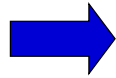


# INF1040 – Digital representasjon



22. august 2007

## Praktisk informasjon Kapittel 1

<http://www.uio.no/studier/emner/matnat/ifi/INF1040/h07/>

# INF1040 - Digital representasjon

av tekster, tall, former, lyd, bilder og video



### Forelesere:

- Fritz Albregtsen
- Ragnhild Kobro Runde

[ragnhilk@ifi.uio.no](mailto:ragnhilk@ifi.uio.no)

[fritz@ifi.uio.no](mailto:fritz@ifi.uio.no)

### Forelesninger:

- Onsdager, 12:15 – 14:00, Sophus Lies Auditorium.

### Pensum:

- Hele læreboka
- Stoff gjennomgått i forelesninger, oppgaver og obliger

### Obligatorisk oppmøte i dag

- Registrer deg i pausen!

## Gruppeundervisning – I

- Undervisningsuken starter med forelesningen
- Terminalstueundervisning (PO-bygget)
  - Velg 1 av 8 grupper
  - Dere løser oppgaver
  - Gruppelærerne hjelper og veileder

	Onsdag	Torsdag	Fredag	Lør	Søn	Mandag	Tirsdag
08-09							G5 (PS) Baktus
09-10							
10-11		G1 (INF) Karius	G3 (INF/LAP) Baktus	G8 (TOOL) Rom 121			G104 (INF) Baktus
11-12							
12-13	Forelesning SL	G7 (DIG) Rom 121		G2 (INF) Karius			
13-14							
14-15						G6 (DIG) Rom 121	
15-16							

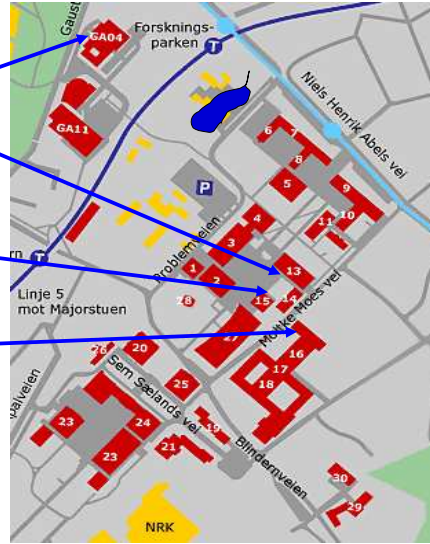
## Gruppeundervisning - II

- Plenumsundervisning – velg 1 av 4 dobbeltimer
  - To og to gruppelærere
  - Gjennomgang av teori, eksempler, løsninger ...

	Onsdag	Torsdag	Fredag	Lør	Søn	Mandag	Tirsdag
08-09							
09-10							
10-11							
11-12							
12-13	Forelesning SL	Ø3 Aud 1, VB	Ø2 Aud 1, VB				Ø1 Aud 1, VB
13-14							
14-15							
15-16							
16-17						Ø104 ? Aud 1, VB	
17-18							

## Litt Blindern-geografi

- ❑ **Dere finner oss på Ifi**
- ❑ **Vilhelm Bjerknes ligger her**
  - Plenums-grupper
- ❑ **Sophus Lie ligger her**
  - Forelesninger
- ❑ **PO-bygget ligger her**
  - Terminalstue-grupper



## Forelesningsnotater og oppgaver

- ❑ **Foilene til forelesningene**
  - legges ut mandag før forelesning
  - Pdf-filer 1:1 for de som vil lese foiler på P/TV
  - Pdf-filer 1:4 for utskrift på papir
  - NB: Foilene er ment som et hjelpemiddel i undervisningen, og dekker ikke nødvendigvis alt som skjer på forelesningen!
- ❑ **Opgaver legges ut før forelesningen**
  - Løsningsforslag kommer uken etterpå
- ❑ **Flervalgstester på nettet legges ut i etterkant av forelesningen**

## ”Obliger” og eksamen

- ❑ **Tre ”obliger” MÅ godkjennes**
  - 1 leveres inn fredag 5/10
  - 2 leveres inn fredag 2/11
  - 3 leveres inn onsdag 21/11
  - Fristene skal overholdes!!!
- ❑ **Skriftlig eksamen**
  - Fredag 7/12 09:00-12:00
  - Bokmål / nynorsk / engelsk
  - Ingen hjelpemidler
  - Bokstavkarakterer (A-F)

