

Digital Teknologi

Forelesning nr 1

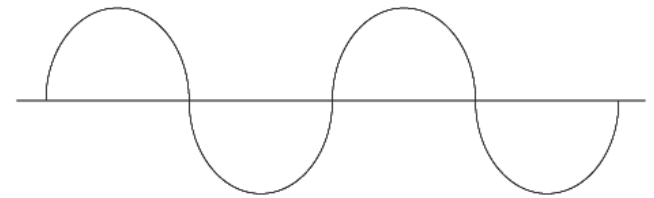
Hovedpunkter

- Desimale / binære tall
- Digital hardware-representasjon
- Binær koding av bokstaver og lyd
- Boolsk algebra
- Digitale byggeblokker / sannhetstabell
- Generelle porter
- Fysisk innpakning

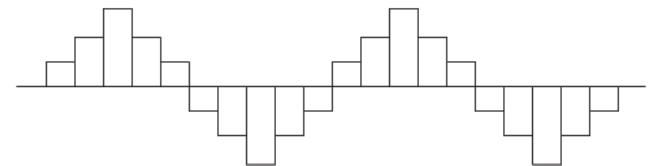
Digitale Teknologi?

- Teknologi som opererer med digitale signaler, eller diskrete data.
- Vi skal se at det er mange fordeler med digitale systemer

Analog

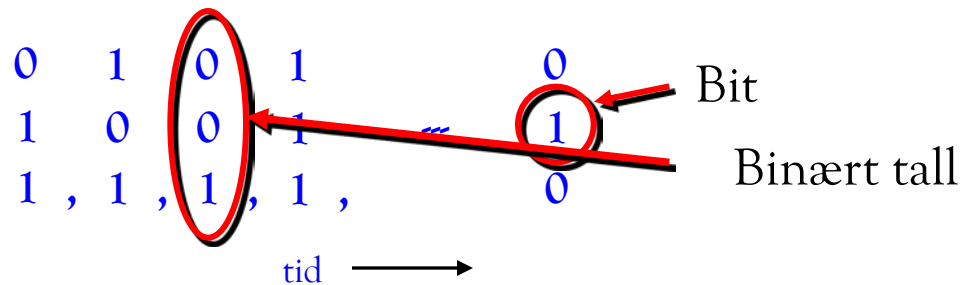


Digital



Digitale Teknologi?

- Som oftest basert på Bincære tall (ord):
 - Tall som kun er representert ved symbolene 0 og 1 (bit's). Nøyaktighet gitt av antall bit. (avrundingsfeil)
- Digitale signaler:
 - Sekvenser av bincære tall i tid. Hvert tall kan bestå av ett eller flere "bit"

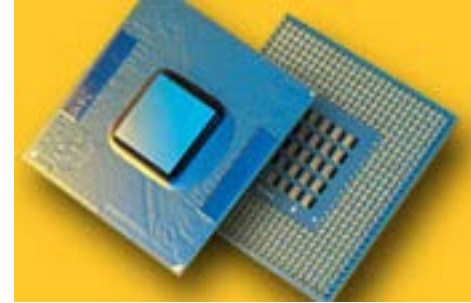


- Digitale system:
 - System som håndterer digitale signaler

Den digitale revolusjon 1

Digitale systemer

- CPU
 - Datamaskinen
 - Internett



Den digitale revolusjon 2

Digitale systemer overtar for analoge systemer

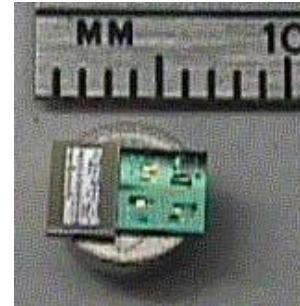
- Eksempler:
 - Stereoanlegg (CD / vinylplate)
 - Video (DVD / VHS)
 - Foto (digitale kamera / 35mm)
 - Mobiltelefon (GSM)
 - Radio (DAB / FM)
 - OSV...

Den digitale revolusjon 3

Trådløse sensorsystemer

Mikroelektronikk gjør det mulig å lage små avanserte trådløse systemer.

- Eksempler:
 - Sensorer/display i klær/stoffer/bøker
 - Fysiske "widgets"
 - Kroppsmonitorering
 - Internet of Things (IoT).
 - Osv...



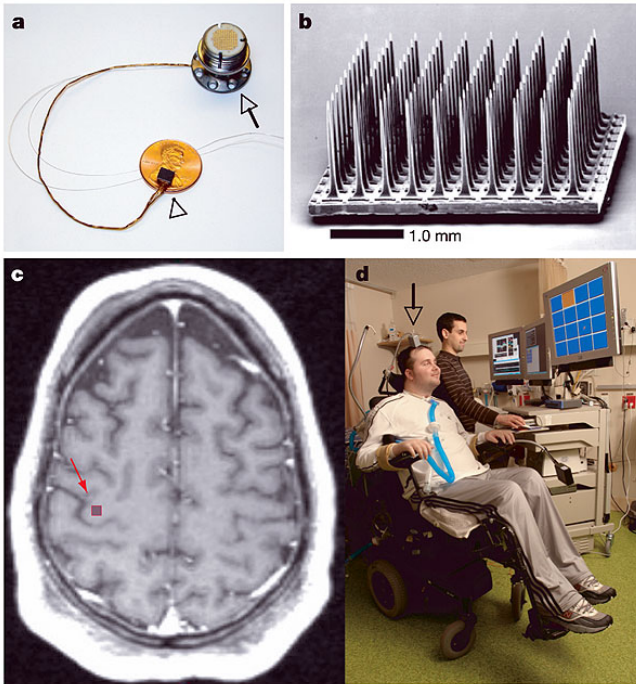
Trådløs sensor node



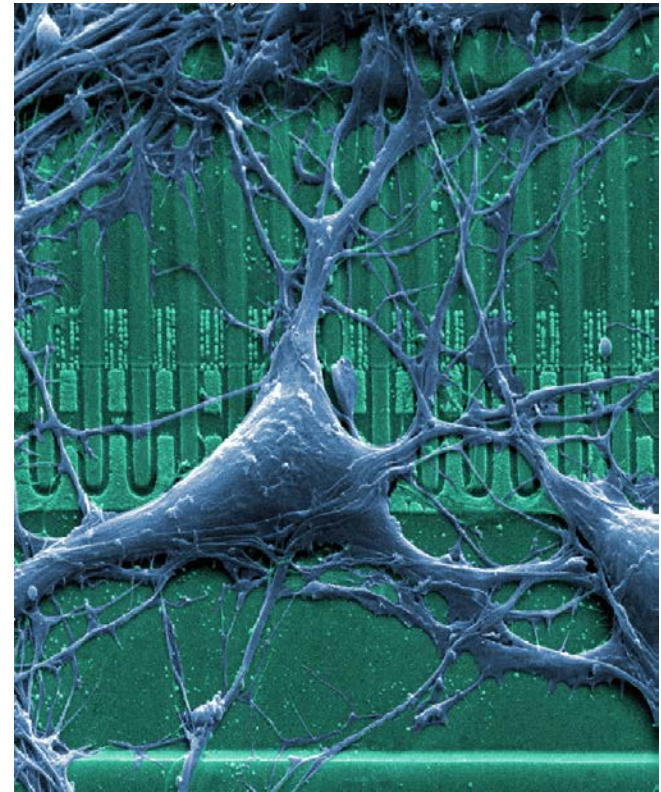
Pillekamera

Den digitale revolusjon 4

Silisium er biokompatibelt.



