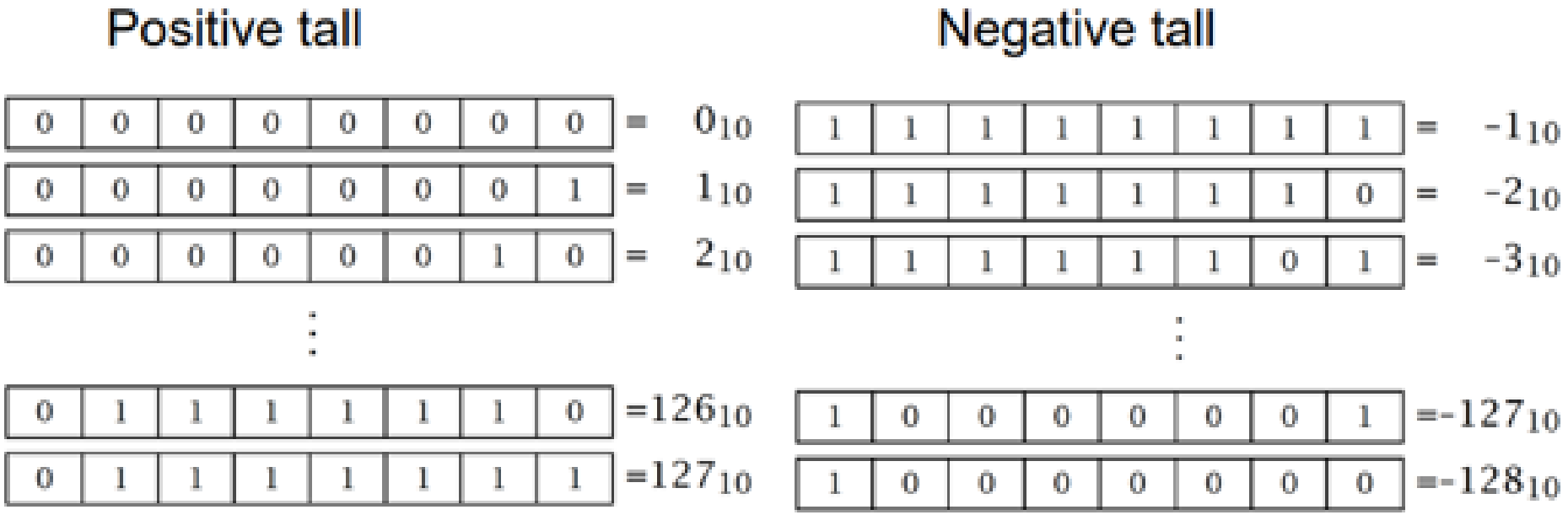
Desimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binær	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