## Arbeidsmengde

- ❑ **Det blir mye stoff per forelesning**
- ❑ **Gå på terminalstuegruppene**
  - Gjør hederlige forsøk på å løse ukeoppgavene
- ❑ **Gå på plenumsgruppene og få**
  - Alternativ gjennomgang av teori
  - Eksempler, løsningsforslag etc
- ❑ **La ikke ett enkelt kurs ta (nesten) all din tid!**
  - Du skal (stort sett) ha eksamen i tre kurs
- ❑ **Arbeid minst 12 timer per uke per kurs**
  - 6 timer + 2 timer forelesning + 4 timer grupper

## Er dette vanskelig?

- **Dere er en sammensatt gruppe**
  - Noen kan mye fra før, andre litt mindre
  - Vi skal ha med oss (nesten) alle
- **Innsatsen betyr mer enn hva du kan fra før**
  - Arbeid jevnt – hele semestret
- **Still spørsmål**
  - Til foreleserne og til gruppelærerne
- **Vi er her for dere – ikke omvendt !!!**

## Kapittel 1: Innledning

Lagring og overføring av informasjon

Koding

Analog vs digital representasjon

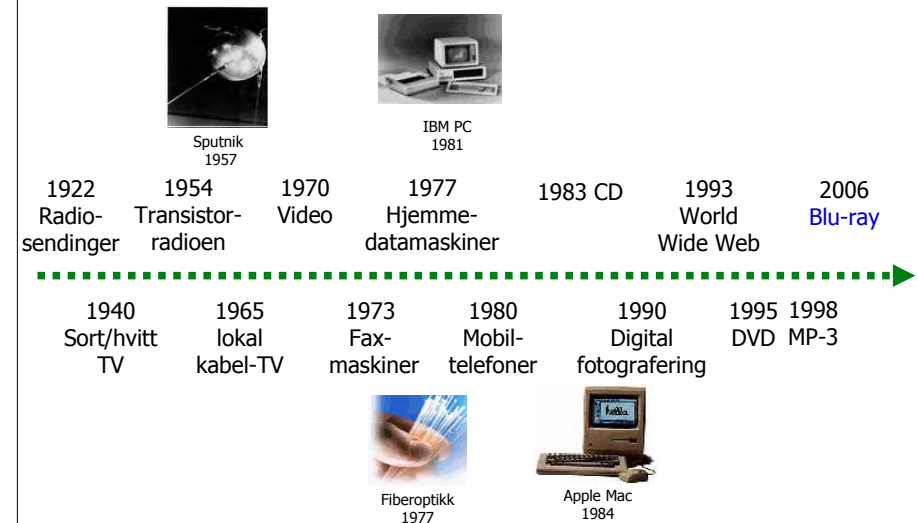
Lagring

Overføring

## Lagring og formidling av informasjon



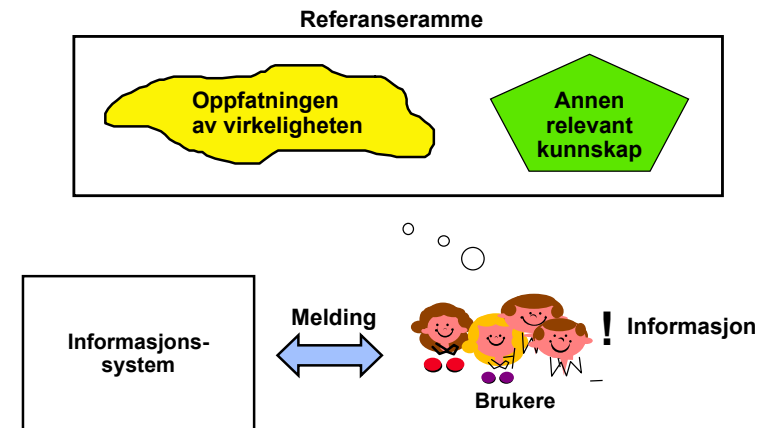
## Lagring og formidling av informasjon (forts.)



## Hva er det vi lagrer og overfører?

- Meldinger, kodet i et alfabet
  - Røyksignaler
  - Morsekode
  - Varselsignaler 
  - Vanlig tekst
- Analog informasjon
  - Kontinuerlige fenomener (temperatur, vindstyrke, vindretning, vannføring, vannstand, posisjon, ...)
  - Lyd
  - Bilder

## Meldinger krever tolkning



## Koding av meldinger




Krav til kodingen:

- Den må entydig representere meldingen slik at den kan gjenskapes i sin opprinnelige form
- Den må være standardisert slik at den kan brukes for ulike formål
- Den må egne seg for billig og pålitelig teknologi

