



Digital representasjon: Hvorfor bare '0' og '1'?

- Nesten all elektronikk inneholder *digital* elektronikk:
 - PC'er, mobiltelefoner, MP3-spillere, DVD/CD-spillere, biler, kjøleskap, TV osv osv.
- For å utføre beregninger trengs byggesteiner som kan *addere*
 - Andre regningsarter avledes fra addisjon
- De første datamaskiner brukte *radiorør*
 - Størrelse og effektforbruk som en vanlig lyspære
 - En meget enkel datamaskin laget av radiorør bruker mange kW og avgir nesten alt som varme
- Revolusjon: Oppfinnelsen av *transistoren*
 - Grunnlaget for vår digitale hverdag!



UNIVERSITETET
I OSLO

Transistorens virkemåte (1)

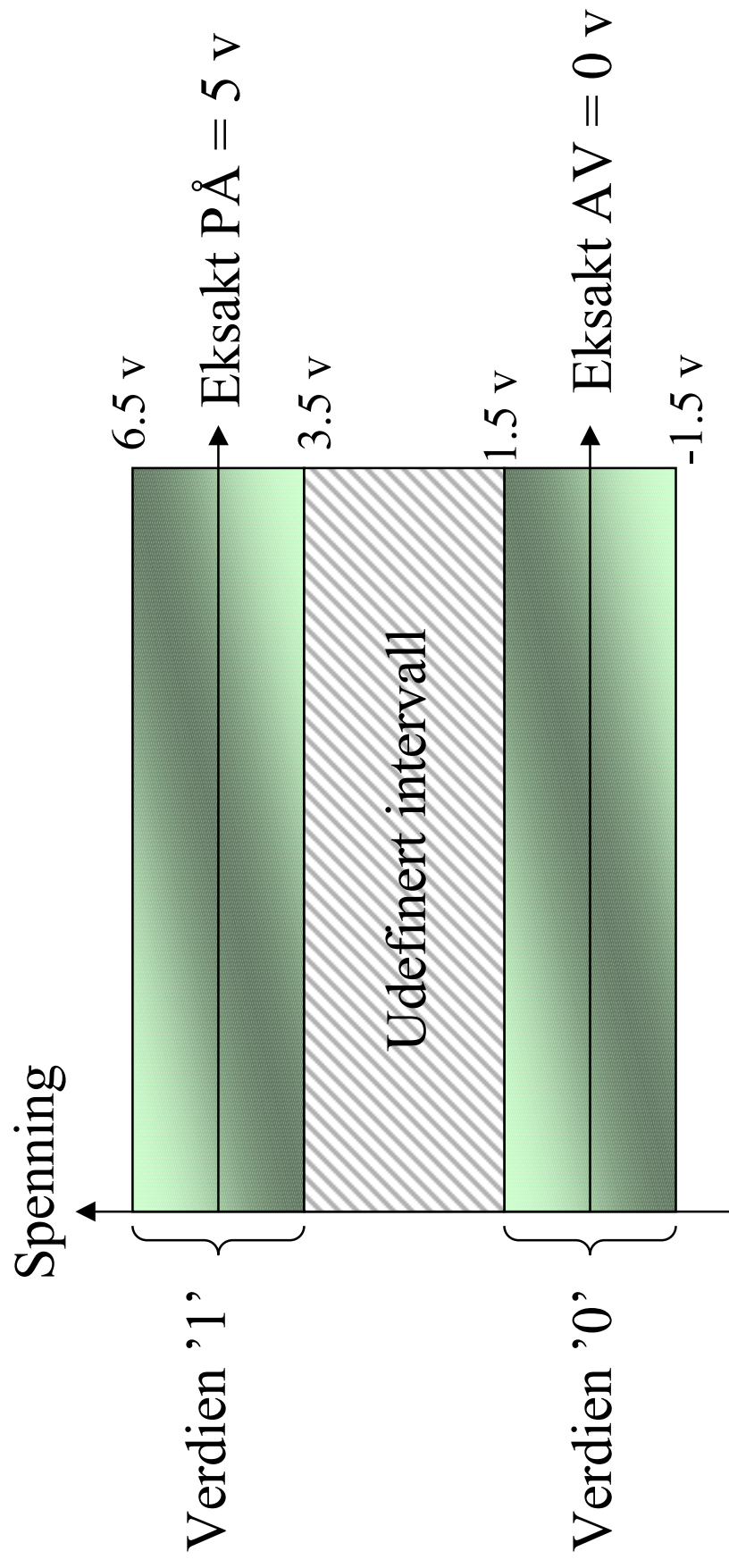
- En transistor kan sammenlignes med en elektrisk bryter
 - Bryteren er enten AV eller PÅ
 - Kan også brukes til å lagre elektrisk ladning
- En transistor er svært liten: 0,000000018 meter bred
- På et lite areal får man plass til mange millioner transistorer
 - Svært mye regnekraft på størrelse med et fyrstikkhode!
 - Pentium har ca 450 millioner
- Transistoren bruker svært lite strøm når den enten er AV og PÅ
 - Men allikevel noe, og hvilestrømmen varierer fra transistor til transistor



