

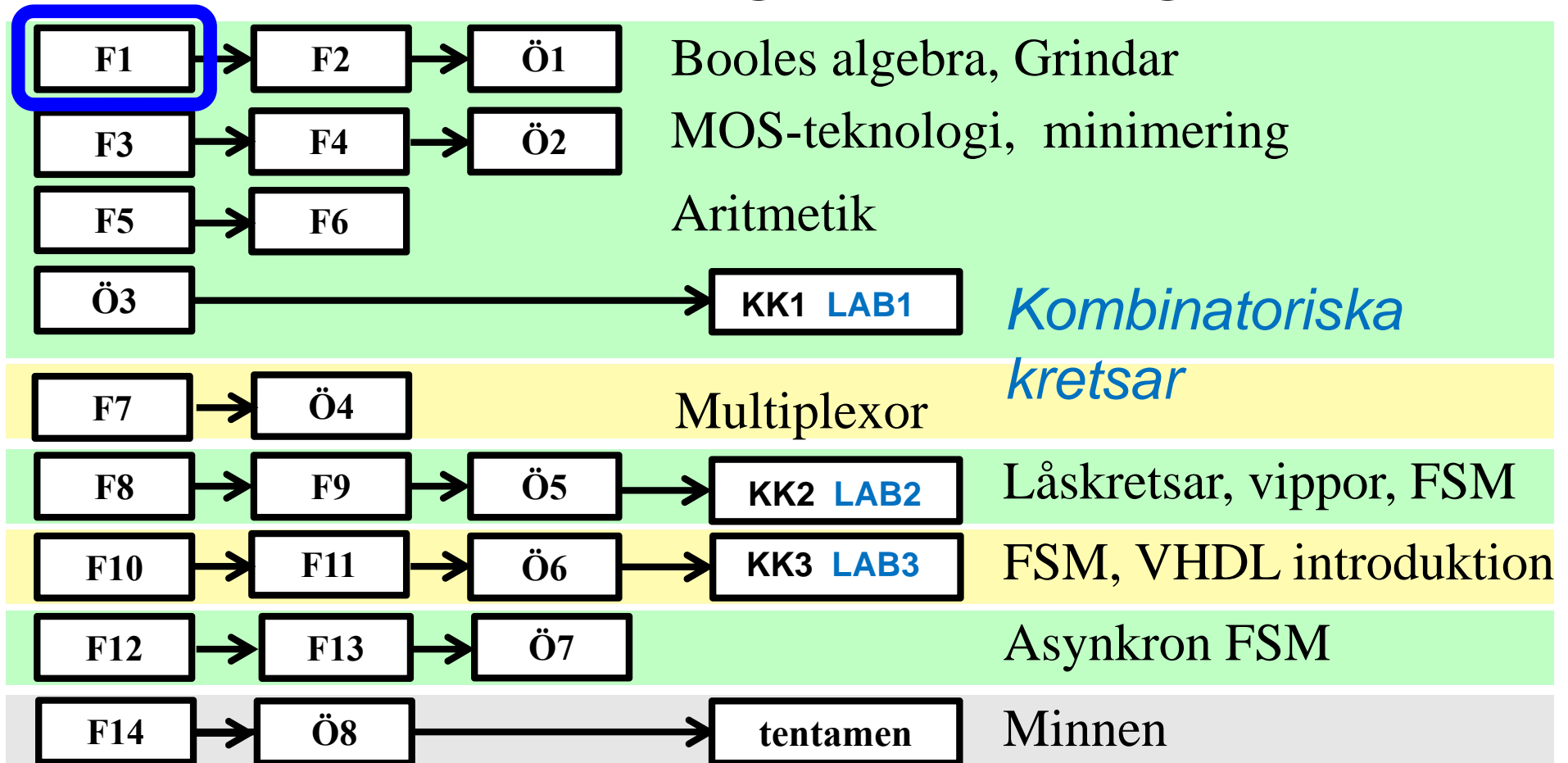
# Digital Design IE1204

Baserat på föreläsningsbilder av William Sandqvist

## **F1** Introduktion till Digitaltekniken

**Carl-Mikael Zetterling**  
**bellman@kth.se**

# IE1204 Digital Design



*Föreläsningar och övningar bygger på varandra! Ta alltid igen det Du missat!  
Läs på i förväg – delta i undervisningen – arbeta igenom materialet efteråt!*

