

## ***IKT historikk***

Geir Ove Rosvold

1. februar 2011

Opphavsrett: Forfatter og Stiftelsen TISIP

L restoffet er utviklet for fagene LO216D Datamaskinteknikk  
og LO246D Datateknikk

## **IKT historikk**

*Resym : I denne leksjonen presenteres en del bakgrunnsmateriale som er nyttig for resten av kurset. Vi ser datamaskinhistorie og trekker linjer b de fremover og bakover.*

### **Innhold**

1.1.	DATAMASKINHISTORIE .....	2
1.1.1.	von Neumann arkitektur .....	4
1.2.	GENERASJONER AV DATAMASKINER.....	5
1.3.	MASKINKLASSER .....	6
1.4.	DATAMASKINFAMILIER .....	6
1.4.1.	Intel og IBM PC .....	7
1.4.2.	Mikromaskiner .....	8
1.5.	UTVIKLINGSTREKK .....	8
1.5.1.	Informasjonsteknologi.....	9
1.5.2.	Moore's lov .....	9
1.6.	KOMMUNIKASJONSTEKNOLOGI.....	9
1.6.1.	Internett.....	10
1.7.	INFORMASJONS- OG KOMMUNIKASJONSTEKNOLOGI (IKT) .....	10

## 1.1. Datamaskinhistorie

På mange måter gjennomsyrrer datateknikken hverdagslivet i det moderne samfunn. Til tross for dette har den moderne datamaskinen en relativt kort historie. Det var først etter den andre verdenskrig at vi kan begynne å snakke om noe som ligner de moderne datamaskinene. Men lenge før den tid har mennesket laget redskap som utførte regneoperasjoner.

Hvis vi ser bort fra kulerammen, som ikke kan utføre regneoperasjoner selv, er en av de eldste regnemaskinene en mekanisk adderer som Pascal utviklet i 1642. Ved at man dreide på tannhjul og akslinger kunne denne maskinen utføre addisjoner. Den kunne representere negative tall - dermed kunne man også utføre subtraksjoner - og den overførte automatisk mente. I 1671 forbedret Leibniz denne addereren slik at den også greide multiplikasjon og divisjon. Mekaniske regnemaskiner som var basert på de samme prinsipper ble utviklet og forbedret i de påfølgende århundrer, og fra midten av 1800-tallet utviklet det seg en stor industri med mange aktører og stor konkurranse. Små bærbare mekaniske regnemaskiner var et vanlig syn langt inn i 70-tallet ved norske regnskapskontor og høgskoler. På NTH ble slike brukt i kurset *Numerisk matematikk* etter 1970.

På 1700/1800-tallet ble det utviklet teknologi som tilsynelatende ikke har noe med regnemaskiner å gjøre men som likevel fikk betydning for datamaskinen, nemlig hullkortet. En av industrigrenene som drev den industrielle revolusjon fremover var tekstilindustrien, og i denne perioden ble det utviklet stadig mer automatiserte vevstoler. I 1801 laget Jacquard en vevstol hvor all kraft ble tilført mekanisk og hvor vevingen ble 100 % styrt med hullkort. Mønstrer som ble vevd var dermed bestemt av et "program" som fantes i form av en bunke hullkort, og hvor hullene - eller evt fravær av hull - på forskjellige plasser på kortet bestemte bevegelsene til bestemte deler av vevstolen slik at det ble vevd et forhåndsbestemt mønster. Det mest berømte programmet besto av 24.000 hullkort og vevde et portrett av Jacquard. På digitale datamaskiner var hullkort vanlig brukt til å angi hvilke instruksjoner et program besto av helt opp til ca 1980.

Hvis vi ser på utviklingen av vitenskapelige regnemaskiner (til forskjell fra de tidligere nevnte små mekaniske regnemaskiner), så er det naturlig å stoppe ved engelskmannen Babbage som i 1822 utviklet en såkalt differensialmaskin. Dette var en mekanisk regnemaskin som først og fremst skulle være til hjelp under fremstilling av matematiske og nautiske tabeller hvor de samme beregningene skulle utføres for mange tall. Maskinen kunne utføre beregningene som skulle inn i en tabell og overføre resultatene til en trykkeriplate. Maskinen kunne drives av en passende motor, gjerne en dampmaskin.

