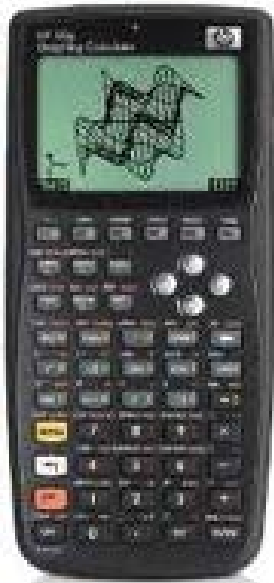


Översikt, kursinnehåll

- Specifikation av digitala funktioner och system
- Digitala byggelement
- Kombinatoriska system
- Digital Aritmetik
- Synkrona system och tillståndsmaskiner
- Asynkrona system och tillståndsmaskiner
- Lite större digitala system – om processorn och datorer
 - VHDL ingår inte i någon större utsträckning – eftersom det är en hel kurs i sig.*

Digital Design IE1204

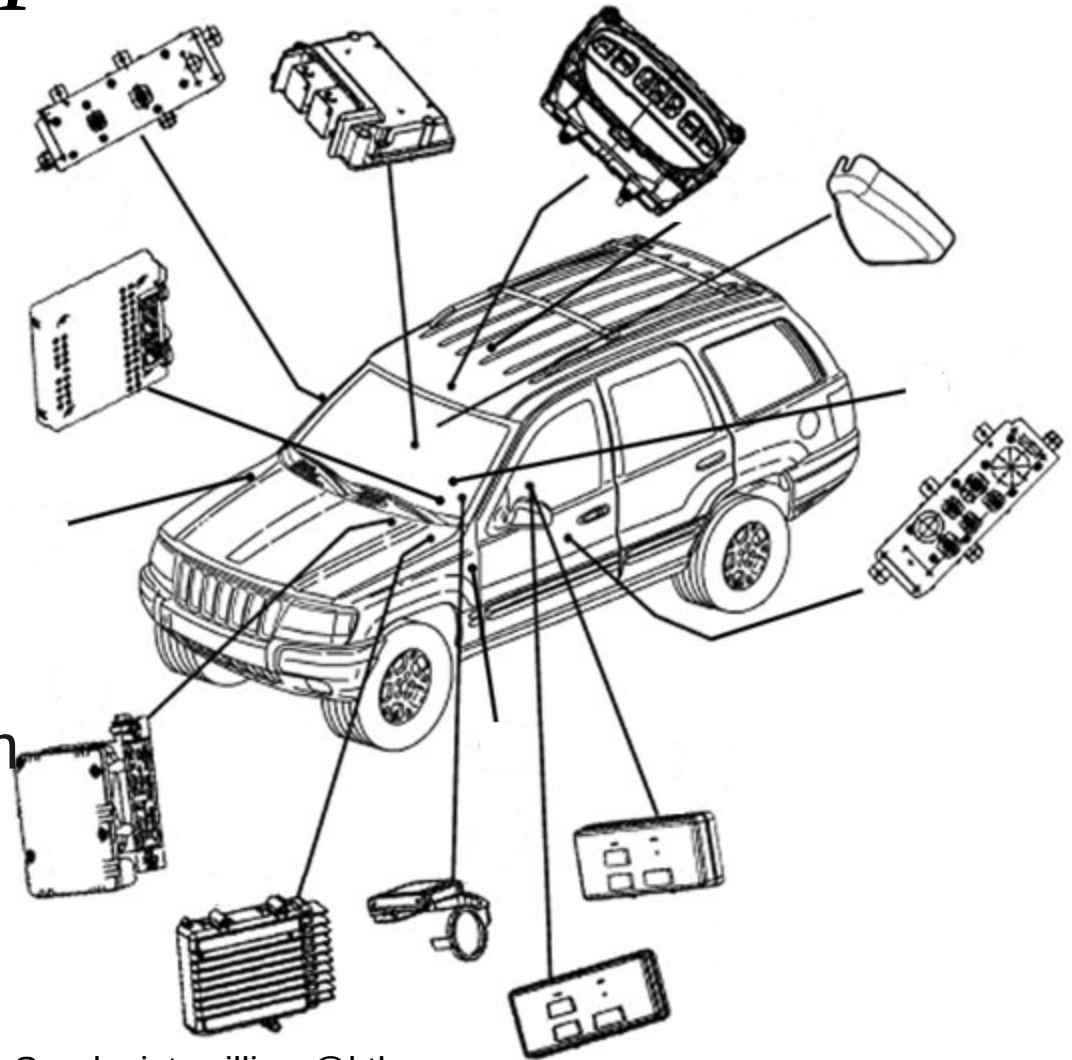


Digital Design
Överallt!



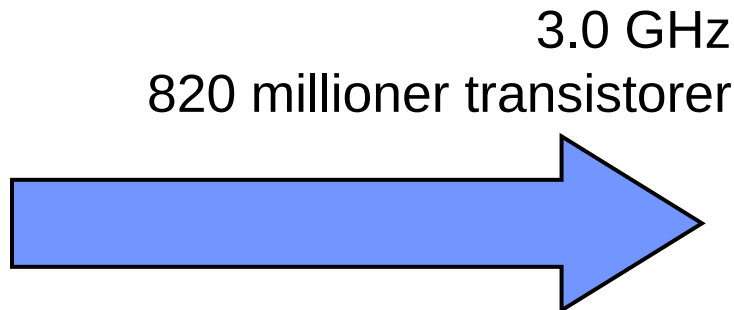
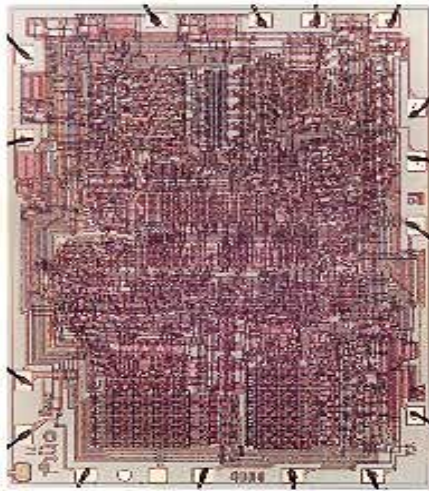
40-100 mikroprocessorer i en bil!

- Ignition system
- Emission control system
- Anti-lock brakes
- Dashboard display
- Entertainment system
- Navigation system
- ...



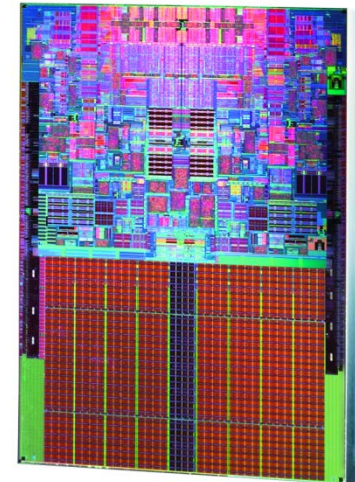
Utvecklingen av elektroniken

Intel 4004
(1971)



108 KHz
2,300 transistorer

Intel Xeon 5400
(2008)



Om man hade haft motsvarande utveckling för **bilhastigheten** så skulle man nu kunna köra från **San Francisco** till **New York** på ca 13 sekunder (Intel).

Varför är digitaltekniken så framgångsrik?

