
Datamaskinens oppbygning

Håkon Tolsby

Innhold

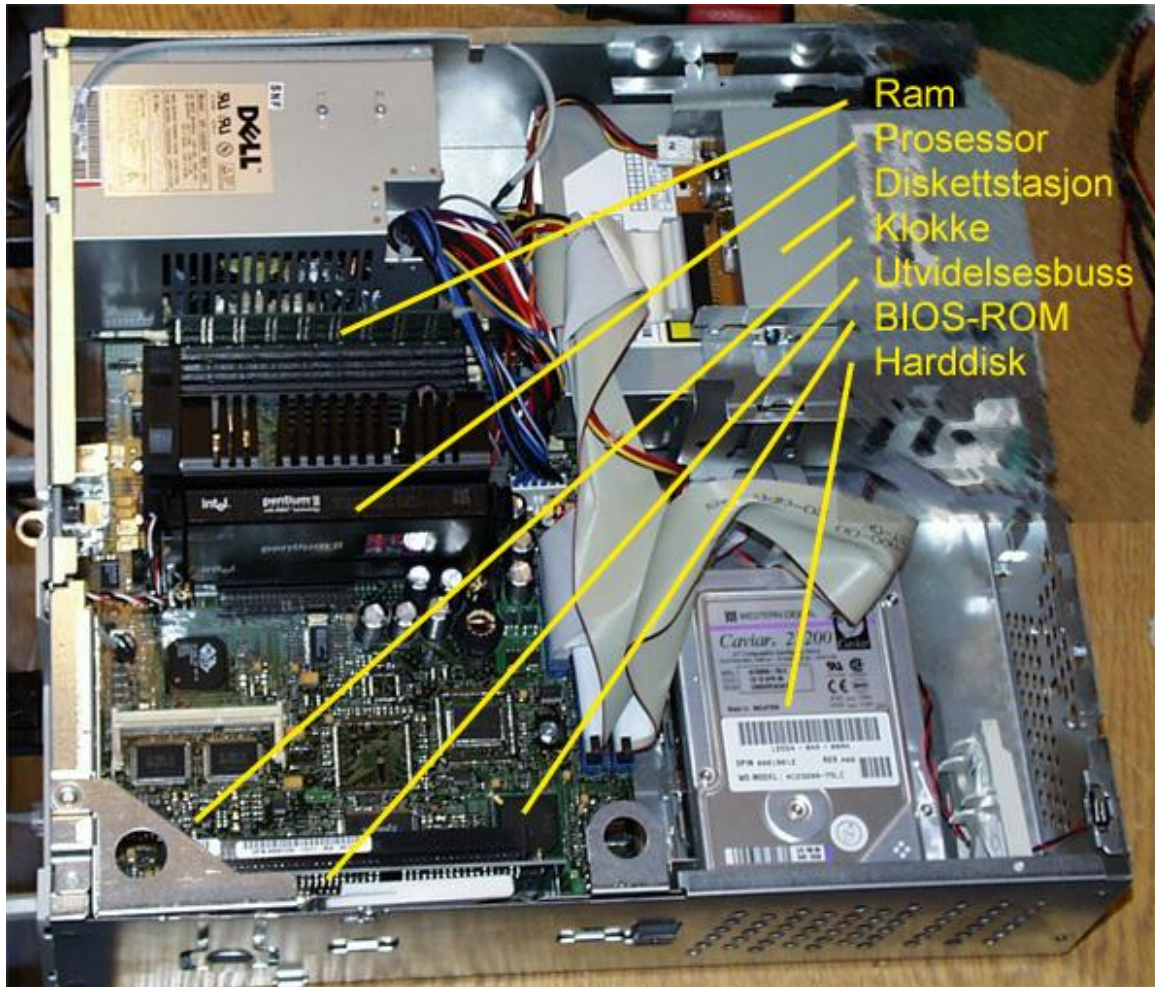
- Hovedenheten
- Hovedkort
- Prosessor
- CISC og RISC

Datamaskinens bestanddeler

- Hovedenhet
- Skjerm
- Tastatur
- Mus



Inni datamaskinen



- Hovedkortet
- Prosessor (CPU)
- Internminne (RAM),
- DVD-spiller,
- Harddisk
- Strømtilførsel
- Diverse utvidelseskort
- Kontakter for tilkoblingen av skjerm, tastatur, mus og andre inn- og ut-enheter (i/u-enheter).

