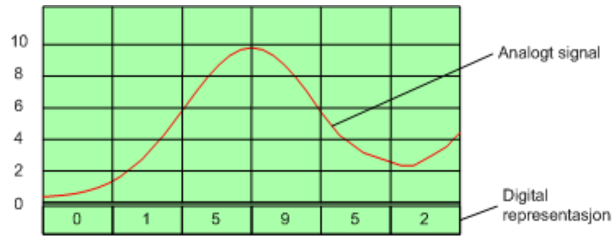
Tallsystemene

- **ANALOGE OG DIGITALE SIGNALER**
- **TALLSYSTEMENE**
- **TCP/IP, Sub nett**

Innholdsfortegnelse

Analoge signaler

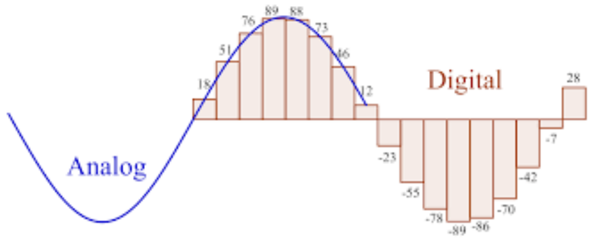
Analoge signaler er signaler der informasjonen vi overfører er på en trinnløs, uendelig skala. Informasjonen har en nesten ubegrenset kompleksitet, og det er måleutstyret som setter begrensninger for detaljnivået vi kan hente ut.



Det analoge signalet i eksempelet med vibrasjoner i forskjellige medier er på en trinnløs skala. Det har en ubegrenset kompleksitet. Når vi omgjør et analogt signal fra et medium til et annet, mister vi noe av innholdet i det originale signalet. Det vil si at vi mister noe av kvaliteten. Vi lager et nytt analogt signal, men dette nye signalet er ikke identisk med det originale.
Et godt eksempel på dette er de første opptakene av lyd.

Digitalisering

Prosessen med å gjøre om et signal fra analog til digital form kaller vi for digitalisering. Vi bruker en avgrenset, trinnet skala med digitale signaler. Vi gir hvert trinn, en tallverdi som oftest oppgis binært.



Siden datamaskiner har en så viktig rolle i dagens samfunn, bruker vi begrepet digitalisering når vi snakker om å gjøre analoge signaler tilgjengelige for våre datasystemer. Siden våre datamaskiner er *binære* (bruker det binære tallsystem med verdiene 0 og 1), må også informasjonen vi skal ha inn i disse systemene ha denne formen.

Digitalisering er blitt et begrep for å gjøre informasjon tilgjengelig i datasystemer, og begrepet digital beskriver informasjon som er i datasystemer.

Sampling

Hvis vi ser på et analogt signal hvor forandringene er ekstreme, et lydsignal er et slikt signal. Lyd som vi mennesker kan høre, er bølger i lufta på mellom 20 og 20 000 bølger per sekund, eller 20–20 000 Hz (hertz) om du vil. Hvis vi ønsker å gjøre et digitalt opptak av lyden, trenger vi å gjøre mange samplinger per sekund. Resultatet kan bli som på bildet under.

Når vi digitaliserer lyd, ønsker vi at den digitale lyden skal høres mest mulig identisk ut med det originale signalet. Fordi vi bruker en trinnet skala til å måle med, kan vi ikke gjengi det analoge signalet perfekt. Det vi kan gjøre, er å gjøre mange nok gode målinger til at vi som mennesker ikke klarer å høre forskjellen.

Valg vi må gjøre når vi digitaliserer lyd

Når vi skal digitalisere lyd, er det to viktige valg vi må gjøre.

- Vi må bestemme hvor mange sampler vi skal gjøre av signalet per sekund. Dette kalles også *samplerate*. I modellen over vil dette være antallet stolper bortover.
- Vi må bestemme hvor mange steg skalaen vi måler med skal ha. Dette kalles også *bitdybde*. I modellen over vil dette være hvor findelt skalaen er oppover for hver stolpe.