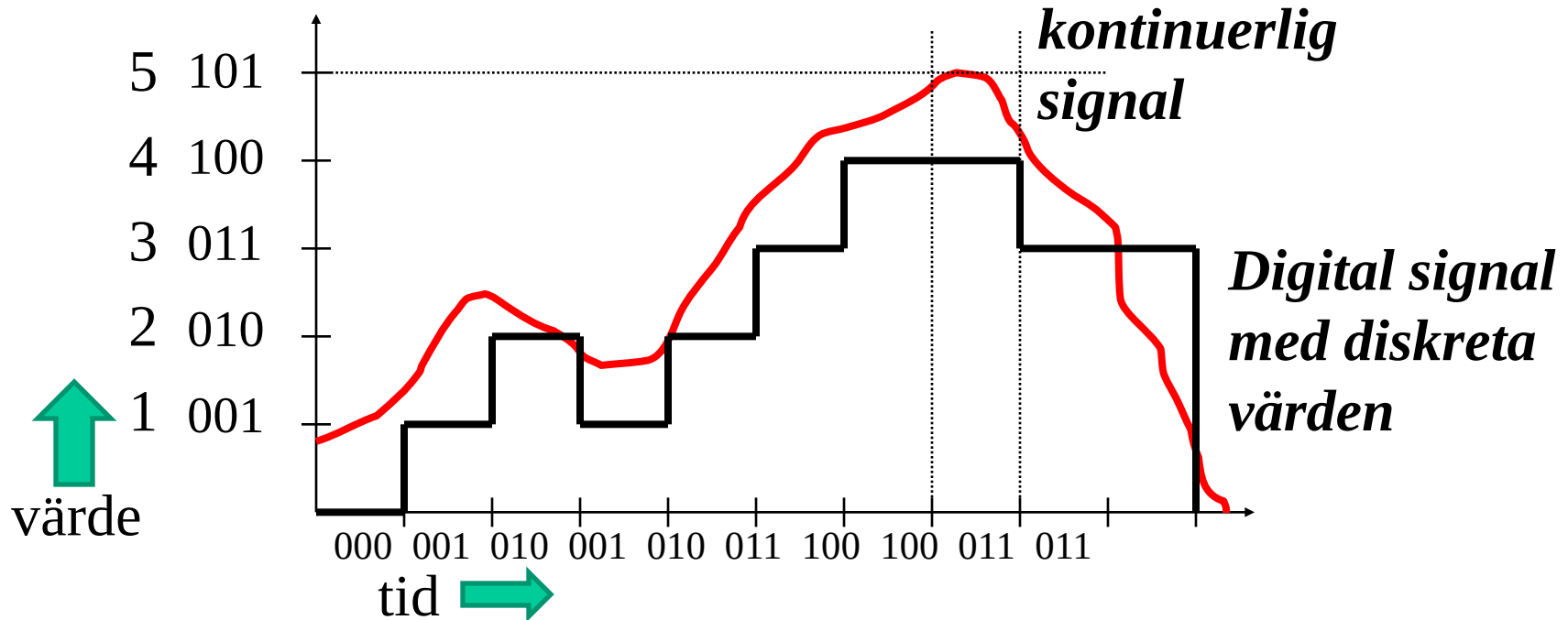


Enkelhet, störningssäkerhet

- **Enkel** matematisk modell
 - med bara 1:or och 0:or som värden
 - Boolesk algebra
- **Störningsokänslig, effektiv implementering** av den matematiska modellen
 - Transistorer – Integrerade kretsar
 - Framsteg i halvledartechnologin
- Effektiva **designmetoder** och **verktyg**

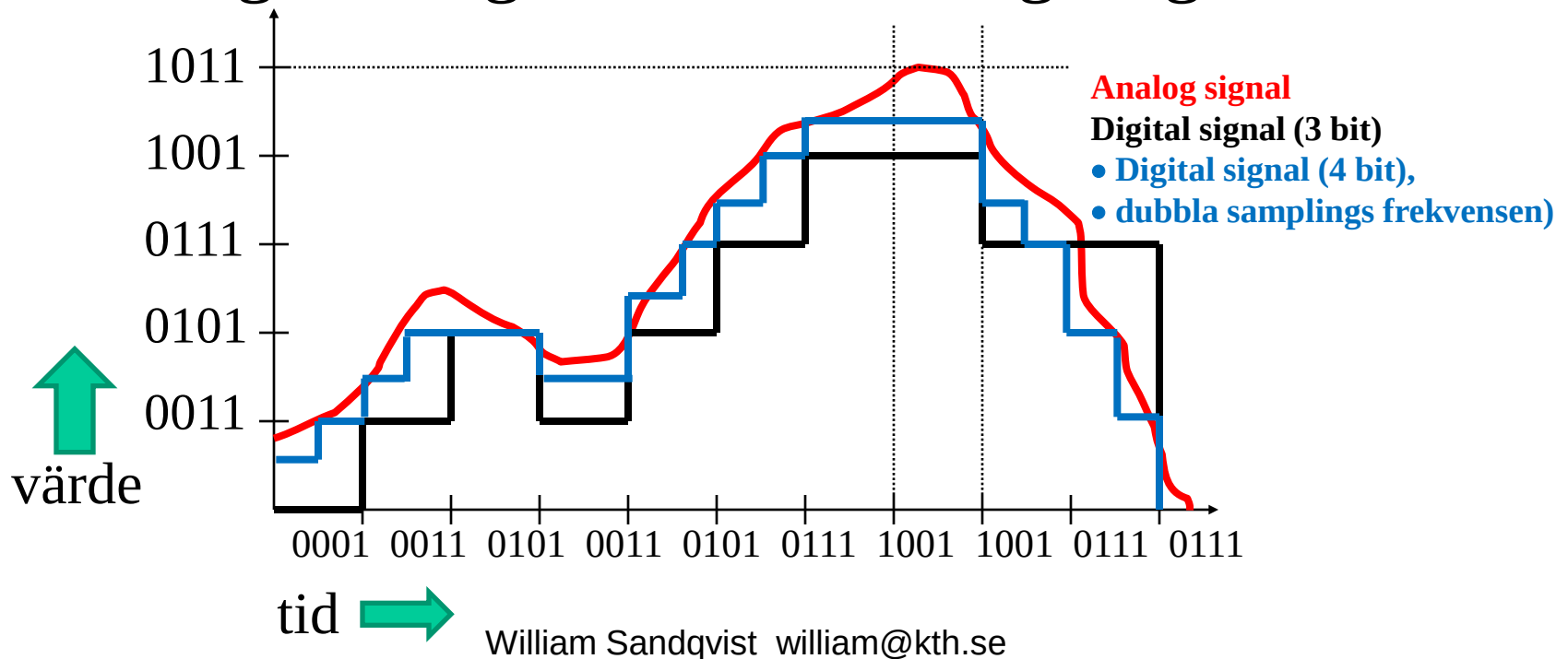
Analoga eller Digitala signaler?

En analog signal kan anta *kontinuerliga* värden, medan en digital signal bara kan anta *diskreta* värden (här 0...5)