I 1834 utviklet den samme Babbage noe som kaltes en analytisk maskin. Det var også en rent mekanisk regnemaskin. Den hadde en mekanisk "kvern" som kunne utføre de fire grunnleggende regneoperasjonene, et mekanisk minne som kunne lagre en del tall, og i tillegg kunne maskinen presentere resultatene på en lett lesbar måte. Den inneholdt også en automatisk sekvensieringsmekanikk som gjorde at instruksjonsrekkefølgen kunne bestemmes mens man kjørte. Dermed kunne man utføre betingede hopp (if-setninger, while-setninger osv - man kunne bl.a. foreta hopp i programmet avhengig av fortegnet til et tall). Etter modell av Jacquards vevstol ble den analytiske maskinen programmert med hjelp av hullkort. Maskinens arkitektur var bemerkelsesverdig lik dagens datamaskiner på mange måter. Den hadde en prosesserende enhet, et lager, i/o-utstyr, og kunne programmeres med et programmeringsspråk - senere i kurset skal vi se at alt dette også er typiske trekk med moderne maskiner. En av de mest kjente programmerere som laget program for den analytiske maskinen var Ada Lovelace. Det moderne programmeringsspråket *Ada* er oppkalt etter henne.

Etter Babbage oppsto det to ulike retninger innen vitenskapelige regnemaskiner. Den ene retningen utviklet analoge maskiner. Dette er maskiner som viser glidende overganger mellom to verdier, og dermed ikke forholder seg til et endelig antall ulike (distinkte) verdier. Slike maskiner kunne for eksempel produsere grafer som viste tidevann, planetbaner og mye annet. Mange vitenskapelige milj er foretrakk analoge maskiner. Blant annet kunne de analoge maskinene l se differensial ligninger p  en hurtig og grei m te, og differensial ligninger er viktige innenfor mange vitenskapelige grener. Store analoge maskiner ble brukt til langt inn i 1960-tallet. Blant annet ble det bygget en stor differensial analysator p  Astrofysisk institutt ved Universitetet i Oslo i f rste halvdel av 50-tallet.

Den andre retningen behandlet tall; f rst representert med tannhjul i bestemte stillinger og senere elektromekaniske releer. Siden maskinene forholdt seg til tall, kaltes de ofte siffermaskiner. Det var denne retningen som senere utviklet seg til dagens moderne digitale maskiner. Digital betyr nettopp at man representerer et endelig antall ulike verdier.

P  30- og 40-tallet var elektromekanikk et godt utviklet industriomr de som serieproduserte p litelige releer og radiatorer. Under andre verdenskrig ble det stort behov for regnekapasitet, spesielt innen kryptografi, ballistikk og atomteknologi. Kombinasjonen av stort behov og gode tekniske muligheter, ga utviklingen av siffermaskiner et kraftig puff. Man konsentrerte seg om elektro-mekaniske maskiner som var bygget opp av mekaniske deler og elektro-magnetiske rel er.

Det kan v re naturlig   nevne tre pionerprosjekter fra denne tiden:

1. I 1938 utviklet tyskerne Konrad Zuse og Helmut Schreyer regnemaskinen *Z1*. Dette er den f rste automatiske og programmerbare siffermaskinen. Den baserte seg p  mekaniske releer, og hadde et minne som kunne lagre 16 flyttall   24 bits. Prinsippene for programmeringsspr ket som Zuse utviklet til maskinen dannet grunnlaget for mange moderne programmeringsspr k.
2. Amerikanerne Vincent Atanasoff og Clifford Berry har f tt  ren av   ha utviklet den f rste elektroniske digitale regnemaskinen. Denne maskinen var bygget opp av radiatorer. Selve regneenheten besto av 300 radiatorer, og kunne bare addere og subtrahere. I historisk perspektiv er minneteknologien de brukte viktigere enn regneenheten. De lagret nemlig bin re verdier i s kalte kondensatorer. En kondensator er en elektronisk komponent som husker en spenning en kort tid. Etter hvert mister den spenningen, og m  lades opp igjen. Atanasoff og Berry plasserte kondensatorene p  to tromler, og etter hver omdreining ble kondensatorene ladet opp. Senere skal vi se at dagens RAM-minne ogs  bruker kondensatorer til   lagre bit-verdier.
3. Under krigen overtalte Howard H. Aiken IBM til   investere 1 mill. dollar for   bygge maskinen *Mark 1*. I likhet med Zuse's maskiner var *Mark 1* en elektromagnetiske regnemaskiner som ble styrt med hjelp av elektromagnetiske rel er og inneholdt mange mekaniske deler. *Mark 1* kunne gj re en enkel multiplikasjon p  rundt 6 sekunder, mens en divisjon tok 12 sekunder. Den hadde et lager som kunne lagre 72 tall. Lagringsformatet var 23 desimalers tall som ble lagret i 10-tallsystemet.