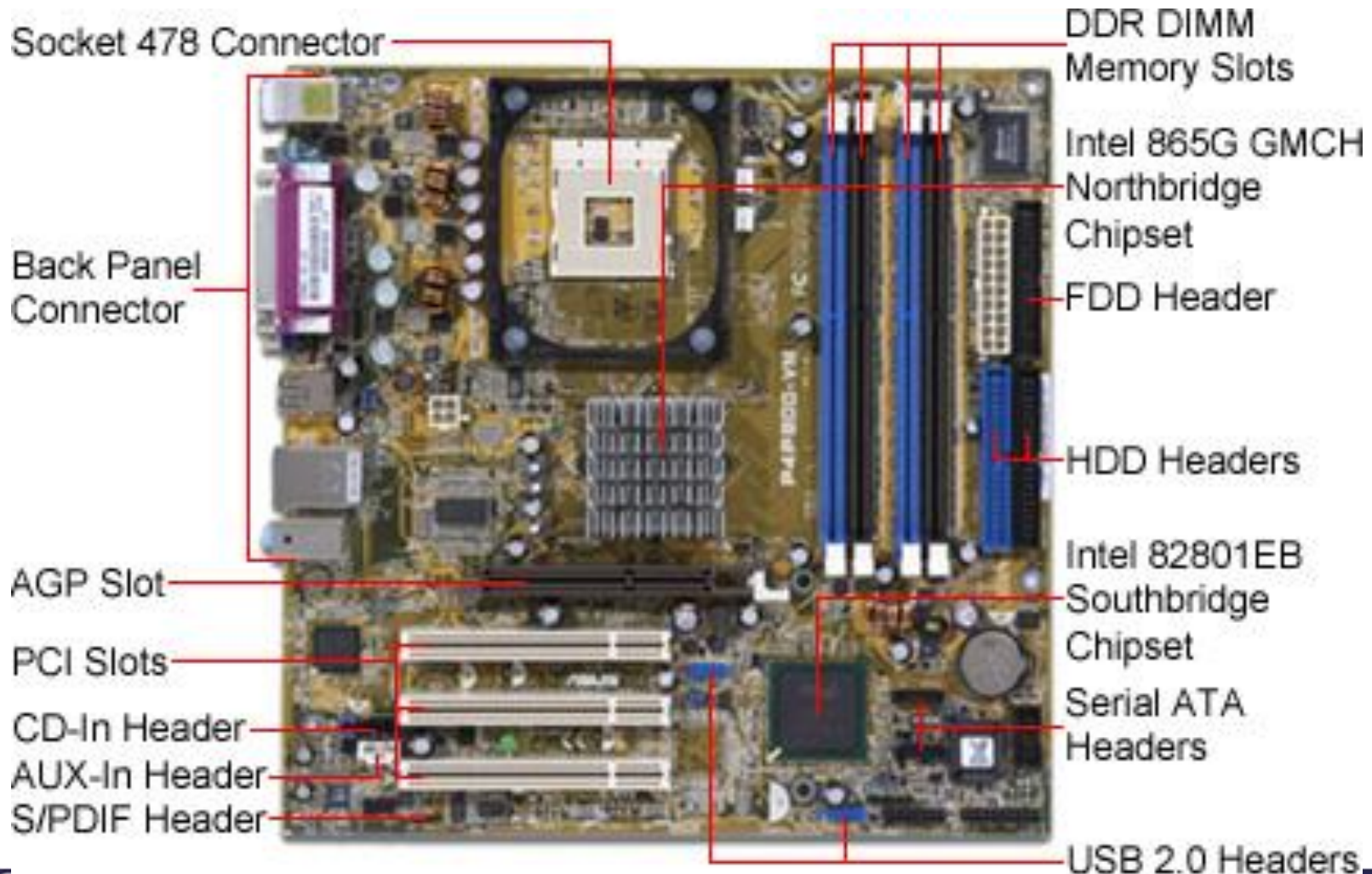
Hovedkort - motherboard

- De viktigste komponentene er samlet på hovedkortet
- Fordeler med å samle komponenter på ett kort er:
 - Mindre avstand mellom komponenter = raskere overføring av data
 - Enklere og rimeligere å produsere datamaskiner.

