Gruppetime IN1020 - uke 4

Prosessoren og LMC

Foiler tatt fra forelesningene i uke 38

2(a)

Binære tall 1

Verdien $\mathbf{30_{10}}$ (dvs 30 i 10-tallsystemet) kan også representeres i andre tallsystemer. Hvilke av disse verdiene er lik 30_{10}?

Velg ett eller flere alternativer:

- [ ] 11110_2
- [ ] 111_5
- [ ] 1000_3
- [ ] 42_7

IN1020 eksamen høst 2018

2(a)

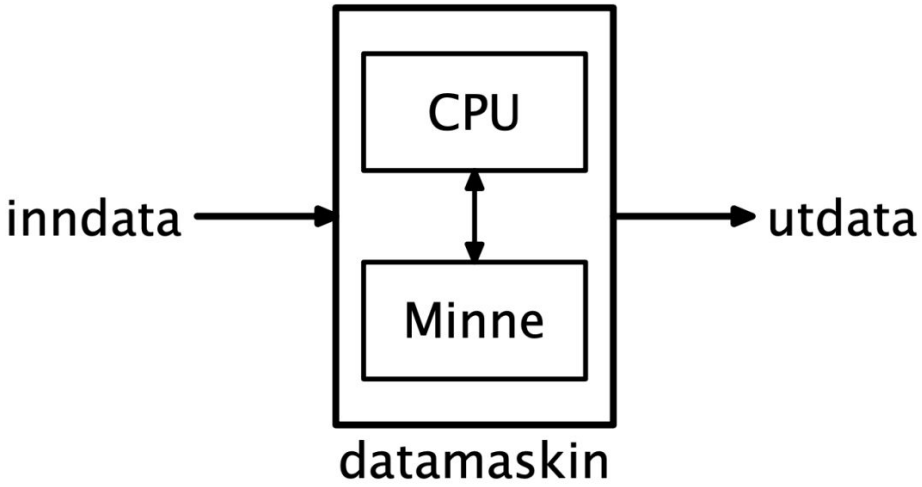
Binære tall 1

Verdien $\mathbf{30_{10}}$ (dvs 30 i 10-tallsystemet) kan også representeres i andre tallsystemer. Hvilke av disse verdiene er lik 30_{10}?

Velg ett eller flere alternativer:

- 42_7 ✔
- 1000_3
- 11110_2 ✔
- 111_5

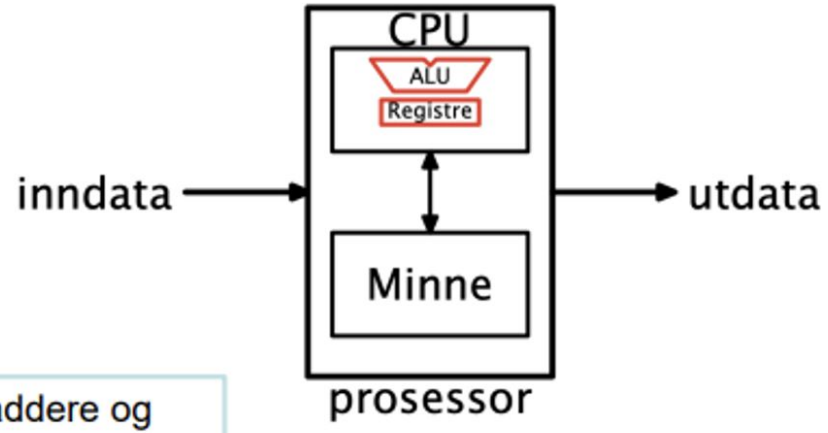
Von Neumann-arkitekturen



Den store erkjennelsen er at minnet holder på både data og instruksjoner til CPU-en (altså programmet). Dette er en arkitektur som fortsatt er i bruk i dag.

CPU («Central Processing Unit») er den sentrale delen i en datamaskin. Den inneholder:

- ALU («Arithmetic Logic Unit») til aritmetiske og logiske beregninger
- Registre
- Kontrollogikk til å utføre instruksjonene



ALU-en (regneenheten) i LMC kan addere og subtrahere heltall

ALU-en i en moderne datamaskin kan alle vanlige regnearter for heltall av ulik lengde, og for flyt-tall.