Alle de tre maskinene inneholdt betydelige mengder mekanikk – og mekanikk er langsomt sammenlignet med elektronikk. Dersom man kunne bygge maskiner der b de lager og regneenhet var helt elektronisk ville maskinene bli mye hurtigere. Med hjelp av *radiatorer* var det mulig   lage hele regnemaskinen elektronisk.

I 1946 ble den amerikanske regnemaskinen ENIAC ferdigbygd. Dette var den f rste helt *elektroniske regnemaskin*. Den veide 30 tonn, trengte 15.000 kvadratfot gulvplass og inne-

holdt 18.000 rador r. Det ble brukt brytere og pluggen n r man programmerte den. Men den kunne utf re 5.000 addisjoner pr sekund. Ogs  denne brukte imidlertid 10-tallsystemet.

### 1.1.1. von Neumann arkitektur

I en rapport som kom ut i 1945/46 foreslo den amerikanske matematikeren *John von Neumann* i en ny design p  regnemaskiner. I rapporten beskrives en datamaskinarkitektur som best r av f lgende deler:

- Et *arbeidslager* som skulle inneholde b de data og instruksjoner p  *bin r form*
- En *aritmetisk/logisk enhet* som kunne utf re en del matematiske og logiske operasjoner
- En *kontrollenhet* som tolket instruksjonene i minnet og s rget for at de ble utf rt
- *Inn/ut-enheter* som s rger for kommunikasjon mellom bruker og kontrollenhet

Alle moderne datamaskiner bruker denne arkitekturen den dag i dag, og rapporten har derfor hatt en formidabel betydning.

I etterkant h stet von Neumann betydelig kritikk fordi han ikke nevnte at oppbyggingen opprinnelig var forsl tt av to av ENIACs konstrukt rer, nemlig John W. Mauchly og J. Presper Eckert jr. Disse to har dermed ikke f tt rettmessig kreditt for sine bidrag til datamaskinutviklingen. P  den andre siden kopierte disse to betydelige deler av Vincent Atanasoffs konstruksjoner uten   gi han kreditt. Dermed oppsto en situasjon som vi har sett mange ganger under datamaskinens utvikling: Sv rt gode hoder samarbeider om   frembringe store fremskritt, men ender opp i stor uenighet.

S  istedenfor   bygge den f rste maskinen som fulgte de viktige prinsippene som von Neumann beskrev, endte de samme personer opp med rettsaker og indre stridigheter.

Von Neuman startet arbeidet med en maskin han kallte IAS, som fulgte prinsippene som var beskrevet i rapporten. Han var imidlertid mer forsker og akademiker, enn han var ingeni r og konstrukt r. S  IAS tok lang tid   realisere. Derimot var von Neuman en ivrig foredragsholder og l rer. En av mange som deltok p  hans foredrag og kurs var Maurice Wilkes fra Cambridge University. Etterp  dro Wilkes tilbake til England og bidro sterkt til   bygge EDSAC. EDSAC sto ferdig i 1949, og var dermed den f rste von Neuman-maskinen.