Utviklingen går mot at flere og flere funksjoner legges til hovedkortet. Håndtering av lyd, grafikk og lokalnett (Ethernet) er eksempler på funksjoner som tidligere lå på egne utvidelseskort, men som nå er integrert i hovedkortet.

- Neste skritt er en komplett PC på en brikke.
- På hovedkortet sitter de to viktigste delene i en datamaskin:
 - Prosessor (engelsk: CPU, Central Processing Unit)
 - Internminne (engelsk: RAM, Random Access Memory og ROM, Read Only Memory)

Hovedkort - Asus P4P800-VM (2003)



Datamaskinens arkitektur

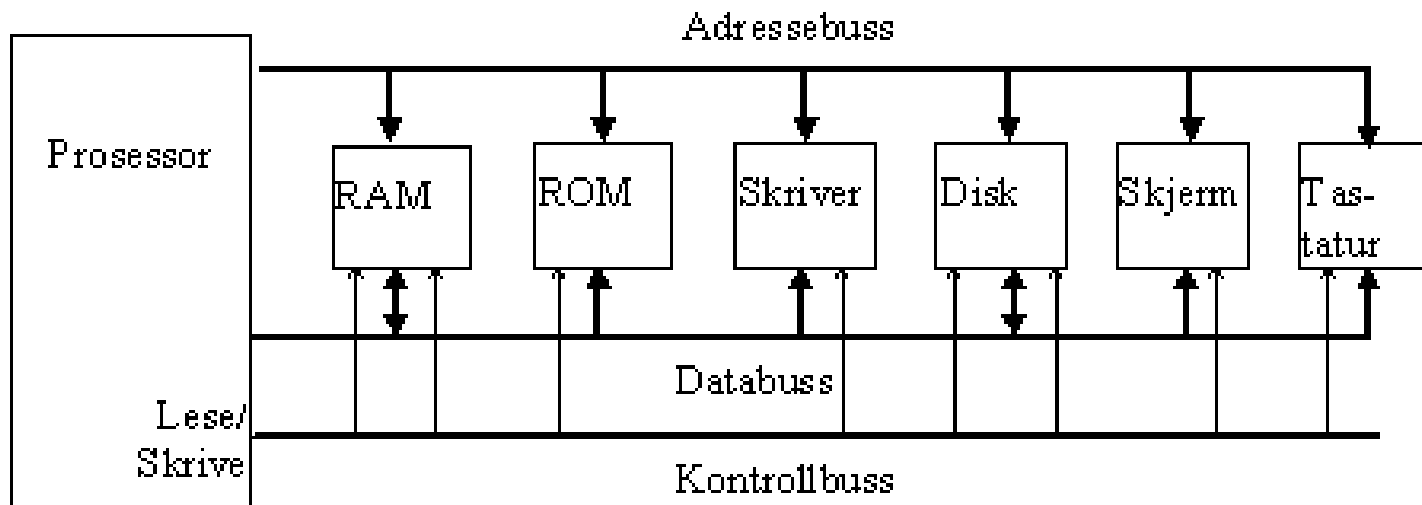
- Skiller mellom minne og prosessor
 - vesentlig trekk ved datamaskinens arkitektur.
Datamaskinen er en generell maskin som kan kjøre ulike programmer.
- Program er kun data som kan lagres sammen med annen data i datamaskinens minne. Og det kan byttes ut etter behov, bare prosessoren kan tolke programmets instruksjoner.
- Denne arkitekturen stod i sterk kontrast til de første datamaskiner hvor programmet var en del av selve datamaskinen. Skulle man omprogrammere datamaskinen, måtte man til dels bygge den om.