For lyd er det for eksempel vanlig å gjøre 48 kHz (48 000 målinger per sekund) med bitdybde på 24 bit. Dette betyr at vi gjør 48 000 målinger per sekund med lyd, og for hver måling brukes 24 binære tall (bit) for å beskrive lyden akkurat da. Når vi ganger dette sammen, blir det 1 152 000 bit per sekund, eller 1,152 Mbit/s. Dette kaller vi *bitraten* når vi arbeider med lyd og video. Utregningen er for et monosignal, hvis vi skal ha et stereosignal må vi gjøre alt dobbelt. Uten noen form for komprimering vil ett minutt av lyd ta 17,28 MB med lagringsplass. En høyere samplerate og bitdybde vil kreve mer lagringsplass enn en lavere.

Det Binære tallsystemet

Digital elektronikk arbeider med signaler som bare har to mulige tilstander.

Hva er de to tilstandene?

To forskjellige spenningsnivåer, vi kaller dem **HØY** og **LAV**.
Som betyr «spenning» og «ikke spenning», også kalt «**1**» og «**0**»

Andre måter å representere dette på er ved hjelp av lamper/lys, **lys på** og **lys av**. Et annet alternativ er basert på forskjellig strømnivå, eksempelvis 230V av eller på. Normalt arbeider mikroprosessoren med 3,3V eller 5V.

Kombinasjonen av 1 og 0 vil representere tall, bokstaver, symboler og andre typer informasjon eller funksjon. For at vi skal forstå dette bedre må man beherske det digitale tallsystemet og overgangen mellom det desimale og digitale tallsystemet.

Oppgave

La oss så si at du har to utelamper på huset ditt, en utenfor hoved
døren og en utenfor bakdøren. Så skal du lage et lite varslingssystem sammen med naboene dine i fellesskap. Tenk ut et system som viser dette:

- Inne i huset
- Ikke i huset
- Ute på en kort tur
- Ute, tilbake neste dag

Tall	1	2	4	8	16	32	64	128	256	512
Toerpotens	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9

Vi kan erstatte de forskjellige kombinasjonene av 0 og 1, altså de binære tallene, med Vanlige tall. Har vi to bit for eksempel 11, begge på, gir vi ett-tallet på plassen lengst til venstre verdien 2, og plassen til høyre få verdien 1. Er de av, altså 0, får begge plassene verdien null.

Med to utelamper kan vi da telle fra null til tre med vanlige tall. Det gjør vi slik:
00 gir 0 + 0 = 0.
Mens 01 gir 0 + 1 altså 1.
Ser vi på 10, så har sifferet 1 verdien 2 siden det står først, og det gir 2 + 0 = 2. Tilsvarende blir 11 = 2 + 1 = 3, siden begge er slått på og den første har verdien 2 og den neste verdien 1.
Et slikt oppsett kaller man for **Boolsk algebra og sannhetsverdi-tabell**.

Britiske matematiker George Boole

TO BIT BINÆRT TALL	VERDI: 2	VERDI: 1	SUM, VANLIG TALL
00	0	0	0
01	0	1	1
10	1	0	2
11	1	1	3

Når vi regner, bruker vi titallsystemet (desimaltall).

Systemet består av 10 tall som er: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Eksempel 2347 =
2347 = 2 x 1000 + 3 x 100 + 4 x 10 + 7 x 1=2000 + 300 + 40 + 7

$2347 = 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 = 2000 + 300 + 40 + 7$

Det binære tallsystemet har bare 2 tall, som er «0» og «1».
Eksempel: 1010 =

$1010 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 8+0+2+0 = 10$

1010 = 1 x 8 + 0 x 4 + 1 x 2 + 0 x 1= 8+0+2+0 = 10

- Det binære tallsystemet virker på samme måten som titallsystemet
- Alle datamaskiner bruker det binære tallsystemet for utregning, svaret blir deretter oversatt til titallsystemet og vist på skjermen. I oversettelsen fra titalsystemet bruker vi ASCII tabell.

Antall muligheter

Antall muligheter ved tre bits finner vi slik: 2^3, eller $2 \times 2 \times 2 = 8$

BIT	1	2	3	4	5	6	7	8	9	10
MULIGHETER	2	4	8	16	32	64	128	256	512	1024

8 bit er en byte

8 bit blir kalt 1 byte. I mange datamaskiner er 1 byte den minste plassen du kan sette av til br uk i minnet. De 256 mulighetene som en byte gir, beskriver ofte en farge, et lydnivå eller et tegn i et tegnsett i en datamaskin.

Tallet 112 og skal skrive det som 8 bit.