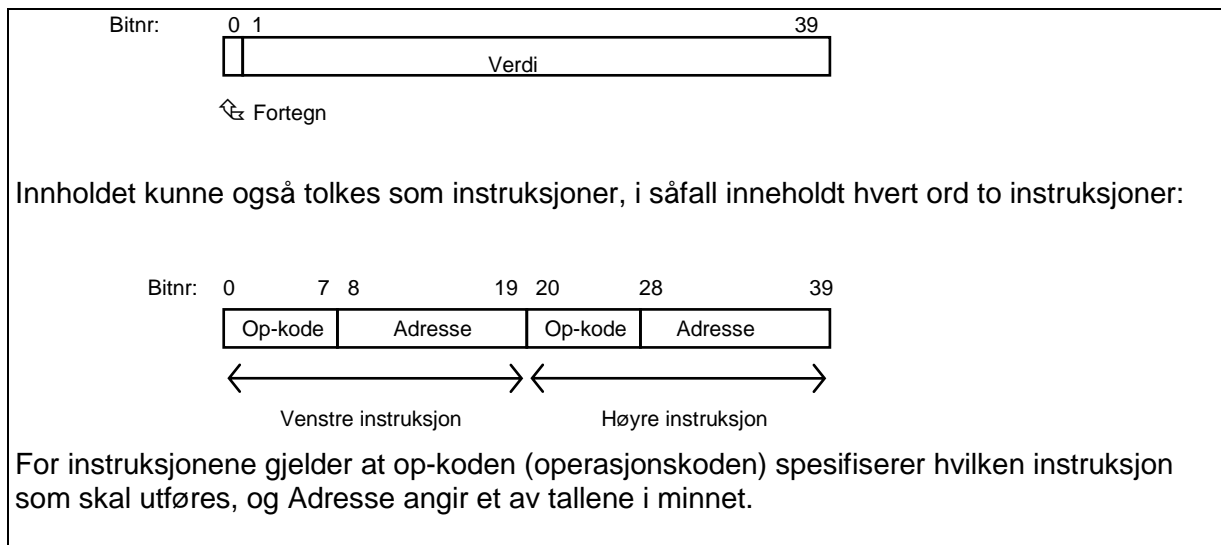
F rst i 1952 var von Neumanns IAS ferdig. Minnet til maskinen besto av 1000 *ord* som hver var 40 bits. Minnet inneholdt bin re koder som enten kunne tolkes som data eller som instruksjoner.

Senere har det blitt vanlig   bruke maskiner med ordbredde lik en potens av to, f.eks 8-bits, 16-bits, 32-bits eller 64-bits-maskiner.

Utover 50-tallet ble datamaskinene kommersielle produkter, og dataindustrien begynte   vokse. Denne utviklingen har n  p g tt i over 50  r. P  denne tiden har informasjonsteknologi blitt en av de st rste og mest hurtigvoksende n ringene p  global basis.

#### Eksempel p  minneformat til IAS

Hvis det 40-bits innholdet av en minnelokasjon skulle tolkes som et heltall var det gitt p  f lgende form:



## 1.2. Generasjoner av datamaskiner

Rador ret ble oppfunnet i 1906, og de datamaskinene som ble bygget med *rador r* kalles 1. generasjon datamaskiner. Slike maskiner ble laget i perioden 1946-57. Typisk antall operasjoner kunne v re 40.000 pr sek. Den f rste kommersielle datamaskinen med en viss suksess var UNIVAC (Universal Automatic Computer). Med den ble flere trekk utviklet som fortsatt kjennetegner dataindustrien. Blant annet at ny teknologi hurtig blir tatt i bruk, men man beholder bakover kompatibilitet med tidligere produkter. Da kan kunden bruke samme programvare, og man beholder gamle kunder n r man kommer med nye produkter.

Transistoren ble utviklet i 1947, og i l pet av 50- rene fikk den innpass i all slags elektronikk. Mellom 1958 og 1964 ble datamaskinene bygget med denne teknologien. Slike maskiner kalles 2. generasjons datamaskiner, og antall operasjoner kom opp i rundt 200.000 pr sekund. P  disse maskinene ble det introdusert mye mer avanserte *aritmetisk-logiske-enheter* (ALU - Arithmetic Logic-Unit) og man begynte ogs    snakke om h yniv  programmeringsspr k. De f rste versjonene av programmeringsspr ket FORTRAN (FORMula TRANslator) ble for eksempel utviklet p  denne tiden.

En transistor er en s kalt diskret komponent, og 2. generasjonsmaskinene kunne best  av flere hundre tusen slike enkeltst ende komponenter. B de produksjon, montering, feils king og feilretting ble sv rt komplisert. I 1958 ble imidlertid *integrerte kretser* (IC) utviklet. En integrert krets er, som navnet sier, en krets hvor flere komponenter er samlet (integrert) p  samme enhet. En slik enhet kalles *brikke*, eller chip p  engelsk.