Little Man Computer (LMC)

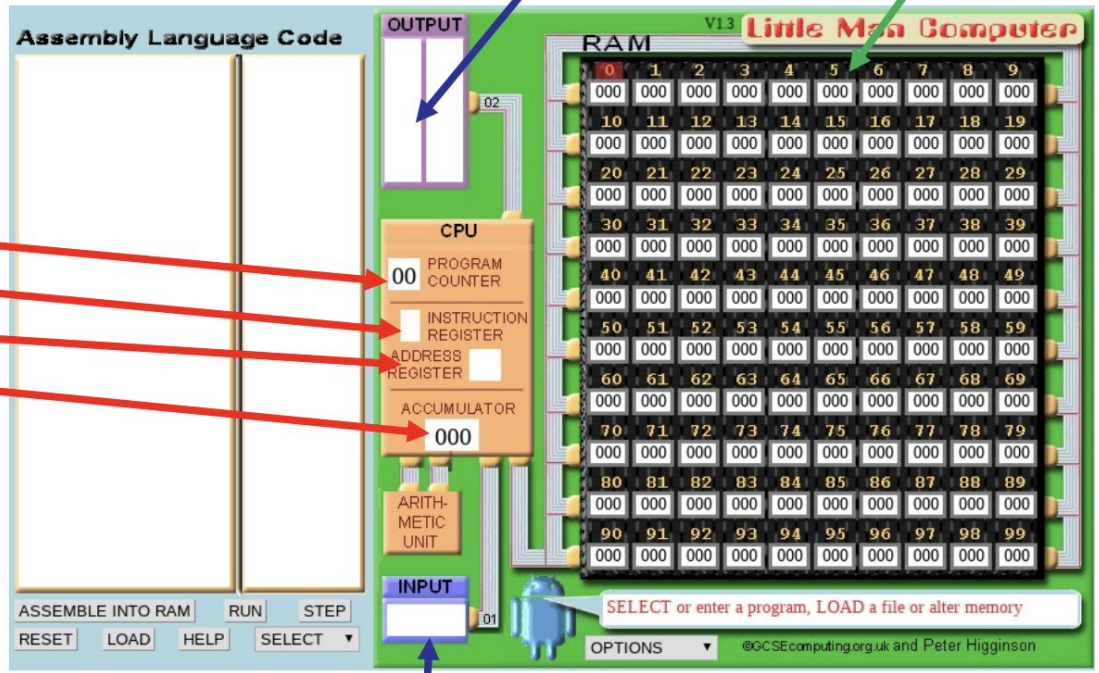
- Simulator
- Desimaltall istedenfor binærtall
- Teoretisk korrekt
- Fin måte å få innblikk i hvordan prosessoren fungerer

LMC-overblikk

Resultater skrives ut her

Minne med 100 celler for program og data

Fire registre
- Program counter
- Instruction register
- Address register
- Accumulator



LMC kan lese tall som brukeren (vi) skriver inn

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	STO	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 til angitt adresse hvis akkumulatoren er 0
8xx	BRP	Hopper til angitt adresse 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)

Assemblerkode

- Direkte oversettelse numerisk <--> tekstlig representasjon
- Kronglete å lese numeriske koder - bedre at vi kan navngi instruksjonene
- Oversettes til numerisk kode før kjøring
- Mye enklere å lese og skrive for mennesker
- Kan også sette navn på celler i minnet, slik at det blir lettere å holde styr på data i minnet
- Assemblerkode skrives og lagres som vanlig tekstfil

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

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

Skrive ut bokstaver og tegn

H: $48_{16} = 4*16 + 8 = 72_{10}$

e: $65_{16} = 6*16 + 5 = 101_{10}$

i: $69_{16} = 6*16 + 9 = 105_{10}$

ASCII

Første vellykkede forsøk på standardisering var ASCII (American Standard Code for Information Interchange) i 1963; den har 128 tegn:

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

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.

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.
- [ ] 21
- [ ] 50
- [ ] 0
- [ ] 7

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

Ukesoppgaver og/eller oblig 2