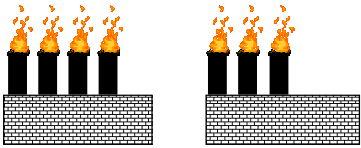
## Alfabetet

- Kodingen gjøres i et tegnssett med et endelig antall tegn, kalt et *alfabet*

- Eksempler på alfabetet

-  ikon
  - japansk kanji
  - ji ten sha japansk "latin"
  - 6A 69 20 74 65 6E 20 73 68 61 japansk "latin" heksadesimal
  - 01101010 01101001 00100000 01110100 01100101  
10011110 00100000 01110011 01101000 10010001  
japansk "latin" binær
- Her koder vi først i ett alfabet, og så koder vi hvert tegn i et annet alfabet

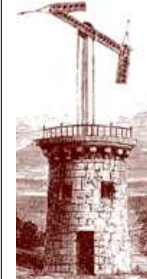
## Koding – Polybius' system



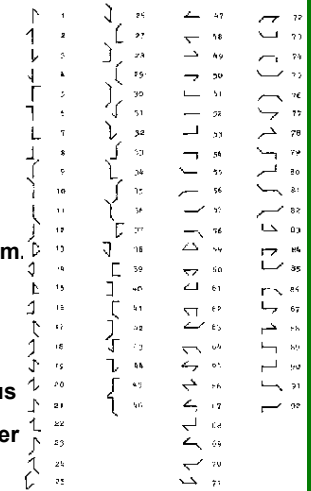
	A	B	Γ	Δ	E
	Z	H	Θ	I	K
	Λ	M	N	Ξ	O
	Π	P	Σ	T	Υ
	Φ	X	Ψ	Ω	

- Gresk historiker (ca 203-120 f. Kr)
- To tårn, hvert av dem med fem fakler
- Antall brennende fakler på hvert tårn pekte ut en bokstav i det greske alfabetet, arrangert i en 5X5-matrise
- Faklene ble hevet/senket, evt mørklagt med en skjerm mellom hver bokstav

## □ happe's semafor-telegraf

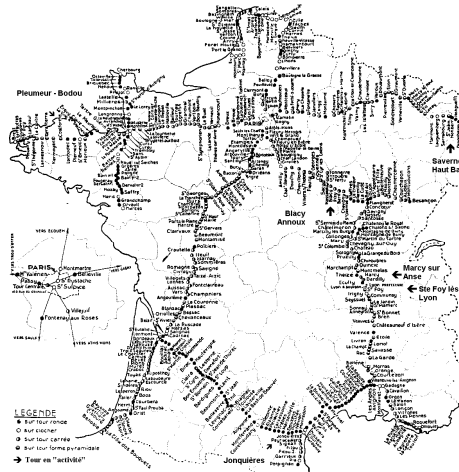


- Fransk oppfinner (1763-1805)
- Tårn plassert ca hver 10. km
- 1794: Paris – Lille (230 km)
- 1810: Paris – Venezia og Paris Amsterdam
- 1844: 25 byer, 556 stasjoner, 4800 km.
- 196 mulige symboler (7x7x4).
- Kodebok basert på symbol-par:
  - En utvidelse av ideen til Polybius
  - 92 sider a 92 linjer ga 8 464 koder for tegn, ord eller hele fraser.

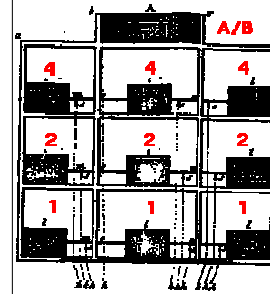
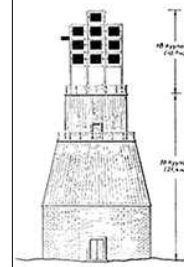


## Napoleon – en bruker av teknologi

- Innså den militære fordel i å sende informasjon raskt fra sted til sted.
- Bygget opp et nett av linjer.
- Brukte mobil optisk telegrafi i sine felttog.
- Kunne kordinere styrker og logistikk over lengere avstander enn noen annen armé i samtiden.
- Hastighet ca 500 km/time.

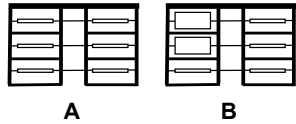
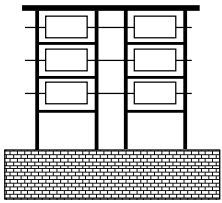


## Edelcrantz' optiske telegraf



- Svensk akademiker (1754-1821).
- I praktisk bruk fra 1794 til 1881.
- Hver stasjon hadde 10 metall-lukkere.
  - Dobbelt så raskt som □ happe's system.
- Et binært kodesystem med 10 biter.
- Kunne kode 1024 forskjellige signaler.
- Oktal koding av signaler.
  - Fra A000 til B777
  - Veldig enkelt å lese av og sende videre
  - Mer robust mot feil enn semafor
- En forløper for moderne systemer.