# Kursens mål

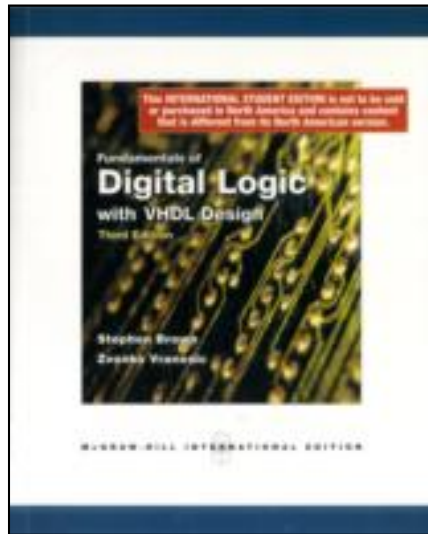
- Introducera studenterna till engelska och engelsk kurslitteratur

- *Nästan all relevant litteratur inom ämnet är på engelska*

- *Engelska är arbetspråk i alla större svenska internationella bolag*

- *Att tala engelska (någorlunda) flytande är en förutsättning för en framgångsrik karriär som ingenjör*

# Lärobok på engelska



Brown/Vranesic, ***Fundamentals of Digital Logic with VHDL Design*** (3rd edition), Mc-Graw-Hill, 2009  
(eTextbook: at [CourseSmart](#))

Boken finns också på KTH  
Kårbokhandel (Valhallavägen)  
Eller från Adlibris

( *Eller om Du  
så vill, kan Du  
använda en  
annan lärobok  
som är på  
svenska.*  )



# ***Två övningsgrupper på svenska***

- Tillfälle 1: Efternamn A - K  
Första övningen 1/11 8-10 i **sal 205**
- Tillfälle 2: Efternamn L - Ö  
Första övningen 1/11 13-15 i **sal B**

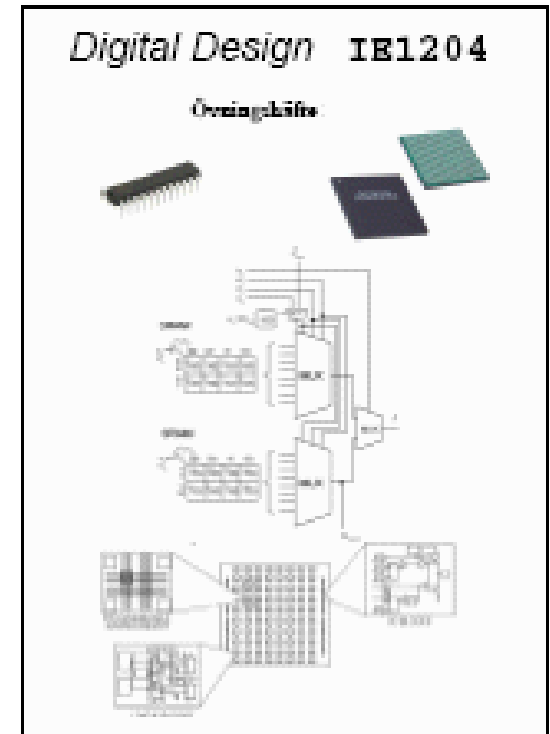
# Övningshäfte

Vid övningarna tar vi uppgifter från ett övningshäfte (svensk text).

Du laddar ned häftet från kurswebben. 🇸🇪 / 🇬🇧

- Föreläsningspresentationer finns att ladda ned från kurswebben. 🇸🇪 / 🇬🇧
- Övningspresentationer finns att ladda ned från kurswebben. 🇸🇪 / 🇬🇧


*( Observera att övnings-assistenterna är fria att lösa uppgifterna på annat sätt än i presentationerna).*





# Laborationer

***Observera! Ett mycket omfattande förberedelsearbete krävs inför laborationerna!***


- **Kombinatoriska kretsar**
- **Sekvens kretsar**
- **VHDL intro**

*Labbhäften finns att ladda ner från kurswebben.  / *

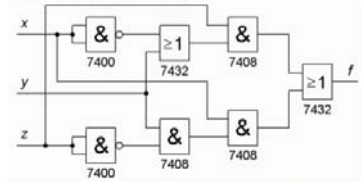
För att få laborera måste Du ha gjort ditt **kunskapskontrollhäfte** före varje lab (=100%), och gjort alla **förberedelseuppgifter** ( $\approx 99\%$ ).  / 