- En Byte = 8 bit
- Vi kan telle fra 0 til 255 med én byt



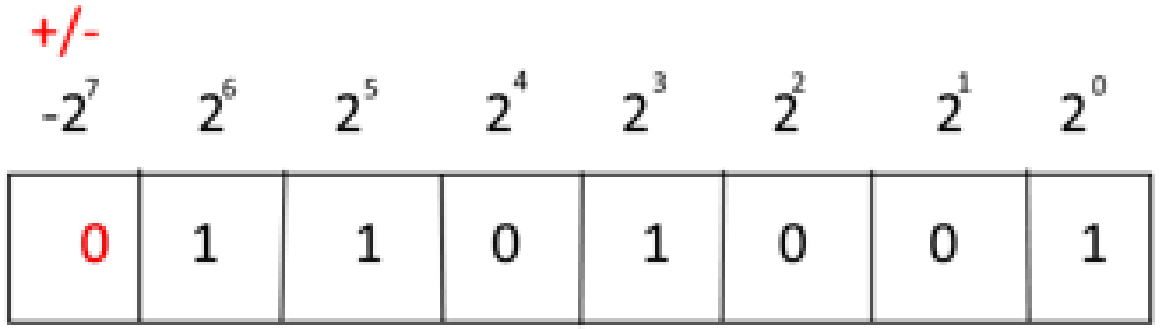
Negative tall/2-er-komplement

- Hvordan representerer vi negative tall?
- Fortegnsbit

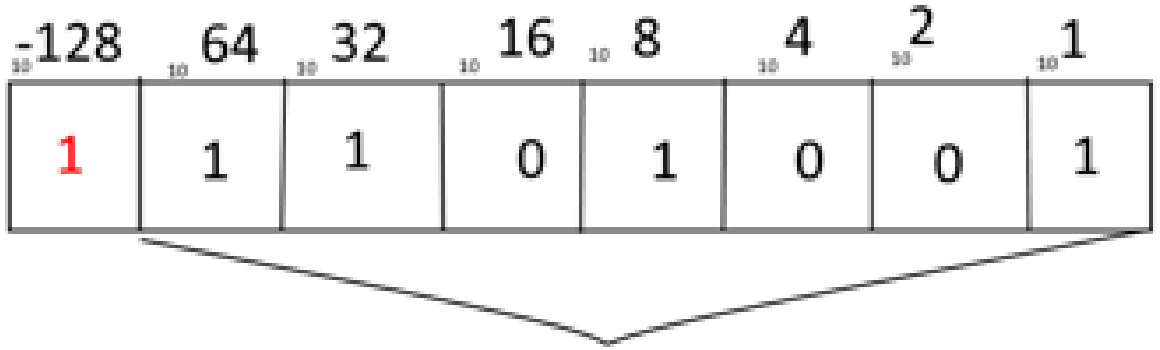


2-er-komplement

8 bit:



=0*(-2^7)+1*(2^6) + 1*(2^5)+0*(2^4)+1*(2^3)+0*(2^2)+0*(2^1)+1*(2^0)
=64+32+8+1=107



-128 +107

Verdien på det nederste tallet blir

$-128_{10} + 107_{10} = -21_{10}$

Hvordan lagrer vi tegn?

- Maskinen husker bare 1 og 0
- ASCII
- Lett å bruke
- Lett å se om siffer

- Eks 0x45 = «E»