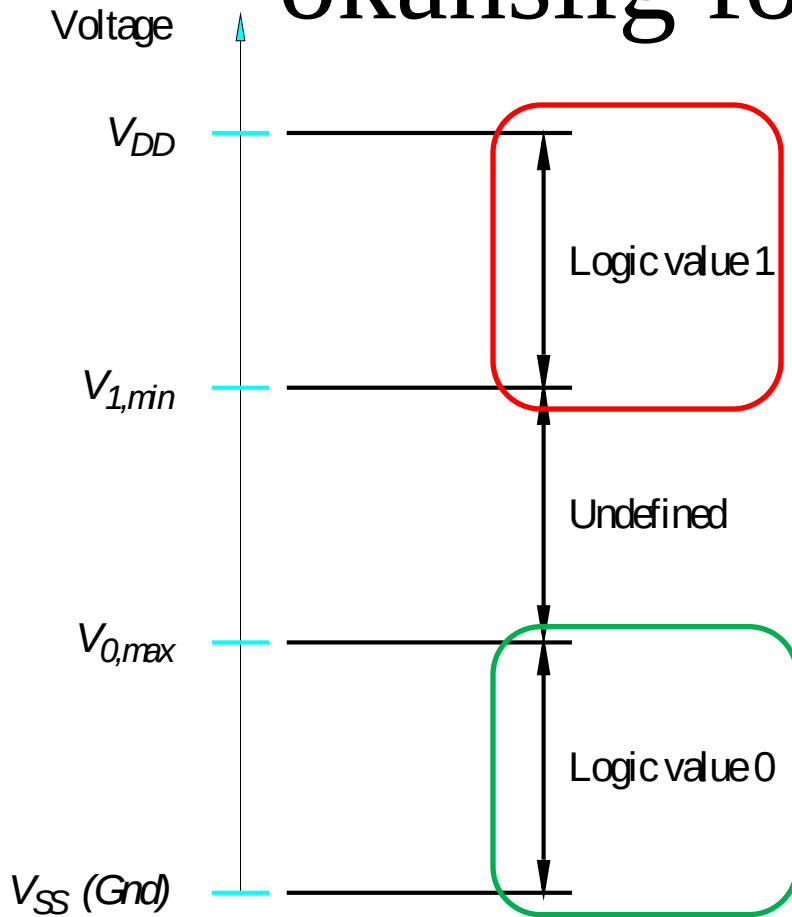


Digitalt borde vara sämre?

Men har man tillräckligt *många* bitar och tillräckligt *hög* samplingsfrekvens *efterliknar* den digitala signalen den analoga signalen

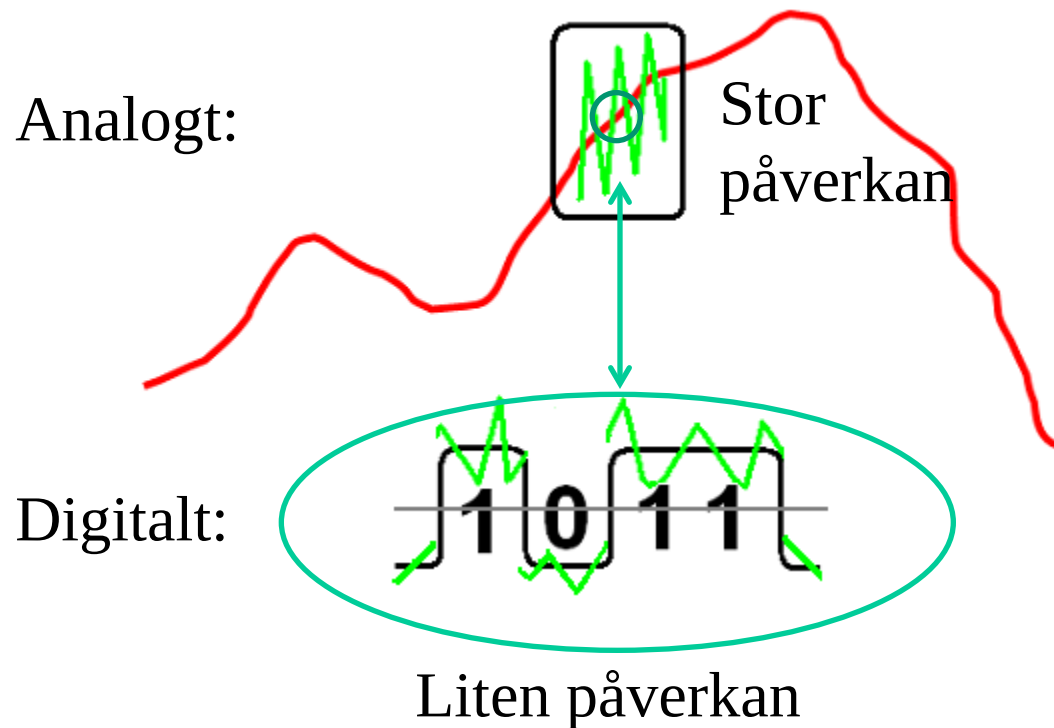


Digitaltekniken är mycket okänslig för störningar



- Det är inte bara ett spänningsvärde som översätts som 1 eller 0 utan ett helt spänningsområde
- En störning om några mV kan påverka värdet på en analog signal mycket, men gör ingen skillnad inom digitaltekniken