Transistorens virkemåte (2)

- Transistoren bruker mer strøm når den enten slås AV eller PÅ
 - Kan endre tilstand i løpet av nanosekunder (10^{-9} sekunder)
 - Svitsjetid og -strøm varierer fra transistor til transistor
- Umulig å lage en transistorer som helt AV eller helt PÅ
- Umulig å lage to transistorer som er identiske
 - Variasjon i hvilestrøm, svitsjetid og -strøm
- Konklusjon: Trenger informasjons-representasjon som er robust
 - Må tolerere at transistoren aldri er helt AV eller PÅ
 - Må også tolerere variasjoner mellom transistorer
- Løsning: 2-tallsystemet som har to verdier: '0' og '1'

Transistorens virkemåte (3)





Boolsk algebra

- Boolsk algebra: Regneregler for variable med verdiene '0' og '1'
- '0' og '1' benevnes også *TRUE* og *FALSE*
- En boolsk variabel har enten verdien '0' eller '1'
 - Må ikke forveksles med tallene 0 og 1 i tallsystemet
- Boolsk algebra har tre basale operasjoner: *AND*, *OR* og *NOT*

a	b	$a \times b$
0	0	0
0	1	0
1	0	0
1	1	1

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

a	a'
0	0
0	1
1	1
1	1



Boolsk algebra (forts.)

- Andre boolske funksjoner defineres vha AND, OR og NOT:

$$\begin{aligned} a \text{ XOR } b &= (a \text{ AND } (\text{NOT } b)) \text{ OR } ((\text{NOT } a) \text{ AND } b) \\ &= a' \times b + a \times b' \end{aligned}$$

$$\begin{aligned} a \text{ XNOR } b &= (a \text{ AND } b) \text{ OR } ((\text{NOT } a) \text{ AND } (\text{NOT } b)) \\ &= a \times b + a' \times b' \end{aligned}$$

$$a \text{ NAND } b = \text{NOT } (a \text{ AND } b)$$

$$= a' + b' \text{ (Merk denne!)}$$

$$a \text{ NOR } b = \text{NOT } (a \text{ OR } b)$$

$$= a' \times b' \text{ (Merk denne ogs\aa!)}$$

- De to siste reglene bruker *de Morgans* teorem:

$$(a \times b)' = a' + b'$$

$$(a + b)' = a' \times b'$$



UNIVERSITETET
I OSLO

De Morgans teorem

- Kan vise de Morgans teorem ved å sette opp tabell for hver side

a	b	$(ab)'$

a	b	$a'+b'$

a	b	$(a+b)'$

a	b	$a'b'$



Boolske regneregler

- Andre regneregler i boolsk algebra

$$a + a' = 1$$

$$a \times a' = 0$$

$$a + a = a$$

$$a \times a = a$$

$$a + 0 = a$$

$$a \times 1 = a$$

$$a + 1 = 1$$

$$a \times 0 = 0$$

$$(a')' = a$$

- Regnereglene kan brukes til å forenkle uttrykk:

.Eksempel 1: $ab + ab' =$

.Eksempel 2: $abc + a'bc + a'bc'(a + c) =$



Boolsk funksjoner

• Beskrives med *funksjonsuttrykk* eller *sannhetsverditabell*

• Eksempel 1: Funksjon med 2 variable

$$F = ab + a'b'$$

a	b	F

• Eksempel 2; Funksjon med 3 variable

$$G = abc + a'c + ab'$$

a	b	c	G

Boolsk funksjoner (forts.)

- Funksjonsuttrykk egner seg best ved få variable og ledd
- Sannhetsverditabell egner seg for funksjoner mange ledd og variable, og for maskin-representasjon (eks datamaskin-asisstert design)
- Enkle funksjoner representeres grafisk ved hjelp av symboler kalt (logiske) porter:

