

## Digitalelektronik

### Det binære- og hexadecimale talsystem

#### Den digitale verden

I den digitale verden er alt udtrykt med "0" og "1". I tabellen nedenunder ses en række måder at omtale henholdsvis "0" og "1":

Digital 0	Digital 1
Slukket	Tændt
Off	On
Lav	Høj
False	True
0 V	Vcc (forsyningsspænding)

Men hvis man kun har "0" og "1" til rådighed, hvordan kan man så indikere et 5-tal? Svaret er, at man bruger "0" og "1" i en kombination, der gør, at man kan repræsentere tallet "5".

#### Det binære talsystem

Det binære talsystem bruger "0" og "1" fra den digitale verden, og gør os i stand til at repræsentere almindelige tal som vi kender dem.

Hvert ciffer kan enten være "0" eller "1", og et ciffer kaldes for en *bit*. Har man 2 cifre til rådighed, har man dermed 2 *bit*.

En bit kan enten være 0 eller 1, hvilket giver os to tilstande. Har vi to bit kan disse kombineres på 4 forskellige måder. Har vi tre bit kan disse kombineres på 8 forskellige måder, og så fremdeles. I nedenstående tabel ses hvordan fire bit kan kombineres på 16 forskellige måder, og dermed repræsentere tallene fra 0 til 15:

Binært	Decimalt
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10

1011	11
1100	12
1101	13
1110	14
1111	15

Bemærk, at bit'et længst til højre skifter hver gang, mens bit'et længst til venstre kun skifter én gang. Bit'et længst til højre kaldes *det mindst betydende bit* (På engelsk *Least Significant Bit / LSB*), og bit'et længst til venstre kaldes *det mest betydende bit* (På engelsk *Most Significant Bit / MSB*).

Vi kan nu, ved at kombinere de fire bit, tælle fra 0 og op til 15. Har vi brug for at tælle højere op, tilføjer vi flere bit. Man udregner antallet af kombinationer der kan laves ud fra antallet af bit, således:

$$\text{Antal kombinationer} = 2^{\text{antal bit}}$$

Antal bit	Antal kombinationer	Tælle fra 0 til...
1	$2^1 = 2$	1
2	$2^2 = 4$	3
3	$2^3 = 8$	7
4	$2^4 = 16$	15
5	$2^5 = 32$	31
6	$2^6 = 64$	63
7	$2^7 = 128$	127
8	$2^8 = 256$	255

Når man skriver et tal i det binære talsystem, så skal man gøre opmærksom på, at det er binært og ikke et decimaltal. Det gør man enten ved at skrive "0b" foran tallet, eller skrive "b" efter tallet.

Ex: decimaltallet 12 skrives som: 0b1100 eller 1100b. Gør man ikke opmærksom på at det er et binært tal, står der jo 1100 (et tusinde et hundrede) og det er jo ikke helt det samme som tallet 12.

## Hexadecimaltal

Når man arbejder med mange bit, kan det være en fordel at arbejde med bit'ene i klynger. 8 bit kalder man en byte, og 4 bit kaldes en halv byte.

Hexadecimaltalsystemet går ud fra at man tager 4 bit sammen, og repræsenterer disse 4 bit med enten et tal eller et bogstav. Tallene fra 0 til 15 bliver i hexadecimalsystemet til følgende:

Binært	Hexadecimalt	Decimalt
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4

0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Hexadecimaltalsystemet ligner decimalsystemet fra 0 til 9, men i stedet for (decimal)tallet 10, skrives A; dermed kan man nøjes med ét ciffer til at repræsentere tallene fra 0 til 15.

Efter hexadecimaltallet "F" kommer endnu et ciffer på (dvs. 4 bit mere), og man kan nu tælle helt op til 255.

Man indikerer, at det er et hexadecimaltal ved at skrive "0x" eller "0h" foran tallet, eller "h" efter tallet.

Ex: 0x1F = 31 decimalt.