I perioden 1965 til 1971 laget man 3. generasjon datamaskiner med slike ICer. N  kom man opp i hastigheter p  rundt 1.000.000 operasjoner pr sekund.

Disse oppfinnelsene representerer rene kvantesprang i teknologien. Senere har utviklingen imidlertid fortsatt jevnt og trutt med stadig bedre integrerte kretser, uten at vi har opplevd slike plutselige hopp i teknologiutviklingen. Hvis vi likevel vil bruke generasjonsbegrepet, kan vi si at vi har gjennomg tt 4. og 5. generasjon med s kalte LSI- (Large Scale Integration) og VLSI- (Very Large Scale Integration) kretser. Fra   kunne samle noen ti-talls transistorer p  en brikke, har teknologien utviklet seg til   kunne samle flere ti-talls millioner transistorer p  en og samme brikke.

### 1.3. Maskinklasser

Datamaskiner karakteriseres ofte etter ytelse. Strengt tatt er grensene mellom de fleste typene s vrt s  flytende, men tradisjonelt har vi grovt sett delt opp i stormaskiner, minimaskiner og mikromaskiner. I tillegg kommer supermaskinen.

*Stormaskiner* representerer den "tradisjonelle" l sningen. Den best r av store system hvor en maskin betjener s vrt mange brukere. Stormaskiner krever store investeringer fordi den er dyr og den krever driftspersonell og spesielle maskinrom. Typisk for slike system er at det foreg r enormt mye I/O. L sningen brukes fortsatt, spesielt i en del forretningsl sninger - for eksempel innen bank, forsikring og bookingsystemer for reiser - med store sentraliserte databaser, mange terminaler og mange transaksjoner. IBM er den st rste leverand ren av slike l sninger.

*Minimaskiner* oppsto da teknologien tidlig p  60-tallet gjorde det mulig   fremstille relativt sm  og billige maskiner (i forhold til stormaskiner). I en del arbeidsoppgaver egner en stormaskin seg d rlig, og minimaskinene ble blant annet brukt i prosessstyring og av forskergrupper som ikke trengte den store regnekraften til stormaskinen. DEC (Digital Equipment Corp.) var den st rste leverand ren av minimaskiner. En annen kjent akt r i minimaskin-markedet var det norske flaggskipet Norsk Data.

*Mikromaskiner* er maskiner av den typen de fleste av oss tenker p  som datamaskiner. Det vil si personlige maskiner, som PC og Mac. Den opprinnelige definisjonen av mikromaskin var en maskin som har samlet den *aritmetisk/-logisk enhet* og *kontrollenheten* i  n eneste integrert krets (en s kalt mikroprosessor).

Det er verdt   merke seg f lgende historiske trekk:

P  70-tallet ble stormaskinen utsatt for hard konkurranse fra minimaskinene. Dette f rte til nye og sterkt forbedrede stormaskiner. Denne trenden har holdt seg, og det utvikles stadig bedre og bedre stormaskiner.

P  slutten av 80-tallet og p  90-tallet utviklet mikromaskiner i datanettverk seg til   bli s vrt gode alternativer til b de minimaskinl sninger og stormaskinl sninger. Dette gjorde at minimaskinl sningene etter hvert forsvant. Minimaskiner selges ikke lenger. Det er bare for fullstendighetens skyld at minimaskiner nevnes her.

Stormaskinene holder imidlertid stand. Det selges fortsatt stormaskiner, og for mange bransjer er den fortsatt et naturlig valg. Bransjer som banker, reisereservasjoner og store Internett-baserte nettverksl sninger kj rer fortsatt p  stormaskiner. Den st rste leverand ren er IBM, som tjener gode penger p  dette markedssegmentet.

*Supermaskinen* er den siste hovedklassen av maskiner. I motsetning til de andre typene er ikke dette en generell maskin. Denne maskintypen er laget for en bestemt type beregninger, nemlig vitenskapelige beregninger. N rmere bestemt flyttallsberegninger med h y presisjon. Supermaskiner kan utf re slike beregninger p  store mengder tall. Slike maskiner er dyre og det selges ikke mange av dem.