## Murrays optiske telegraf - 1796

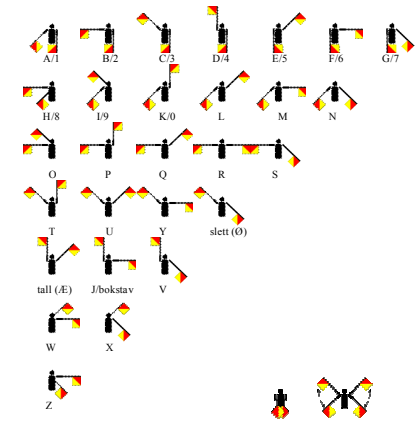


- Engelsk lord og biskop
- Seks plater dreies med flaten eller kanten mot observatøren
- En binær kode
- 64 mulige symboler
- Korte meldinger kunne sendes over 100 km fram og tilbake på 15 minutter.
- Kompresjon var ofte brukt:
  - H.M.S. ROYAL SOVEREIGN ANCHORED AT SPITHEAD YESTERDAY SAILING FOR PLYMOUTH TOMORROW
  - RSOV ANCHOR SPITD YESDA SAIL PLYTH TMRO

## Semafor med to flagg

- Hvor mange forskjellige tegn kan vi kode med to flagg, når hvert flagg kan ha 8 posisjoner?
- $$7 + 6 + 5 + 4 + 3 + 2 + 1 = \frac{7 \times 8}{2} = 28$$

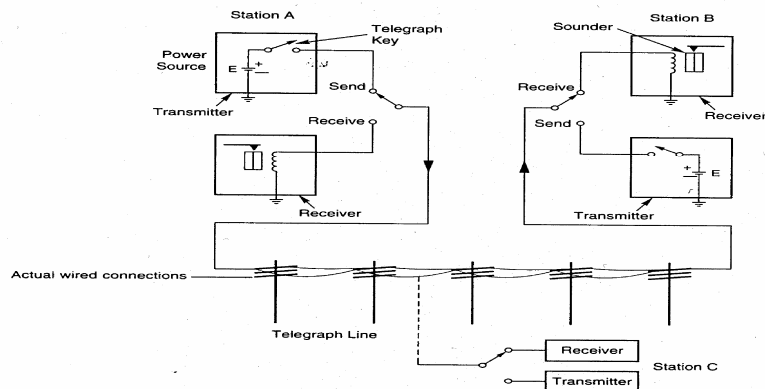
- Nok til alfabetet og tallene
  - hvis vi har koder for å skifte mellom tegn og tall



I tillegg er det en startposisjon og et kallesignal

## Så kom elektrisiteten...

- Rundt 1800: Tidlige forsøk på å signalisere ulike tegn ved hjelp av et system med ulike spenninger ble på grunn av feilkildene erstattet av av et binært system (spenning på eller av)



## Morsealfabetet (1844)

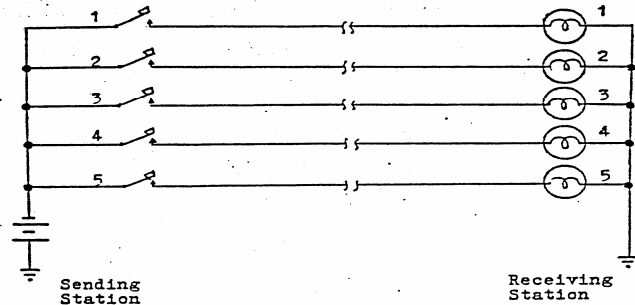
- Konstruert av Morse, amerikansk maler (1791-1872) og/eller Vail, amerikansk maskinist (1807-1859)
- Hvert tegn representert med en sekvens av prikker (kort signal) og streker (langt signal), atskilt av kort pause (mellom tegn) eller lang pause (mellom ord)
- Koden bygd på statistikk: Hyppigst brukte tegn har de korteste sekvensene (observasjon av typekassetene i et trykkeri)
- Eksempel: ··· --- ··· SOS (Internasjonalt maritimt nødsignal)

A	· —
B	· · · ·
C	— · · ·
D	— · ·
E	·
F	· · — ·
G	— — ·

Morsealfabetet har vært i bruk helt opp til ganske nylig!