Von Neumann arkitektur

- Forskeren, John von Neumann (1903-57), har fått æren av å ha designet den moderne datamaskinen.
- I 1945 presenterte han en rapport som organiserte datamaskinen i fem deler:
 1. Aritmetisk enhet
 2. Kontrollenhet
 3. Hukommelse
 4. Inputenhet
 5. Utputenhet
- Siden har dette vært den rådende modellen for alle datamaskiner.

Bussystemet

- Prosessoren er koblet sammen med internminnet og i/u-enhetene (tastatur, mus, skjerm og disk) ved hjelp av remser med ledninger som er etset inn i hovedkortet.
- Disse kalles for busser
- I alle datamaskiner er det tre busstyper: **adressebuss, databuss og kontrollbuss.**

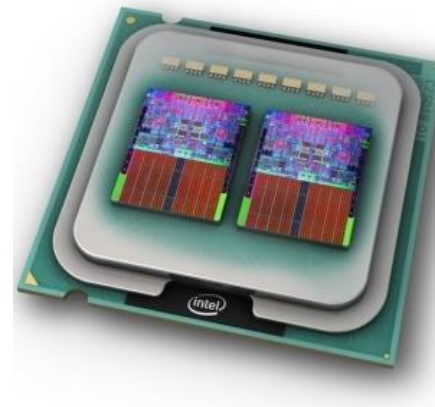


Prosessoren

- Prosessoren (CPU - Central Processing Unit) er selve hjernen i datamaskinen. Den utfører alle beregninger og styrer de andre enhetene.

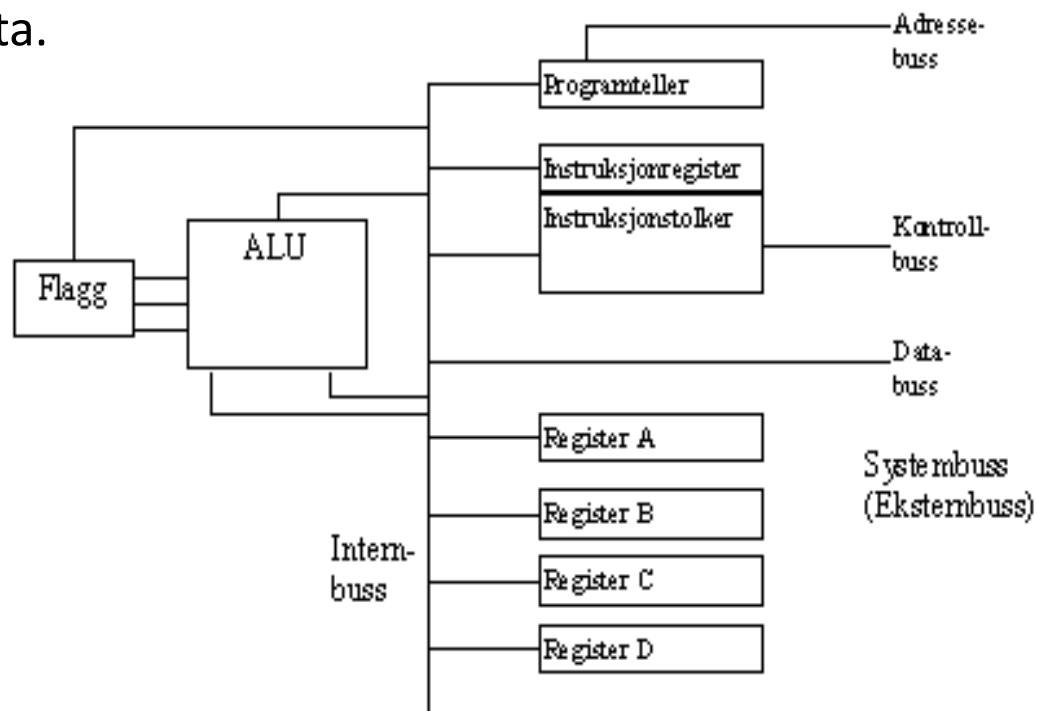


Image courtesy of CPU-Zone.com. Used with permission.