"Braingate" USA

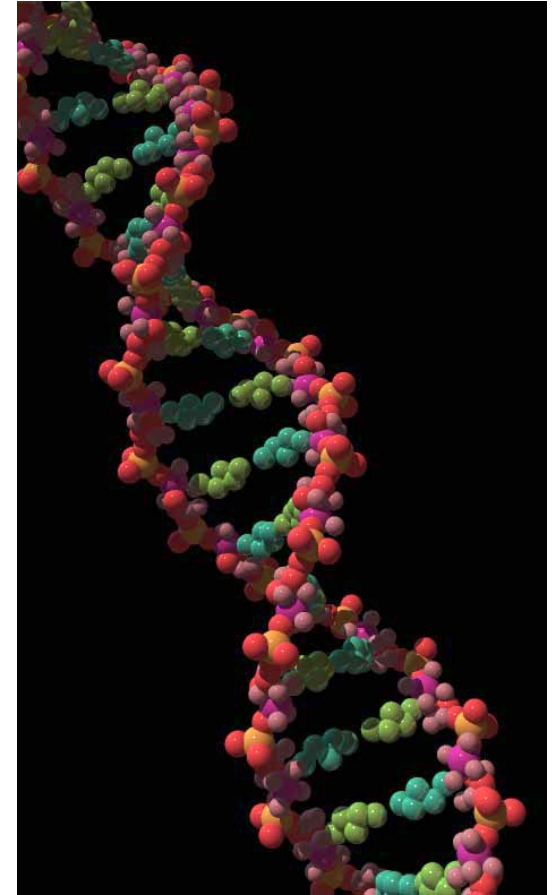
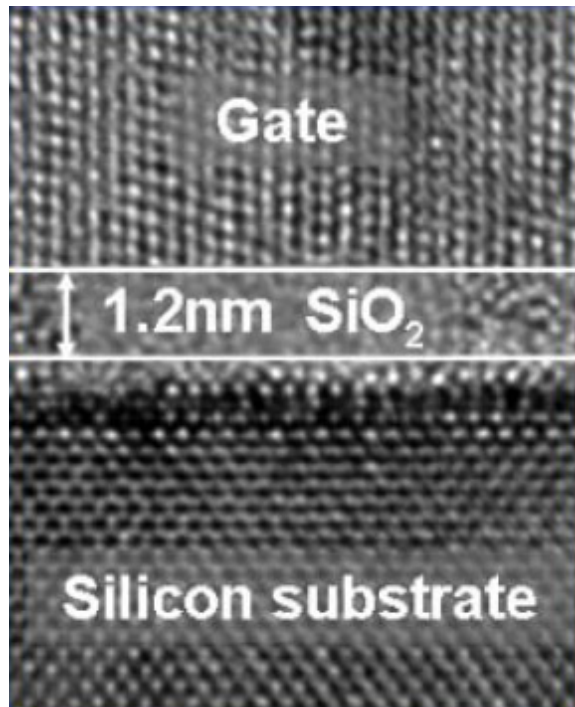


Neuron / μ chip

M.H

Den digitale revolusjon 5

Sammensmelting av
mikroelektronikk og nanoteknologi



Hva blir forskjellen på maskin og biologi?

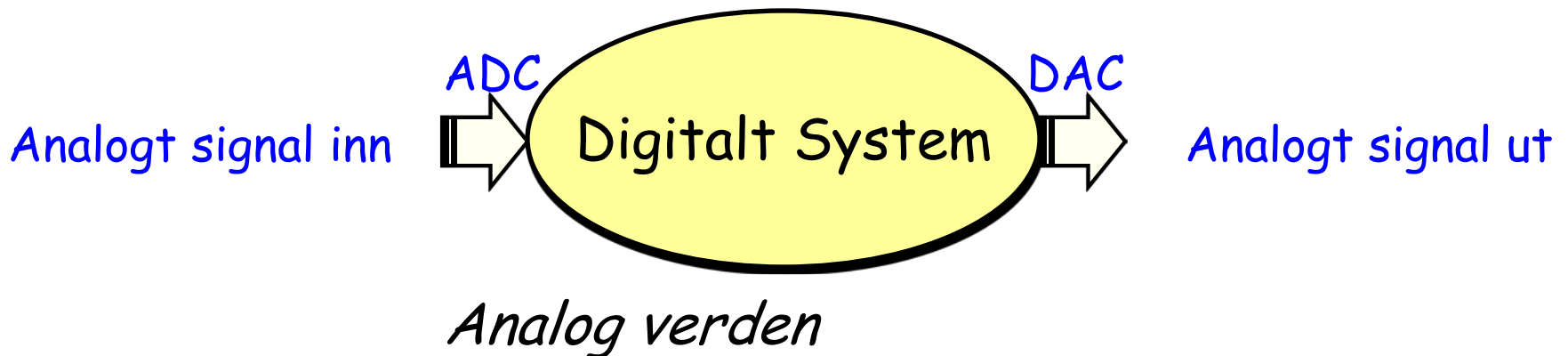
Hvorfor er digitale system bedre enn analoge system?

- Tapsfri lagring signaler (kopiering)
- Tapsfri signalbehandling
- Kraftigere muligheter for manipulering (filtrering)
- Enklere design
- Billigere...

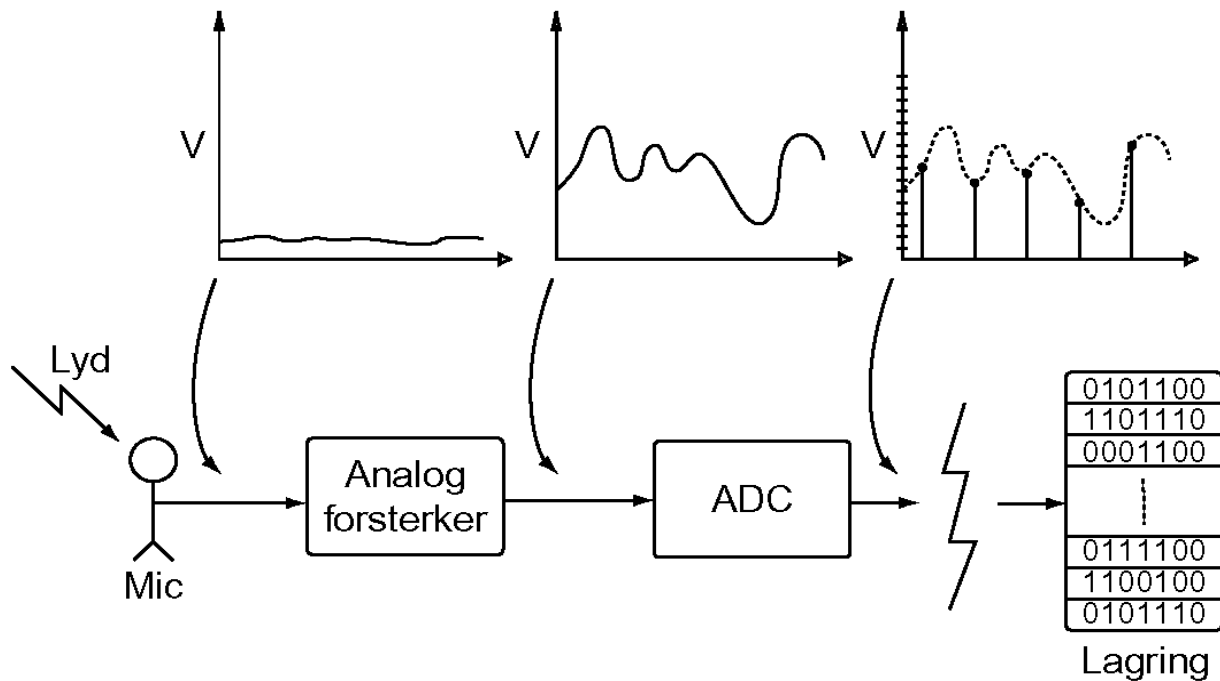
Det digitale "egget"

Verden vi lever i er "stort sett" analog

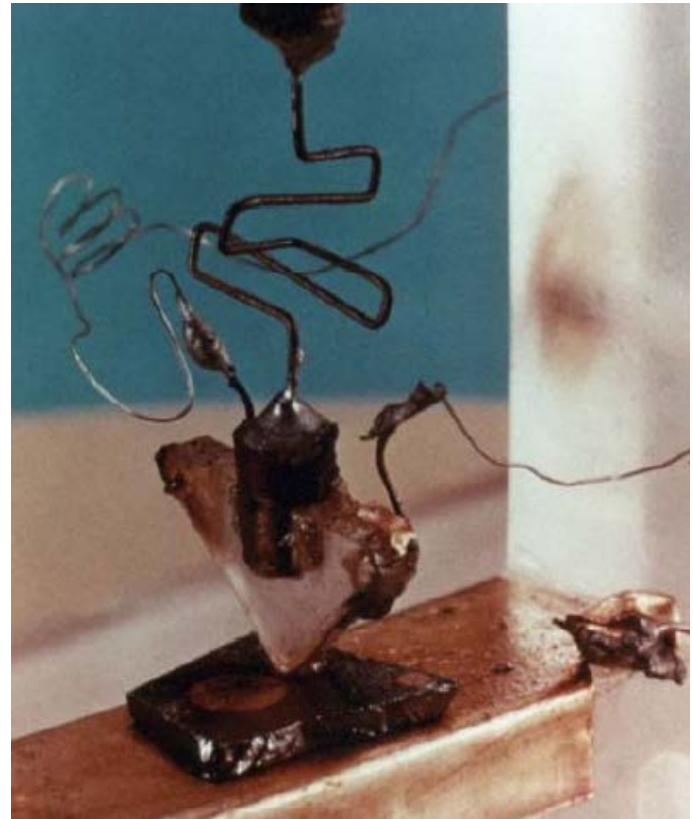
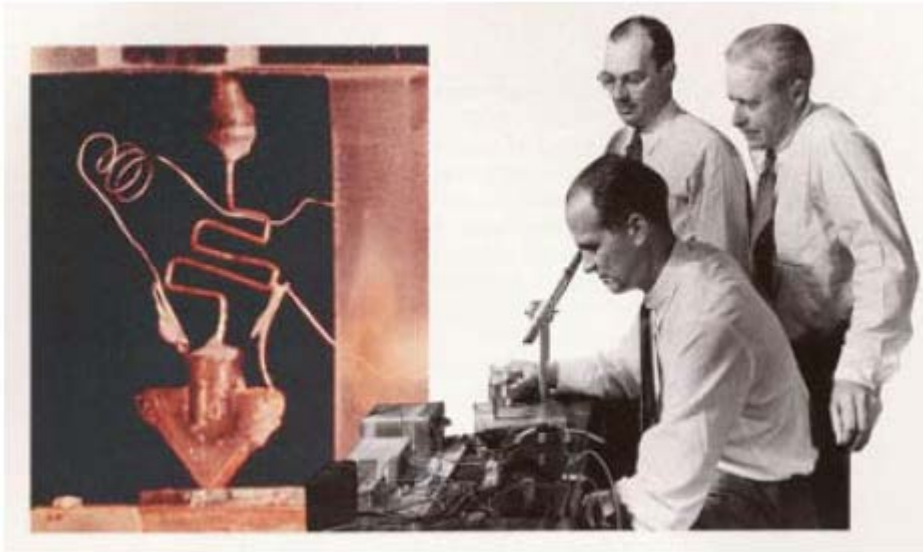
Skal vi bruke digitale system til å håndtere analoge fenomen trenger vi en **Analog-til-Digital konverter** + en **Digital-til-Analog konverter**