## Baudots femlednings-system (1874)

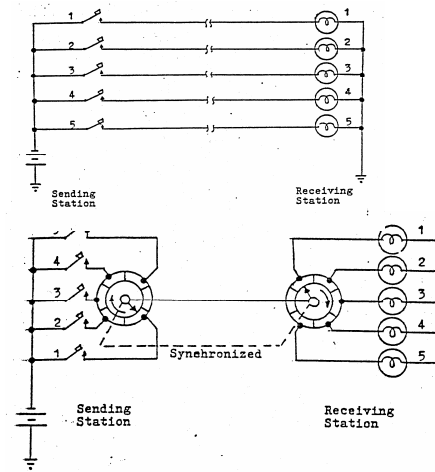
- Fransk ingeniør (1845-1903)
- Brukte fem ledninger for å overføre ett tegn
- Første eksempel på fast antall biter per tegn i den elektriske verden
- Et genuint binært system!



© Institutt for informatikk – 22. august 2007

INF1040-Intro-25

## Baudot – parallell vs. seriell overføring



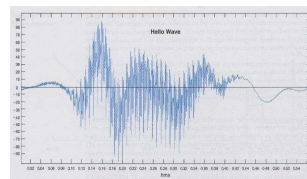
Navnet lever videre i "Baud", en enhet for båndbredden på en overføringslinje

© Institutt for informatikk – 22. august 2007

INF1040-Intro-26

## Analog virkelighet – analog representasjon

- Analog – "som er basert på fysiske, kontinuerlig varierende størrelser"
- Virkeligheten er stort sett analog
  - Tid, temperatur, posisjon, lysintensitet, lydstyrke, ...
- Den mest direkte løsningen er å lagre og overføre informasjon ved hjelp av analoge signaler
- Eksempler
  - Konvensjonell radio og TV, telefon
  - Grammofonplater, kassettbånd



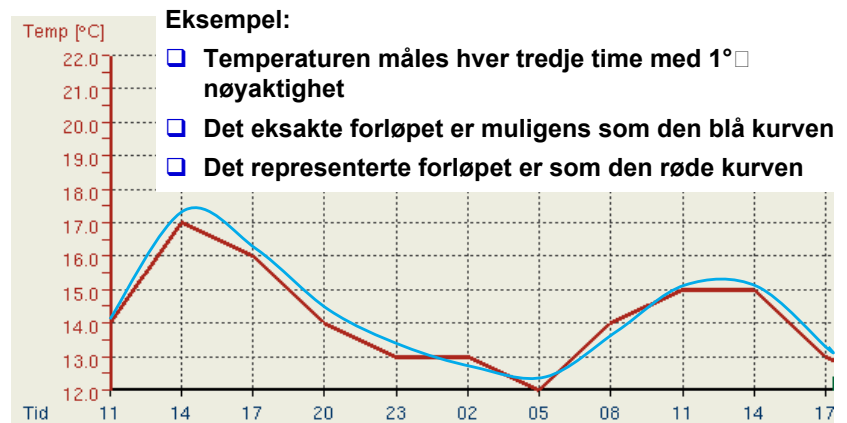
Analogt signal: Lydbølger for "hello"

© Institutt for informatikk – 22. august 2007

INF1040-Intro-27

## Analog virkelighet – digital representasjon

- Digital – "som gjengir fysiske størrelser med diskrete tegn"
- Forutsetter *diskretisering* og *kvantisering*



