Introduksjon

TK1100

Digital teknologi

O'te forelesning

Gjennomgås i dag

- Kursopplegget
- Datamaskinens bruksområder
- Datamaskinens bestanddeler
- Datamaskinen og PCens historie

Om Bengt Østby

- C-programmerer
- Windows system drivere
- 16 års erfaring fra anti-virus bransjen
- Jobbet med avansert video overvåking

- Lead Programmer, Norman
- Enterprise Architect Security CoE, AVG
- Foreleser NITH og Westerdals
- Security Concepts Group
- Head of OffSec, Capgemini Cybersecurity

TK1100 – forkunnskaper

- Ingen formelle krav til forkunnskaper
- Har man lite forkunnskaper vil man måtte jobbe MYE mer med emnet
- Alle senere emner bygger på dette emnet

- Datamaskin virkemåte
- Binær matematikk
- Nettverksteknologier

Kurs-opplegget

- 12 uker undervisning og øving
  - 2-3 timer forelesning tirsdager fra 15:15, fulgt av
  - 1-2 timer veiledet øving

- Eksamen
  - Deleksamen 1, 2 timer Skriftlig eksamen
    - Dekker de 5 første forelesningene, teller 25% av karakter
    - Introduksjonen, Binære formater, tegnsett og enkoding, maskinarkitektur, og datamaskinens oppbygging
  - Deleksamen 2, 3 timer Skriftlig eksamen
    - Dekker de siste forelesningene, teller 75% av karakter
  - **NB!!! Dersom du har rett til ekstra tid pga lese- / skrive-vansker o.l. SØK OM DET NÅ MED EN GANG!!!!**

- Egenarbeid er VIKTIG (aktiviser kunnskap!)

Egenarbeid

- Fagplanen sier 200 timer arbeid i dette emnet
- 48 timer forelesning og øving
- Feks 110 timer forbredelse og arbeid ifm forelesninger; 10 timer til HVER forelesning
  - 2 timer hver mandag; repetisjon forrige forelesning
  - 8 timer hver tirsdag formiddag/kveld; forberedelse og etterarbeid for å sikre læring
- Feks 42 timer forbredelse til eksamen
  - 5 dager med 8 timers full jobbing til eksamen
  - 2 timer repetisjon siste kveld / morgen

Forbered deg på mye jobbing i TK fag!

- De første 4 forelesningene er det (for noen) vanskelig matematikk
- De neste 6 forelesningene er det MYE ny læring, og ganske mye pugging (TCP/IP ++)
- TK fagene er det flest studenter stryker på, hvis man ikke er i forelesning er det vanskelig å bestå (også de siste forelesningene!)
- Hvis man ikke jobber 10 timer ekstra i uken er det vanskelig å bestå
- Dette er et vanskelig fag, forbered deg på det!

Veiledere

- Alexander Bredesen
- Hans Ludvig
- Aleksander Omer Hauabakk-Anwar
- Jason Williams
- Sander Sjøthun
- Ben Nicholay Gyllenhaal Johansen
- Kjell-Olaf Slagnes
- Heidi Fikkan

Pensum?

- «Pensum» er fra Latin:
  - Pendere = en bestemt mengde ull man (typisk: en slave) skulle bearbeide i løpet av en dag.
  - M.a.o OPPGAVE
- Pensum er alle forelesninger i emnet (det er ikke noen egen lærebok som må kjøpes)
- Pensum er også å kunne besvare alle oppgaver som blir gitt i emnet i løpet av høsten
- Oppgavene er ofte en oppsummering av forelesning, og en del sett er multiple choice oppgaver av forskjellig oppbygging.
- I tillegg vil det være en del mer åpne oppgaver, både som egenstudier og tekstoppgaver som ligner mer på hva dere får på eksamen
- Det blir lagt ut anbefalte linker til websider som kan hjelpe til i oppgaveløsning

- Det ligger *kompendium* på emnesiden som kortfattet dekker flertallet av de viktigste kompetansemålene.
  - Skrevet av tidligere foreleser Bjørn Olav Listog, bedre enn å finne en lærebok i emnet da disse går spesifikt mot læringsmålene i TK1100

KOMPTANSESMÅL

TEMA i emnet TK1100

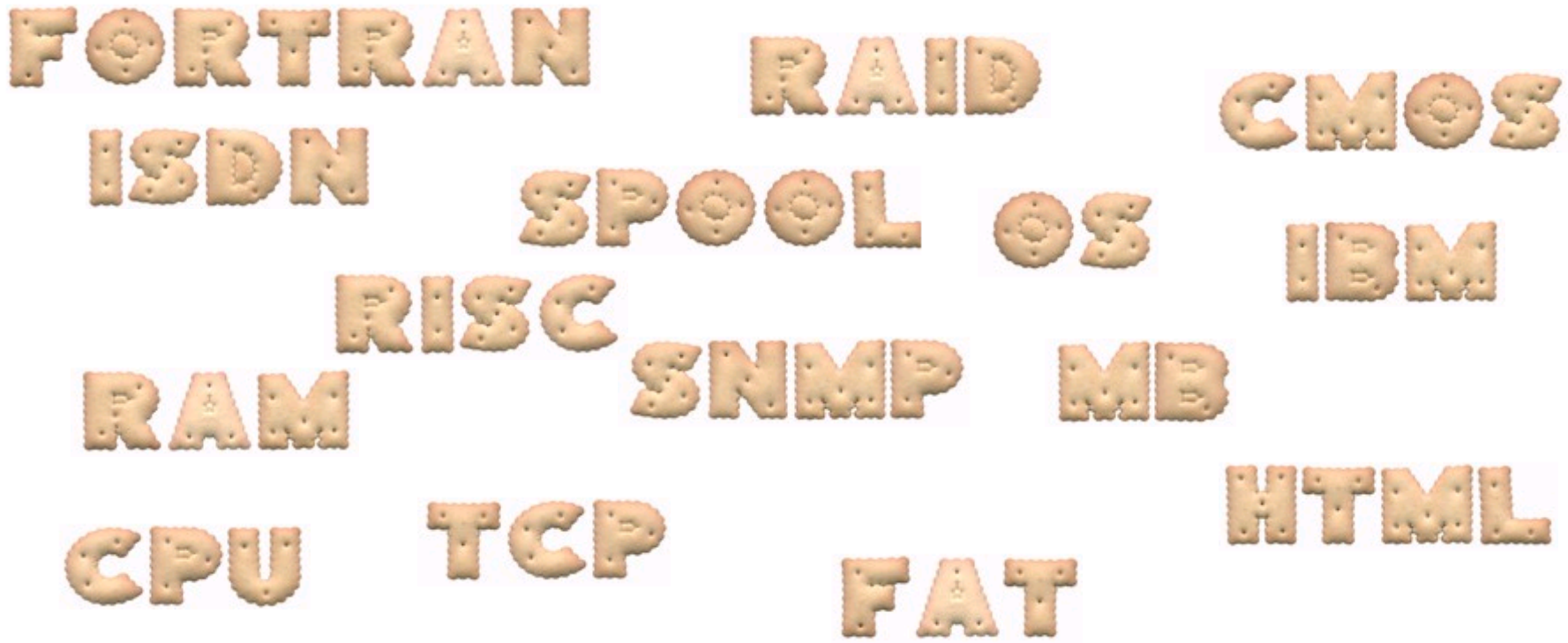
1. Digital representasjon av data
   - Tall, tekst, lyd, bilder, …
   - primitive datatyper (de maskinvaren støtter direkte: boolsk, heltall og flyttall)
   - Alt annet representeres ved hjelp av de primitive typene og forhåndsdefinerte formater/standarder (= KODING!)
2. Maskin-arkitektur og –organisering
   - Oppbyggingen av maskinvaren og gangen i beregningene
3. Operativsystemet
   - Hva, hvorfor og hvordan
4. Internett
   - Oppbygging og protokoller

Oversikt over forelesninger

- Introduksjon til emnet (28. aug)
- 1: Binære data (4. sept)
- 2: Tegnsett, enkoding og media (11. sept)
- 3: Datamaskin arkitektur (18. sept)
- 4: Datamaskin oppbygging (25. sept)
- Eksamen del 1 (dato kommer)
- 5: Operativsystem (2. okt)
- 6: World Wide Web (9. okt)
- 7: Applikasjonslaget (16. okt)
- 8: Transportlaget (23. okt)
- 9: Nettverkslaget (30. okt)
- 10: Linklaget (6. nov)
- 11: Spørretime (26. nov) - Mandag
- Eksamen del 2 (dato kommer)

Forkortelser (bokstavkjeks)

- Det er mange forkortelser i data-verdenen!
- Og ofte benyttes samme ord som betegnelse på vidt forskjellige ting!



Mål: Begreper

- For å beherske et fag må man (dessverre?) lære seg fagspråket
  - Bare slik kan man kommunisere med andre profesjonelle.

- Nivåer av kunnskap (forenklet)
  i. Kunne ordene og forstå hva de betyr (definere)
  ii. Kunne formulere egne påstander (teorier) og finne urimeligheter i andres (forstå teorier).
  iii. Kunne lage ting selv (anvende).
  iv. Kunne vurdere kvaliteten på andres arbeid.

Ex: Et begrep og en ”teknikk”

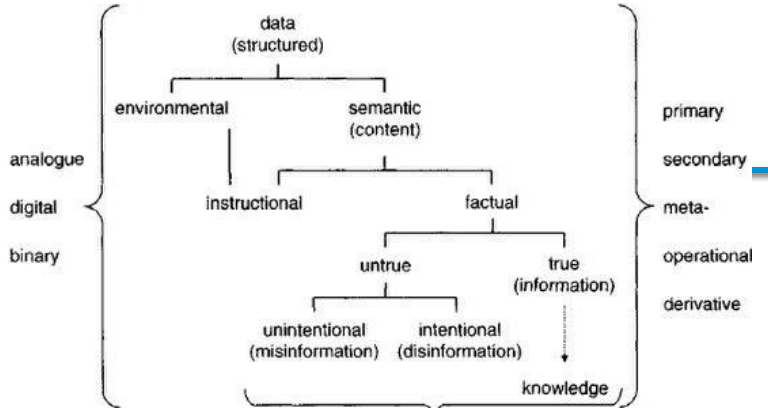
- System
  - Et sett (en mengde) med komponenter som henger sammen på en slik måte at en endring i en komponent medfører en endring i en eller flere andre komponenter
- Sort boks (”Black box”)
  - ”Det vi tar for gitt, ikke bryr oss om hvordan innmaten fungerer i, og kun bryr oss om hva vi kan legge i og få ut av.”
  - Påstand: De fleste brukere ”black-boxer” PCen sin.

DATA, INFORMASJON, SYSTEM

En systemforståelse av et PC-system

- Ytelse er like mye å flytte data og instruksjoner raskt mellom komponenter, som å prosessere dem raskt
- Kjapp CPU på et hovedkort med trege og trange busser gir dermed (likevel) dårlig ytelse, og vice versa.
- Målet er et balansert system

Hva er informasjon?



- Brukes i mange mange forskjellige betydninger – to ytterpunkter er:
  - Informasjonsteoretisk (Shannon, 1948): Entropi (egen-informasjon) H = k*lg(N), N = antall tegn i tegnsettet, k antall tegn i meldingen.
  - Folkelig: data forstått i en sammenheng, ”meningsfulle data”

Hva vi kommer til snakke om

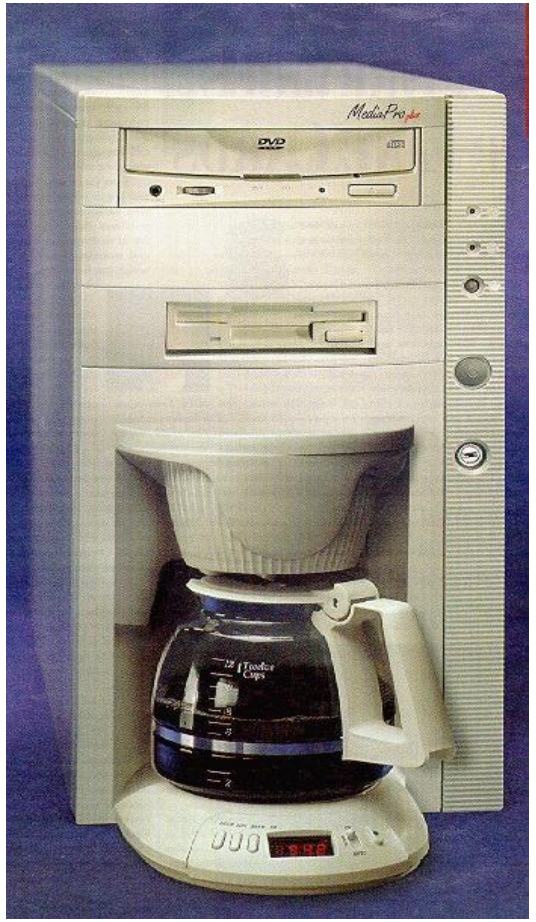
- Generell bruk
  - Datamaskinen brukes innen mange forskjellige områder
- Elektronisk
  - Datamaskinen trenger elektrisitet for å fungere
- Digital
  - Datamaskinen er basert på binære logiske kretser (alt er representert som tall på bunnen)
- Menneske-laget
  - Datamaskinen er konstruert av mennesker for å gi resultater som mennesker er interessert i

Hovedmomenter med datamaskiner

- **Hastighet:** De utfører oppdrag meget fort
- **Pålitelighet:** De gjør ikke feil (bortsett fra når noe er galt med datamaskinen eller et menneske har gjort en tabbe, eller...!)
- **Lagringsevne:** De kan lagre store mengder informasjon (data) over lang tid
- **Pris:** De blir bedre og bedre til lavere pris
- **Størrelse:** De blir stadig mindre

Bruksområder for datamaskiner

- Brukes idag "overalt"
- Barnehager --> forskning
- Banker --> tungindustri
- Internett (email) --> mobiltelefon
- Trafikklys --> kjøleskap
- .....

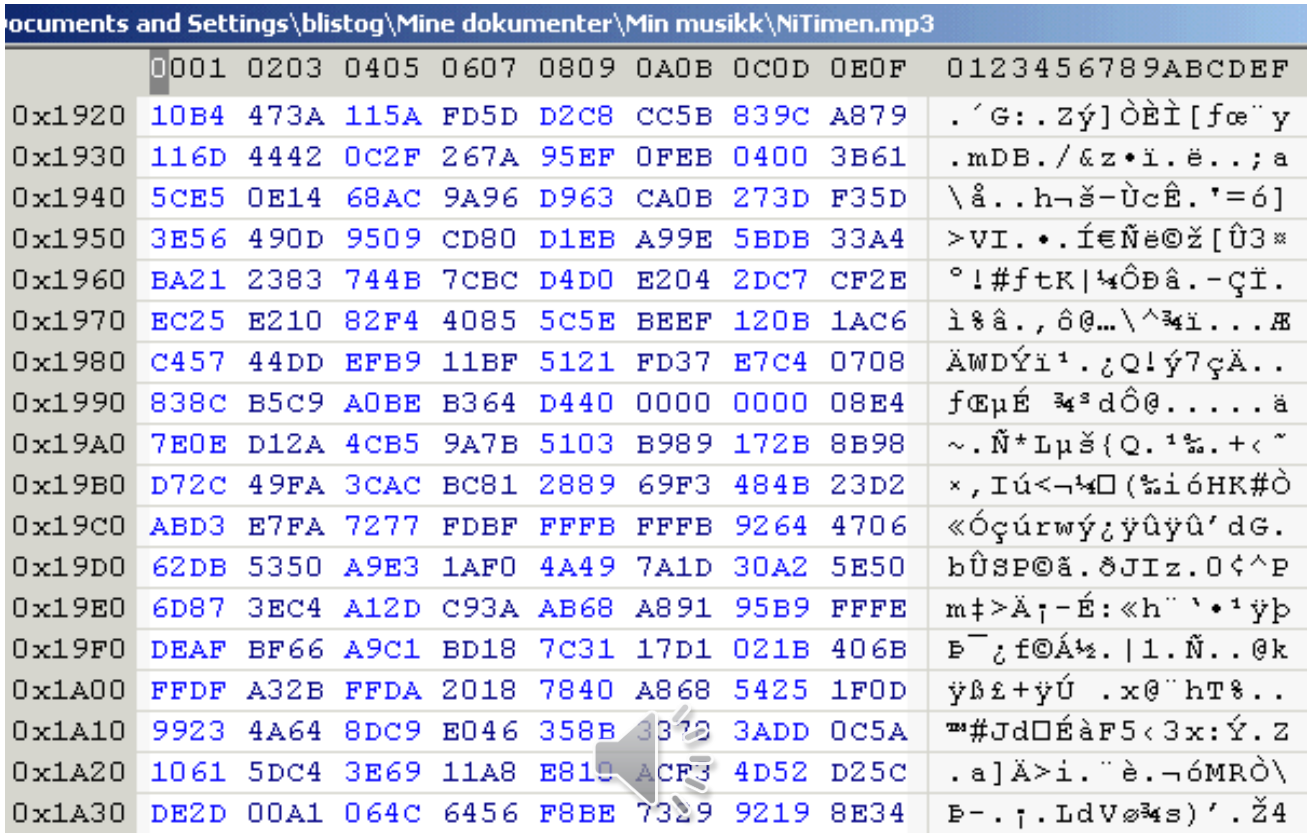


Datamaskin (PC) med tilbehør

DIGITALT?

Digitalisering

- I datamaskinen er alt representert som tall
  - Tall, bokstaver, bilder, lyd, film, instruksjoner…
- F.eks. så vil en MP3-fil være et langt tall som tolkes i forhold til et forhånds-definert format



Tallsystemer - desimaltall

- Desimaltall er basert på antall fingre (tær)
- Dusin, snes, og andre eldre måter å telle på
- Romertall
  - MCMLXXIV = 1974 (I=1, V=5, X=10, L=50, C=100, M=1000)
- 0 oppfunnet ca. 600 e. Kr. (indiske tall)
- Innført i Europa av Leonardo fra Pisa (Fibonacci) ca. 1200
- 1904 = **1***1000 + **9***100 + **0***10 + **4***1
- = **1***10^3 + **9***10^2 + **0***10^1 + **4***10^0

Tallsystemer - binærtall

- Basert på to tilstander (digital; digit = finger eller tå)
  - finger oppe, finger nede
  - av, på (lavt signal, høyt signal)
  - 0, 1
- $25_{10}=11001_2=\mathbf{1}*16 + \mathbf{1}*8 + \mathbf{0}*4 + \mathbf{0}*2 + \mathbf{1}*1$
- $= \mathbf{1}*2^4 + \mathbf{1}*2^3 + \mathbf{0}*2^2 + \mathbf{0}*2^1 + \mathbf{1}*2^0$
- Blir fort svært mange sifre
- Et binært siffer kalles **bit** (binary digit) (b)
- En gruppe på 8 binære sifre kalles **byte (B)**
  - (En byte består av to **nibble**)

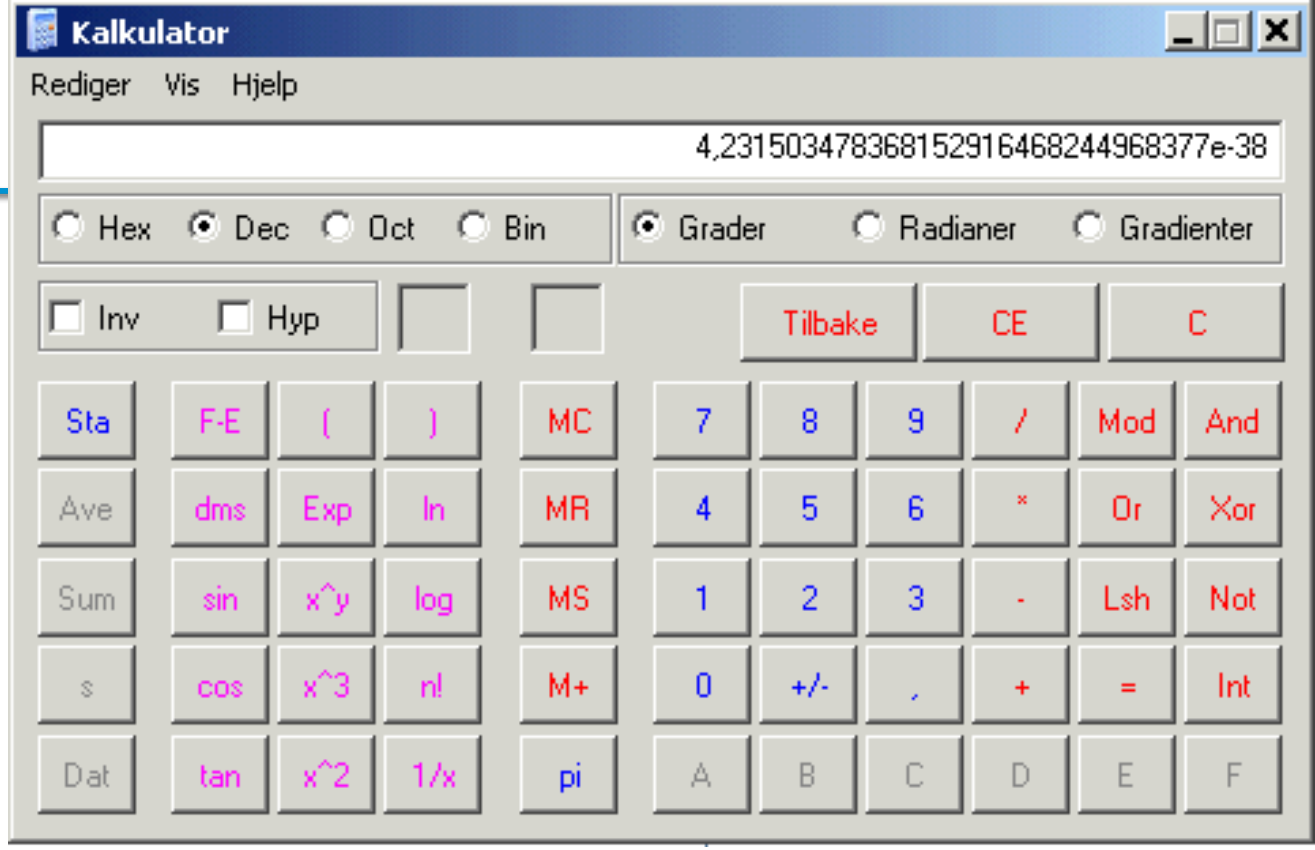
Å telle med binære tall

0	01	10	11
100	101	110	111
1000	1001	1010	1011
1100	1101	1110	1111

Flyttall

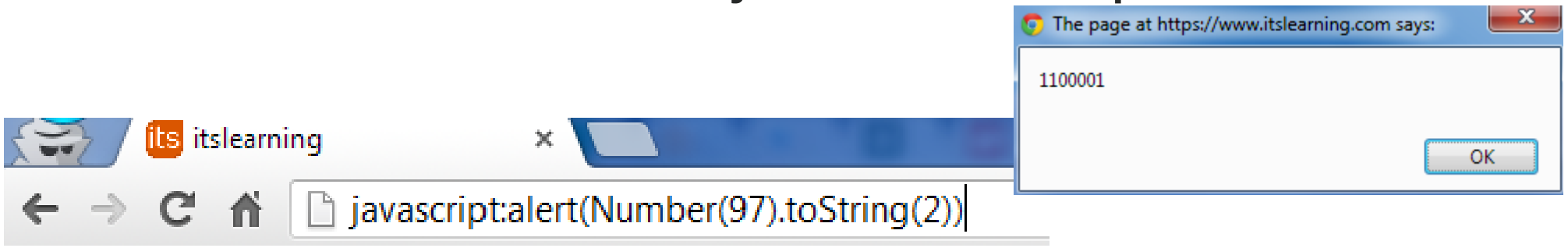
- Windows-kalkulatoren hevder at

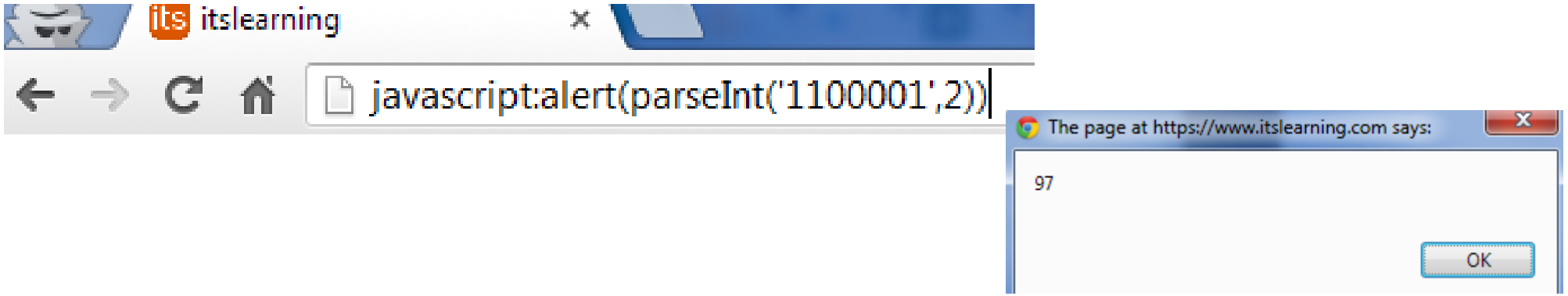
  $(\sqrt{2})^2-2=?$
- Computere arbeider ”alltid” med endelig presisjon!!!
- Flyttall er et kodingsformat!



To «binære tricks»

- Browseren din kan kjøre Javascript…





COMPUTERE

Enheter og størrelser

- Bit (b) - 0 eller 1
- Byte (B) = 8 bit
- Kilo = 10^3 = 1000 ≈ 1024 = 2^{10}
  - 1 km = 1000 m, 1 mm = 1/1000 m
- For enkelhets skyld ”jukser” vi litt !
  - k = 1000, Ki = 1024, (anta at K = 1024 også)
- Kilobyte/KibiByte (KiB) = 2^{10} byte = 1024 byte
- Megabyte/MibiByte (MiB) = 2^{20} byte = 1024 KB = 1048576 byte
- Gigabyte/GibiByte (GiB) = 2^{30} byte = 1024 MB = 1073741824 byte
- Herz (Hz) = hendelser pr. sekund
- MIPS = Mega instruksjoner pr sekund.
- FLOPS = Mega flyttalloperasjoner pr sekund
- kbps = 1000 bit / sekund (bitrate, «båndbredde»)

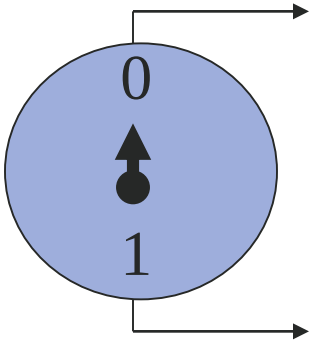
Multiples of bytes v · d · e

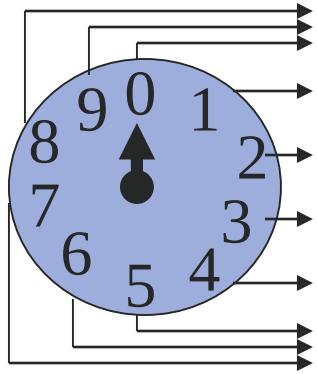
SI decimal prefixes		Binary usage	IEC binary prefixes	
Name (Symbol)	Value		Name (Symbol)	Value
kilobyte (kB)	10^3	2^{10}	kibibyte (KiB)	2^{10}
megabyte (MB)	10^6	2^{20}	mebibyte (MiB)	2^{20}
gigabyte (GB)	10^9	2^{30}	gibibyte (GiB)	2^{30}
terabyte (TB)	10^{12}	2^{40}	tebibyte (TiB)	2^{40}
petabyte (PB)	10^{15}	2^{50}	pebibyte (PiB)	2^{50}
exabyte (EB)	10^{18}	2^{60}	exbibyte (EiB)	2^{60}
zettabyte (ZB)	10^{21}	2^{70}	zebibyte (ZiB)	2^{70}
yottabyte (YB)	10^{24}	2^{80}	yobibyte (YiB)	2^{80}

1 EiB tilsvarer en 50.000 år lang video (DVD-kvalitet)!

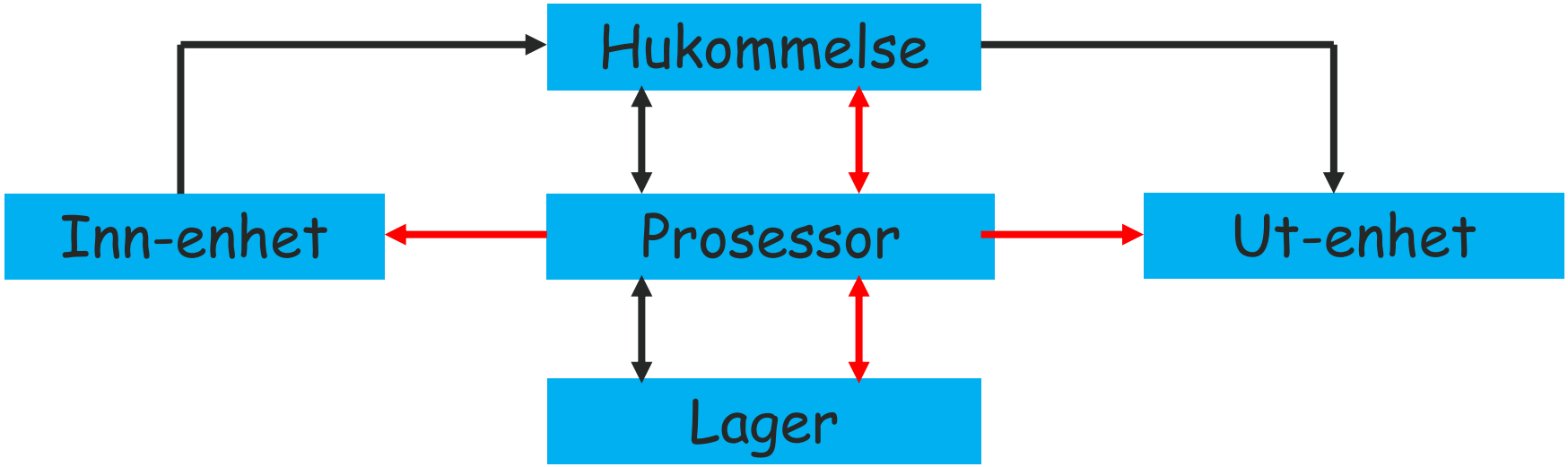
Hvorfor bruker datamaskiner binær logikk?

- Det binære systemet er enkelt og pålitelig
- Enkle kretser gir billige kretser
- Større kompleksitet i basis byggesteiner øker sannsynligheten for feil
- Det er det (teoretisk sett) mest effektive symbolsystemet som finnes!





Von Neumann arkitekturen

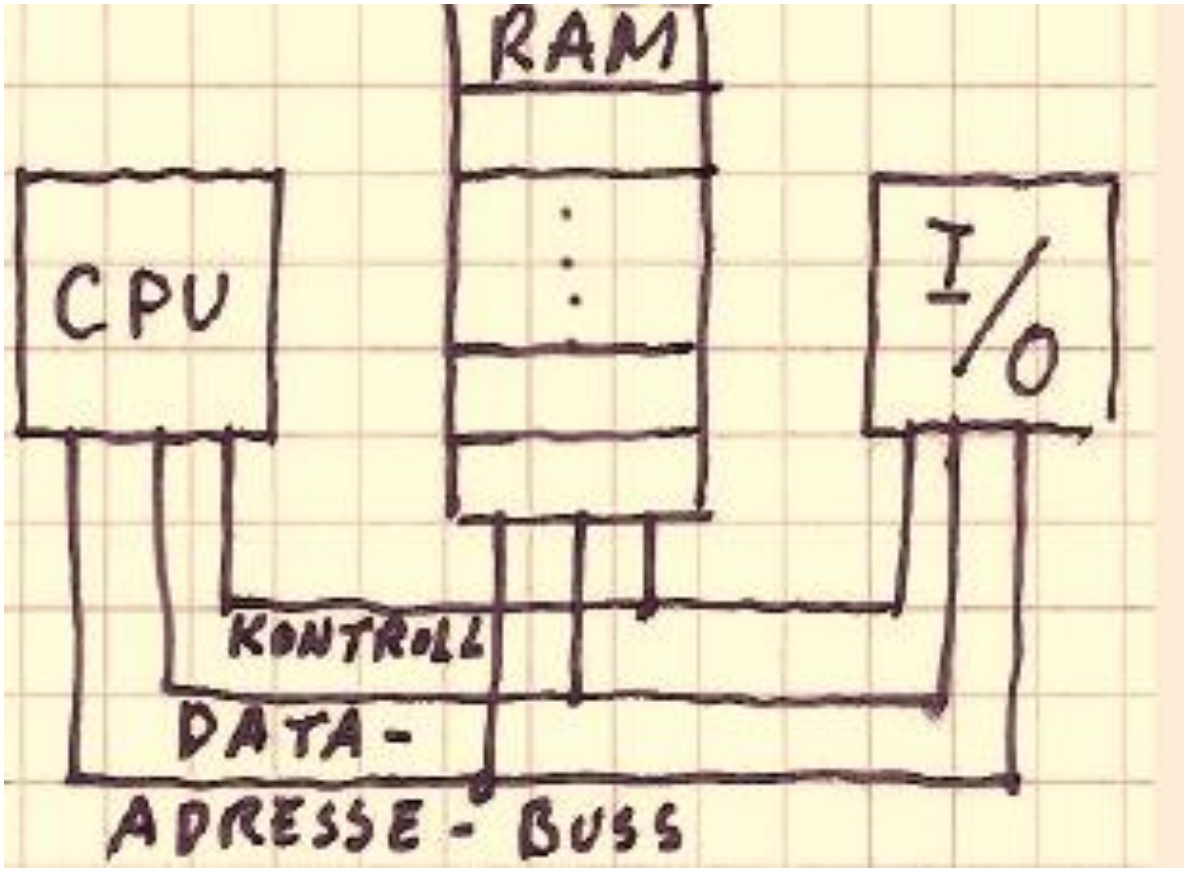


Informasjon

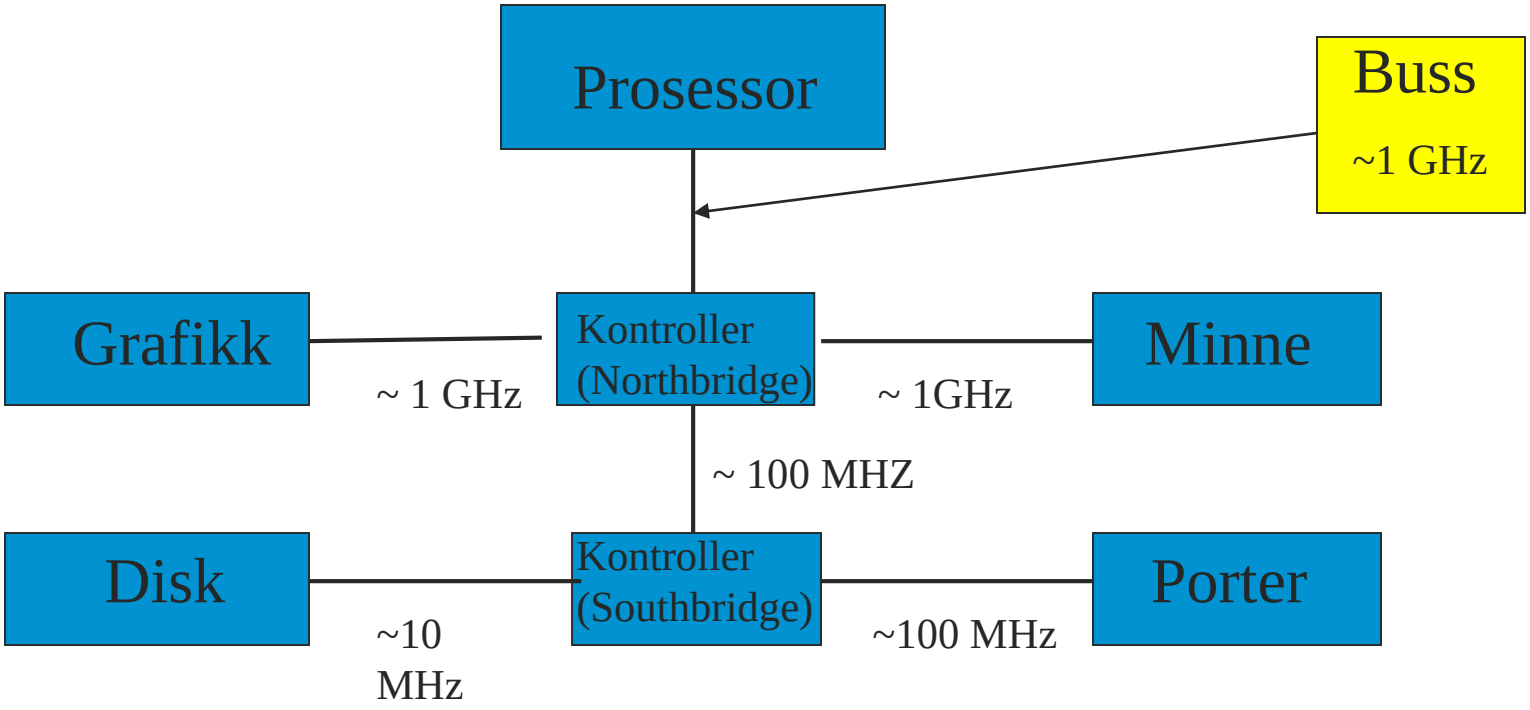
Kontroll

Von Neumann (utvidet)

- Vi legger inn hovedtypene busser også
- Kun en prinsippskisse, men den hjelper deg å tenke klart



PCens hovedbestanddeler (eksempel)



Datamaskin - hovedinndeling

- Datasystemet kan tenkes delt i 2 hoved-deler
- Maskinvare (hardware)
  - *Elektroniske* og *elektro-mekaniske* komponenter som får computer-miljøet til å fungere
  - Input/Output, Lagring (HD, RAM), Beregning (CPU, GPU)
- Programvare (software)
  - Applikasjoner som utnytter de muligheter som maskinvaren gir for å utføre spesielle oppgaver
  - Applikasjoner er programmer som er laget i språk som datamaskinen ”forstår”

Funksjonsorientert modell

- Lag 5 Brukerprogramnivå
- Lag 4 Kompilatornivå/interpreter
- Lag 3 Operativsystemnivå
- Lag 2 Instruksjonsnivå
- (Lag 1 Mikroinstruksjonsnivå)
- Lag 0 Digitalt *krets*nivå

- Her fokuserer vi på hva slags oppgaver som løses på ulike nivåer.
- 0-2 er HW
- 3-5 er SW

Maskinvareorientert modell

Navn på nivået	Komponenter	IC tetthet	Informasjons-enheter	Tidsenheter
Portnivå (Gate)	Logiske kretser, Flip-Flops,..	SSI	Bit	10^{-10} - 10^{-8} Sek
Registernivå	Registre, sekvensielle og kombinatoriske kretser	MSI	Ord (Words)	10^{-9} - 10^{-6} Sek
Prosessornivå	CPU, Minne, busser, I/O	LSI/VLSI	Grupper av ord	10^{-6} - 10^{3} Sek

Applikasjoner

- Tekstbehandling (Word, Notepad, ...)
- Regneark (Excel,....)
- Database-systemer (Oracle, Access, ...)
- Grafikk (PhotoEditor, ...)
- Nettleser (Chrome, Internet Explorer, ...)
- Elektronisk post (Outlook, ...)
- Spesial-programmer
  - Virus-sjekker, multi-media, programmerings-verktøy, ...

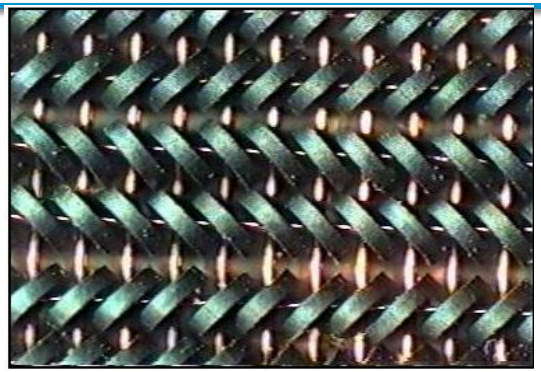
CPU - Central Processing Unit

- **Prosessoren**
- Den "utførende" del av datamaskinen
- Kalles ofte datamaskinens «hjerne», men kan like gjerne oppfattes som en hovedmotor («regnemølle»).
- Består av kretser og elektronikk som kan utføre oppdrag (instruksjoner)
- Instruksjonene kan bearbeide data
- Begrepspar: instruksjon <-> data

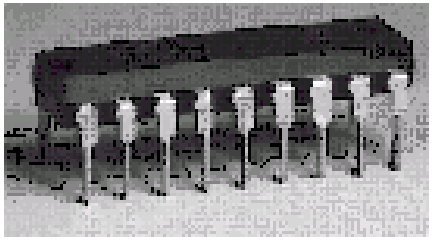
Prosessorens virkemåte

- **Instruksjoner til prosessoren = programmering**
- Hentes inn i prosessoren fra minnet (RAM)
- Maskinspråk (IA32, IA64, m.fl.)
- Assemblerspråk (hver kommando tilsvarer en maskininstruksjon: `mov eax, [00AF3B13]`)
- Lavnivåspråk
  - C, C++ (og flere)
  - Kompileres over til maskin-instruksjoner
- Høynivåspråk
  - C#, JAVA (og veldig mange andre)
  - Interpreteres i et miljø (feks en virtual machine)
- Andre ”språk”
  - Excel, SQL, skriptspråk

Minne



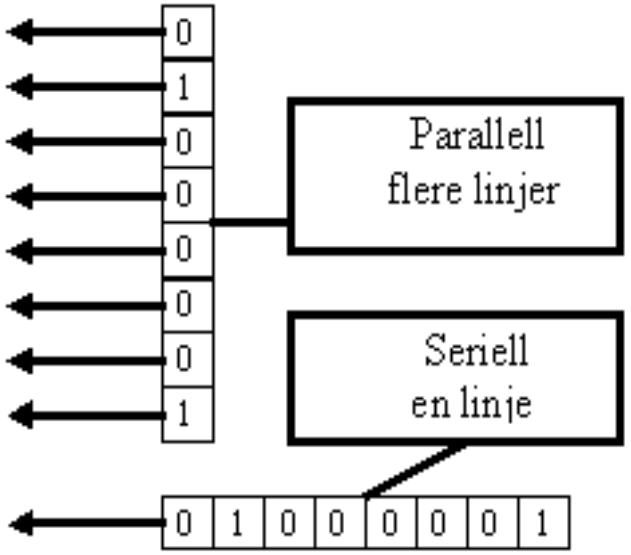
- Lagring av data og programmer
  - Ferritt-kjerner
  - RAM (Random Access Memory)
  - ROM (Read Only Memory), PROM, EPROM, Flash-RAM
  - «Virtuelt minne»



- Raskt mellomlager (cache)
  - Level 1 – internt (on-die, on-chip); ~KB
  - Level 2 – internt/eksternt; ~ MB
  - Level 3 – eksternt ~10 MB

Transportsystemet

- ”Kabler” med parallelle ”ledninger”
- Bredde * Hastighet = Båndbredde/bitrate
  - 64 bit (8 Byte) * 133 MHz = 1064 MiBps



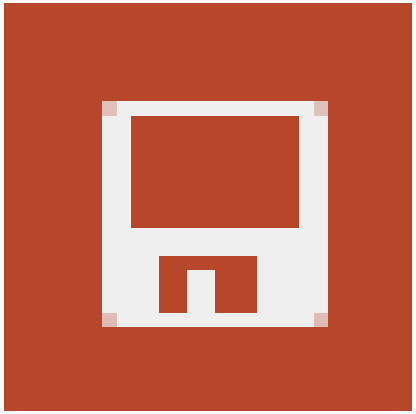
Masselager

- Brukes for permanent lagring av store datamengder
  - Hullkort
  - Hullbånd
  - Magnetbånd (spolebånd, kassett)
  - Diskett
  - Disk
  - USB stick
  - CD, DVD

- Overføring til prosessoren via en kontroller
  - IDE/EIDE, SATA, SCSI,…

Diskusjonsoppgave: Hva er dette?

- Ikonet for lagring (Save) ser slik ut:



- Hva er dette bilde av, og hvorfor betyr det ikonet å lagre filen man har endret?

Periferiutstyr

- Skjerm
- Tastatur
- Mus
- Skriver
- Modem
- Annet utstyr
  - Høyttaler, mikrofon, joystick, plotter, scanner.....

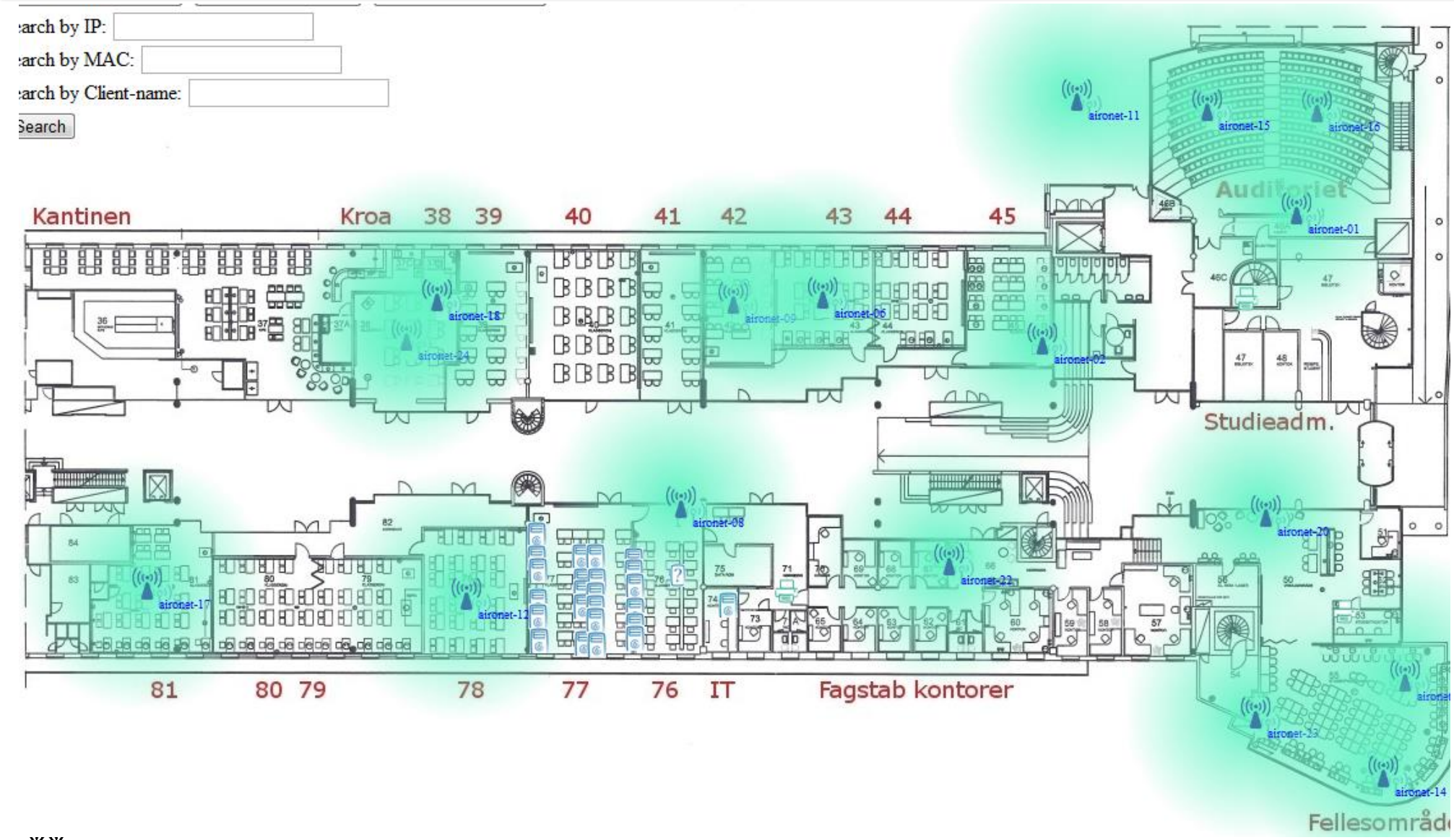
Operativsystemet

- Kjernen er det eneste som har full tilgang til Hardware!!!
- Gir et brukergrensesnitt
- Gir muligheter for sikkerhet og pålitelighet
- Kjører applikasjoner (tilbyr et API)
- Administrerer ressursene
  - Prosessor, hukommelse, eksterne lager, I/O-enheter
- Håndterer nettverk
- Ikke alle anvendelser av datamaskiner trenger et OS

Datamaskiner i nettverk

- Datamaskiner kan koples sammen i nettverk
  - Samme bygning = LAN (Local Area Network)
  - Hele verden = Internett (nettverk av nettverk)
- Kan overføre data mellom maskinene - protokoller
  - Tekst, bilder, musikk, mail, ...
- Strekker egne kabler (LAN), benytter generelt tilgjengelige linjer(telefon, kabel-tv, ...) eller bruker trådløse nett
- WWW (World Wide Web) er den mest brukte tjenesten på Internett = Filoverføring ihht HTTP-protokollen

WLAN skisse (fra gamle skolebygget)



LITT HISTORIE

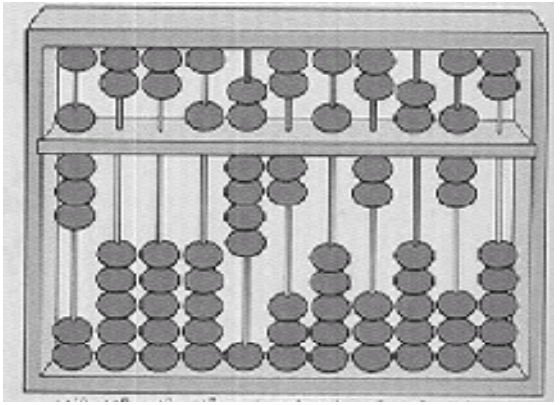
Historie

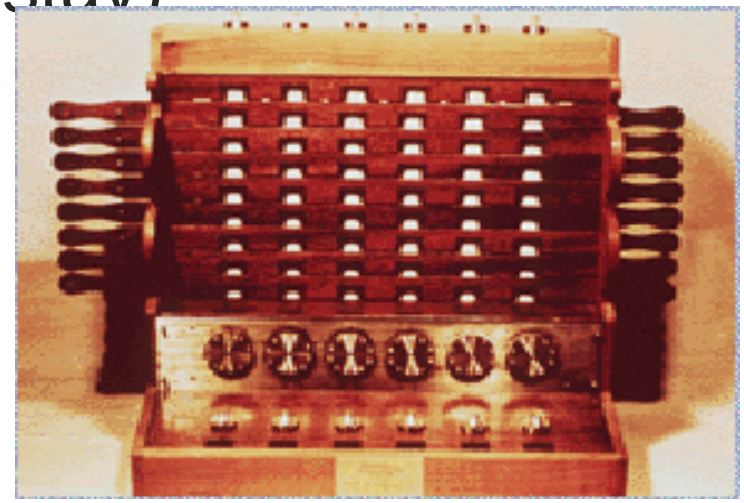
- Den moderne datamaskinen kan ses på som en løsning på tre (historisk sett) forskjellige problemer:

1) Beregningsproblemet
   – hvordan utføre kompliserte beregninger raskt og pålitelig
2) Massedataproblemet
   – Hvordan lagre og behandle store mengder data
3) Reguleringsproblemet
   – Hvordan styre og automatiser industrielle o.a. prosesser

Regnemaskiner

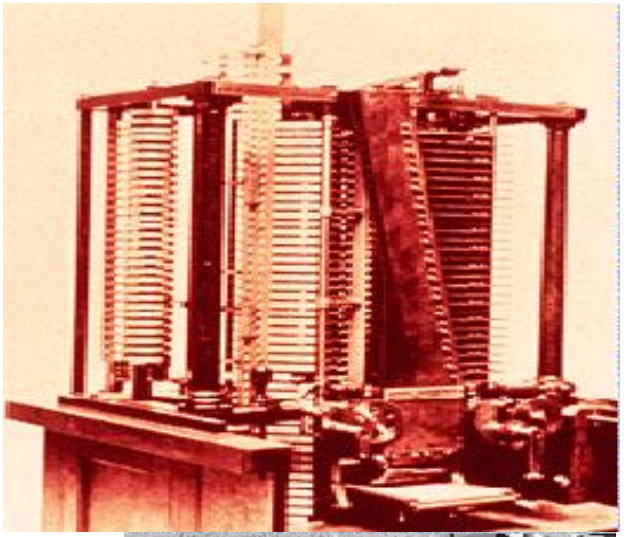
- Abacus (ca -3500)
- ”Tabeller”
  - John Napier, 1600 (staver)
  - Willian Oughtree, 1622 (regnestav)
- ”Tannhjul-maskiner”
  - Pascal, 1642
  - Leibniz, 1694

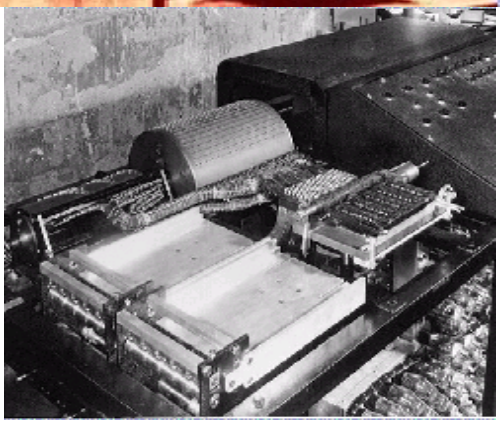




Regnemaskiner (2)

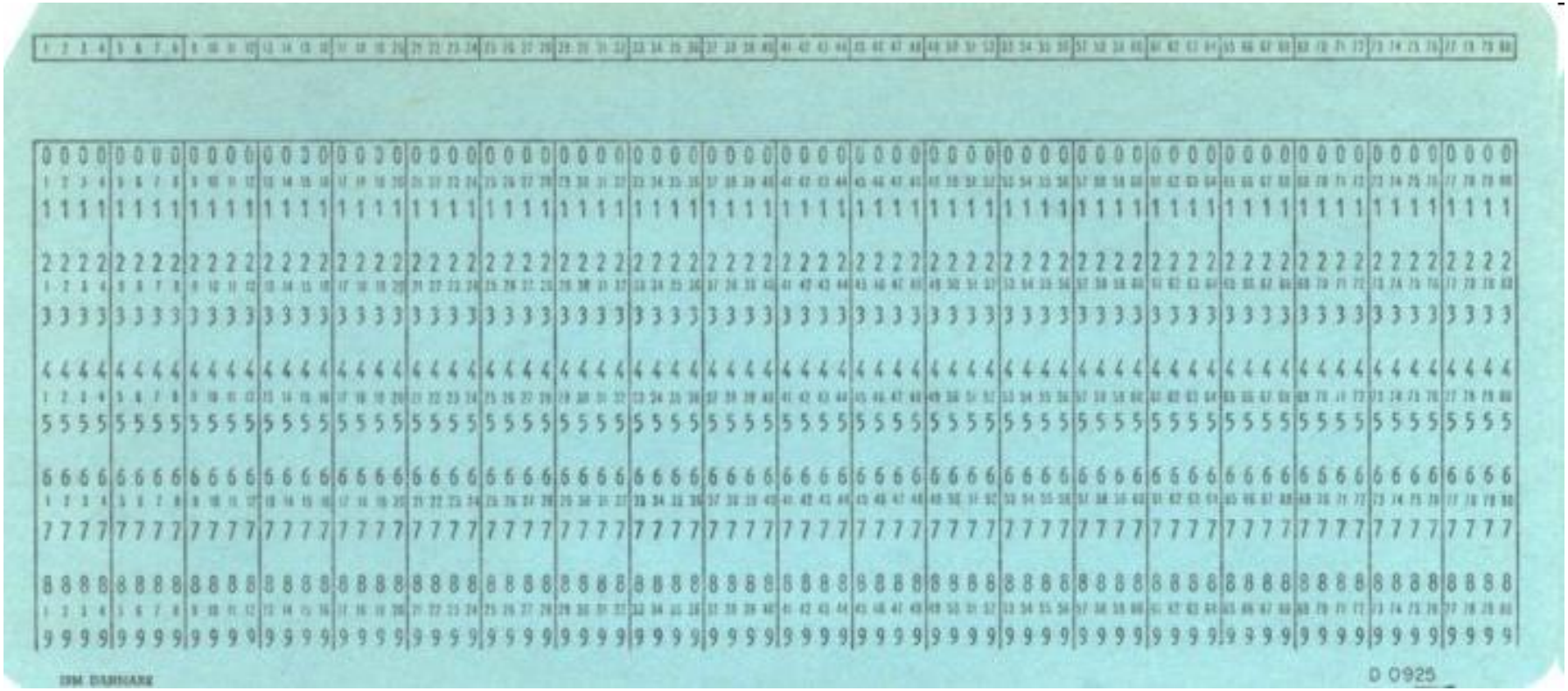
- Mekaniske
  - Charles Babbage (1791 - 1871)
  - Difference Engine - 1822
  - Analytical Engine - 1833
  - Augusta Ada King, countess of Lovelace
- Elektromekaniske
  - Zuse, 1936-
  - Atanasoff, 1940-





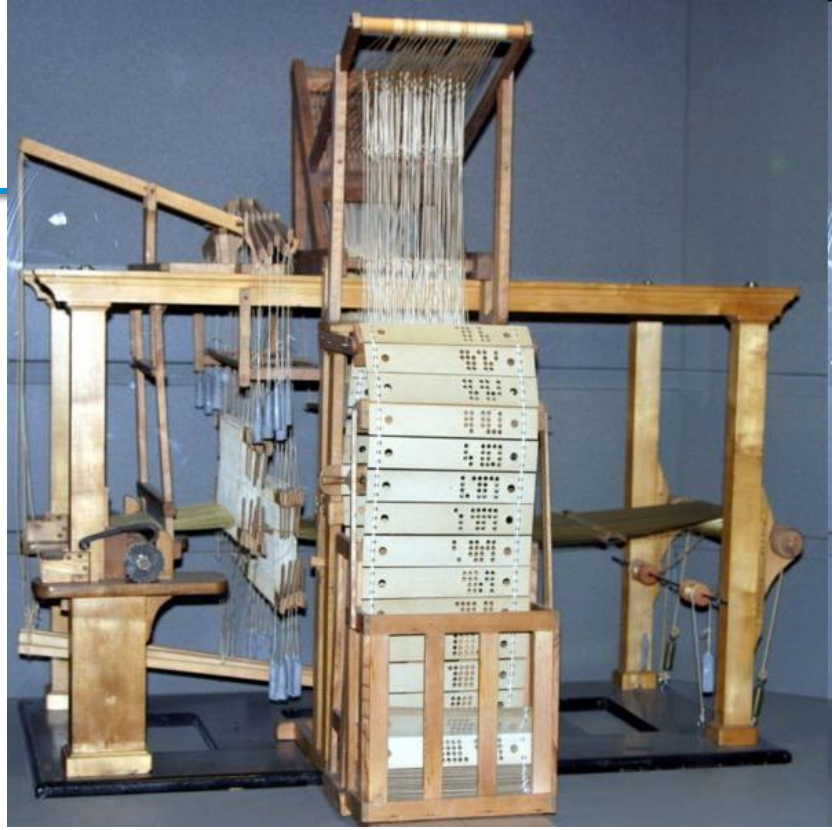
Massedata

- Herman Hollerith (hullkort) – 1890
  - Folketelling
  - Forløper til IBM



Regulering

- Jacqard-veven (ca. 1810)
  - hullkort/automatisering
- Maxwell, ca. 1880
  - teori/diff. Lign for maskinstabilitet
- Norbert Wiener, 1946
  - kybernetikk (reguleringsteknikk)
- Mikrokontrollere overalt!
  - 1971 og utover