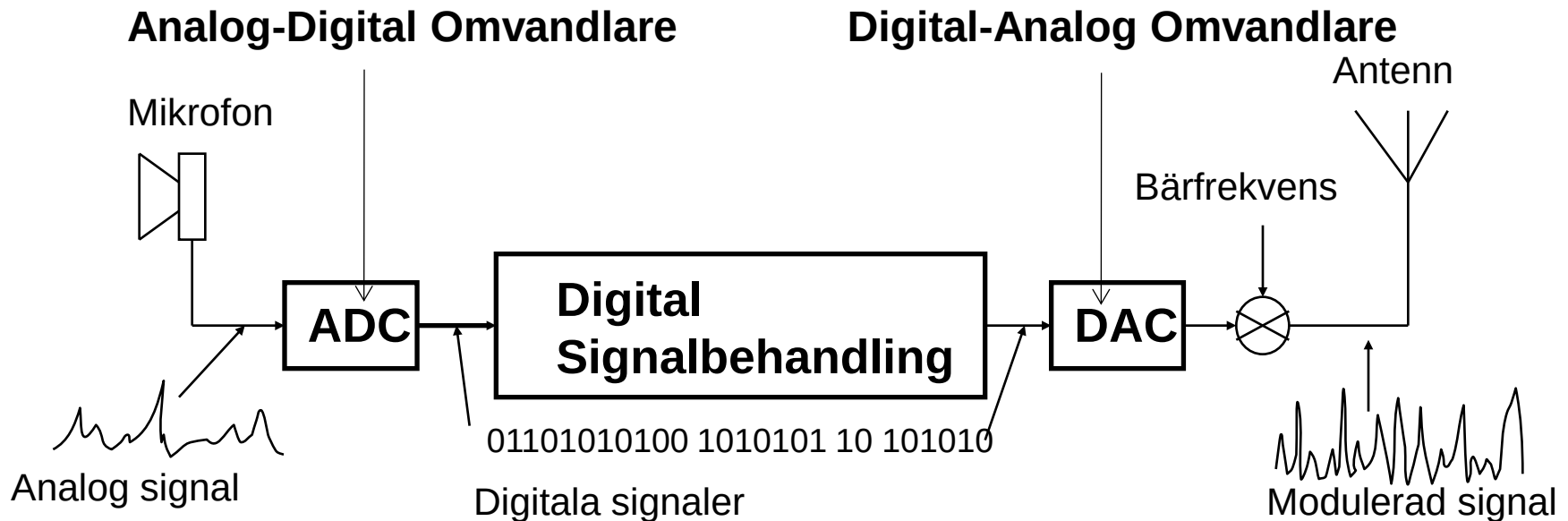
Störningsokänslighet



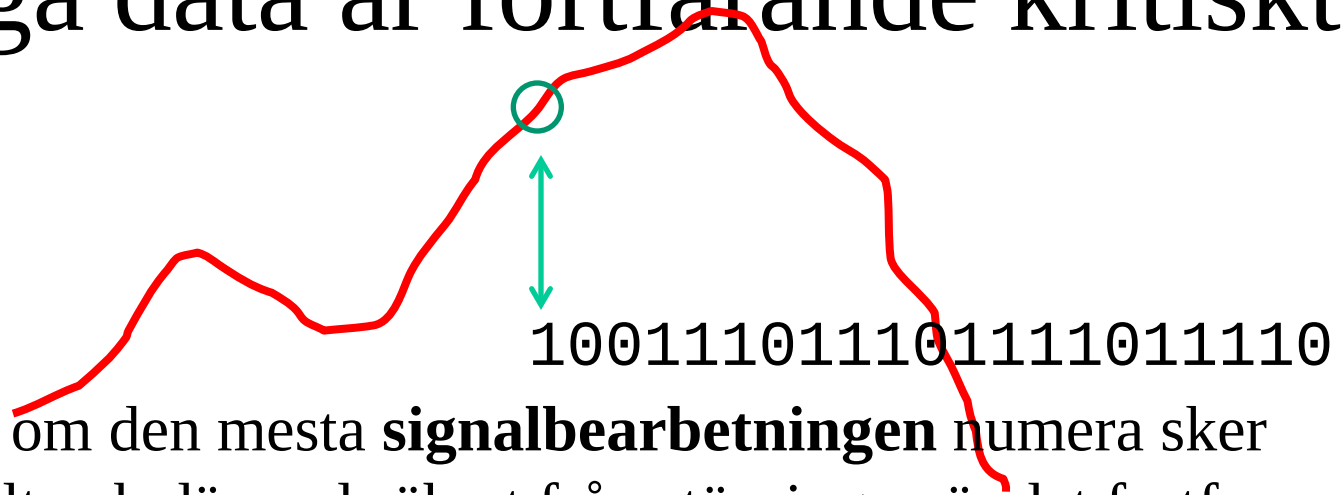
- Digital **databearbetning** kan ske störningsokänsligt!

Digital signalbehandling

Om det är möjligt så görs idag **all** signalbehandling digitalt



Fånga data är fortfarande kritiskt!



Även om den mesta **signalbearbetningen** numera sker digitalt och därmed säkert från störningar, är det fortfarande kritiskt att ”fånga” data på ett bra sätt – analog/digital-omvandlingen!

De ”störningar” som följer med in i det digitala systemet får man sedan leva med!

Matematisk modell - Boolesk algebra: axiom

- I boolesk algebra finns det bara **1** (sann) och **0** (falsk) som värden
- Följande operationer är definierade: **AND** ($\cdot$), **OR** ($+$), **NOT** (x)
- Följande axiom definierar den booleska algebran

Axiom	
(1a) $0 \cdot 0 = 0$	(1b) $1 + 1 = 1$
(2a) $1 \cdot 1 = 1$	(2b) $0 + 0 = 0$
(3a) $0 \cdot 1 = 1 \cdot 0 = 0$	(3b) $1 + 0 = 0 + 1 = 1$
(4a) If $x = 0$, then $\bar{x} = 1$	(4b) If $x = 1$, then $\bar{x} = 0$

Booleska Algebra: **räknelagar**

Räknelagar kan härledas ur axiomen

- många av räknelagarna stämmer överens med våra vanliga algebra!
- bekvämt att man kan fortsätta att räkna som man gjort i grundskolan!
- men se upp! En del räknelagar blir anorlunda och nya.

• *Mer snart i kursen ...*

Den tekniska bakgrunden

När **telefonväxlarna** automatiserades användes den booleska algebran som ett verktyg för beräkna hur kontaktnäten kunde förenklas.

Kontakterna ritades i opåverkat tillstånd.

Slutande kontakt

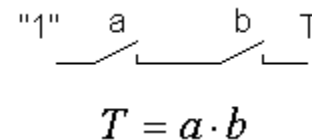
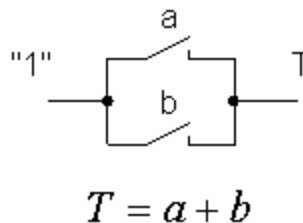


Brytande kontakt, "icke"

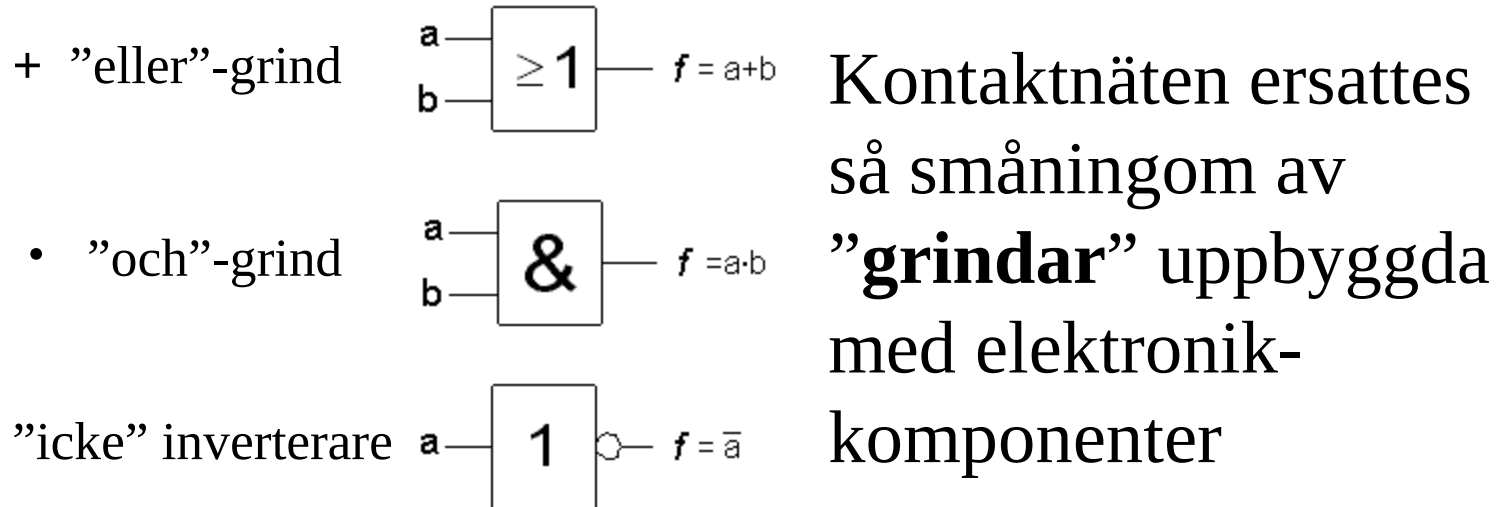


• för seriekoppling
"och"-funktion

+ för parallellkoppling
"eller"-funktion



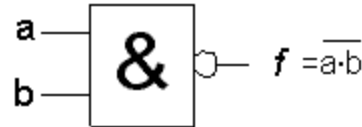
Grindar i stället för kontakter



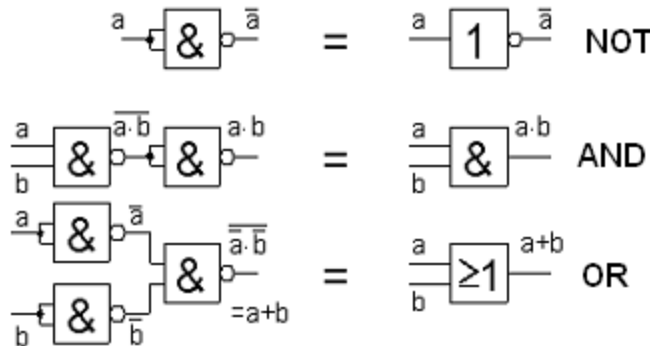
Med dessa tre grundläggande grindtyper: **OR**
AND NOT kan **alla** logiska funktioner utföras.

• *Mer snart i kursen ...*

Bara en enda grindtyp!



Det räcker faktiskt med **en enda** grindtyp, **NAND**, för att tillverka alla andra!