	128	64	32	16	8	4	2	1
H		1	1	1				
L	0				0	0	0	0

Eksempel på utregning

112 er mindre en 128, så første det blir 0 i første kolonne og et 1-tall settes i kolonnen for 64.
112– 64 = 48, så neste 1 tall settes i kolonne for 32.
48 – 32 = 16, så neste 1 tall går i kolonne for 16.
Da er det 0 igjen, så resten, 8, 4, 2, 1 blir 0

Lagringsplass og hastighet

BYTE	BIT	BINÆRT TALL
En byte	8 bit	11001100
To byte	To 8 bit	11001100 11001111
Tre byte	Tre 8 bit	11001100 11001111 11001101
En kilobyte = 1000 byte	Tusen 8 bit	11001100 ...999 ganger
En megabyte = 1000 000 byte	En million 8 bit	11001100 ...999 999 ganger
En gigabyte = 1000 000 000 byte	En milliard 8 bit	11001100 ...999 999 999 ganger

Lagringsplass oppgis i byte, mens hastigheten på nettverk oppgir vi i bit per sekund.

Kan 0 være eksponenten i en potens? Hva betyr at en eksponent er et negativt tall?

Definisjon
For ethvert tall a er $a^0 = 1$, forutsatt $a \neq 0$.

0 som eksponent

Vi har undersøkt brøker på formen $\frac{a^m}{a^n}$ der m > n og funnet at $\frac{a^m}{a^n} = a^{m-n}$.
Men hva hvis $m = n$?

Eksempel

Vi vet at når vi dividerer to like tall, får vi 1 som svar. For eksempel

$$\frac{2^3}{2^3} = 1$$

Men hvis regneregelen over skal gjelde, blir jo også

$$\frac{2^3}{2^3} = 2^{3-3} = 2^0$$

Vi kan konkludere med at vi kan få to svar hvis $2^0 = 1$. Dette gjelder også for alle andre grunntall. Også når vi regner med bokstaver, gjelder det samme, $\frac{a^m}{a^m} = 1$ og $\frac{a^m}{a^m} = a^0$.

Vi har en generell bestemmelse at et grunntall opphøyd i null er lik 1.

Fra binærtall til desimaltall

Øvelser
Regn om fra binært til vanlig tall (desimalt)

a. 101
b. 1101
c. 10011

Øvelser
Regn om fra binært til vanlig tall (desimalt)

a. 110
b. 1110
c. 10110--
d. 111001

$2^8 = 256$

$2^7 = 128$

$2^6 = 64$

$2^5 = 32$

$2^4 = 16$

$2^3 = 8$

$2^2 = 4$

$2^1 = 2$

$2^0 = 1$

Tallkonverter

Fra desimaltall til binærtall

Eksempel 1: 23 =?

$1 \text{ x } 2^4$	$0 \text{ x } 2^3$	$1 \text{ x } 2^2$	$1 \text{ x } 2^1$	$1 \text{ x } 2^0$
1	**0**	**1**	**1**	**1**

23 er større enn 16 men mindre enn 24.
7 er større enn 4 men mindre enn 8
3 er større enn 2 men mindre enn 4
1 blir restverdi

Eksempel 2: 37 =?

$1 \text{ x } 2^5$	$0 \text{ x } 2^4$	$1 \text{ x } 2^3$	$1 \text{ x } 2^2$	$1 \text{ x } 2^1$	$1 \text{ x } 2^0$
1	**0**	**0**	**1**	**0**	**1**

37 er større enn 32 men mindre enn 64.
5 er større enn 4 men mindre enn 16 og 8
1 blir restverdi

$2^8 = 256$

$2^7 = 128$

$2^6 = 64$

$2^5 = 32$

$2^4 = 16$

$2^3 = 8$

$2^2 = 4$

$2^1 = 2$

$2^0 = 1$

Oppgave
Regn om fra vanlige tall til binære tall

a) 2, b) 8, c) 9, d) 10, e) 42, f) 70

Det hexadesimale Tall systemet

Navnet «**heksadesimal**» **er** en hybrid sammensatt av det greske hexa (έξι (exi)) for «seks» og desimal fra det latinske ordet for «ti». Tallsystemet har 16 ulike siffer:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E og F.