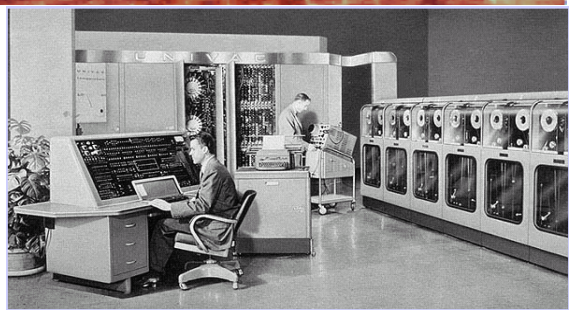
Typisk historisk fremstilling

- Inndeling i generasjoner/epoker basert på:
  - Ny hardware
  - Nye anvendelser

- «Koevolusjon av maskin- og myk-vare»

1. Generasjon (1940-1950)

- Elektromekaniske reléer og radiorør (vacuum tube)
- Conrad Zuse
  - Z1- 1936; Z4 – 1945
  - Plankalkül
- Alan Turing
  - Universell Turingmaskin - 1937
  - Colossus - 1943
- Howard H. Aiken
  - (Mark 1) - 1944
- John Presper Eckert & John W. Mauchly
  - ENIAC - 1944
- John von Neumann
  - EDVAC - 1945
  - UNIVAC - 1951



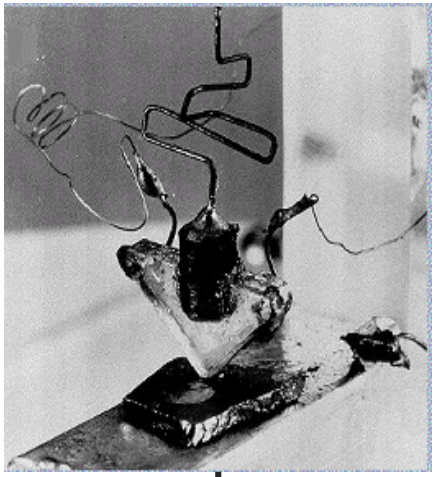
Sitat: Thomas Watson 1943

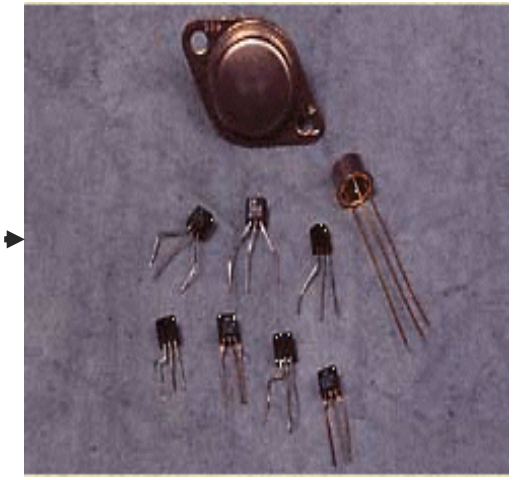
- Thomas Watson, president IBM, 1943:

“I think there is a world market for maybe five computers”

2. Generasjon (1950 - 1964)

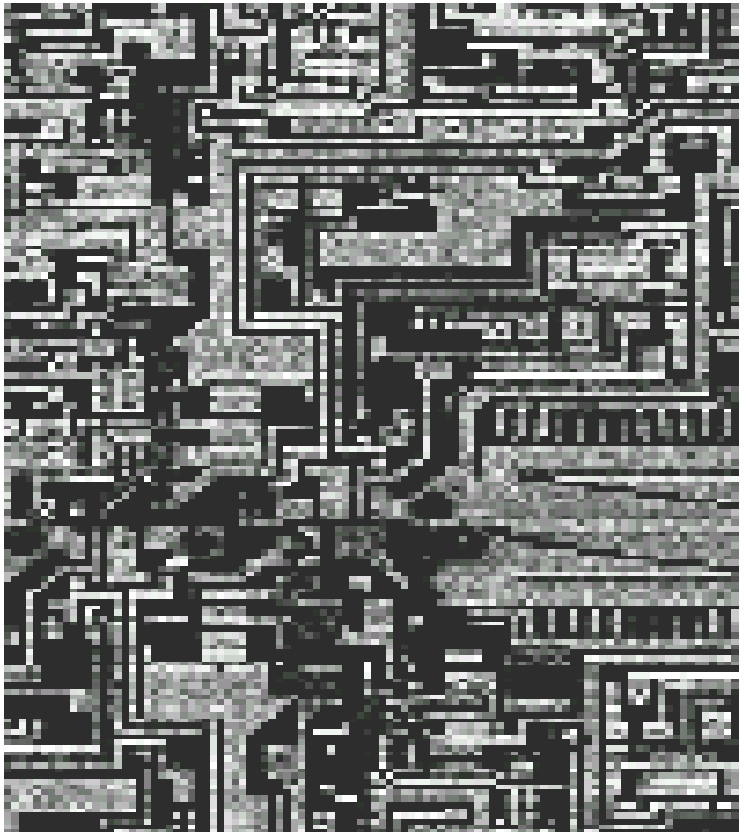
- Transistor – 1947
- Den første kompilatoren (A-0, Grace Hopper) – 1951
- IBM 701 - 1953
- IBM og ”de syv dvergene”: Sperry-Rand, Burroughs, Control Data, Honeywell…
- Fortran - 1957
- Cobol, Lisp, Algol,…
- IBM 1401 - 1960. Datamaskinens ”T-Ford”





3. Generasjon (1964 - 1971)

- Integrerert krets - 1958
- Basic (interpretert språk!) – 1964
- IBM 360 - 1964
- Intel – 1968
- Interaktive terminaler!!!
- ARPAnet - 1969
- PDP, DEC, Data General,..



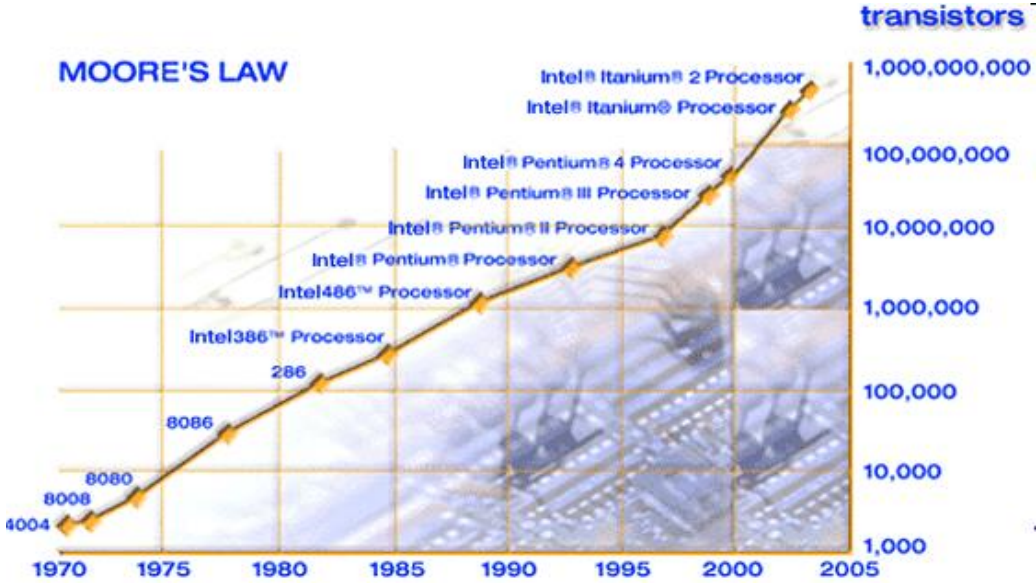
4. Generasjon (1971 -)

- Mikroprosessor
- Intel 4004 - 1971
- IBM 370 - 1971
  - MVS, VM
- UNIX - 1971
- Altair 8800 – 1975
- Apple, Radio Shack, Commodore
- VisiCalc - 1979
- WWW - 1991

(5.) PC (1981-)

- IBM PC, DOS - 1981
- Apple Macintosh - 1984
- Windows 1 - 1985; 3.0 - 1991
- Linux 1.0 – 1994
- Java 1.0 – 1995

- Moore's "lov"
  - Prosessor-"ytelsen" dobles (ca) hvert annet år
  - Antall transistorer pr cm2 dobles ca hvert 2. år



Prosessor - CPU

- 1971 - Intel 4004
- 1973 - Intel 8008, 8080 (Altair)
- 1981 - IBM PC, 8086, 8088, 4 MHz
  - 8086: 16 bit databuss, 20 bit adressebuss = 1 MB
  - 8088: 8 bit databuss
- 1982 - 80286, 8 MHz
  - 24 bit adressebuss = 16 MB
- 1985/86 – 80386, 12-40 MHz
  - 1-4 GB virtuelt minne
  - 0,8 µm
  - Multitasking; Real Mode, Protected Mode

Prosessor - CPU

- 1985 - 80386, 12,5 - 30 MHz
  - 32 bit databuss og adressebuss(4 GB)
  - Virtual Real Mode
  - SX (1988); 16 bit ekstern databuss, 24 bit adressebuss
- 1989 - 80486, 25 - 50 MHz(DX2, DX4)
  - Innebygget flyttall ko-prosessor
  - 8 KiB cache
- CISC og RISC

Prosessor - CPU

- 1993 - Pentium, 75 MHz →
  - RISC nærmer seg CISC
  - Dual Independent Bus, L1 og L2 cache, Dynamic Execution
  - MMX (Multimedia = masseregning parallellt.
- 2001
  - Pentium 4 passerer 2 GHz
  - Intel *Itanium* (Merced), 64 bit prosessor → McKinley
  - AMD tar det private 64 bits markedet
- Ikke bare Intel
  - AMD, VIA (Cyrix), +++
  - Apple (MacIntosh), Alpha, SPARC, +++

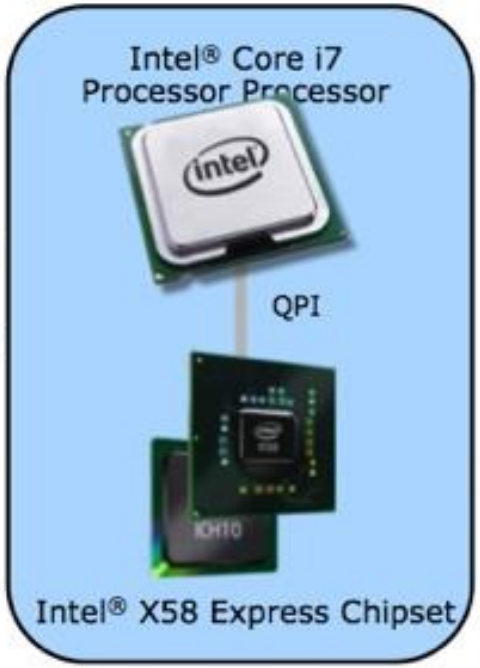
- Pr 2013 er Skjermkort i gang med å overta for supercomputere!