# Laborationer

1. Kombinatoriska kretsar
2. Sekvens kretsar
3. VHDL intro

**Laboration Kombinatoriska kretsar** 

Digital Design IE1204/5

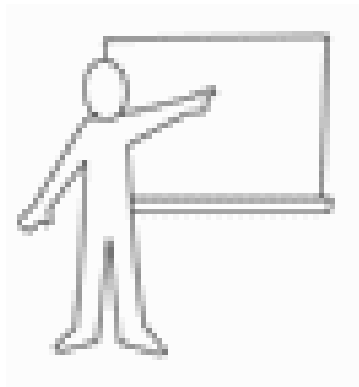


*Eftersom detta är ditt labkvino måste Du fylla i tabellen med blåck.*

Namn:	
Personnummer:	
• <b>Förkunskapstest (Web-frågehäfte)</b>	
Häfte nr:	Datum:
Lab-lärarens kvittens:	
• <b>Förberedelseuppgifter i labhäftet</b>	
Lab-lärarens kvittens:	
• <b>Laborationens genomförande</b>	
Laborationen utförd den:	
Lab-lärarens kvittens:	



# Muntlig presentation inför gruppen



Den här ikonen i labb-häftena betyder att Du kan bli utlottad att få presentera denna förberedelseuppgift muntligt inför gruppen vid laborationen. Om din lab-assistent är engelskspråkig så försöker Du på engelska.

Det hela är en nyttig övning, i din kommande yrkesroll kommer detta hända ofta.

***Glöm inte bort att här är Du bland vänner. Ingen sömnlös natt över detta – men kom förberedd.***

# Kunskapskontroll inför labb

Inför laborationerna har vi kunskapskontroller. Du kan göra denna på webben när som helst, och prova att svara på frågorna hur många gånger som helst.

Därför kräver vi **alla rätt!**

- För att få utbyte av laborationerna måste Du ha tillräckliga förkunskaper.

- När Du gör kunskapskontrollerna så läser Du samtidigt in viktiga kursavsnitt inför tentamen. Alla vet att tentamensveckan *inte* räcker till för detta.

# Kunskapskontroll inför labb

Välj rätt **frågehäfte** – det med ditt nr (se Uppgifter i Canvas)

 TIDAB TIEDB ht 2017 P2 DD1

( [Change to english text](#) )

Att ge "Rätt svar" utan att *själv* ha utfört *alla* beräkningar kan verka **fördummande!**

DD1 = Labb 1

DD2 = Labb 2


DD3 = Labb 3

<a href="#">DD1-201</a>	<a href="#">DD1-202</a>	<a href="#">DD1-203</a>	<a href="#">DD1-204</a>	<a href="#">DD1-205</a>
<a href="#">DD1-206</a>	<a href="#">DD1-207</a>	<a href="#">DD1-208</a>	<a href="#">DD1-209</a>	<a href="#">DD1-210</a>
<a href="#">DD1-211</a>	<a href="#">DD1-212</a>	<a href="#">DD1-213</a>	<a href="#">DD1-214</a>	<a href="#">DD1-215</a>
<a href="#">DD1-216</a>	<a href="#">DD1-217</a>	<a href="#">DD1-218</a>	<a href="#">DD1-219</a>	<a href="#">DD1-220</a>
<a href="#">DD1-221</a>	<a href="#">DD1-222</a>	<a href="#">DD1-223</a>	<a href="#">DD1-224</a>	<a href="#">DD1-225</a>
<a href="#">DD1-226</a>	<a href="#">DD1-227</a>	<a href="#">DD1-228</a>	<a href="#">DD1-229</a>	<a href="#">DD1-230</a>
<a href="#">DD1-231</a>	<a href="#">DD1-232</a>	<a href="#">DD1-233</a>	<a href="#">DD1-234</a>	<a href="#">DD1-235</a>
<a href="#">DD1-236</a>	<a href="#">DD1-237</a>	<a href="#">DD1-238</a>	<a href="#">DD1-239</a>	<a href="#">DD1-240</a>
<a href="#">DD1-241</a>	<a href="#">DD1-242</a>	<a href="#">DD1-243</a>	<a href="#">DD1-244</a>	<a href="#">DD1-245</a>
<a href="#">DD1-246</a>	<a href="#">DD1-247</a>	<a href="#">DD1-248</a>	<a href="#">DD1-249</a>	<a href="#">DD1-250</a>
<a href="#">DD1-251</a>	<a href="#">DD1-252</a>	<a href="#">DD1-253</a>	<a href="#">DD1-254</a>	<a href="#">DD1-255</a>