Eksempel: Lydinnspeiling



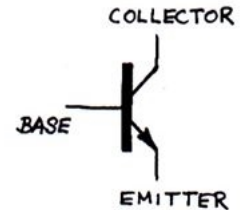
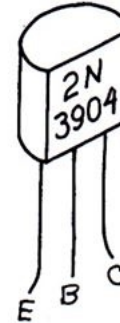
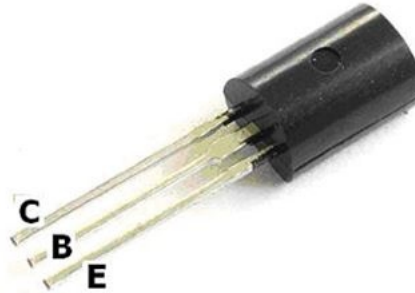
Transistoren 1947



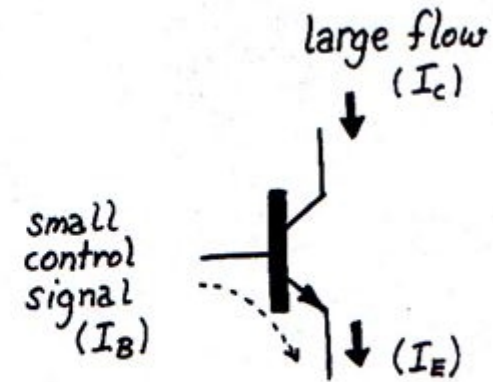
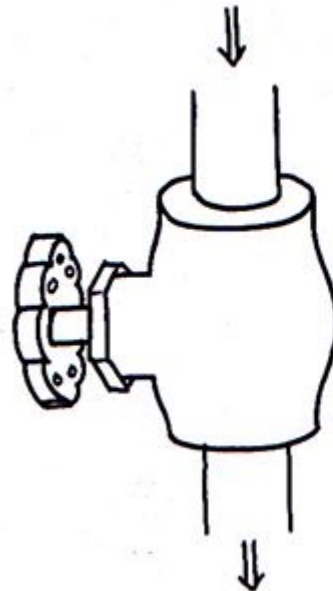
- Første transistor fra 1947. Bell labs. William Shockley
- Erstattet store og strømkrevende radiorør.
- Kan fungere som av/på bryter.

Transistor som strømbryter

- Transistoren kan vi benytte for å slå av og på strøm.



- Spenningen på B kontrollerer strømmen mellom C og E
- Tilsvarende som en vannkran.



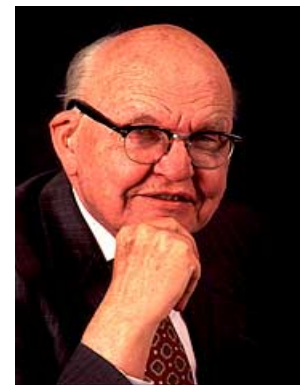
Mikroelektronikk

- Den egentlige årsaken til teknologiutvikling
- Integreerte kretser
 - Texas Instruments - 1958
 - germanium



- Fairchild
 - Første kommersielle krets i silisium
- Intel - 1968
 - Første mikroprosessor i 1971

Kanskje tidenes mest imponerende og viktigste industrielle revolusjon



Jak S. Kilby



*Andy Grove
Robert Noyce
Gordon Moore*

Fra radorør til mikrochip

- Mindre størrelse
- Mindre strømforbruk
- Billigere
- Mulighet for å lage store digital systemer