© Institutt for informatikk – 22. august 2007

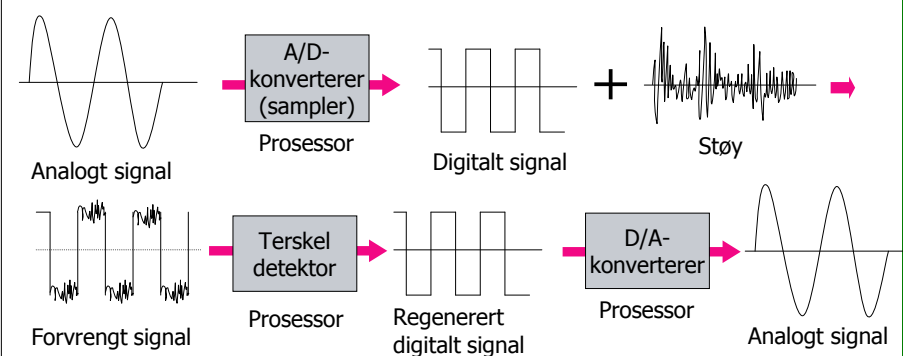
INF1040-Intro-28



## Digital vs. analog teknologi

- ❑ Digitale representasjoner kan lagres og overføres digitalt
- ❑ Digitale representasjoner kan kopieres med 100 % nøyaktighet
- ❑ Digitale representasjoner kan bearbeides
  - komprimeres for effektiv overføring og lagring
  - krypteres for økt sikkerhet
  - renses for å oppnå bedre kvalitet
- ❑ Digitale representasjoner gir muligheter for lokalisering av eventuelle feil og automatisk feilretting
- ❑ Digitale overføringer kan multiplekseres for økt kapasitet
- ❑ Digital teknologi er mye billigere
- ❑ Digitale overføringer er ikke utsatt for forvrengning og unøyaktigheter på grunn av interferens med andre signaler (støy)

## Analoge og digitale signaler – og støy



## Datamaskinverdenen er binær digital

- ❑ Digitale datamaskiner arbeider med 2 diskrete verdier, 0 og 1
- ❑ 0 og 1 kalles binære tall – binary digits – bits eller biter
- ❑ To verdier er tilstrekkelig for å kode tekst, tall, lyd, bilder og video!
- ❑ Alt er representert ved sekvenser av biter – bitmønstre
- ❑ De fysiske signalene kan være varierende elektriske spenninger, lyspulser, magnetiseringsretninger, hull/ikke-hull...

### Hvorfor bare to verdier?

- ❑ To verdier passer bra med de elektroniske på/av-brytere som finnes i digitale datamaskiner
- ❑ Jo færre verdier, jo lettere er det å skille dem fra hverandre
- ❑ Et binært system er mer robust

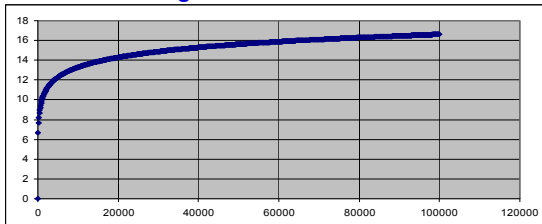
## Hvor mange ulike bitmønstre?

- ❑ 1 bitposisjoner – 2 bitmønstre {0, 1}
- ❑ 2 bitposisjoner – 4 bitmønstre {00, 01, 10, 11}
- ❑ 3 bitposisjoner – 8 bitmønstre {000, 001, 010, 011, 100, 101, 110, 111}
- ❑ 4 bitposisjoner – 16 bitmønstre {0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111}
- ❑ 5 bitposisjoner – 32 bitmønstre {00000, 00001, 00010, ... , 11110, 11111}
- ❑ ...
- ❑ For hver ekstra bitposisjon doubles antall mulige bitmønstre
- ❑  **$n$  bitposisjoner gir  $2^n$  mulige bitmønstre**
- ❑ Moderne datamaskiner arbeider gjerne med grupper på 8 biter.
  - En slik gruppe på 8 biter kalles en byte, forkortet B.



## Hvor mange bitposisjoner trengs?

- Anta at vi har  $K$  ulike ting (objekter)
- Anta at hvert objekt skal representeres med et unikt (entydig) bitmønster
- Anta at alle bitmønstre må ha en fast lengde  $n$
- Da er  $n$  gitt ved logaritmen til  $K$  med grunntall 2:
 
$$n = \log_2(K) \quad (\text{se side 351 i boka})$$
  - "Toerlogaritmen til  $K$  er det tallet du må opphøye 2 i for å få  $K$ "



Prøv funksjonen  
 $\text{LOG}(K;2)$  i et  
regneark!

## Store tall

- For å håndtere store tall i tallsystemet bruker vi en-bokstavs SI-symboler som betegner potenser av 1000
  - $k$  (kilo) =  $10^3$ ,  $M$  (mega) =  $10^6$ ,  $G$  (giga) =  $10^9$ ,  $T$  (tera) =  $10^{12}$ ,
  - $P$  (peta) =  $10^{15}$ ,  $E$  (exa) =  $10^{18}$ ,  $Z$  (zeta) =  $10^{21}$ ,  $Y$  (yotta) =  $10^{24}$

(Merk at vi her bruker  $k$  for 1 000, fordi  $K$  i SI-systemet er en temperatur.)
- Anta at vi har et digitalt bilde med  $1\,024 \times 1\,024$  piksler (bildeelementer)
  - La hvert piksel representeres med 1 byte
  - Bildets størrelse blir oftest angitt til 1 MB
  - Men bildet er jo  $1\,024 \times 1\,024 \times 1$  byte = 1 048 576 byte  $\approx 1.05$  MB
  - Denne feilen øker jo større tall vi snakker om!