<a href="#">DD1-301</a>	<a href="#">DD1-302</a>	<a href="#">DD1-303</a>	<a href="#">DD1-304</a>	<a href="#">DD1-305</a>
<a href="#">DD1-306</a>	<a href="#">DD1-307</a>	<a href="#">DD1-308</a>	<a href="#">DD1-309</a>	<a href="#">DD1-310</a>
<a href="#">DD1-311</a>	<a href="#">DD1-312</a>	<a href="#">DD1-313</a>	<a href="#">DD1-314</a>	<a href="#">DD1-315</a>
<a href="#">DD1-316</a>	<a href="#">DD1-317</a>	<a href="#">DD1-318</a>	<a href="#">DD1-319</a>	<a href="#">DD1-320</a>
<a href="#">DD1-321</a>	<a href="#">DD1-322</a>	<a href="#">DD1-323</a>	<a href="#">DD1-324</a>	<a href="#">DD1-325</a>
<a href="#">DD1-326</a>	<a href="#">DD1-327</a>	<a href="#">DD1-328</a>	<a href="#">DD1-329</a>	<a href="#">DD1-330</a>
<a href="#">DD1-331</a>	<a href="#">DD1-332</a>	<a href="#">DD1-333</a>	<a href="#">DD1-334</a>	<a href="#">DD1-335</a>
<a href="#">DD1-336</a>	<a href="#">DD1-337</a>	<a href="#">DD1-338</a>	<a href="#">DD1-339</a>	<a href="#">DD1-340</a>
<a href="#">DD1-341</a>	<a href="#">DD1-342</a>	<a href="#">DD1-343</a>	<a href="#">DD1-344</a>	<a href="#">DD1-345</a>
<a href="#">DD1-346</a>	<a href="#">DD1-347</a>	<a href="#">DD1-348</a>	<a href="#">DD1-349</a>	<a href="#">DD1-350</a>
<a href="#">DD1-351</a>	<a href="#">DD1-352</a>	<a href="#">DD1-353</a>	<a href="#">DD1-354</a>	<a href="#">DD1-355</a>

# Kunskapskontroll inför labb

Change to English

 Bundle: [DD1-398](#)

( Klicka på länken för att öppna svarssidan! )

## Digital Design IE1204, för TIDAB/TIEDB 2017/(2018) p2

### *Kunskapskontroll inför laborationen,* Kombinatoriska kretsar

Detta är ditt eget unika **frågehäfte**. Normalt kräver systemet siffersvar, så frågehäftet innehåller JAVA-script som räknar fram siffersvar när svaren avges på a  
*Detta kräver att Du tillåter din webbläsare att köra JAVA-script.*

Till frågehäftet hör en **svarssida** med samma nummer. Med det provar Du siffersvaren, och får reda på om Du svarat *rätt* eller *fel* på frågorna i frågehäftet.

Du kan svara hur många gånger som helst, men skriv gärna upp alla dina rätta svar i frågehäftet eftersom svarsrutan försvinner när Du svarar rätt. Uppgifterna

**Observera att Du måste ha svarat på alla frågorna innan Du laborerar! Du rekommenderas därför att göra kunskapskontrollen omg**

Om Du vill jobba med papper och penna så kan Du skriva ut buntan och anteckna svaren i rutorna, men observera i så fall att alla svar måste avges till svarssi  
(JAVA-scripten måste Du naturligtvis köra med webbläsaren för att få fram siffersvaren.)

- I frågehäftet finns sedan "hjälp"-knappar som pekar ut vilka kursavsnitt som är lämpliga att läsa på för att lösa frågorna!

## Svarssidan.

När svaret är det rätta bockas rutan av och tas bort.

När alla rutor i hela kunskapskontrollen blivit avbockade är man klar.

DDA-001

✖ [Excercise 1](#)

✖ [Exc](#)

✖ [Exc](#)

DDA-001 / [Excercise 1](#)

✖ [Question 1](#)

✖ [Qu](#)

DDA-001 / [Excercise 1](#) / [Question 1](#)

a)

✔ Correct answer.

b)

c)

d)

Submit

# Kunskapskontroll inför labb

*Lös uppgifterna ...*

## Exercise: 1, question: 1

*Decimala, binära, hexadecimala och oktala tal* [Hjälp?](#)

Omvandla följande hexadecimaltal,  $3E_{16}$  till

- a) binärtal?
- b) oktaltal?
- c) decimaltal (heltal)?

Omvandla följande reella binärtal,  $0110.0101_2$  till

- d) reellt decimaltal (tal med decimaler, systemet kräver decimalpunkt när Du svarar)?