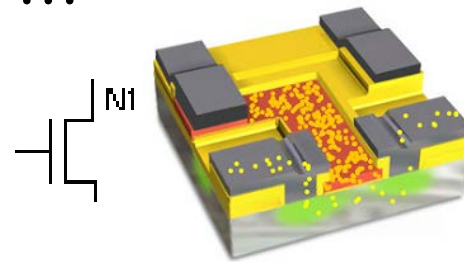
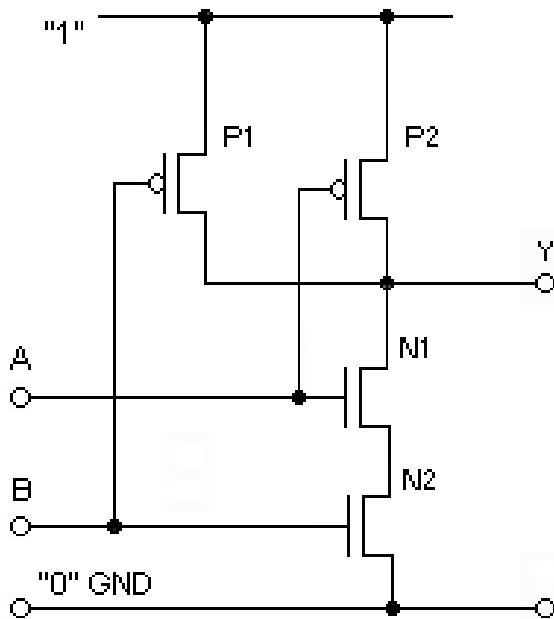


*Extrem förenkling,
en enda grindtyp
räcker till **allt** !*

• *Mer snart i kursen ...*

CMOS NAND - grind

En **effektiv** implementering av CMOS NAND-grinden kräver bara ***fyra transistorer*** ...



En N-MOS transistor.

Finess: CMOS-grindar förbrukar bara effekt vid omslag.

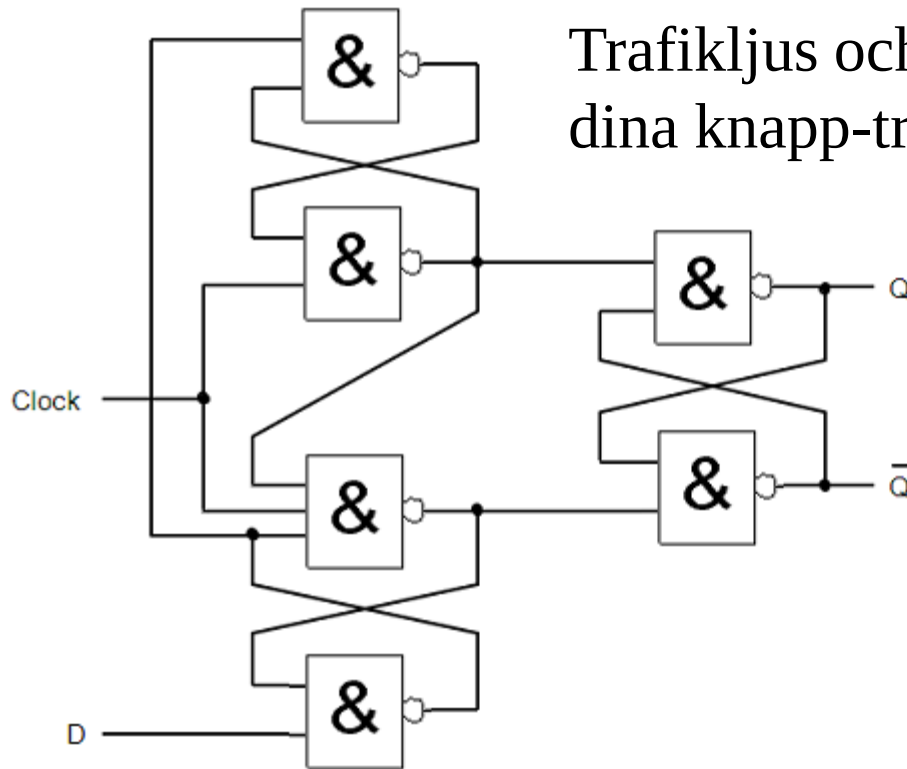
Tur det, annars skulle dagens datorkretsar bli ***glödheta***.

• ***Mer snart i kursen ...***

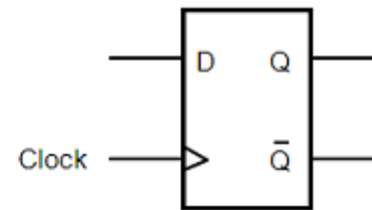
Minnesfunktion?

Trafikljus och hissar måste ”**minnas**” dina knapp-tryckningar.

Detta utförs av ”vippor” eller ”låskretsar” – som man också kan göra med NAND-grindar.