## Prefikser for potenser av 1 024

- SI-prefiksene  $k$ ,  $M$ ,  $G$  osv er desimale enheter og har ingen mening som potenser av 1 024.
- IEC publiserte i 1999 en standard for potenser av 1 024.
- Navnene er satt sammen av de to første bokstavene i SI-prefiksene pluss  $bi$  for "binær".

- Les mer om dette i  
Appendiks B  
i læreboka!

Navn	Symbol	Potens av 2 og verdi i tallsystemet
kibi	Ki	$2^{10} = 1\,024$
mebi	Mi	$2^{20} = 1\,048\,576$
gibi	Gi	$2^{30} = 1\,073\,741\,824$
tebi	Ti	$2^{40} = 1\,099\,511\,627\,776$
pebi	Pi	$2^{50} = 1\,125\,899\,906\,842\,624$
exbi	Ei	$2^{60} = 1\,152\,921\,504\,606\,846\,976$
zebi	Zi	$2^{70} = 1\,180\,591\,620\,717\,411\,303\,424$
yobi	Yi	$2^{80} = 1\,208\,925\,819\,614\,629\,174\,706\,176$

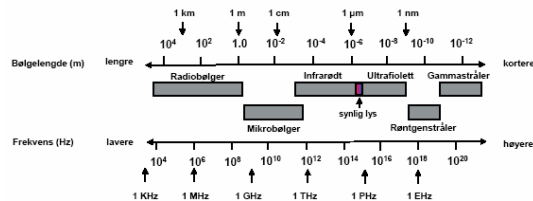
## Noen konvensjoner det er nyttig å kjenne ...

- Størrelsen på RAM, ROM eller flash-minner gis som regel i binære enheter.
- Kapasiteten til harddisker og lagre som betraktes som en stor disk oppgis i desimale enheter.
  - Sektorstørrelsene på en disk gis nesten alltid i toerpotenser, siden de mapper direkte til RAM.
  - Det finnes en forvirrende hybrid, der en "megabyte" betyr 1000 "kilobytes" a 1024 byte.
  - En "1.44 MB" diskett er verken  $1.44 \times 2^{20}$  byte eller  $1.44 \times 10^6$  byte, men  $1.44 \times 1\,000 \times 1\,024$  bytes (som er ca 1.406 MiB, eller 1.475 MB).
- Dette kan også gjelde disk-lignende flashminner (toerpotens multipler av desimale megabyte!)
- Kapasiteten til en CD er alltid gitt i binære enheter.
  - En "700 MB" CD har en nominell kapasitet på 700 MiB.
- Kapasiteten til en DVD er gitt i desimale enheter.
  - En 4,7 GB DVD har en nominell kapasitet på 4,38 GiB.
- Overføringskapasitet uttrykkes som bps (biter per sekund) eller Bps (byte per sekund)
  - angis alltid i tallsystemet, med SI-prefikser
  - eksempel: kBps ( $10^3$  byte per sekund), MBps ( $10^6$  byte per sekund), osv

## Hvordan overføres bitene?

### □ Gjennom fritt rom

- Radiobølger
- Infrarødt lys



### □ Gjennom et fysisk medium

- Elektrisitet
  - Lys (fiberoptikk)
- Vi ønsker at minst mulig av signalet skal bli borte på veien

## Fysikkens lover gir begrensninger

- Begrenset signalhastighet
  - $3 \cdot 10^8$  m/s (lyshastigheten) i fritt rom
  - typisk ca 2/3 av dette i et fysisk medium (optisk eller elektrisk)
- Begrenset overføringskapasitet – typisk fra 10 Mbps til mange Gbps
- Begrenset overføringslengde pga. demping ("attenuation"), støy og interferens
  - jo lengre og jo tynnere kabel, jo større demping



*I de fleste tilfeller er begrensninger i overføringskapasiteten mer plagsomt enn begrensninger i signalhastigheten*