*Du kan skriva ner dina svar i tabellen för att ha det till hands senare.*

a) Svar: [ binärtal ]	b) Svar: [ oktaltal ]	c) Svar: [ decimalt heltal ]	d) Svar: [ reellt decimaltal ]

*Länk till användbara kurssidor*

[Hjälp?](#)



# **Vi kontrollerar om Ditt nummer registrerats**

**Innan laborationerna kontrollerar vi om Ditt kk-nummer registrerats. Kontrollera därför själv att Din bunt har alla uppgifter lösta!**

**Blir Du sedan *inte* klar med labbförberedelserna i tid gör du laborationen vid något annat tillfälle (eller bli omdeltagare vid nästa kursomgång)**

# Kan man lura systemet?

Kan man "lura" systemet?

- Nej det är ju automatiskt och "bryr sig därför inte om hur Du gör" så själva systemet blir *inte* lurat, ***det är alltid dig själv Du lurar.***





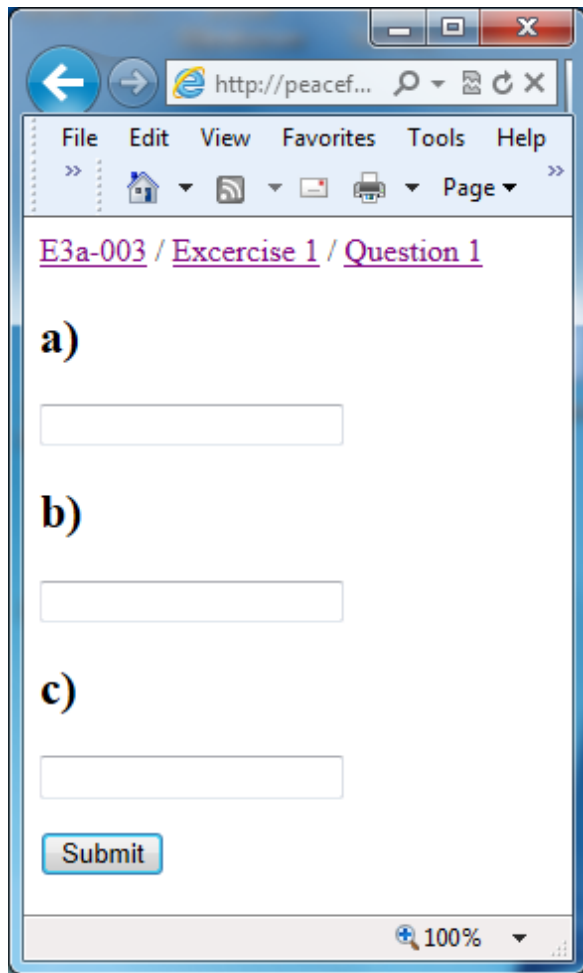
# Hjälper varningstexter?



- Hur är det? Har alla problem upphört i och med varningstexterna?

# Hjälper varningstexter?

## *Vår varningstext*



The screenshot shows a web browser window with the address bar displaying "http://peacef...". The browser has a menu bar with "File", "Edit", "View", "Favorites", "Tools", and "Help". Below the menu bar is a toolbar with icons for home, back, forward, and search. The main content area shows the URL "E3a-003 / Exercise 1 / Question 1" in purple. Below the URL are three questions labeled a), b), and c), each followed by a text input field. At the bottom of the form is a "Submit" button. The status bar at the bottom of the browser shows "100%" zoom.

Att ge "Rätt svar" utan att *själv* ha utfört *alla* beräkningar kan verka **fördummande!**

*Tveksamt om varningstexter hjälper, men trots det så försöker vi ...*

# Bokade labb-tider i Canvas

Från början har Du fått **slumpvalda** labbtider i Canvas. Vid dessa tider är Du **garanterad** din plats.

Om någon tid är olämplig så maila kursansvarig för att få boka om. OBS! Vi gör *inga sena ombokningar* i labb-veckorna, så maila i god tid före labbtillfället.

Efter det att labben har börjat vet vi hur många som kommit – det kan då finnas några **lediga platser** för **obokade studenter**. (Vi kan inte garantera detta)

# Högskolestudier – eget ansvar

**Planera** den tid Du lägger på kursen.

- Närvaro på föreläsningar/lektioner är *inte obligatorisk* – *planera*.
- Till laborationerna hör obligatorisk kunskapskontroll på webben – *planera*.
- Till laborationerna hör obligatoriska förberedelse-uppgifter – *planera*.