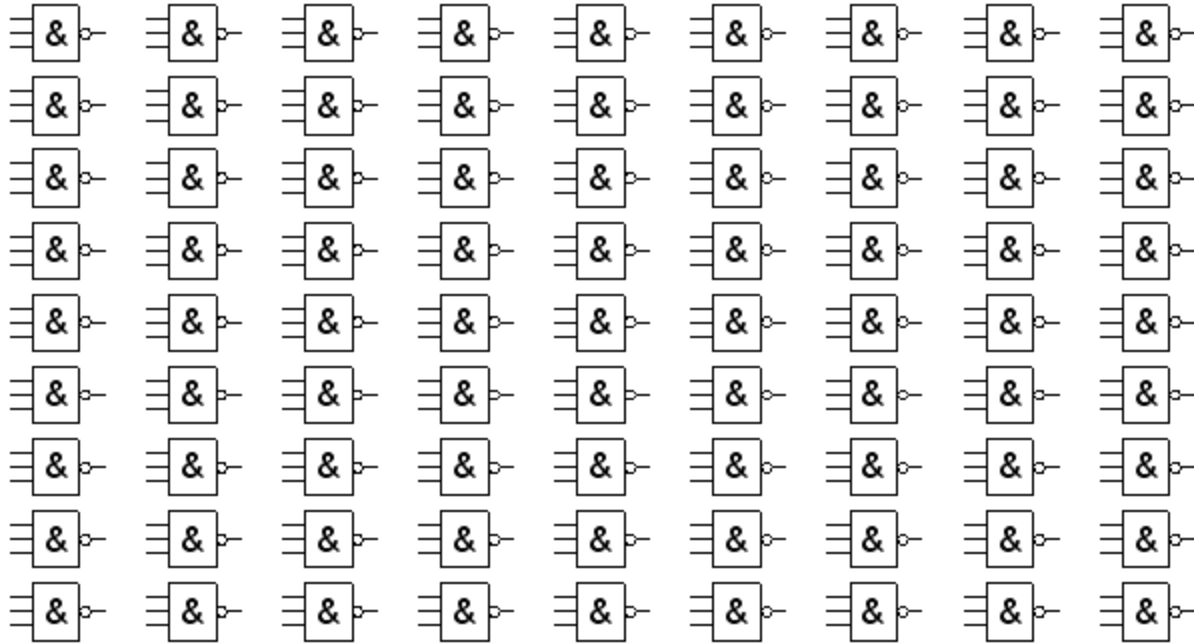
(a) Circuit



(b) Graphical symbol

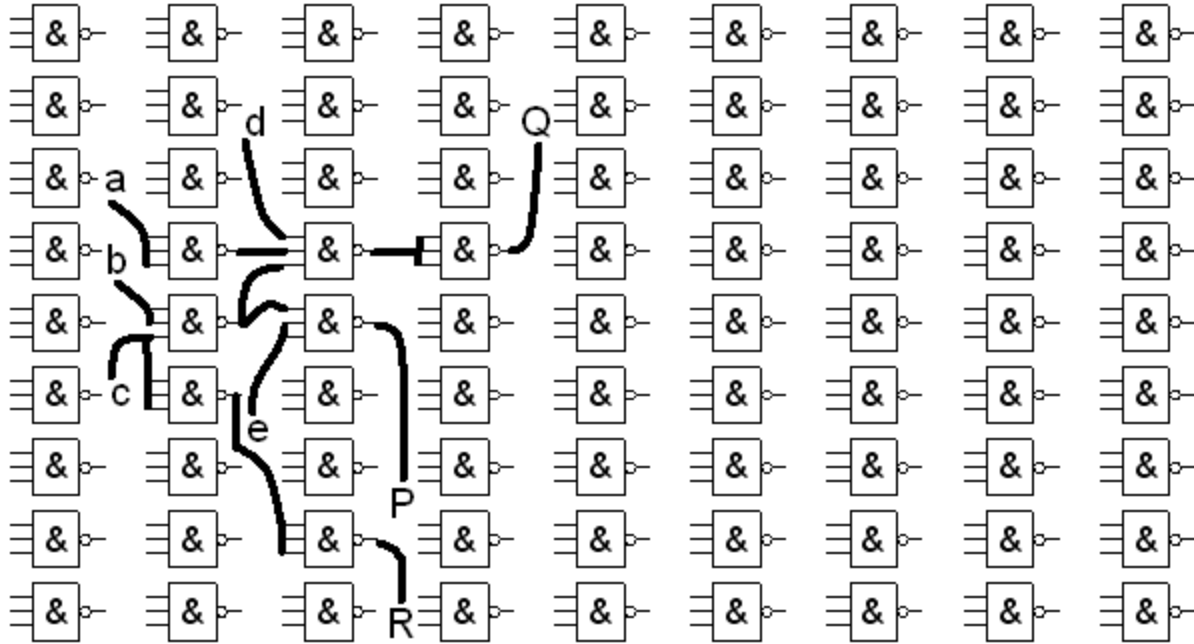
• *Senare i kursen ...*

See-of-Gates



Eftersom bara en enda typ av grindar behövs, kan man göra vad som helst med ”ett hav av grindar” som får ett lednings-mönster utformat för en egen specifik funktion.

See-of-Gates

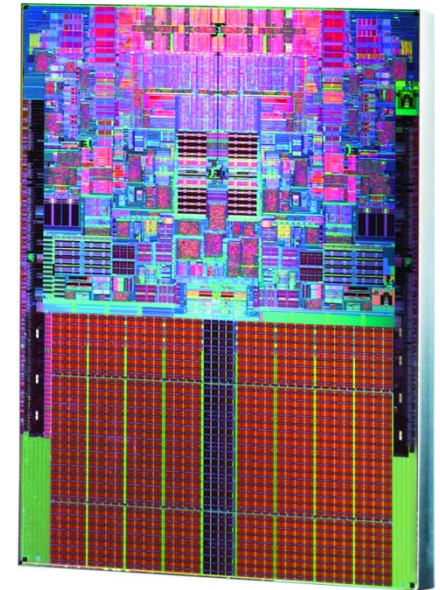


Eftersom bara en enda typ av grindar behövs, kan man göra vad som helst med "ett hav av grindar" som får ett lednings-mönster utformat för en egen specifik funktion.

En miljard grindar?

En modern processor kan innehålla en miljard grindar – man kan inte rita ett sådant kretsschema ...

Det behövs andra metoder för att beskriva ett sådant system!

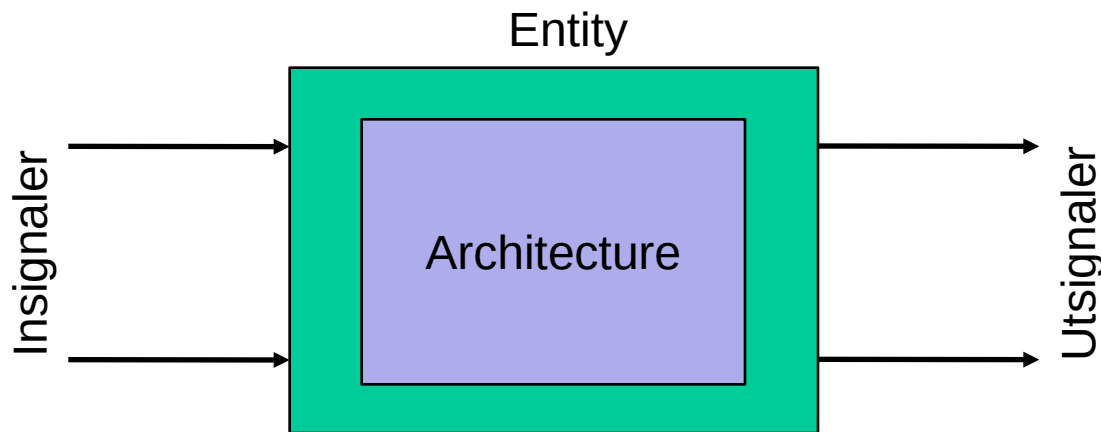


CAD-verktyg

- Ett CAD-verktyg är ett program som hjälper ingenjören att konstruera (tex en integrerad krets)
- CAD-verktyg kan vara helt automatiserade eller interaktiva
- CAD-verktyg är baserade på algoritmer som definierar ordningen på en sekvens av metoder som skall appliceras

Hårdvarubeskrivande språk

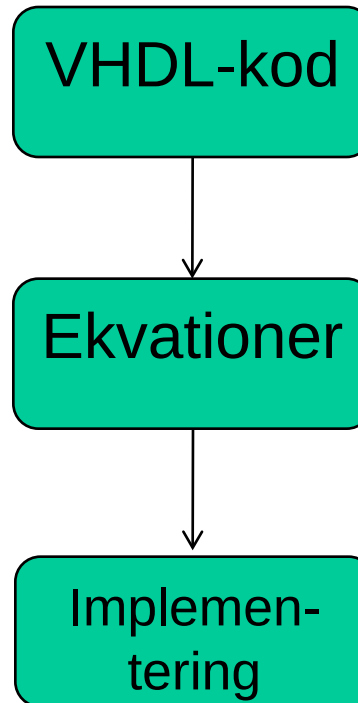
Designen beskrivs med ***entity*** (black box) och ***architecture*** (innehållet i boxen)



Design-verktyg

Steg 1: VHDL-Beskrivningen av den önskade hårdvaran översätts till booleska ekvationer

Steg 2: Booleska ekvationer översätts till den tillgängliga hårdvaran



I kursen får Du prova designverktyget
Quartus II

Ex. på VHDL-kod

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY mux4to1 IS
    PORT( w0, w1, w2, w3 : IN STD_LOGIC ;
          s                : IN STD_LOGIC_VECTOR(1 DOWNT0 0) ;
          f                : OUT STD_LOGIC ) ;
END mux4to1 ;