### 1.4. Datamaskinfamilier

I 1964 var IBM allerede en av de st rste datamaskinleverand rene. P  denne tiden kom den nye IC-teknologien, og IBM tok sjansen p    markedsf re en helt ny maskin som ikke var kompatibel med de tidligere IBM-maskinene. Det vil si; egentlig kom de med mange maskiner som spente over et stort spekter av av ytelse og pris. Tanken var at alle disse

maskinene skulle kunne kj re samme programvare, slik at hvis kunden fikk st rre databehov kunne han skifte maskin men beholde programvaren. Dette er det f rste eksemplet p  en datamaskinfamilie, og konseptet f rte IBM opp i en suveren markedsledelse. Den dag i dag har moderne IBM stormaskiner arkitektur basert p  denne datamaskinfamilien som kalles System 360, eller S/360-arkitekturen.

En datamaskinfamilie er alts  maskiner

- Som har samme, eller i hvert fall sv ert likt, instruksjonssett
- Som har samme, eller i hvert fall sv ert likt, operativsystem
- Som har maskiner som spenner over flere prisklasser
- Hvor dyrere maskiner er hurtigere
- Hvor dyrere maskiner kan takle mere I/O
- Hvor dyrere maskiner har mere minne

#### 1.4.1. Intel og IBM PC

I 1968 ble Intel Corporation grunnlagt. P  bestilling fra en kalkulatorfabrikant produserte de i 1971 verdens f rste mikroprosessor, 4-bits prosessoren 4004. Et  r senere kom den f rste 8-bits mikroprosessoren, 8008, som ogs  var spesialbestilt. Dette var prosessorer laget for spesielle form l. P  grunn av uventet stor interesse fra flere hold satte de i gang med   lage en generell (general purpose) prosessor. Dette resulterte i 8080-prosessoren som tok markedet med storm. Millioner av brikker ble solgt.

I 1978 kom den f rste 16-bits prosessoren, 8086, og kort tid etter 8088 som var en billigere utgave av 8086. Den hadde en 8-bits databuss istedenfor den originale 16-bits databussen. IBM valgte 8088-prosessoren til den originale IBM PC som ble introdusert i 1981. IBM friga ogs  den tekniske dokumentasjonen til maskinen. Dette f rte til at andre kunne lage "kloner" og konkurrere p  PC markedet. Dette har nok hatt stor medvirkning til at IBM-standarden ble, og fortsatt er, den mest utbredte innen mikromaskiner.

8088-prosessoren kunne ikke bruke mer enn 1 MB minne. Dette ble etter hvert et s  stort problem at Intel kom med en ny prosessor, 80286, som var kompatibel med 8088. Organiseringen av minnet var annerledes og ganske tungvindt nettopp fordi den m tte v re bakoverkompatibel. Dette er ofte et krav da mange er avhengig av   kj re gammel programvare. Alle Intel prosessorene er bakoverkompatible helt ned til 8088-prosessoren. Det er i det hele tatt sjeldent i denne bransjen at man bryter helt med fortiden.

Motorola gjorde likevel dette i 1979, da de kom med den helt nye 68000-prosessoren som med sine 32-bits registre var godt i forkant av Intel, som f rst i 1985 introduserte sin f rste 32-bits prosessor, nemlig Intel 80386.

Intel 80386 var en 32-bits prosessor med 32-bits buss. I 1989 kom s  80486 prosessoren som er en raskere versjon av 80386. Den st rste forskjellen er tillegget av en flyttalls-ko-prosessor og cache.

I 1993 kom Pentium-prosessoren, og de siste  rene har den nyeste generasjonen kommet. Det er Pentium Pro, Pentium II, Pentium III og Pentium IV (som alle tilh rer samme generasjon, men har forskjellig ytelse).

Det har v rt en enorm utvikling innen mikroprosessorteknologi siden 4004 prosessoren ble introdusert. Et eksempel er antall transistorer. Fra rundt 2.300 p  4004-prosessoren til rundt

800 millioner p  de nyeste Pentium-prosessorene. 8086-prosessoren utf rte under 1 MIPS (millioner instruksjoner per sekund) mens Pentium Pro utf rer rundt 250 MIPS. En PC i dag kan ha st rre regnekapasitet enn en IBM stormaskin fra 10  r tilbake.

Dette har f rt til at flere og flere applikasjoner som tidligere var reservert for stormaskiner, n  ogs  kan kj res p  mikromaskiner i nettverk. Ved   koble mikromaskinene sammen i nettverk, kan informasjon b de deles og effektivt flyttes rundt i bedriften. Ogs  flerbrukerapplikasjoner kan kj res i PC-nettverk.

En annen  rsak til mikromaskinens inntog er de vennlige brukergrensesnittene som har blitt utviklet. Utviklingen av grafiske brukergrensesnitt (som blant annet brukes i Windows). Det hele startet med et forskningsprosjekt hos Xerox, men den f rste maskinen med et grafisk brukergrensesnitt var Apple Computer sin Mac. Gode brukergrensesnitt, tilfredstillende beregningskapasitet og bra ytelse p  kommunikasjon over nettverk gj r at mikromaskiner meget vanlig p  alle slags arbeidsplasser og ogs  i hjemmet.

#### 1.4.2. Mikromaskiner

Det har v rt mange akt rer p  mikromaskinmarkedet. Kjente mikromaskiner er for eksempel IBM/PC med prosessorer fra Intel og Apple Macintosh som bruker prosessorer fra Motorola. B de IBM/PC og Apple Macintosh er eksempler p  *personlige datamaskiner*. Det vil si maskiner som betjener en bruker i slengen. De kan ogs  kobles i nettverk, og p  den m ten betjene mange brukere.

De siste  rene har mikroprosessorene oppn dd en regnekapasitet p  h yde med hva stormaskiner hadde for f   r siden. Eksempler p  slike kraftige prosessorer er Intels Pentium prosessorer, de kraftigste PowerPC prosessorene fra Motorola (utviklet i samarbeid mellom Motorola, IBM og Apple), og Intels nye Itanium prosessorer (som er utviklet sammen med Hewlett Packard). Alle disse prosessorer representerer den maksimale ytelse med dagens teknologi.

### 1.5. Utviklingstrekk

Det har v rt brukt mange begrep for   beskrive bruken av datamaskiner.

For en del  r tilbake var det vanlig   snakke om EDB som st r for elektronisk databehandling. Utrykket forteller at teknologien blir brukt til   behandle data; det vil si   i stor grad bruke datamaskinen som en regnemaskin. EDB-begrepet forbindes med et betydelig teknologisk fokus.

Etter hvert ble det vanlig   bruke datamaskiner til   behandle informasjon p  langt mer generelt grunnlag. Fokus ble flyttet fra teknologi og over til bruken av teknologien, og det ble vanlig   snakke om informasjonsteknologi, eller IT, isteden.

Parallelt med utviklingen av datamaskiner har det ogs  skjedd en enorm utvikling innen datakommunikasjon. Dette har gjort datanettverk til en naturlig del av v r hverdag. De fleste firma har lokale nettverk, og mange er ogs  koplet opp mot det globale Internett. Dette gj r at kommunikasjon er en like viktig del av datateknologien som selve informasjonsbehandlingen.

Derfor har det de siste  rene blitt vanlig   snakke om IKT, som st r for informasjons- og kommunikasjonsteknologi.

Vi skal se litt p  hovedtrekkene i utviklingen.

### 1.5.1. Informasjonsteknologi

Etter hvert som datamaskinene har f tt stadig st rre regnekapasitet og stadig blitt billigere har bruken av datamaskinen endret seg. Man kan godt skissere f lgende hovedtrekk i utviklingen:

- ”Mange brukere – en maskin” (Stormaskinen og minimaskinen p  50, 60, 70-tallet).
- ”En bruker – en maskin” (Mikromaskiner. PCens inntog, 80-tallet).
- ”En bruker – mange maskiner” (PC i nettverk. Internett fra 90-tallet).

Et annet utviklingstrekk er at det har skjedd en sammensmelting mellom den rene regnemaskinteknologien som datamaskinen har v rt basert p , og andre media som tekst, lyd, bilde, film og animasjon:

- Datamaskinen har blitt en multimedamaskin.

### 1.5.2. Moore's lov

P  grunn av kontinuerlige forbedringer innen teknologien for fremstilling av integrerte elektroniske kretser, er moderne prosessorer inne i en utvikling som det knapt finnes maken til innen den industrielle historien. Gjennom flere ti r har produsentene fortl pende greid   fordoble regnekapasiteten til prosessorene med s  korte mellomrom som hver 18. m ned.