1907

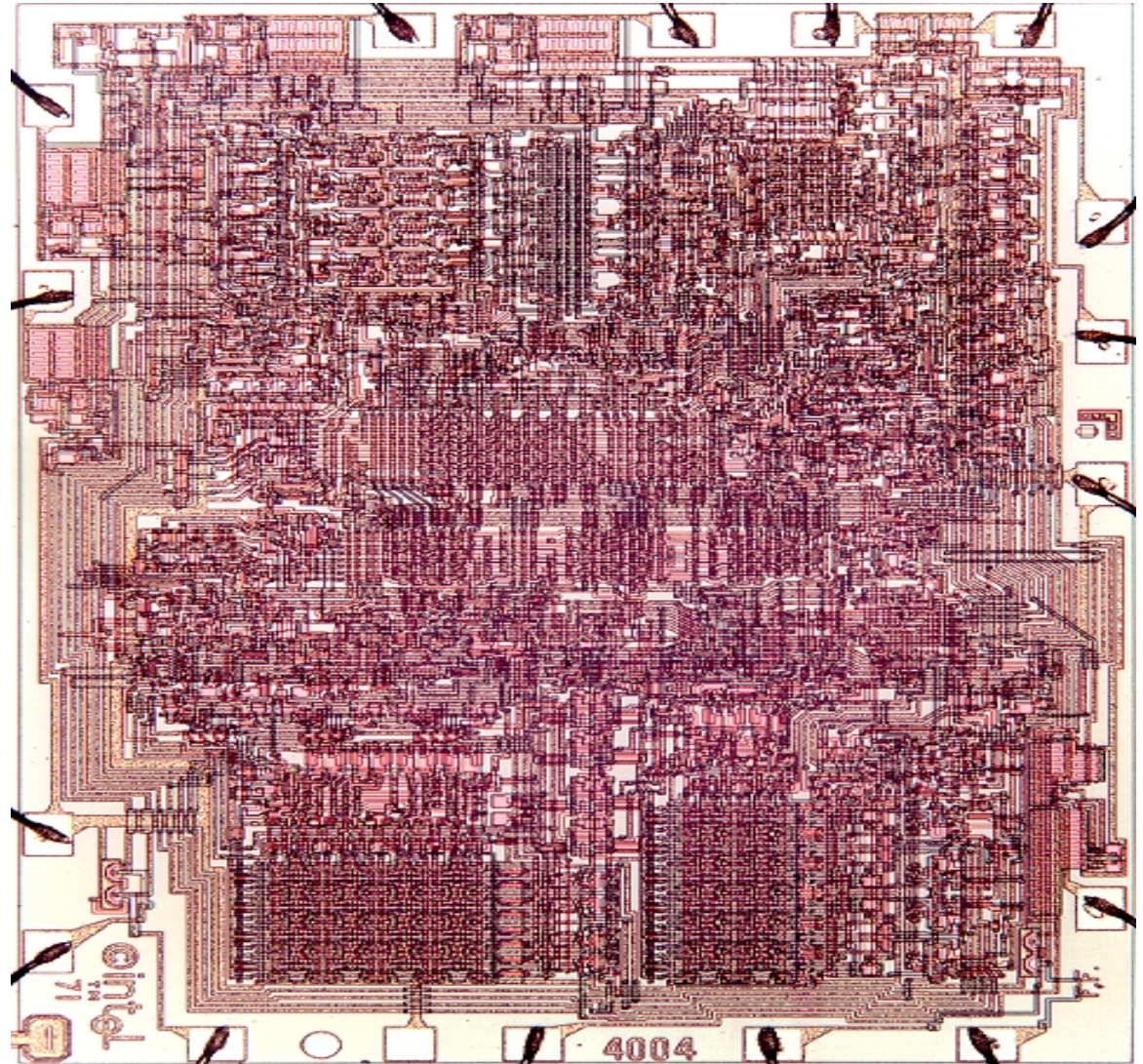


1947



1963

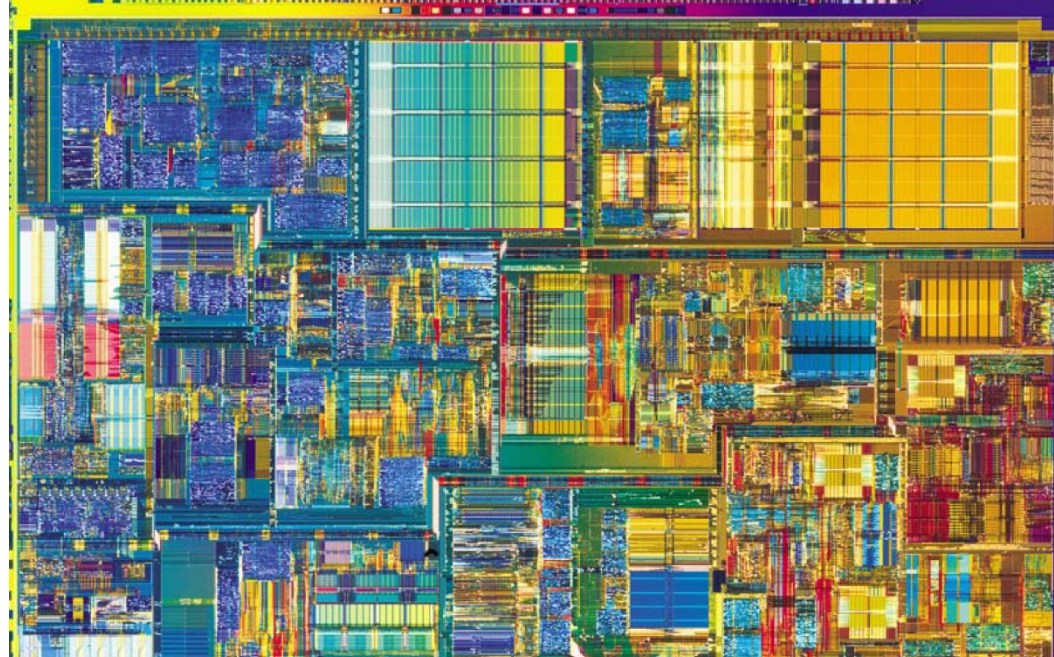
Intel 4004, mikro-chip



- 1971
- 2300 transistorer
- 108kHz klokke
- max 648 byte minne
- 4 bit bus

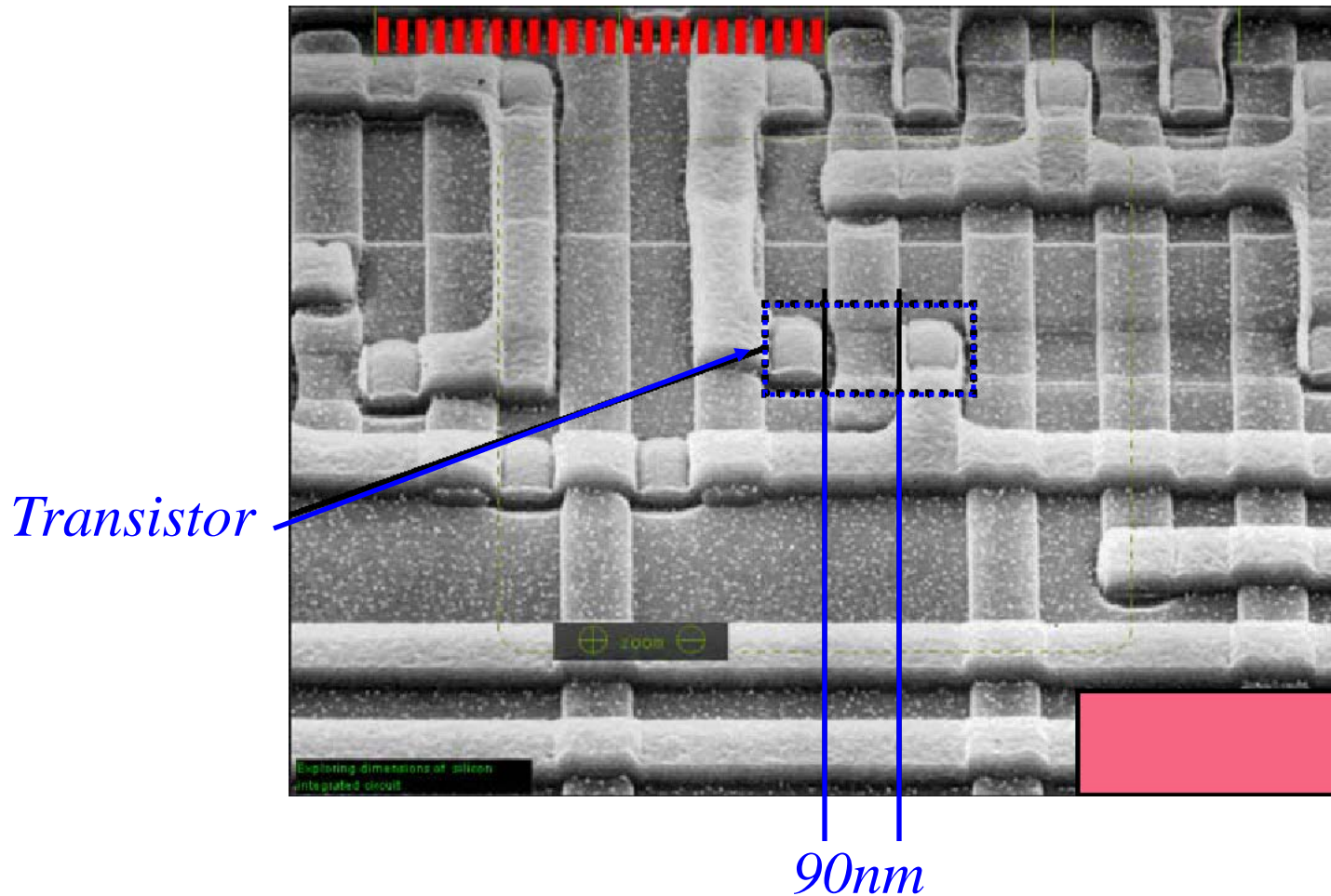
Pentium 4

- Ca. 2008
- >42 000 000 transistorer
- >3GHz klokke
- 90-180 nm



- 2017 AMD EPYC 7401P - 24-core server CPU
- 19,200,000,000 transistorer
- 14 nm teknologi, fortsatt 3GHz