Prosessorens oppbygning

- **ALU**, Aritmetisk Logisk Enhet, som kan utføre enkle aritmetiske (regne) og logiske operasjoner.
- **Registre** som kan mellomlagre data.
- **Internbuss** som kobler registrene og ALU sammen.
- **Programteller** som holder orden på hvilken instruksjon som er den neste som skal hentes fra minnet.
- **Instruksjonstolker** som tolker hva som skal skje når en instruksjon kommer til prosessoren.
- **Flagg** som markerer statussituasjoner for eksempel mente ved addisjon og overfylte registre (engelsk: overflow).

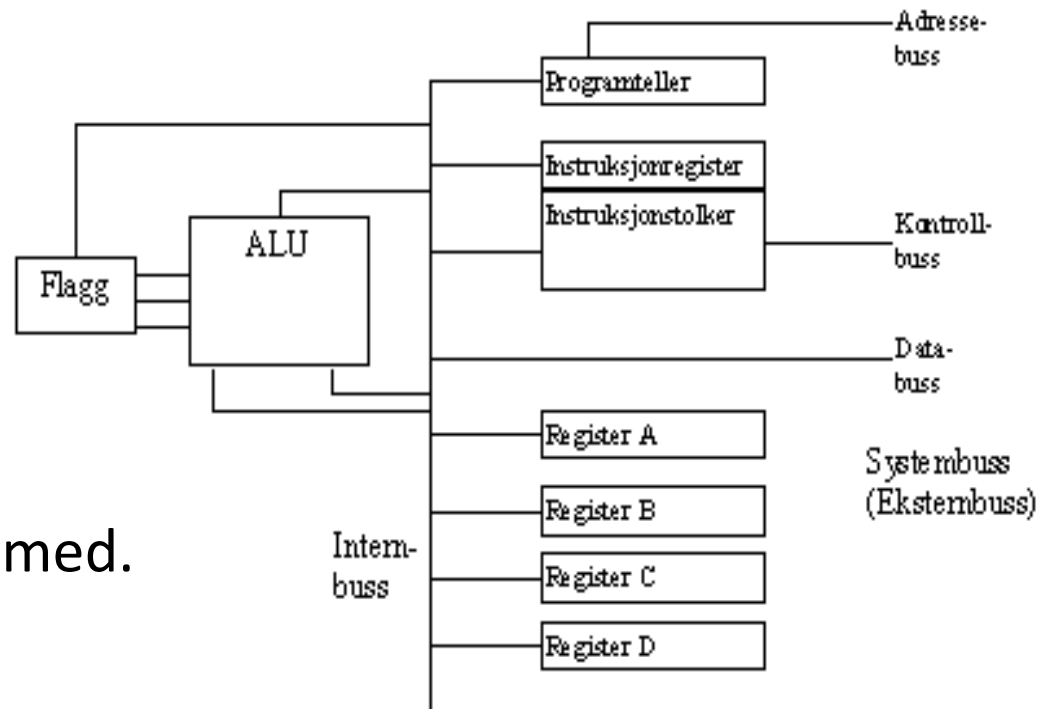


Hvordan prosessoren virker

Prosessoren mottar data fra internminnet gjennom systembussens databuss.

Det er to typer av data:

1. Instruksjoner som prosessoren skal utføre.
 2. Data som den skal arbeide med.
- Resultatet sendes tilbake på systembussen.



Hver instruksjon til prosessoren er egentlig en binær tallkode

Eksempler på instruksjoner for Intel-486			
Kode	Instruksjon	Forklaring	Klokkepulser
0000 0100	ADD	Addisjon	1
0010 0000	AND	Logisk AND	1-3
0011 1101	CMP	Sammenligne	1
1111 0110	DIV	Divisjon	16

Alle programmer består av sekvenser med slike tallkoder. Koding på maskinnivå kalles **Assembly-programmering**. Brukes sjelden.