## Båndbredde, basisbånd vs. bredbånd

### □ Båndbredde

Teoretisk mål for overføringskapasitet – måles i *baud*

### □ Båndbredden kan utnyttes på to måter:

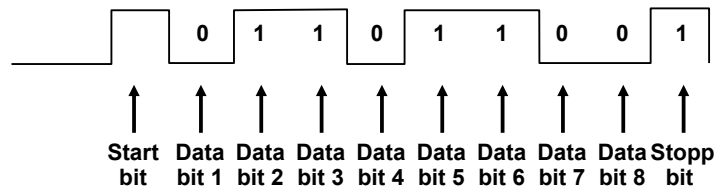
- Basisbånd-signalering
  - » Signalet sendes digitalt og tar hele båndbredden
  - » Brukes f.eks. i Ethernet, videospillere, DVD-spillere og spillkonsoller
- Bredbåndsignalering
  - » Signalet formidles ved å modulere en bærebølge (amplitude, frekvens, faseskift, ...) som overføres analogt (!)
  - Gir flere kanaler
    - flere signaler kan overføres samtidig (multipleksing)
  - » Brukes i de fleste høykapasitets forbindelser (ikke Ethernet)

## Bit-synkronisering

- Det kommer en strøm av biter over overføringsmediet
- Mottaker må være synkronisert med avsender
- Mottaker må få vite hvor hver byte begynner og slutter
- Husk at biter kan mistes på veien – hvordan forhindre at en tapt bit ødelegger hele meldingen?



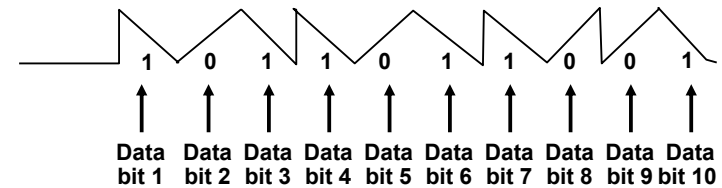
## Asynkron overføring



- Sender og mottaker må være enige om bitraten
- Synkronisering mellom hver byte
- Etter mottak av stoppbit venter mottager på neste startbit og synkroniserer
- I tillegg sendes ofte et paritetsbit (ekstra bit som settes slik at antall 1-ere blir et oddetall)
- Må sende 11 bit for å overføre 8 bit data
- Eksempel på standard: RS-232
- Brukes for printere, modemer, fakser osv

## Synkron overføring

- Klokkingen overføres i parallell med dataene
  - o på separat linje
  - o integrert (for eksempel ved Manchester-koding)



- Brukes for overføring av større datamengder

## Synkron overføring i pakker

Flagg	Adresse	Kontroll	Data	␣R␣	Flagg
01111110	8 biter	16 biter	N * 8 biter	16 biter	01111110

Standard HDL␣ ramme ("frame")

- Eksempel: HDL␣ – High level data link protocol
- En ramme startes og avsluttes med en flaggbyte = 01111110
- Datamengden kan være stor (flere tusen biter)

## HDL␣ Bit -stuffing

- En protokoll som HDL␣ vil feile hvis flagg-byten forekommer også andre steder i bitstrømmen
- Husk at 01111110 = AS␣II/UTF-8 "␣"
- Slutten på et tegn sammen med begynnelsen på et annet, eller f.eks. en RGB-verdi kan også gi 01111110
- For å fikse dette problemet: Etter at flagg-byte er sendt:
  - o Sender: Hvis du har sendt fem 1 etter hverandre, sett inn en ekstra 0
  - o Mottaker: Hvis du har mottar fem 1 etter hverandre, etterfulgt av en 0, fjern 0'en
- Dette kalles Bit Stuffing eller Zero Bit Insertion

## Men hva betyr bitmønsteret?

bit-mønster	betydning
0	nei
1	ja

bit-mønster	betydning
0	♀
1	♂

bit-mønster	betydning
00	♠
01	♥
10	♦
11	♣



*Er disse valgene fornuftige?*

*Blir de forstått av andre?*

*Det er behov for standardisering...*

## Typer av standarder

- **de jure**  
Vedtatt av offisielle standardiseringsorganer
  - ISO – International Organization for Standardization
  - EN – Comité Européen de Normalisation
  - ANSI – American National Standards Institute
  - .....
  - Standard Norge – Standardiseringen i Norge
- **de facto**  
Allment akseptert, gjennom enighet eller markedstyngde
  - åpne – dokumentasjon offentlig tilgjengelig
  - lukkede

## Hva kurset dreier seg om

Hvordan lagrer og overfører vi ved hjelp av bits og bytes på mest hensiktsmessig måte

- tekst
- tall
- lyd
- bilder
- videosekvenser
- .....

?

*Det er viktig å forstå hvordan en gitt teknologi fungerer.*

*Ved å få et innblikk i dette, er du også i stand til bedre å vurdere datamaskinteknologiens muligheter og begrensninger!*

*Vi legger vekt på hvor stor lagerplass en gitt representasjon tar, og hvor lang tid en dataoverføring tar.*