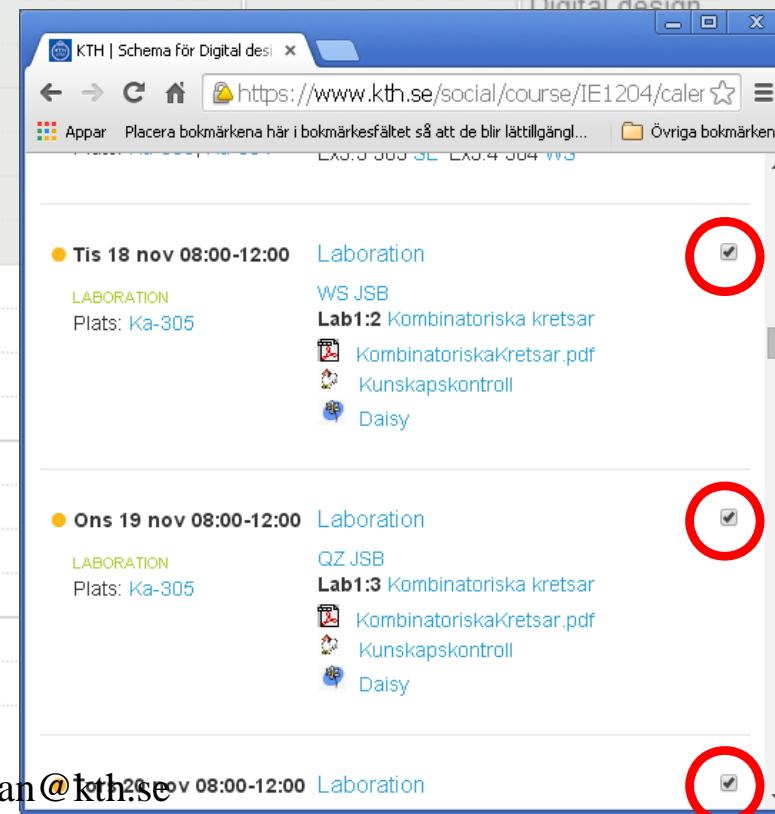
Vecka 47 2014

	Mån 17 Nov	Tis 18 Nov	Ons 19 Nov	Tors 20 Nov	Fre 21 Nov
08:00	08:00 - 12:00	08:00 - 12:00	08:00 - 12:00	08:00 - 12:00	
	● Laboration - Digital design (IE1204)	● Laboration - Digital design (IE1204)	● Laboration - Digital design (IE1204)	● Laboration - Digital design (IE1204)	
09:00	Ka-305	Ka-305	Ka-305	Ka-305	
	Se beskrivning	Se beskrivning	Se beskrivning	Se beskrivning	
10:00					● Föreläsning - Digital design
11:00					
12:00					
13:00	13:00 - 15:00				
	● Föreläsning - Digital design (IE1204)				
	Ka-303, Ka-304				
14:00	Se beskrivning				

# Använd schemat på KTH Social!

Du har tilldelats en av labb-tiderna – de övriga blir ledig tid för andra studier

Anpassa schemat så att bara det som är aktuellt för dig visas



# Studieteknik

- **Läs på översiktligt i förväg.** Gå igenom presentationerna i förväg. Förmodligen verkar mycket då oklart, anteckna de frågor och de funderingar Du har.
- **Ställ frågorna** på lektionen/föreläsningen.  
”Missar” Du lektionen utgår vi alltid ifrån att du tar igen materialet genom att läsa själv hemma.
- **Läs noggrant efteråt.** Gå noggrant igenom presentationerna efteråt. Gå igenom exemplen själv utan att snegla för mycket på lösningarna.
- **Lös uppgifterna i övningshäftet.** Till alla uppgifterna finns lösningsförslag. Kör Du fast snegla på lösningarna, men kom ihåg att Du övar för tentamenssituationen – utan lösningar!

# Kursens mål

- Att lära ut de teoretiska grunderna för analys och konstruktion av kombinatoriska och sekventiella kretsar
- Att genom praktisk problemlösning ge en förståelse för de olika design-/konstruktionsfaserna i syfte att

**kursdeltagarna ska behärska konstruktion av enkla kombinatoriska och sekventiella digitala system**

# Examination

LABA 3.5 hp

- betyg: P/F

TENA 4.0 hp

- betyg: A-E/F

Tentamen har tre delar: del **A1** och del **A2** samt del **B**. Del A2 och del B rättas bara om man är godkänd på del A1. För att bli godkänd på tentamen krävs poäng från både A1 och A2.