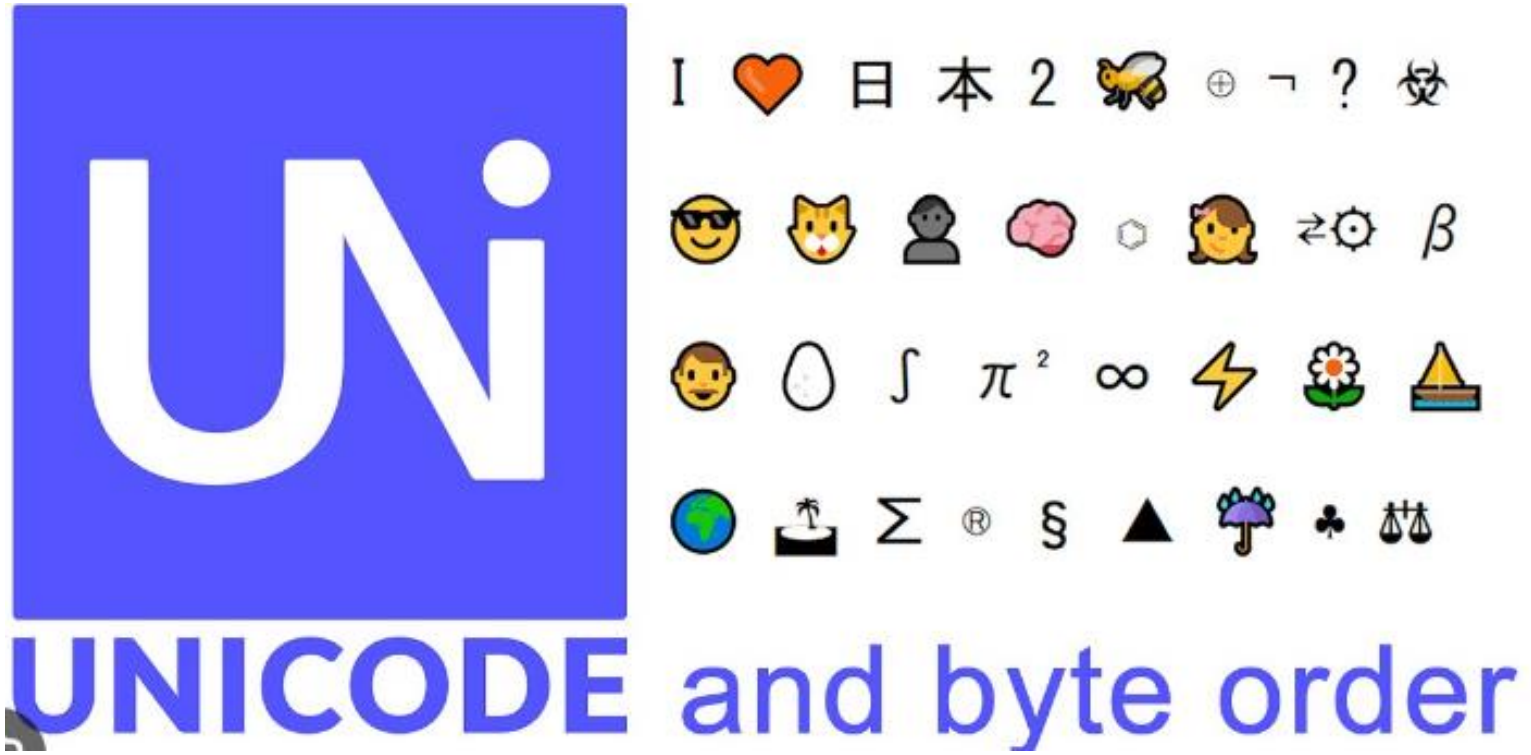
ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

- Var vellykket, men mangler æøå

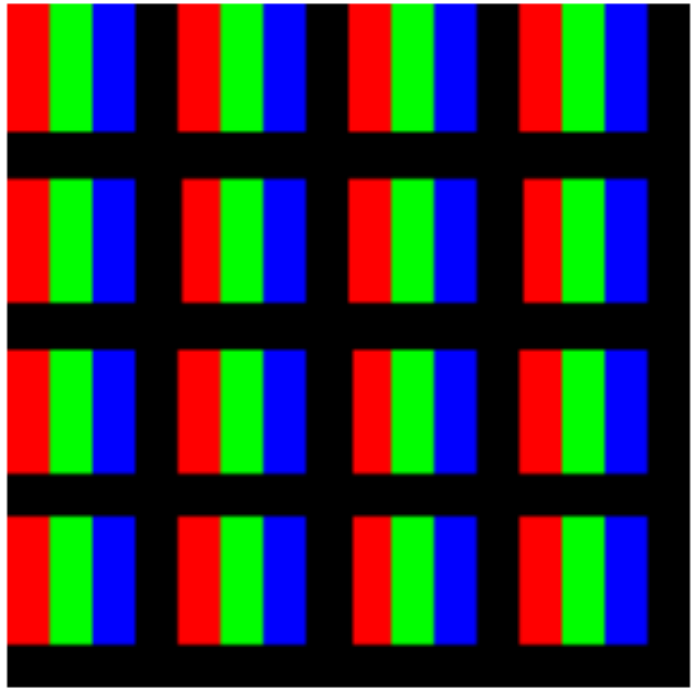
Unicode

- Den ordentlige løsningen
- Alle skriftspråk i verden
- Løsningen som brukes nå



Bilder

- Delt opp i pixler
- Hver pixel består av 3 verdier, rød, grønn, blå (primærfarger)
- Hver delverdi av en pixel går fra 0 til 255, angir intensitet til farge