Digital krets, ca. 10 000 x zoom

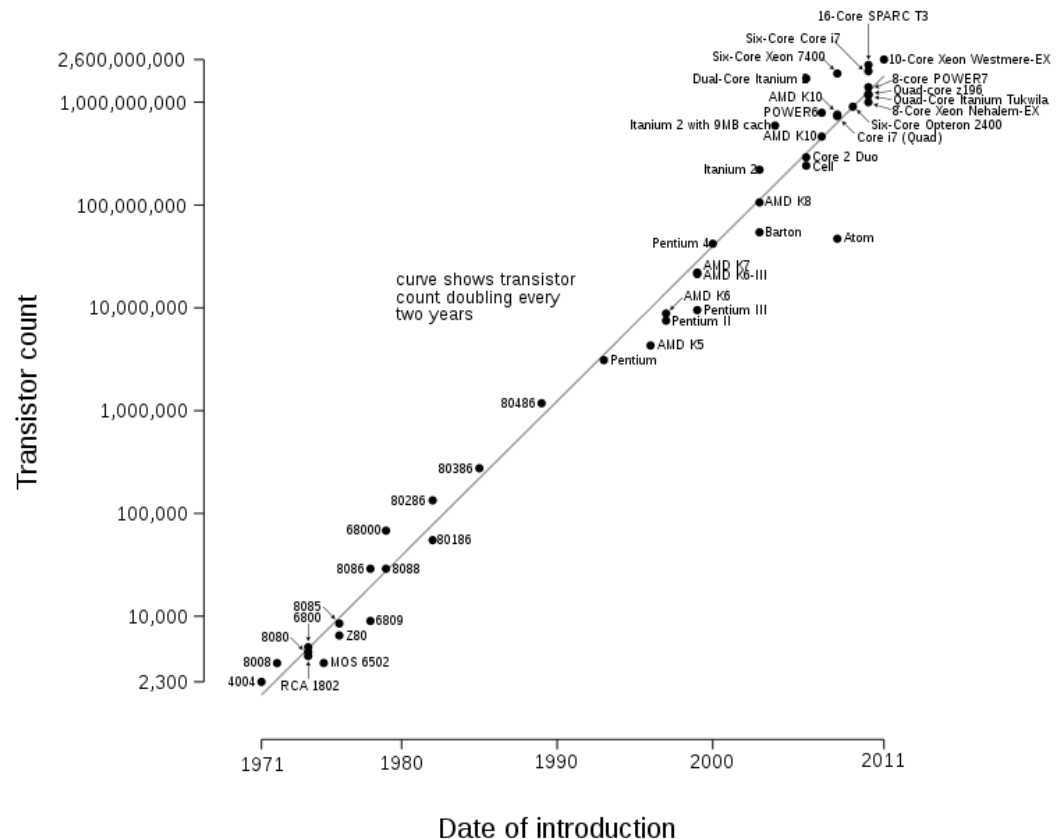


M.H

Moore's law 1965

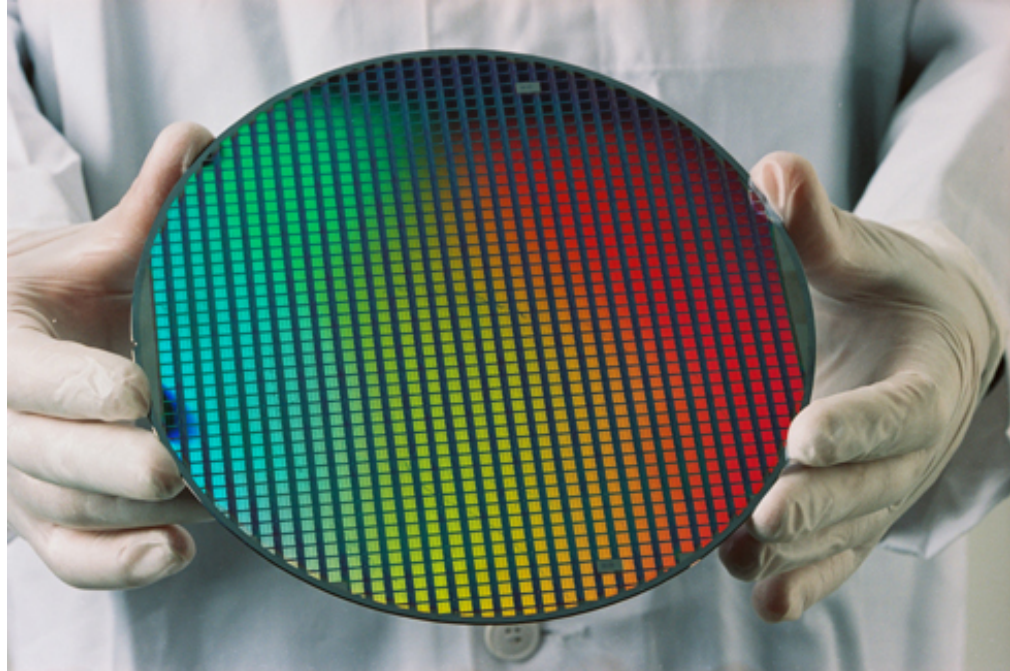
- Antall transistorer per chip doubles hver 18. mnd.
- Sett en eksponensiell vekst, men har begynt å avta.
- 2015: "I see Moore's law dying here in the next decade or so."

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Produksjon

- Silisium wafer
- Identiske chips ferdige til å sages
- Dyr prosess, men billig dersom man masseproduserer.



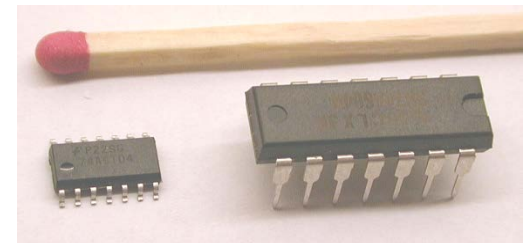
Smarttelefoner overtar utviklingen?

- Selges mer smarttelefoner enn PCer.
- Mye av utviklingen skjer derfor mye innen mobilmarkedet.
- Statig raskere, men også integrasjon av nye «smarte» løsninger.



Definisjoner

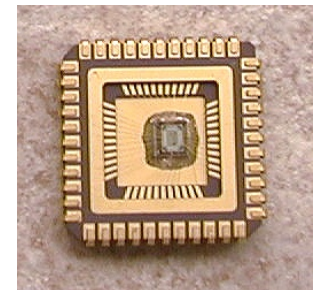
- VLSI (Very-Large-Scale-Integrated-Circuits)
 - Mer enn 100 000 porter på samme chip
- LSI (Large-Scale-Integrated-Circuits)
 - Noen få 1000 porter på samme chip
- SSI (Small-Scale-Integrated-Circuits)
 - Noen få porter på samme chip



Digitale designmetoder I

Hardware-basert design:

- SSI design (Smale-scale integration)
 - Setter sammen SSI pakker på kretskort
- FPGA (Field Programmable Gate Array)
 - Programmerbar logikk, eks. Xilinx, Altera, osv..
- ASIC (Application Specified Integrated Circuit)
 - Skreddersydd logikk (designer på transistor nivå)
 - Strømforbruk / hastighet / areal / spesialfunksjoner / pris / kombinert analog-digital / andre spesielle formål



Digitale designmetoder II

Software kombinert med generell hardware:

- MicroController (uC)
 - Mikroprosessor med tilhørende I/O og memory på en brikke (datamaskin på en brikke)
- DSP (Digital Signal Processor)
 - Microcontroller spesialbygd for rask signalbehandling, eks. video, audio, osv..
- PC / smarttelefon ...
 - Digitale operasjoner kan utføres ved hjelp av passende I/O

Tallsystemer

Desimale tall

Et desimalt tall er representert ved symbolene 0, 1, 2, ... 9

- Kodingen er posisjons bestemt.
- Følger titalssystemet.

Eksempel:

$$(7392)_{\text{dec}} = 7 \cdot 10^3 + 3 \cdot 10^2 + 9 \cdot 10^1 + 2 \cdot 10^0$$

Bincære tall

Tall må generelt ikke representeres ved 10 symboler (antall fingre)

Eksempel: bincære tall

- Et bincært tall er representert ved symbolene **0** og **1**
 - Kodingen er posisjons bestemt

Eksempel:

$$(101)_{\text{bin}} = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

$$(101)_{\text{bin}} = (5)_{\text{des}}$$

Bincær telling

Tilrådighet:
symbolene

0,1

Tallet "3"

Bincær
rep.

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100

Desimal
rep.

00
01
02
03
04
05
06
07
08
09
10
11
12

Tilrådighet:
symbolene

0,1,2,3,4,5,6,7,8,9

Tallet "3"

Oktale tall

Et oktalt tall er representert ved symbolene 0, 1, 2, ... 7
- Kodingen er posisjonsbetinget med grunntall 8

Eksempel:

$$(252)_{\text{okt}} = 2 \cdot 8^2 + 5 \cdot 8^1 + 2 \cdot 8^0$$

$$(252)_{\text{okt}} = (170)_{\text{des}}$$

Heksadesimale tall

Et heksadesimale tall er representert ved symbolene
0, 1, 2, ... 8, 9, A, B, C, D, E, F

- Kodingen er posisjonsbetinget med grunntall 16

Eksempel:

$$(2B9)_{\text{heks}} = 2 \cdot 16^2 + 11 \cdot 16^1 + 9 \cdot 16^0$$

$$(2B9)_{\text{heks}} = (697)_{\text{des}}$$

Oktal og heksadesimal telling

Heksadesimal	Desimal	Oktal	Binær
0 0	0 0	0 0	0 0 0 0 0
0 1	0 1	0 1	0 0 0 0 1
0 2	0 2	0 2	0 0 0 1 0
0 3	0 3	0 3	0 0 0 1 1
0 4	0 4	0 4	0 0 1 0 0
0 5	0 5	0 5	0 0 1 0 1
0 6	0 6	0 6	0 0 1 1 0
0 7	0 7	0 7	0 0 1 1 1
0 8	0 8	1 0	0 1 0 0 0
0 9	0 9	1 1	0 1 0 0 1
0 A	1 0	1 2	0 1 0 1 0
0 B	1 1	1 3	0 1 0 1 1
0 C	1 2	1 4	0 1 1 0 0
0 D	1 3	1 5	0 1 1 0 1
0 E	1 4	1 6	0 1 1 1 0
0 F	1 5	1 7	0 1 1 1 1
1 0	1 6	2 0	1 0 0 0 0
1 1	1 7	2 1	1 0 0 0 1
1 2	1 8	2 2	1 0 0 1 0
1 3	1 9	2 3	1 0 0 1 1
1 4	2 0	2 4	1 0 1 0 0

Tallet $(12)_{des}$

Konvertering fra grunntall "r" til desimal

Generelt:

$$(\dots a_2 a_1 a_0, a_{-1} a_{-2} \dots)_r = \dots + a_2 \cdot r^2 + a_1 \cdot r^1 + a_0 \cdot r^0 + a_{-1} \cdot r^{-1} + a_{-2} \cdot r^{-2} + \dots$$

Eksempel:

$$(1A5,1C)_{16} = 1 \cdot 16^2 + 10 \cdot 16^1 + 5 \cdot 16^0 + 1 \cdot 16^{-1} + 12 \cdot 16^{-2} = (421,1133)_{\text{des}}$$

Konvertering fra desimal til bincær

Prosedyre:

1. Del det desimale tallet på 2
2. Resten etter divisjon, multiplisert med 2 blir LSB
3. Del det nye desimale tallet på 2
4. Resten etter divisjon, multiplisert med 2 blir neste bit
5. Osv.