**Tentamen i Kista Mån 15 jan 2018 14:00-18:00.**

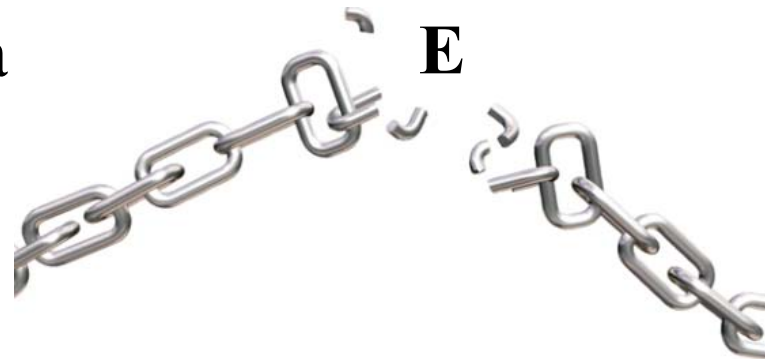
**Obligatorisk anmälan KTH-mina sidor!**



# Examination

Satsa *inte* på betyget **E**! Det är naturligtvist godkänt, men betyder förmodligen att Du saknar massor av nyttiga kunskaper som är viktiga för kommande kurser.

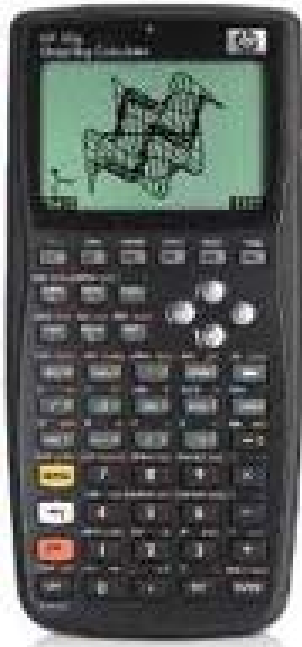
Kurserna följer på varandra  
som länkarna i en kedja.  
Kedjan brister vid den  
svagaste länken.