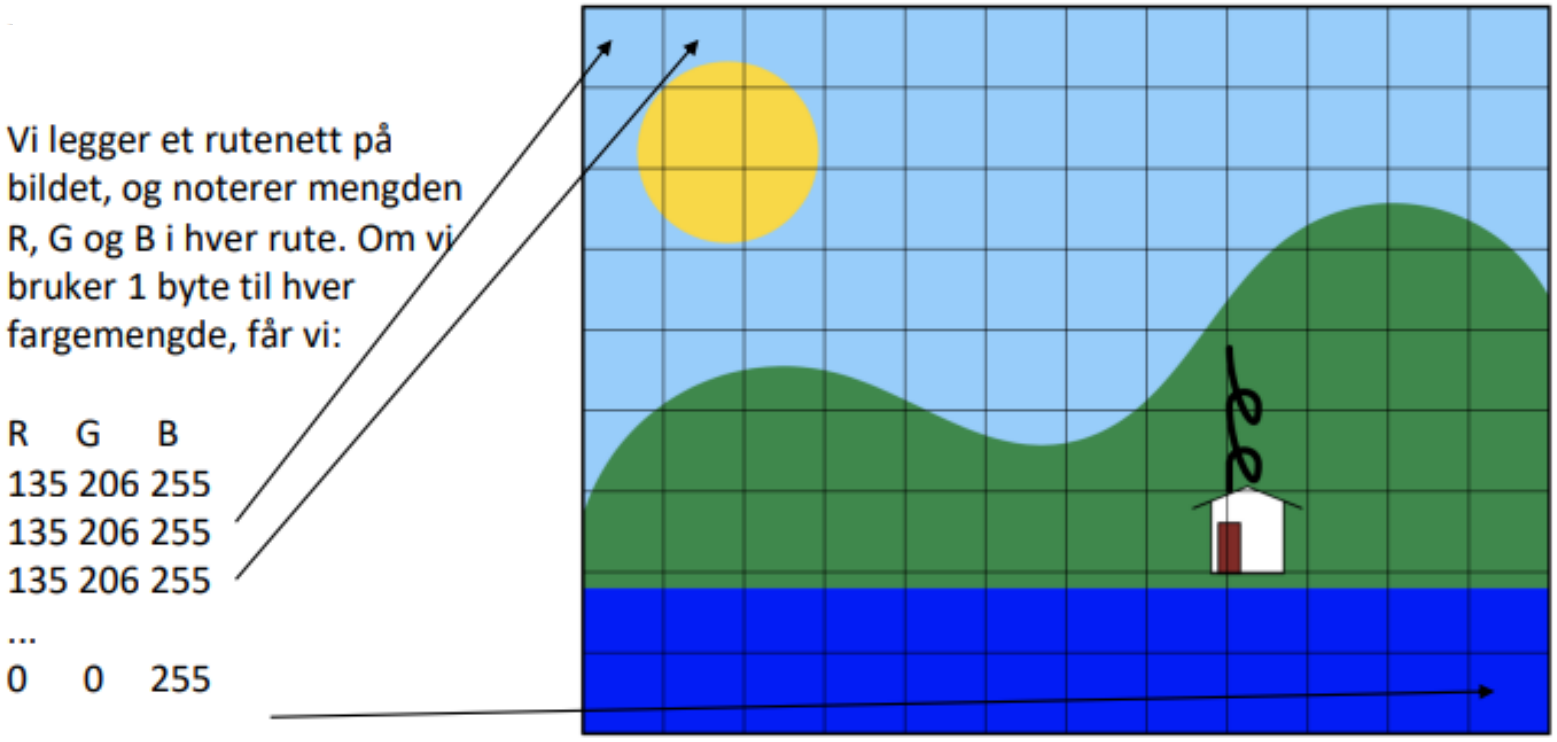


Bilder

Flere pixler = finere bilde

Mindre plass = kompilere (tap)