Konvertering fra decimal til binær

Eksempel:

Konverter tallet $(41)_{\text{des}}$ til binær

$$\begin{array}{rcll} 41/2 & = & 20 + 1/2 & a_0 = 1 \quad \text{LSB} \\ 20/2 & = & 10 + 0/2 & a_1 = 0 \\ 10/2 & = & 5 + 0/2 & a_2 = 0 \\ 5/2 & = & 2 + 1/2 & a_3 = 1 \\ 2/2 & = & 1 + 0/2 & a_4 = 0 \\ 1/2 & = & 0 + 1/2 & a_5 = 1 \end{array}$$

Dermed: $(41)_{\text{des}} = (101001)_{\text{bin}}$

Konvertering fra desimal til grunntall "r"

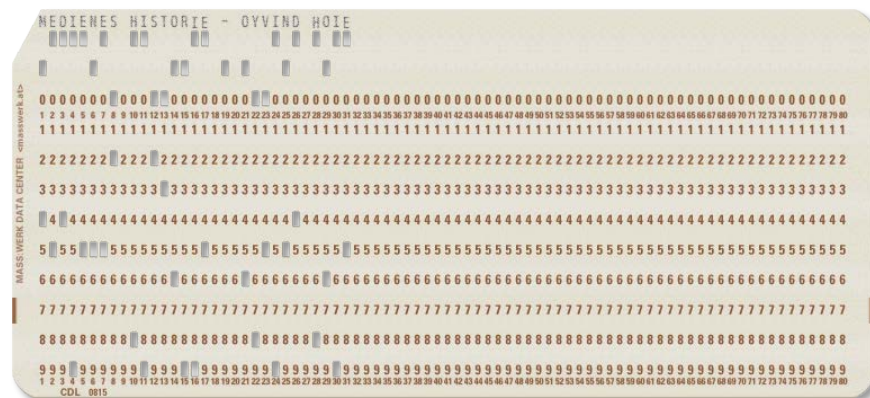
Gjenta prosedyren fra forrige side. Bytt ut grunntallet 2 med r . Resten multiplisert med r blir det aktuelle sifferet

Digital hardvare-representasjon

To «historiske» eksempler

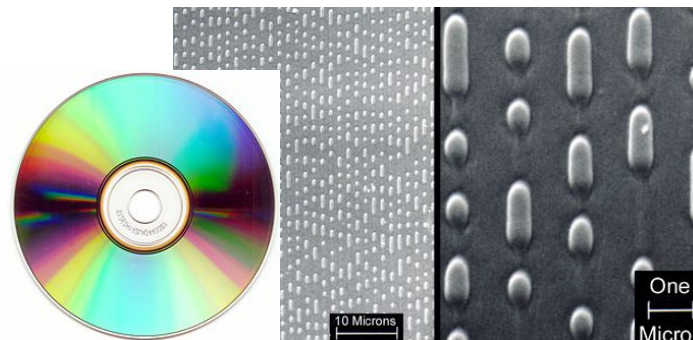
Hullkort

- "0" eller "1" representeres ved hull i gitte områder.



CD plate

- "0" eller "1" representeres ved refleksjon av lys i ett gitt område.



Det binære tallsystemet er en effektiv utnyttelse av plassen når man kun har to kombinasjoner.

M.H



Digital hardware-representasjon

PC og andre elektroniske systemer:

- "1" representeres ved 5V på en ledning
- "0" representeres ved 0V på samme ledning

Harddisk:

- "1" representeres ved tilstedeværelse av magnetisk felt i ett gitt område
- "0" representeres ved fravær av magnetisk felt i samme område

Tilsvarende for SSD (Solid-State Drive)

Digital representasjon

"Alt" kan kodes som binære tall

Eksempel:

Bokstaver

ASCII Table (7-bit)

American Standard
Code for Information
Interchange

Decimal	Octal	Hex	Binary	Value
061	075	03D	00111101	=
062	076	03E	00111110	>
063	077	03F	00111111	?
064	100	040	01000000	@
065	101	041	01000001	A
066	102	042	01000010	B
067	103	043	01000011	C
068	104	044	01000100	D
069	105	045	01000101	E
070	106	046	01000110	F
071	107	047	01000111	G
072	110	048	01001000	H
073	111	049	01001001	I
074	112	04A	01001010	J
075	113	04B	01001011	K
076	114	04C	01001100	L
077	115	04D	01001101	M
078	116	04E	01001110	N
079	117	04F	01001111	O
⋮	⋮	⋮		

M.H

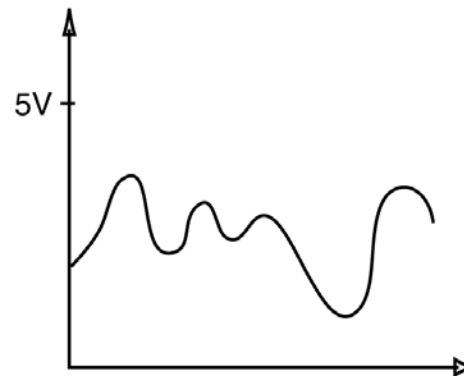
Digitale systemer

"Alt" kan kodes som binære tall

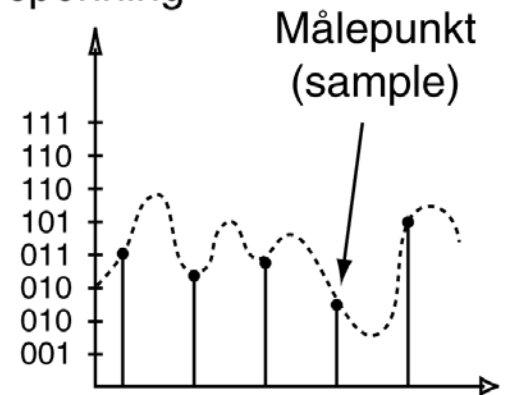
Eksempel: Lyd

Sampling og kvantisering

Analog
spenning
fra mikrofon



Kvantisert
(avrundet)
spenning



Digitale systemer

- På samme måte kan vi representere alle slags dokumenter, bilder, videor osv.
- Men for å kunne behandle binære data effektivt trenger vi regler og verktøy:
 - Binær logikk og boolsk algebra.
- Binær logikk eksempel:
 - "1" / "0" kan tolkes som "yes" / "No"

Bincær logikk - boolsk algebra

Definerer: Variable: "0" og "1", og binære operasjoner.
Ender opp med en fullverdig algebraisk struktur

Definerte basis operasjoner:

AND " . "

OR " + "

NOT " ' "

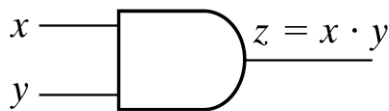
Ved å kombinerer disse 3 operasjonene kan vi lage alle mulige digitale funksjoner

Sannhetstabell

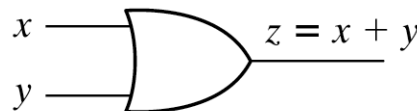
AND		
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

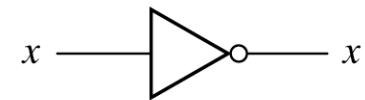
NOT	
X	Y
0	1
1	0



(a) Two-input AND gate



(b) Two-input OR gate

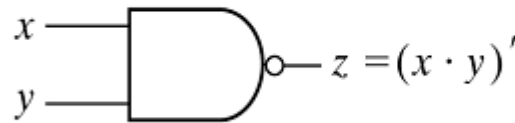
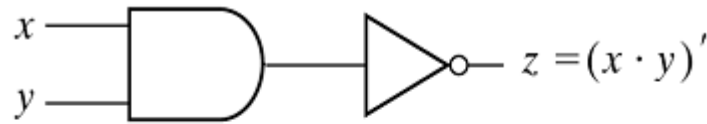


(c) NOT gate or inverter

Alle disse portene kan vi lage ved å sette sammen transistorer på forskjellige måter.