Von Neumann Machine simulator

The simulator interface includes a menu bar with 'Project', 'Guide', 'Info', 'Samples', and 'Settings'. A toolbar at the top right shows playback controls and a 500 ms timer. The main diagram area contains a 'DECODER' block, an adder block labeled '+ 1', a register block labeled '5', and a memory cell block labeled '4'. The right-hand panel is divided into two sections: 'Memory cells' and 'Variables'.

Memory cells		Variables	
0	// X + Y = Z	X	2
1	LOD X	Y	2
2	ADD Y	Z	4
3	STO Z	W	0
4	HLT		+

Made by Lorenzo Ganni

<http://vnsimulator.altervista.org/>

Hva skjer når vi starter et program?

1. Programtelleren blir nullstilt og peker til den første instruksjonen i programmet som ligger i internhukommelsen RAM.
2. Prosessoren leser instruksjonen som programtelleren referer til og overfører den til instruksjonsregisteret.
3. Instruksjonstolkeren tolker instruksjonen og utfører den.
4. Når instruksjonen er ferdig utført, øker programtelleren med 1 og neste instruksjon blir lest (punkt 2)

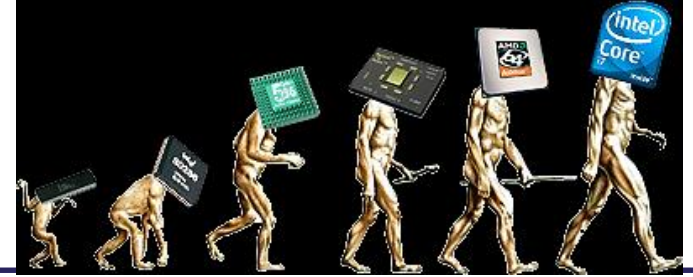
Dette gjentar seg til prosessoren ikke har flere instruksjoner å utføre og programmet avslutter.

Klokkefrekvensen



- Prosessoren kan utfører enkle operasjoner med ekstrem høy hastighet.
- På hovedkortet sitter en liten brikke (krystall) som genererer elektriske pulser. Den kalles datamaskinens klokke og styrer arbeidstakten til alle komponenter i datamaskinen.
- For hver elektrisk puls den sender ut, kan prosessoren utføre en instruksjon.
- Klokkepulsene sørger for at datamaskinen arbeider etter en jevn takt slik at data som fraktes på bussene ikke kolliderer.
- En prosessor med en klokkefrekvens på 600 MHz kan utføre 600 millioner operasjoner i sekundet.
- Problemet med høye klokkefrekvensene er at resten av elektronikken i datamaskinen ikke henger med. De klarer ikke å motta og sende data i et så høyt tempo. Derfor operer man ofte med to klokkefrekvenser, en høy intern klokkefrekvens for prosessoren og en lavere ekstern klokkefrekvens for kommunikasjon mot systembussen og internminnet.

Prosessorens utvikling



år	Prossessor	Intern klokke	Ekstern klokke	Registre	Databuss	Kjerner
1971	4004	0,74 MHz	0,74 MHz	4	4	1
1978	8086	4,77 MHz	4,77 MHz	16	16	1
1982	80286	12 MHz	12 MHz	16	16	1
1985	80386	25 MHz	25 MHz	32	32	1
1989	80486	66 MHz	33 MHz	32	32	1
1993	Pentium 100	100 MHz	66 MHz	32	64	1
1997	Pentium II	333 MHz	66 MHz	32	64	1
1999	Pentium III	5-800 MHz	133 MHz	128	64	1
2000	Pentium 4	1,3 - 2 GHz	400 MHz	128	64	1
2002	Pentium 4	4 GHz		128	64	1
2005	Nehalem	10 Ghz		128	64	1
2006	Core 2 Duo	1,2 -2,33 GHz		128	64	2
2007	Core 2 Quad	1,066 Ghz		128	64	4
2008	Core i7	2,2 – 3,2 Ghz		128	64	4