Dele farger i fargetabell



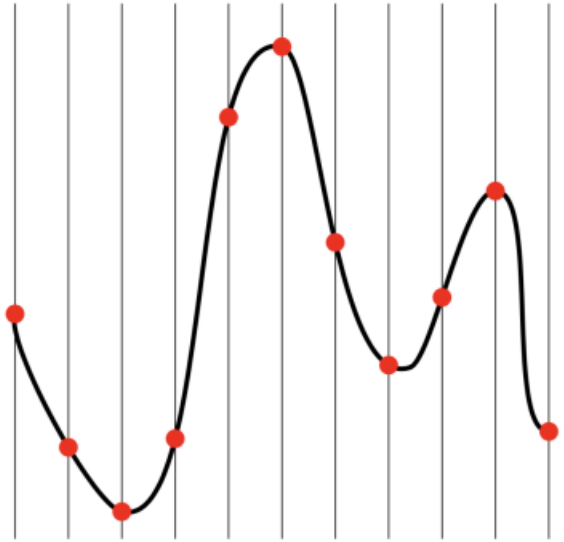
Lyd

- Bølger i luften, som overføres som strøm
- Kvalitet avhenger av antall målinger og trinn

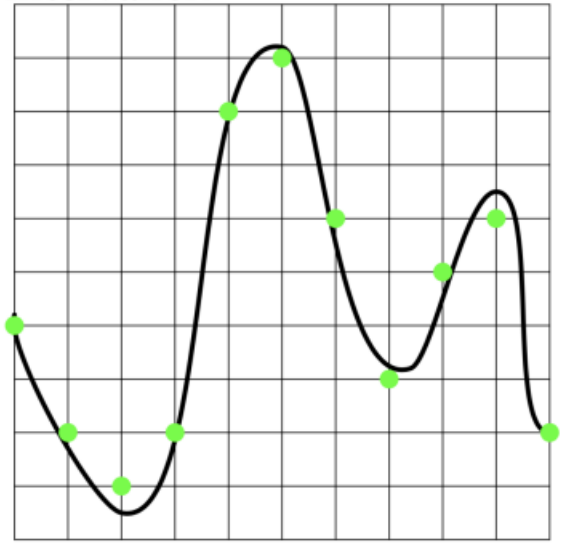
Moderne standarder som **MP2** (brukt i DAB), **MP3** og **AAC** (brukt i DAB+, YouTube, iTunes, . . .) utnytter dette og tillater til dels sterk komprimering:

CD-kvalitet	1410 kb/s
MP2	ca 256 kb/s
MP3	ca 160 kb/s
AAC	ca 128 kb/s

Vi kan lagre lyden ved å måle strømmen med jevne mellomrom:



Men for å lagre digitalt må vi måle styrken i faste intervaller:



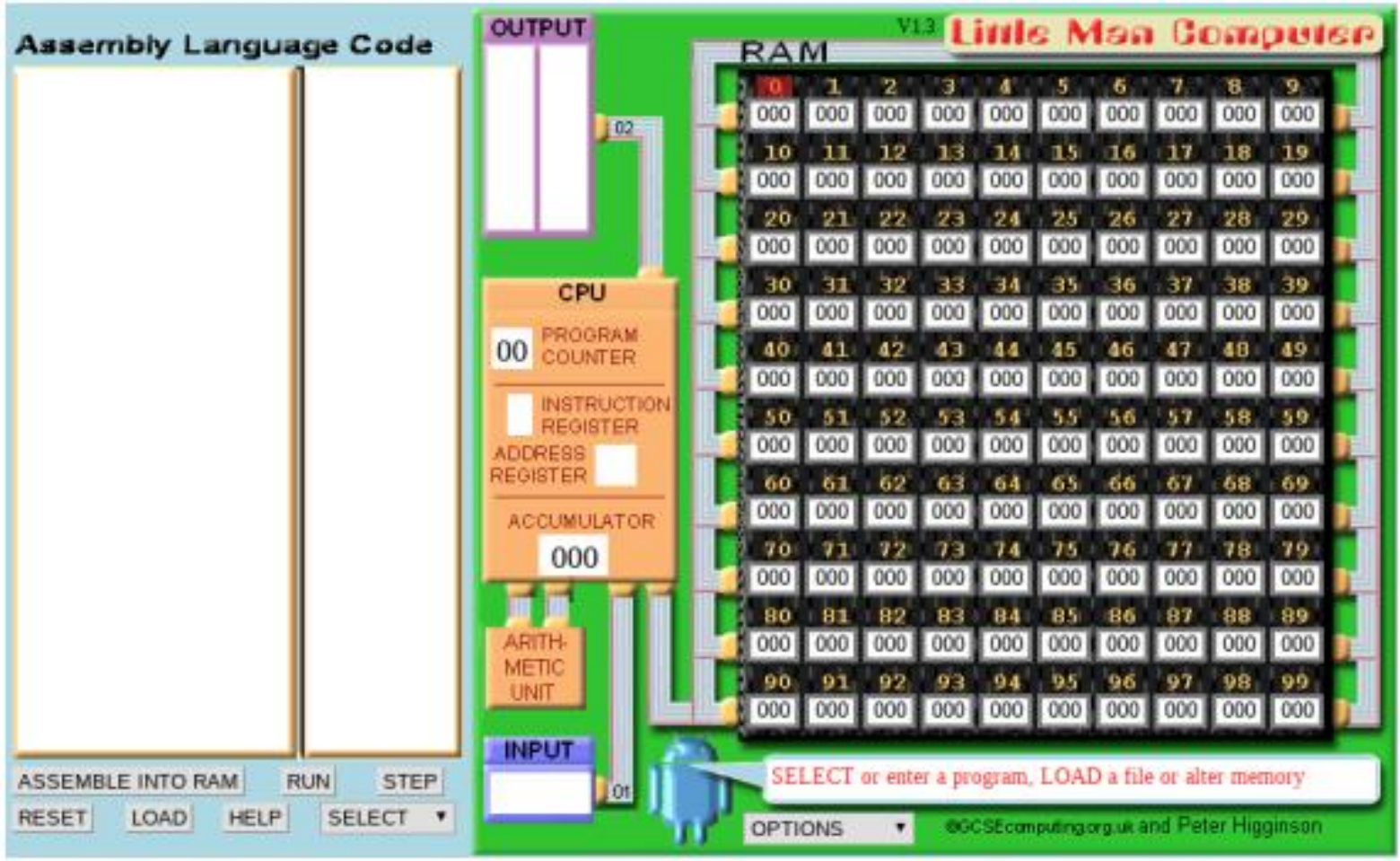
LMC-Little Man Computer

Enkel datamaskin

Bruke desimaltall

Består av 100 celler

Eks. Hylle med 100 skuffer



Ekte maskiner

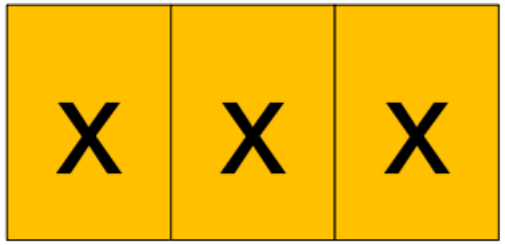
- Ekte maskiner har mer input og output:

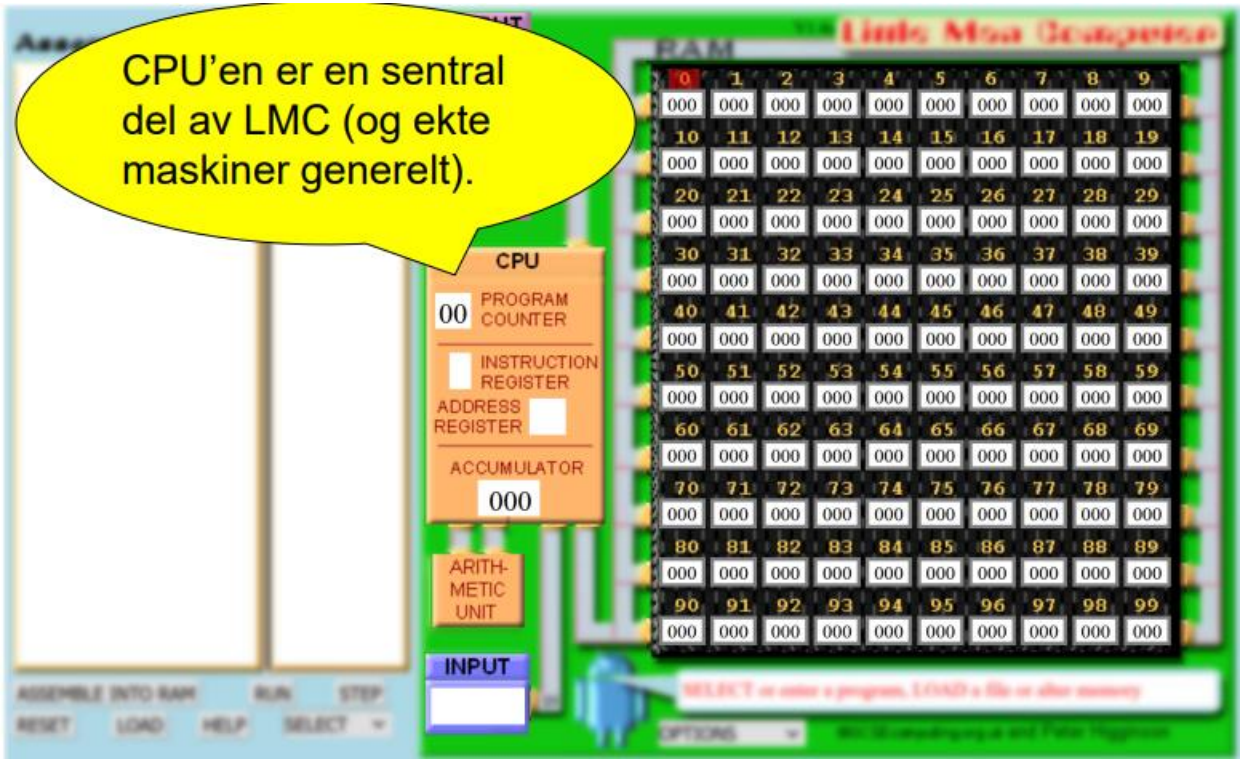
- Input:
- F.eks. mus, touch, skjerm, skanner, mikrofon, kamera