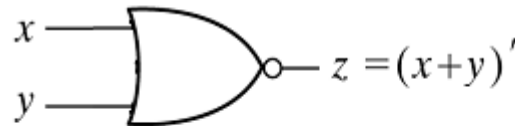
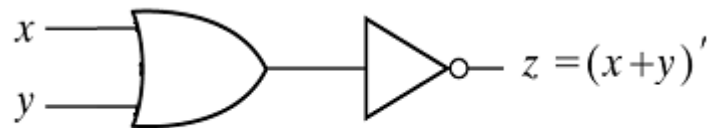
Sannhetstabell

Enda et par vanlige byggeblokker:
NAND og NOR



(a) Two-input NAND gate

NAND	
X Y	Z
0 0	1
0 1	1
1 0	1
1 1	0



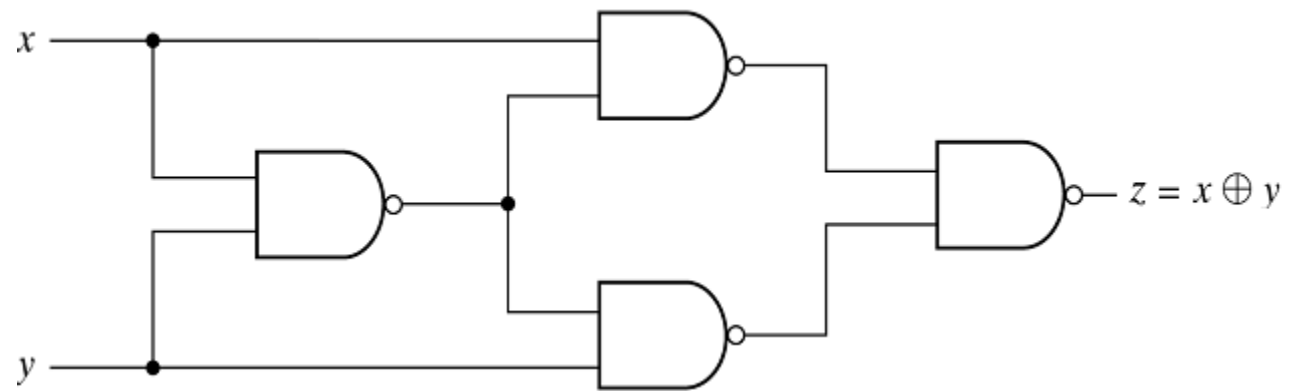
(a) Two-input NOR gate

NOR	
X Y	Z
0 0	1
0 1	0
1 0	0
1 1	0

Sannhetstabell

Den siste, vanlige byggeblokken

XOR		Z
X	Y	
0	0	0
0	1	1
1	0	1
1	1	0

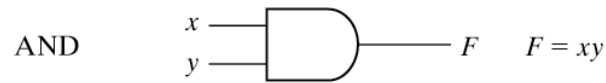


Exclusive-OR Implementations

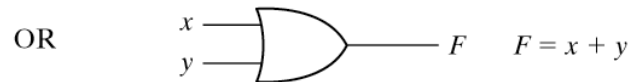
2-inputs byggeblokker oversikt

Nyttig nettside

www.play-hookey.com



x	y	F
0	0	0
0	1	0
1	0	0
1	1	1



x	y	F
0	0	0
0	1	1
1	0	1
1	1	1



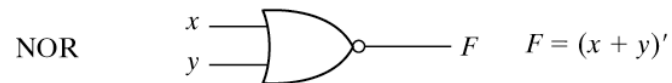
x	F
0	1
1	0



x	F
0	0
1	1



x	y	F
0	0	1
0	1	1
1	0	1
1	1	0



x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

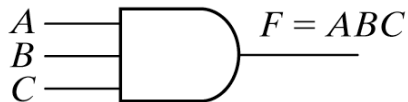
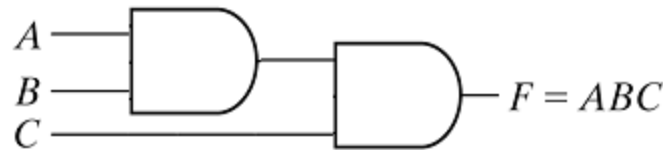


x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

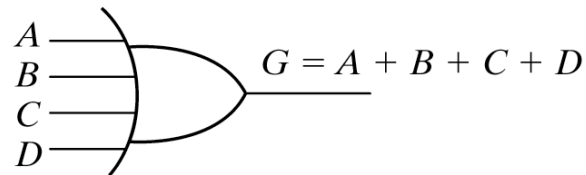
Generelle porter

Kan sette sammen 2-inputs porter til fler-inputs porter

Eksempel:



(a) Three-input AND gate



(b) Four-input OR gate

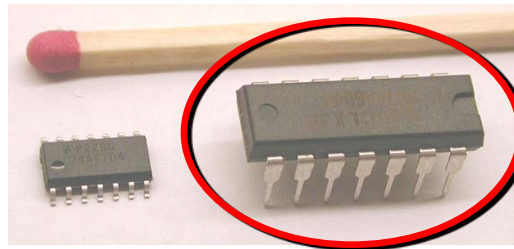
3-input AND

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

SSI - Fysisk innpakning

SSI - Small scale Integration

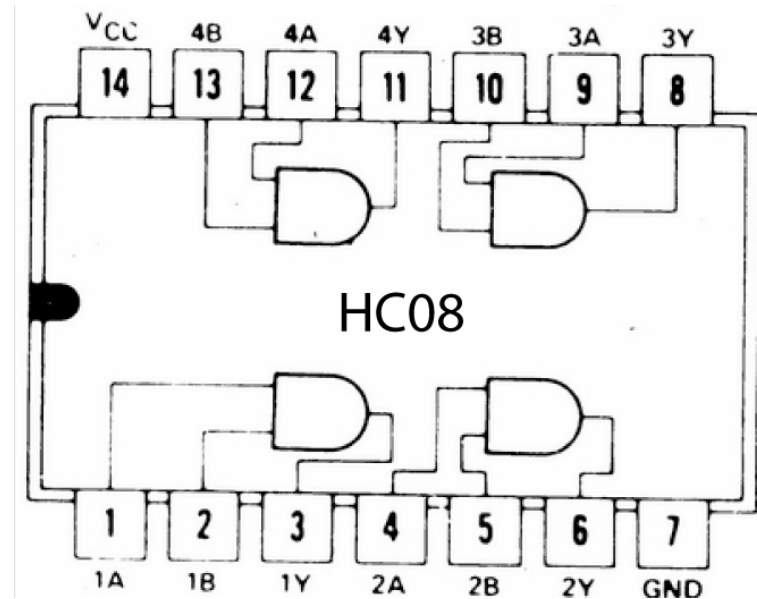
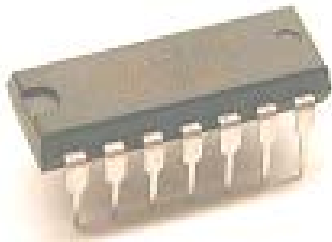
Overflate-montert



Hull-montert

SSI - fysisk innpakning

Eksempel:
kretsen
CD74HC08
2-inputs AND

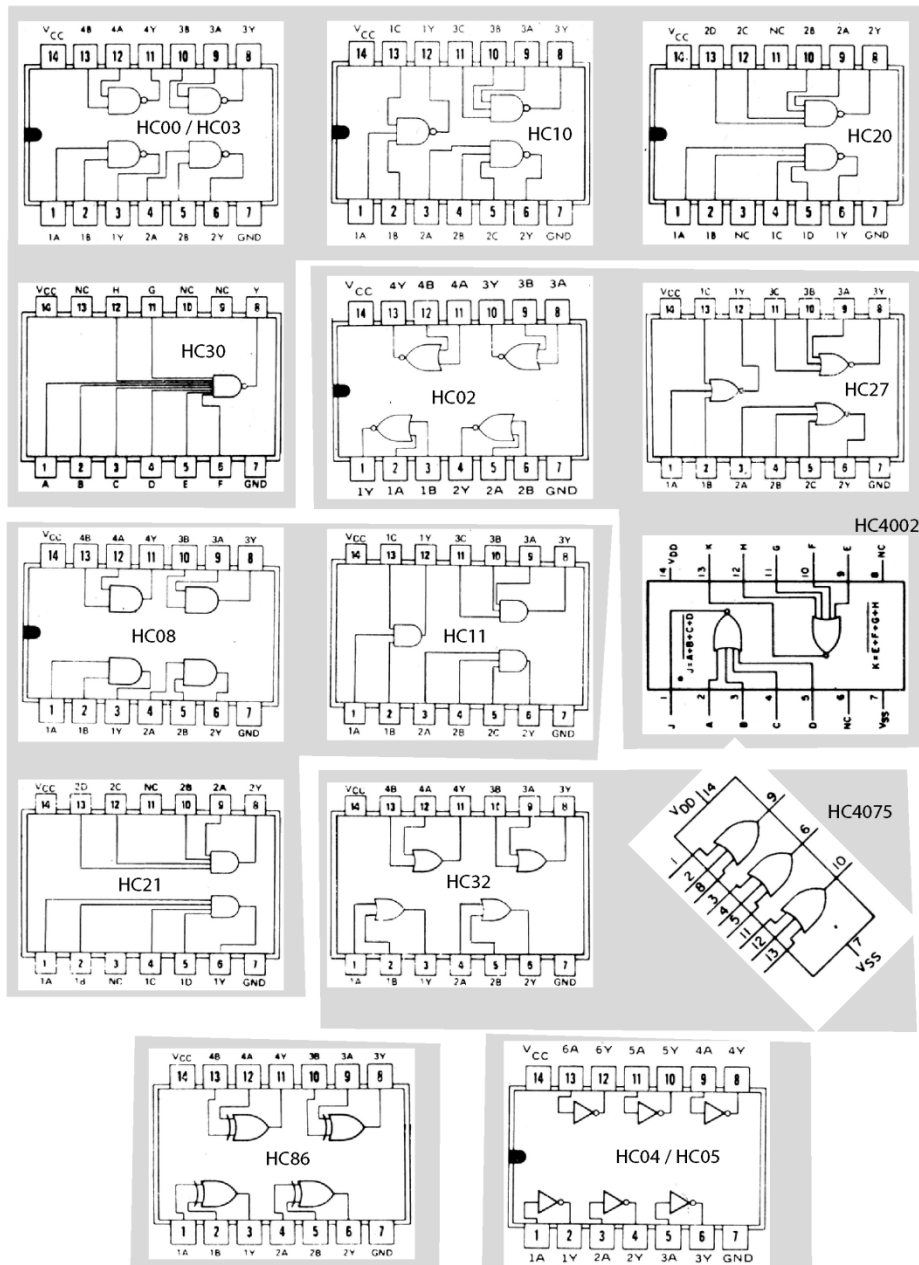


Spenningsforsyning:
"Vcc" eller "Vdd" = 5V
Jord:
"Gnd" eller "Vss" = 0V

M.H

Vanlige pakker

- NOT (inverter)
- 2,3,4 input AND
- 2,3 input OR
- 2,3,4 input NOR
- 2,3,4,8 input NAND
- 2 input XOR