ARCHITECTURE Behavior OF mux4to1 IS
BEGIN
    WITH s SELECT
        f <= w0 WHEN "00",
              w1 WHEN "01",
              w2 WHEN "10",
              w3 WHEN OTHERS ;
END Behavior ;
```

*VHDL är ett mycket
komplext språk – kursen kan
inte rymma så mycket av
detta.*

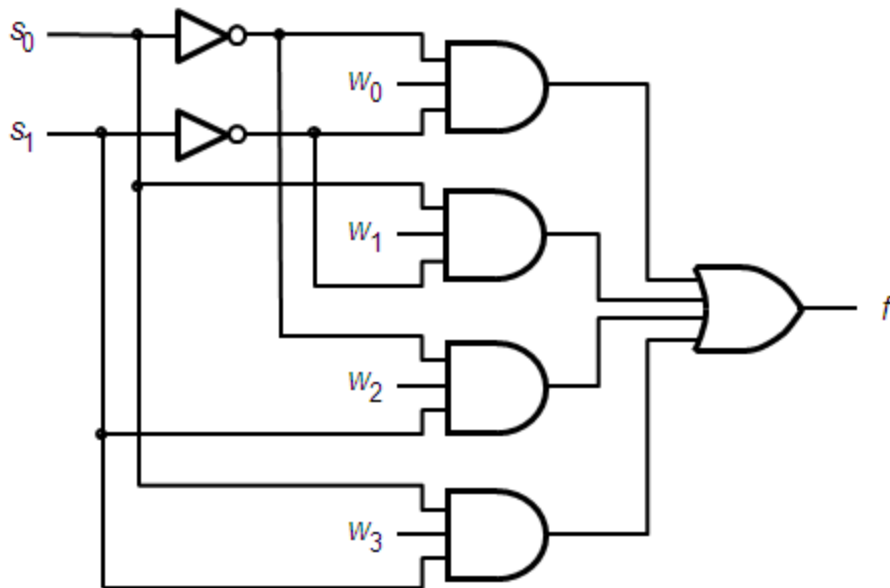
Fortsättningskurser finns!

Boolesk ekvation

$$f = \overline{s_1} \cdot \overline{s_0} \cdot w_0 + \overline{s_1} \cdot s_0 \cdot w_1 + s_1 \cdot \overline{s_0} \cdot w_2 + s_1 \cdot s_0 \cdot w_3$$

- *Mer snart i kursen ...*

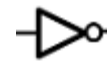
Ett grindnät som implementerar funktionen



*Läroboken använder
Amerikanska
symboler:*

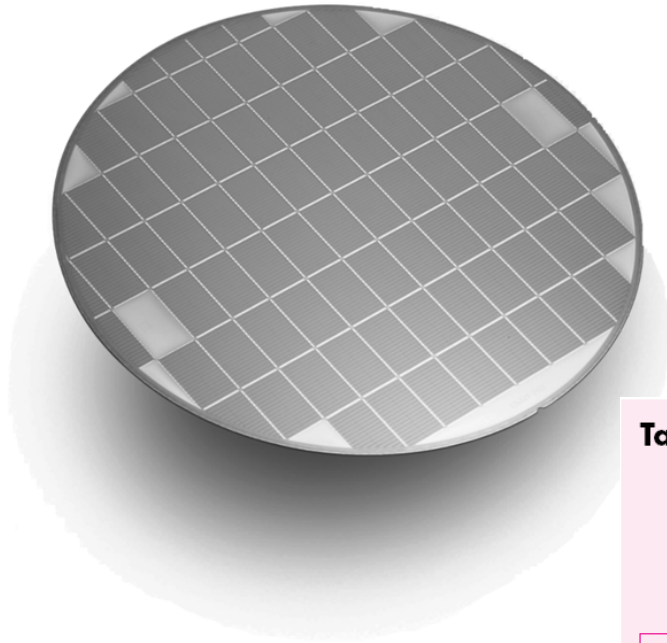
AND 

OR 

NOT 

• *Mer snart i kursen ...*

Teknologiframsteg



Utvecklingen går rasande fort...

Table 1.1

A sample of the International Technology Roadmap for Semiconductors.

	Year					
	2006	2007	2008	2009	2010	2012
Technology feature size	78 nm	68 nm	59 nm	52 nm	45 nm	36 nm
Transistors per cm ²	283 M	357 M	449 M	566 M	714 M	1,133 M
Transistors per chip	2,430 M	3,061 M	3,857 M	4,859 M	6,122 M	9,718 M

Digital hårdvara i en dator

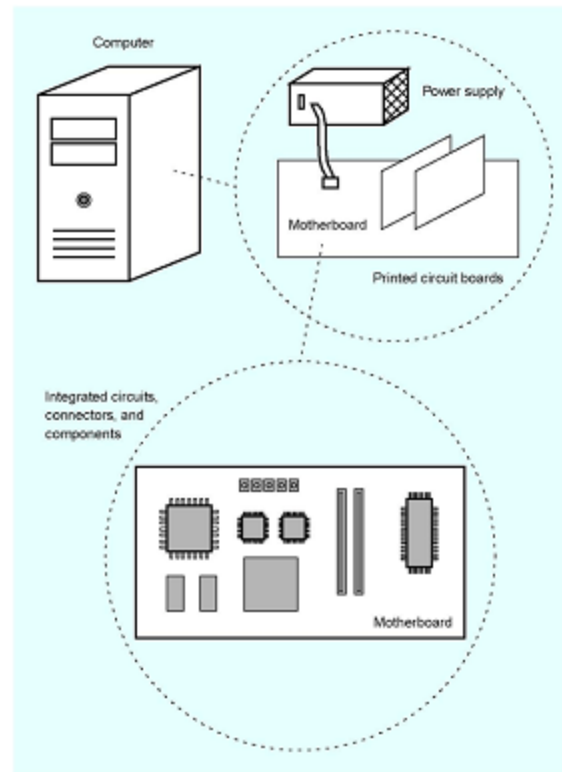
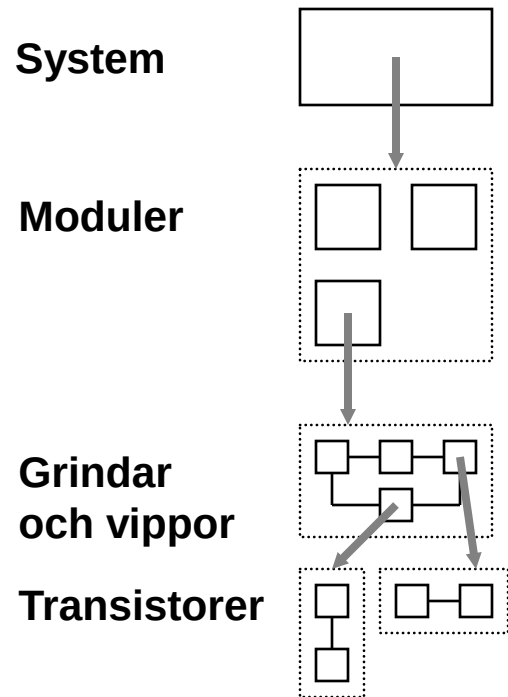


Figure 1.5. A digital hardware system (Part a).

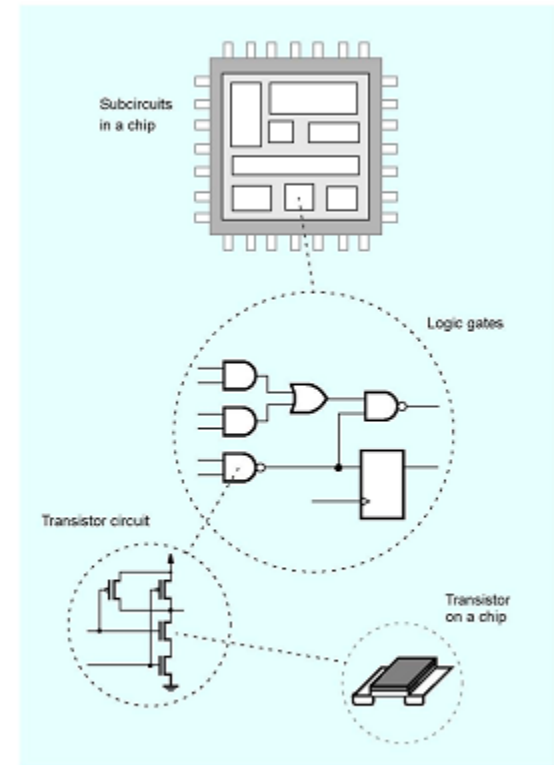
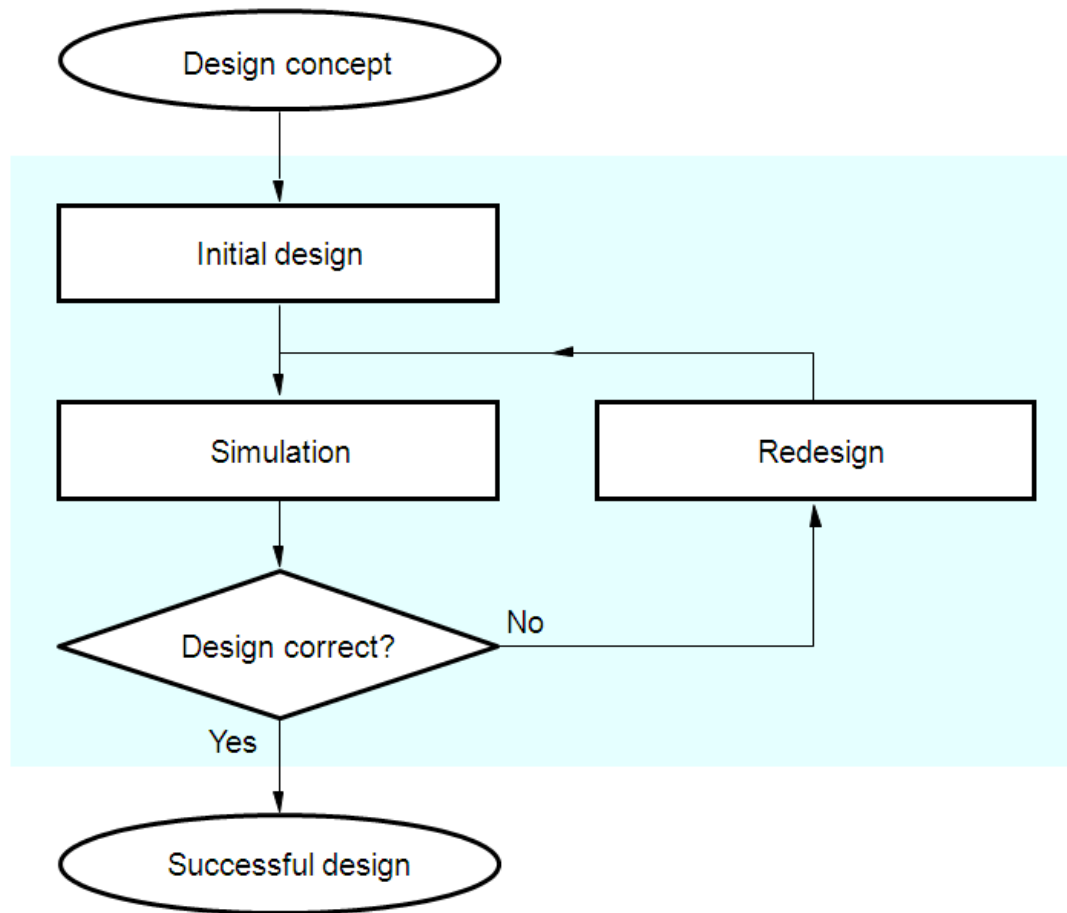


Figure 1.5. A digital hardware system (Part b).

Designprocessen



I kursen simulerar
vi en design med
ModelSim

Binära tal

Digitaltekniken använder bara två siffersymboler: **0** och **1**

– Enkelt att implementera – varje värde motsvarar en spänningsnivå, t ex

0 Volt motsvarar **0**

5 Volt motsvarar **1**

Hur kan man då representera vanliga tal?

Decimala talsystemet

I det decimala talsystemet har man 10 olika siffersymboler: **0** till **9**

Ett decimaltal representeras med en sekvens av siffersymboler

- Positionen i sekvensen ger siffrans vikt och multipliceras med en potens av 10 (*basen* i decimalsystemet är 10)

$$(653)_{10} = 6 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0$$

Decimala talsystemet

Representation av ett heltal

$$N_{10} = x_{m-1} \cdot 10^{m-1} + x_{m-2} \cdot 10^{m-2} \cdots + x_1 \cdot 10^1 + x_0 \cdot 10^0$$

$$(653)_{10} = 6 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0$$

Representation av ett "kommatal"

$$N_{10} = x_{m-1} \cdot 10^{m-1} + x_{m-2} \cdot 10^{m-2} \cdots + x_1 \cdot 10^1 + x_0 \cdot 10^0 + x_{-1} \cdot 10^{-1} + x_{-2} \cdot 10^{-2} + \cdots$$

$$(6.53)_{10} = 6 \cdot 10^0 + 5 \cdot 10^{-1} + 3 \cdot 10^{-2}$$

Binära talsystemet