- Output:
- F.eks. skjerm, høyttaler, printer

CPU

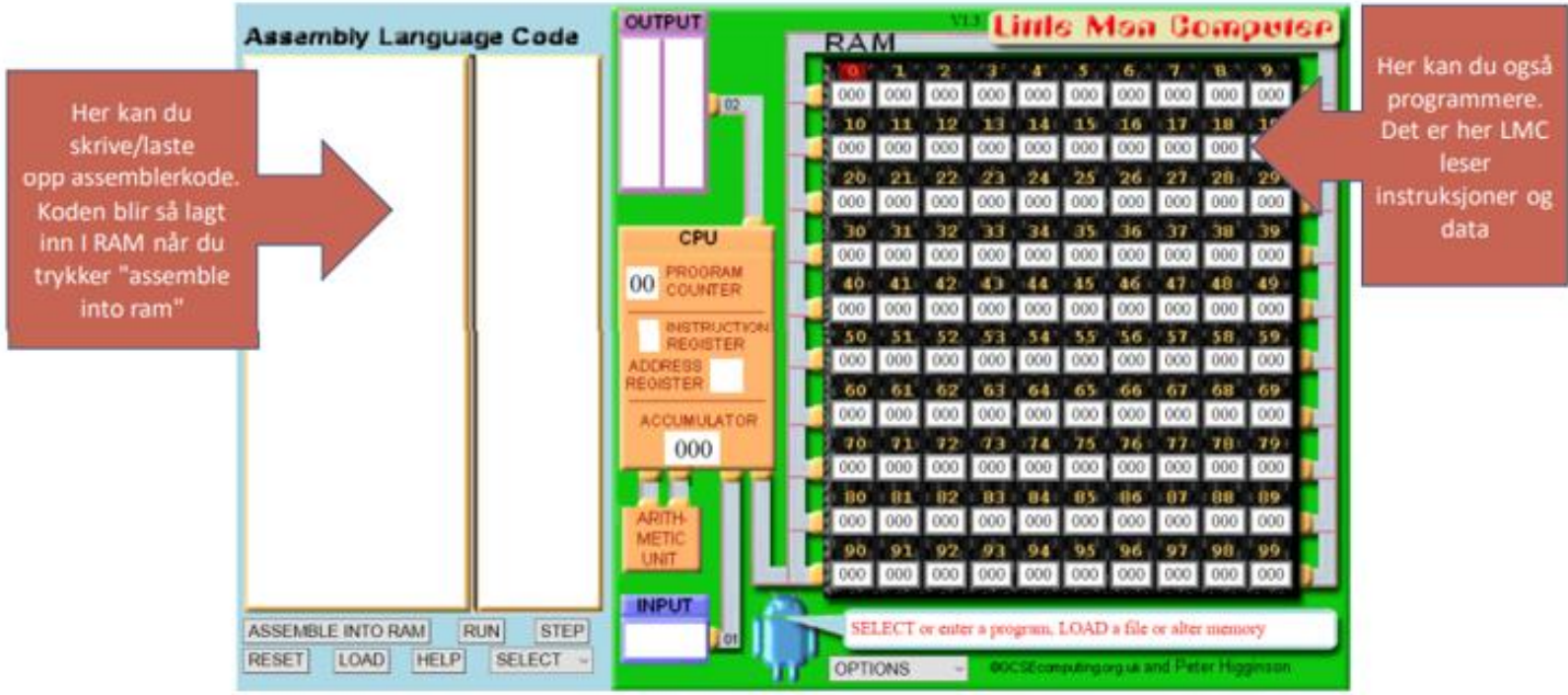
- ALU
- Registre
- Kontrollogikk
- Hver instruksjon består av 3 sifre





LMC

- Kjører sekvensielt
- Skriver ut tall eller tekst
- Input leser tall



Instruksjoner i LMC

- Instruksjonskoden
  - Viser hvilken instruksjon kontroll-logikken skal utføre.

5

- Adressedelen
  - Viser typisk hvilken adresse instruksjonen skal bruke.
  - Med et par unntak..

2 0

Alle instruksjonene i LMC med assemblernotasjon

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)

Enklere ASCII-tabell?

10	LF	39	'	47	/	62	>	93	]	124	|
32		40	(	48	0	63	?	94	^	125	}
33	!	41	)	:		64	@	95	_	126	~
34	"	42	*	57	9	65	A	96	‘		
35	#	43	+	58	:	:		97	a		
36	$	44	,	59	;	90	Z	:			
37	%	45	-	60	<	91	[	122	z		
38	&	46	.	61	=	92	\	123	{		

De mest nødvendige tegnene, i base 10.

Eksamensoppgave:

IN1020 eksamen høst 2018

2(c) **Maskinkode 1**

```
start   INP
        STA   50
        ADD   50
        ADD   50
        OUT
slutt   HLT
```

Hva skrives ut når denne koden kjøres og brukeren gir **7** som inndata?

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

- Det skrives ikke ut noe.
- 50
- 21
- 7
- 0

```
START       LDA     H
            OTC
            LDA     e
            OTC
            LDA     i
            OTC
END         HLT

H           DAT      72  // = 'H'
e           DAT     101  // = 'e'
i           DAT     105  // = 'i'
```