SSI teknologi

To vanlige teknologier:

TTL (transistor-transistor logikk)

- *Bipolare transistorer*
- *Standard TTL (54/74), LS, F,S,AS*

CMOS logikk (Complementary MetalOxide Semikonduktor)

- *Mos transistorer*
- *Standard 4000, HC, AC, LV og HCT, ACT*

Logiske inngangsnivå

Hvilke inngangsspenninger oppfatter en port som "1" og "0"?

TTL:

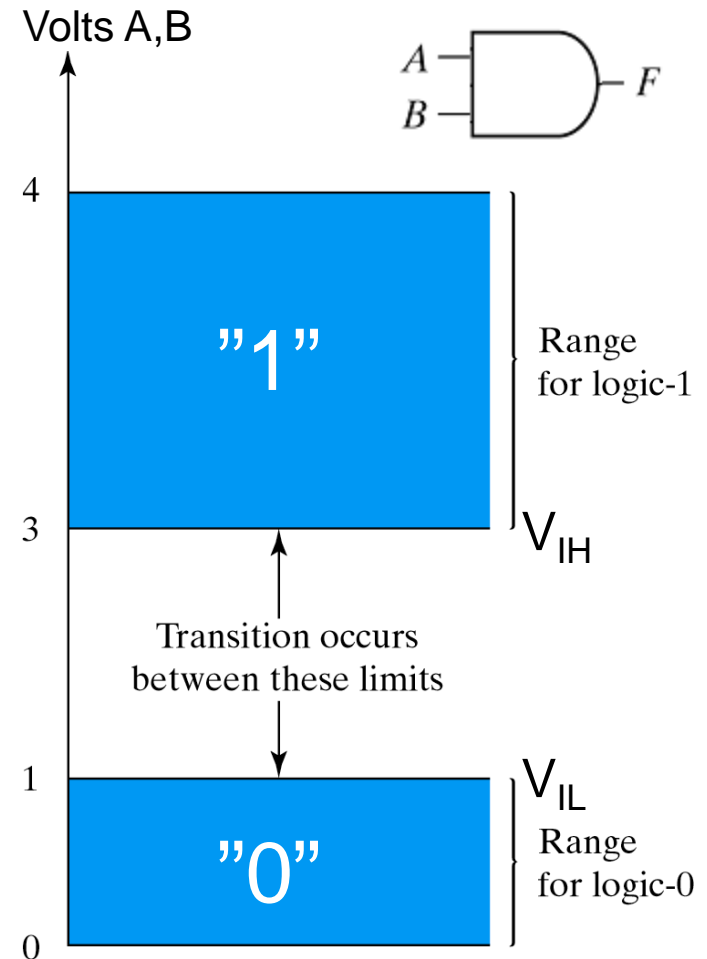
- $V_{IH} = 2V$ (min)
- $V_{IL} = 0.8V$ (max)

CMOS:

- $V_{IH} = 3.3V$ (min)
- $V_{IL} = 1.5V$ (max)

CMOS (HCT/ACT):

- V_{IH} og V_{IL} som for TTL



LSI - Fysisk innpakning

LSI - Large Scale Integration

Eksempel: Programmerbar logikk - 800 frie porter



Flatpack

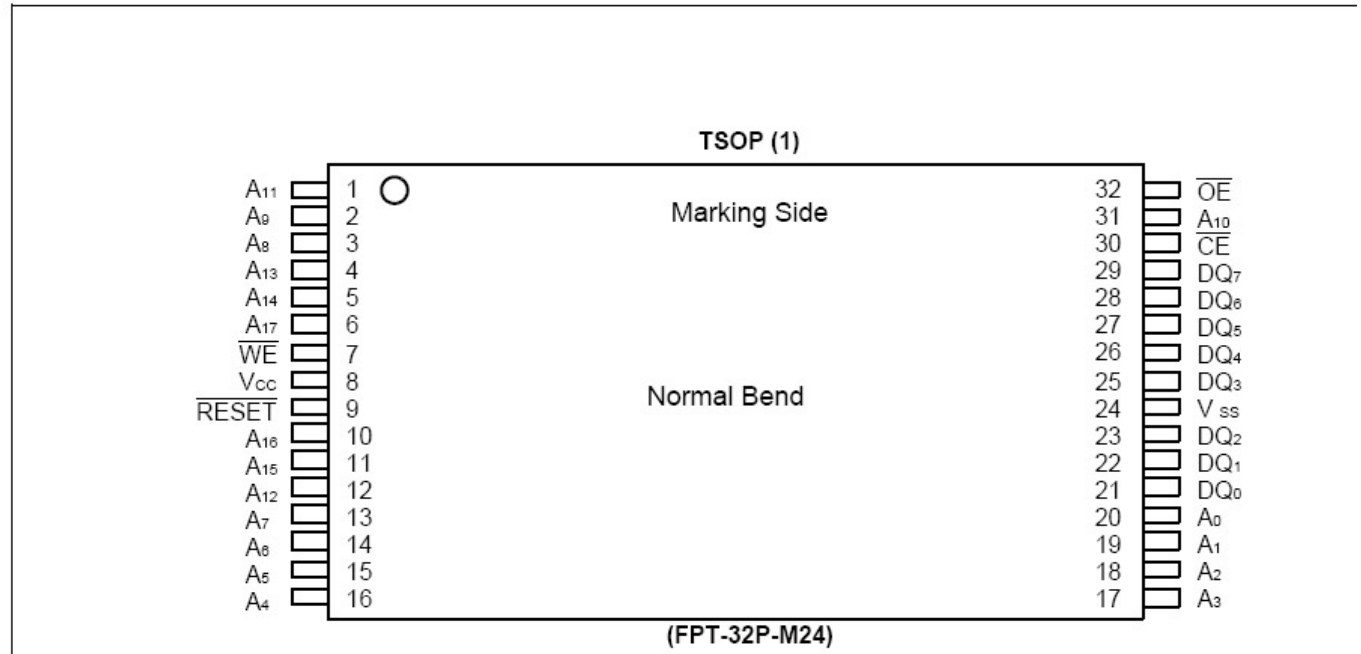
VLSI - Fysisk innpakning

VLSI -Large Scale Integration

MBM29F002TC-55/-70/-90/**MBM29F002BC**-55/-70/-90

Eksempel:
Flash-minne
"FUJITSU"

■ PIN ASSIGNMENTS



Flatpack

M.H

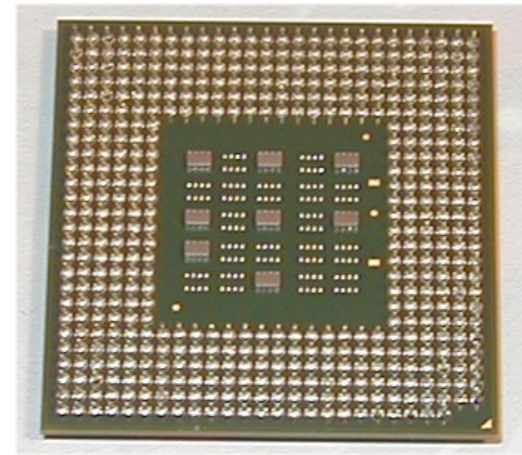
VLSI - Fysisk innpakning

VLSI -Large Scale Integration

Eksempel:
Intel
Pentium4



Overside



Underside

Oppsummering

- Desimale / binære tall
- Digital hardvare-representasjon
- Binær koding av bokstaver og lyd
- Boolsk algebra
- Digitale byggeblokker / sannhetstabell
- Generelle porter
- Fysisk innpakning
- SSI teknologi - CMOS/TTL
- Logiske inngangsnivå - CMOS/TTL