Intel Core i7

Intel® Core™ i7 Processor

Performance/Features:

- 8 processing threads via Intel® Hyper-Threading Technology (Intel® HT)
- 4 cores
- Intel® Turbo Boost Technology operation
- Intel® QuickPath Interconnect (Intel® QPI) to Intel® X58 Express Chipset
- Integrated Memory Controller (IMC) – 3ch DDR3
- 7 more SSE4 instructions
- Overspeed Protection Removed



Socket:

- New LGA1366 Socket

Power:

- 130W TDP
- VRD 11.1

Platform Compatibility:

- Intel® X58 Express Chipset
- ICH10 / ICH10R

Targeted Segment:

- Extreme and performance demanding users
- Ultimate gaming, multimedia creation, compute intensive applications

- Parallellisering
  - 4 kjerner
- Multimedia
- 710 millioner transistorer
- ~ 3GHz
- 45 nm
- L1 & L2 pr kjerner, L3 felles

DISKUTER

- Hva er problemet med denne («hardware-måten») fremstillingen av historien?

SLUTTEMA: PASSORD

Ukens sikkerhetstema: Passord

- En vanlig måte å måle kvaliteten på et passord er bitstyrke
- Uttrykker hvor mange forsøk en tilfeldig angriper ***maximalt*** trenger for å gjette (”brute force”) passordet.

- Bitstyrke = lg_2(forskjellige tegn mulige i passorde)* antall tegn i passordet.
- F.eks. PIN-kode bruker 10 tegn (0-9) og 4 tegn => bitstyrke = $lg_2(10)*4 = 3{,}32*4 = 13{,}28$
  - NB! Praktisk enhet pga kombinatorisk eksplosjon
  - 2^bitstyrke=antall mulige passord som kan lages

- Anbefalt bitstyrke i våre dager er ca 80, mao ca tolv bokstaver og tegn!
  - I tillegg bør man selvsagt unngå alt som er knyttet til din egen person, alle vanlige ord (de som finnes i ordbøker) mm

•“Sikre passord”

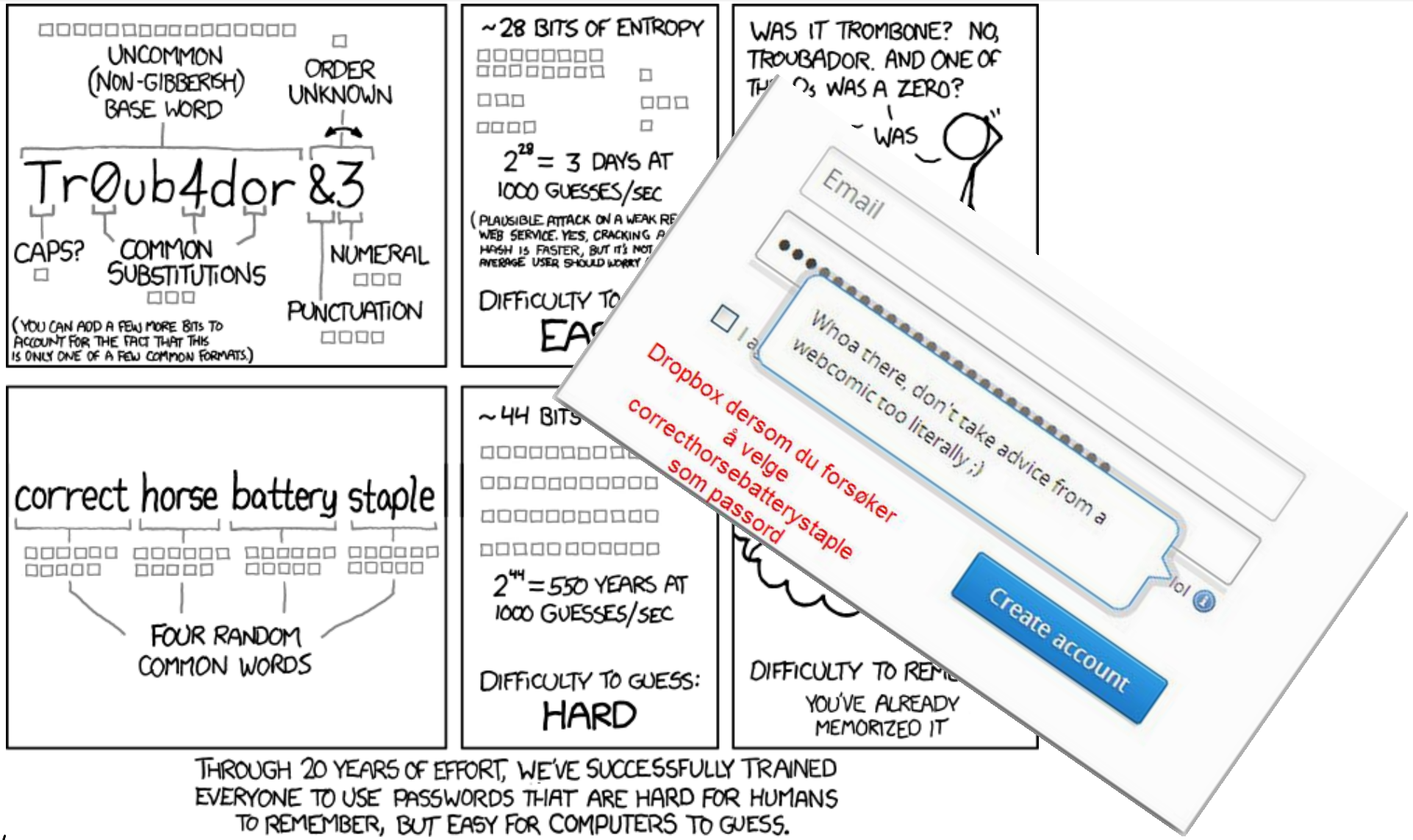
Bør inneholde tegn fra minst tre av gruppene under:

Group	Example
Lowercase letters	a, b, c, ...
Uppercase letters	A, B, C, ...
Numerals	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Non-alphanumeric (symbols)	( ) ` ~ ! @ # $ % ^ & * - + = | \ { } [ ] : ; " ' < > , . ? /
Unicode characters	€, Γ, ƒ, and λ

Passfrase er enda bedre: "I re@lly want to buy 11 Dogs!“

Eller lær deg et fint dikt utenatt!!!

XKCD: kommenterer



Kombinatorisk eksplosjon?

- I sjakk har hvit 20 forskjellige mulige åpningstrekk, svart like mange mulige svar.
  - Etter ett trekk finnes det da 20x20 = 400 mulige stillinger/brett
  - Hvor mange mulige sjakkspill-stillinger finnes det?
  - Ingen vet eksakt. Svaret kalles *Shannon-tallet* og det er av størrelsesorden 10000000000000000000000000000000000000000000000000 = 10^{50}
    - Disse kan anslagsvis inngå i 10^{123} ulike lovlige spill
    - Antall partikler i universet er anslått til 10^{80}
- Det finnes dermed ingen mulighet for at en computer kan teste ut alle mulige spill («traversere alle spilltrær»)
- Se f.eks. Claude Shannon (1950). "Programming a Computer for Playing Chess". *Philosophical Magazine* **41** (314)

- Det finnes mange slike problemer som er oversiktlige, men uløselige innen rimelig tid!

AVSLUTTNING

Oppsummering: «Teori»

- System
- Digitalisering
- Informasjon og data
- Data og Instruksjoner
- Hardware og software
- Abstraksjonsnivåer

Oppsummering: Historikk

- Tre hovedområder: beregning, massedata, regulering
- Moores lov
- Maskinvare og programvare har drevet hverandre frem (”koevolusjon”)

- Wirths lov: ”Dess raskere maskinvaren blir, dess tregre går programvaren” ☺…

Dagens Øving

- *Spørsmålsark, fordel å diskutere i grupper*
- *Skriv ned svar (dere trenger det om 3 mnd…)*
- *Primært repetisjon av forelesningen for å sikre god innlæring*
- *Søk forskjellige kilder på nettet for å lære temaene mer i dybden! Du må i dybden for å FORSTÅ temaet, ikke bare pugge foilene…*
- JA, det dukker opp «nytt stoff» i øvingene!!
- JA, dere kan få (direkte og indirekte) spørsmål fra øvingene på eksamen

Neste gang

- Data-representasjon
- Tall-systemer /Boolsk algebra/Koding/dekoding

- Hjemmelekse
  8-4-2-1, 8.4.2.1., 8 4 2 1 ->pugg!!!!!!!!

  128-64-32-16- 8- 4- 2- 1
  1 0 1 0 0 1 1 0 =????

- Ta med penn og papir!!!!
  - Det meste av det vi gjør neste gang er mye enklere å få til på et ark, enn å forsøke å gjøre på «skjermen»