Det betyr at en ny prosessor er dobbelt s  hurtig som en prosessor som er ett og et halvt  r gammel. Denne utviklingen vil nok fortsette enda en stund, inntil vi treffer p  fysiske begrensninger som ikke lar seg l se med dagens teknologi. Slike begrensninger kan v re at strukturene i de integrerte kretsene kommer ned mot atomst rrelsen, og dermed ikke lar seg forminske mer.

Dette at prosessorene fordobler regnekapasiteten s  ofte ble f rst beskrevet av en kar med navn Moore, og kalles derfor Moores lov:

*Tilgjengelig maskinkraft pr krone fordobles ca hvert 1,5.  r. (Moores lov)*

## 1.6. Kommunikasjonsteknologi

Samtidig med at regnekapasiteten til datamaskinene har  kt, har det skjedd en voldsom utvikling i datakommunikasjonsteknologi og datanettverk.

Dette gj r at nettverk og datakommunikasjon har blitt en viktig del av datautviklingen. Moderne bedrifter har en betydelig kommunikasjon b de internt og eksternt, for eksempel mot Internett.

Viktige utviklingstrekk for kommunikasjonsteknologi er:

- Tidligere har man hatt ulike spesialnettverk for ulike tjenester. For eksempel har man hatt egne nett til datakommunikasjon, telefoni og kabel-TV. Vi er n  inne i en utvikling mot   samle alle typer kommunikasjon i ett og samme nett. Dette kalles tjenesteintegreert nett.
- B ndbredden  ker hurtig. B ndbredde er et m l for hvor mye informasjon som kan overf res pr sekund i et nett. Kommunikasjonsutstyr blir stadig bedre, og det trekkes mye nye nettverk. Det betyr at den tilgjengelige b ndbredde  ker sv rt hurtig. Det er vanlig   regne med at tilgjengelig b ndbredde 3-dobles hvert  r (Gilders lov).

- Det utvikles felles kommunikasjonsstandarder som g r det enkelt   kommunisere mellom ulike typer utstyr og mellom ulike organisasjoner. (eksempel: Internett)

### 1.6.1. Internett

Utviklingen av Internett har akselerert voldsomt de siste  rene. Fra   v re en kommunikasjonskanal for den akademiske verden, har den blitt en alminnelig tilgjengelig informasjonskanal som brukes av sv rt mange. De viktigste grunnene til dette er:

- Internett og web er basert p   pne standarder
- Brukere og tjenester kan kommunisere, samvirke og dele ressurser p  tvers av plattformer
- Effektiv utviklingsmulighet for nye produkter, ressurser og tjenester
- Brukere, tjenester og ressurser knyttes sammen i et system som er tilgjengelig og virksomt uavhengig av tid og rom.
- Enkle, og n rmest universelle, brukergrensesnitt mot tjenester og ressurser i form av en nettleser.
- Ulike datatyper og kommunikasjonsformer konvergerer. De samles, h ndteres og distribueres i en og samme datastr m og datakanal.

I tillegg kan man ta med at det er ingen sentral kontroll og styring med tjenester og ressurser p  Internett. Dette har den ulempen at det finnes mye informasjon som ikke er kvalitetssikret, og at det finnes informasjon som de aller fleste av oss gjerne kunne v rt foruten. Fra et teknologisk st sted har det imidlertid f lgende fordeler:

- Det gir en kooperativ og eksperimenterende praksis i utvikling av tjenester og ressurser.
- Det fremmer kollektiv kreativitet og utpreget tradisjon for deling av kunnskap

## 1.7. Informasjons- og kommunikasjonsteknologi (IKT)

IKT p virker alle i det moderne samfunn, og er gjenstand for samfunnsdebatt og politisk styring. Fra et teknologisk st sted er IKT basert p  f lgende utviklingstrekk:

1. Informasjonsteknologi: Fra regnemaskin til multimediamaskin
2. Kommunikasjonsteknologi: Fra spesialnett til tjenesteintegrerende nett
3. Utvikling av felles kommunikasjonsstandarder og tilgjengelig b ndbredde.