I desimal blir dette:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Eksempel: 5 C 9 $= 5 \times 16^{2} + 12 \times 16^{1} + 9 \times 16^{0}$

$= 5 \times 256 + 12 \times 16 + 9 \times 1 = 1481$

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

Eksempel

a) **Skriv det binære tallet 1011011001 i det heksadesimale tallsystemet**

b) **Skriv tallet i det desimale systemet**

c) **Skriv 2D9 i det desimale tallsystem**

Tips, del tallene inn i 4 og 4; 10 - 1101 – 1001 som gir

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	1	0	1	1	0	0	1
512	0	128	64	0	16	8	0	0	1

2D9 = $2 \times 16^{2\,+}\ 13 \times 16^1 + 9 \times 16^0 =$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Oppgaver

Oppgave 1
a) Skriv det binære tallet 11100101 i det heksadesimale tallsystemet
b) Hvilket tall er dette i vårt tallsystem, vis begge utregningsmåtene

Oppgave 2
a) Skriv det binære tallet 1111010100 i det heksadesimale tallsystemet
b) Hvilket tall er dette i vårt tallsystem, vis begge utregningsmåtene

Oppgave 3
a) Skriv tallet 729 i det binære tallsystemet
b) Skriv svaret i oppgave a i det heksadesimale tallsystemet
c) Kontroller at svaret i oppgave b gir tallet 729

ASCII

ASCII er et tegnsett, det vil si en standard (ISO) for utveksling av tekst mellom datamaskiner.
ASCII benytter 7 bit til koder, noe som tillater koding av 128 mulige verdier. 95 av disse er tilordnet store og små bokstaver i det engelske alfabetet, tallene 0-9 og en del andre vanlig forekommende tegn
Det er også utarbeidet en standard for utvidelse (Extensions) for at land som har mange egne tegn, slik som Norge, har en egen tabell basert på 8 bit koding (ECMA).

Unicode er en nyere og utvidet standard for koding av karakterer. Den benytter 16 biter i stedet for 8. Den kan dermed kode:

$$\mathbf{2^{16} = 65536}\ \textbf{tegn}$$

Den kan definerer de fleste kjente tegnsett i verden (kinesisk, kyrillisk, sanskrit ...).
Ascii-koden utgjør den første delen av Unicode karakterene, så systemene er kompatible en vei.

Eksempel

Når vi har regnet ut desimaltall fra den binære verdien eller fra Hexadesimale verdien og har ASCII tabellen med utvidelse
Binære verdien: 0100 0001 som gir tallverdien 65 i det desimale tallsystemet
Eller som hexadesimale tallsystemet som gir samme verdi 4 1 = $4 \times 16^1 + 1 \times 16^0 = 65$

Tallverdien 65 gir da en stor A
I ASCII tabellen på neste side

128	64	32	16	8	4	2	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	0	0	0	1
0	64	0	0	0	0	0	1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Oppgaver

1. Gjør om den binære tallverdien til desimaltall og finn ASCII kodene

01000001 =
01100001 =
01000010 =
01100010 =
01000011 =
01100011 =

2. Gjør om den binære tallverdien til desimaltall og finn ASCII kodene

0100011111111000011110010010000001101101011001010110110010000100000010
0000101010011010000110100100101001001

Oppgave 3. Skriv fornavnet ditt i binært tallsystem

Oppgave 4. Oversett dette binære til Hex

01100010 01101001 01101110 11100110 01110010 01100101 00100000 01110100
01100001 01101100 01101100 00100000 01100101 01110010 00100000 01101101
01101111 01110010 01110011 01101111 01101101 01110100 00100000 11100101
00100000 01100001 01110010 01100010 01100101 01101001 01100100 01100101
00100000 01101101 01100101 01100100 00100001

Oppgave 5. Vi går fra Hex til ASCII

4b 6c 61 72 74 65 20 64 75 20 64 65 6e 6e 65 2c 20 73 e5 20 65 72 20 64 75 20 69 6e 6e 61
66 6f 72 21

Har du forstått?

Kan du svare ja på alle spørsmålene nedenfor, har du god innsikt i fagområdets viktigste elementer.

______ Jeg kan forklare hva et signal er og gjør.

______ Jeg kan forklare begrepene digital og analog.

______ Jeg kan forklare hva sampler er og knytte det til målinger.

______ Jeg kan beskrive hvordan digitalisering av lyd skjer.

______ Jeg kjenner til og kan forklare flere tallsystemer.

______ Jeg kan forklare hva binære tall, hexadesimale tall, titall og ASCII er.

______ Jeg kan beskrive og forklare hva linje- og pakkesvitsjing er.

______ Jeg kan forklare forskjellen på en trinnløs og en trinnet skala og knytte det til analoge og digitale signaler.