Binärsystemet fungerar på samma sätt som decimalsystemet, men man använder basen **2** i stället för 10!

$$N_2 = x_{m-1} \cdot 2^{m-1} + x_{m-2} \cdot 2^{m-2} \cdots + x_1 \cdot 2^1 + x_0 \cdot 2^0 + x_{-1} \cdot 2^{-1} + x_{-2} \cdot 2^{-2} + \cdots$$

$$(110)_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = (6)_{10}$$

$$(11.01)_2 = 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = (3.25)_{10}$$

Oktala talsystemet

I det oktala talsystemet är basen **8** och därmed används siffersymbolerna **0** till **7**

$$N_8 = x_{m-1} \cdot 8^{m-1} + x_{m-2} \cdot 8^{m-2} \cdots + x_1 \cdot 8^1 + x_0 \cdot 8^0 + x_{-1} \cdot 8^{-1} + x_{-2} \cdot 8^{-2} + \cdots$$

$$(65.3)_8 = 6 \cdot 8^1 + 5 \cdot 8^0 + 3 \cdot 8^{-1} = (53.375)_{10}$$

Hexadecimala talsystemet

I det hexadecimala talsystemet är basen **16** och därmed används siffersymbolerna **0** till **9** och **A** till **F**

$$N_{16} = x_{m-1} \cdot 16^{m-1} + x_{m-2} \cdot 16^{m-2} \cdots + x_1 \cdot 16^1 + x_0 \cdot 16^0 + x_{-1} \cdot 16^{-1} + x_{-2} \cdot 16^{-2} + \cdots$$

$$(AE.8)_{16} = 10 \cdot 16^1 + 14 \cdot 16^0 + 8 \cdot 16^{-1} = (174.5)_{10}$$

Talsystem med basen b

En allmän formulering kan erhållas för basen b

$$N_b = x_{m-1} \cdot b^{m-1} + x_{m-2} \cdot b^{m-2} \cdots + x_1 \cdot b^1 + x_0 \cdot b^0 + x_{-1} \cdot b^{-1} + x_{-2} \cdot b^{-2} + \cdots$$

Heltalen för de olika talsystemen

Bas 2	Bas 8	Bas 10	Bas 16
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7

Bas 2	Bas 8	Bas 10	Bas 16
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10

Snabbfråga

Vilket hexadecimala tal motsvarar det binära talet 00010111 ?

A: 23

B: 13

C: 17

Diskutera fram svaret tillsammans med din bänkgreanne!



Snabbfråga

Vilket hexadecimala tal motsvarar det binär talet 00010111 ?

1 7

A: 23

B: 13

C: 17



Omvandling mellan decimala och binära tal

- Omvandling från binär till decimal är trivial
- Omvandlingen från decimaltal till binärtal görs genom upprepade delning med 2
 - Resten ger siffervärdet
 - Siffrorna kommer i omvänd ordning – Least Significant Bit (LSB) kommer först

$$53 \div 2 = 26 \text{ Rest } 1 \Rightarrow x_0 = 1 \text{ (LSB)}$$

$$26 \div 2 = 13 \text{ Rest } 0 \Rightarrow x_1 = 0$$

$$13 \div 2 = 6 \text{ Rest } 1 \Rightarrow x_2 = 1$$

$$6 \div 2 = 3 \text{ Rest } 0 \Rightarrow x_3 = 0 \quad 53_{10} = 110101_2$$

$$3 \div 2 = 1 \text{ Rest } 1 \Rightarrow x_4 = 1$$

$$1 \div 2 = 0 \text{ Rest } 1 \Rightarrow x_5 = 1 \text{ (MSB)}$$

Sammanfattning

- Det finns olika talsystem
- Digitaltekniken använder det binära talsystemet
- Man kan omvandla tal mellan olika talsystem