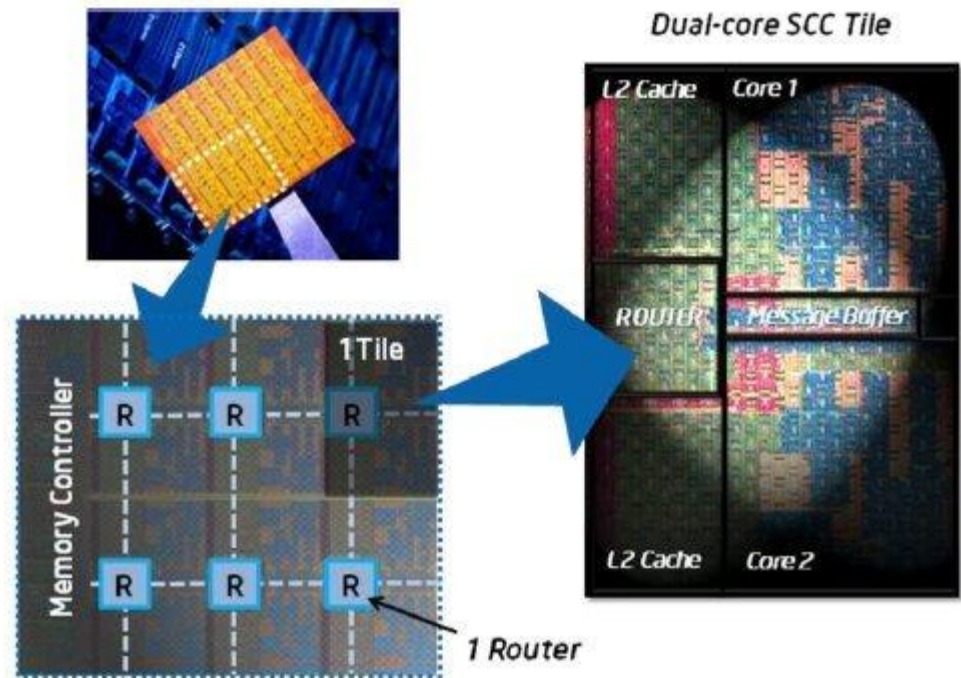
Hva avgjør hastigheten?

- Klokkefrekvensen
- Bredden på databussen
- Størrelsen på registrene
 - Store registre betyr at prosessoren kan utføre operasjoner med store tall uten å måtte dele dem opp, eller at den kan legge sammen flere tall i parallell. Størrelsen på databussen forteller oss hvor mye data som kan fraktes til prosessoren om gangen.
- Antall kjerner 2 og 4, slik at prosessoren kan utføre flere oppgaver samtidig.
- Hyper threading (logiske kjerner x2, x4)
- Instruction-level parallelism (3-5 parallelle instruksjoner)

Mange kjerner

- Windows 10 kan håndtere 256 kjerner (og to prosessorer)

Intel har utviklet en 48 kjerners prosessor med 1,3 milliarder transistorer som de kaller "Single-Chip Cloud Computer". Det er 48 kjerner på en brikke som er knyttet sammen i et eget nettverk med 24 rutere.



Anatomy of the Single-chip Cloud Computer

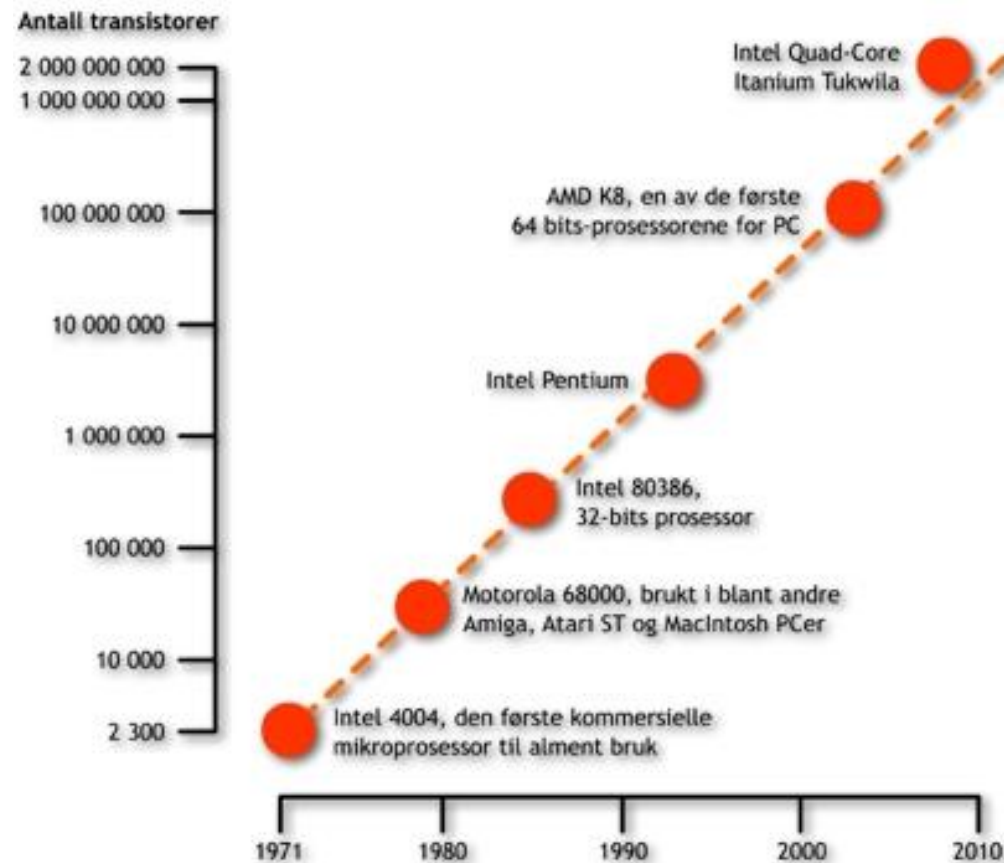
Moore's law

- Antall transistorer i en prosessor vil doble seg hvert andre år.

Moore's law ble fremsatt i 1965 av Gordon Moore som var en av Intels grunnleggere.

Fra 1971 til 2010 har antallet transistorer i mikroprosessen økt fra ca. 2300 til 2 milliarder.

(ca 20 fordoblinger på 40 år)



Hva er grensen???

2012: 220 atomer per transistorer

2020: 50 atomer per transistor

Kan man lage transistorer på 1 atom?

Moore's law justert

- Antall kjerner på prosessorene vil dobles hvert andre år.
- Men hvor mange kjerner trenger man?

CISC og RISC

-to hovedtyper av prosessorer

- **CISC (Complex Instruction Set Computer)**
 - CISC-prosessoren ble utviklet i en tid da RAM var dyrt å kjøpe. Da var det viktig at programmene var små, og at prosessoren gjorde mest mulig for hver instruksjon den fikk. Derfor lagde man prosessorer med mikroprogrammer. En CISC prosessor kan tilby et stort sett av instruksjoner, og mange av instruksjonene er komplekse og trenger flere klokkepulser for å bli utført
 - Ulemper: Dyre å konstruere, flere instruksjoner tar lang tid å utføre, og over 60% av alle transistorene brukes til å lagre mikroprogrammene og mange av de komplekse instruksjonene i brukes aldri eller sjelden.
- **RISC (Reduced Instruction Set Computer)**
 - Redusere antall instruksjoner til et minimum.
 - 95% av alle instruksjonene kan gjennomføres innenfor en klokkepuls
 - Sparer mange transistorer på ikke å ha mikroprogrammer. Dermed kan man legge flere funksjoner i prosessoren, slik som: Grafisk prosessor, mer cache osv.
 - Større programmer fordi noen oppgaver må løses i programkoden i stedet for av prosessoren direkte.
- I dag er de fleste (alle) prosessorer konstruert etter RISC-prinsippet