# 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

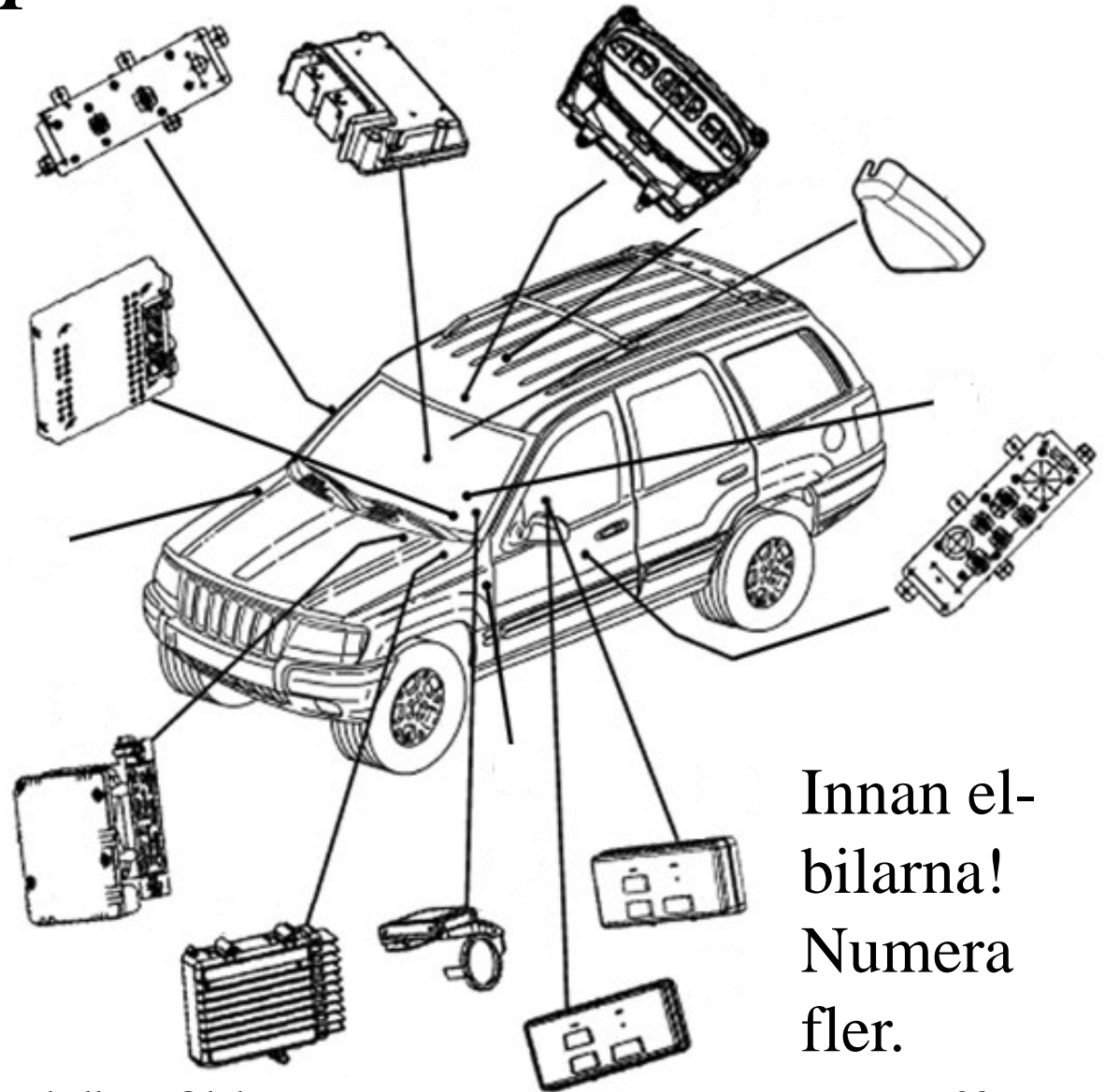


*Digital Design*  
*Överallt!*



*40-100 mikroprocessorer i en bil!*

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

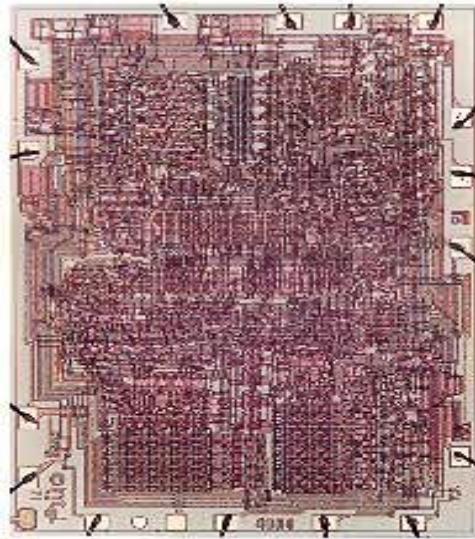


Innan el-  
bilarna!  
Numera  
fler.

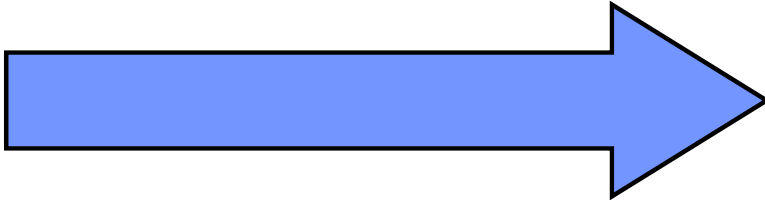


# *Utvecklingen av elektroniken*

Intel 4004  
(1971)



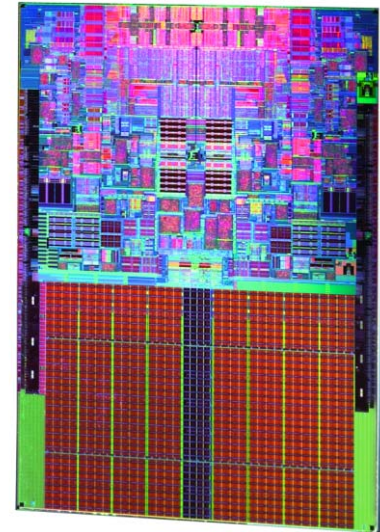
108 KHz  
2,300 transistorer



820 millioner transistorer

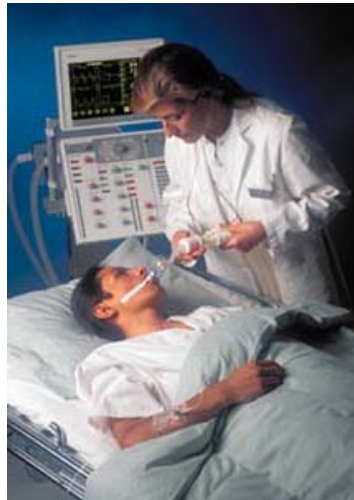
3.0 GHz

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?

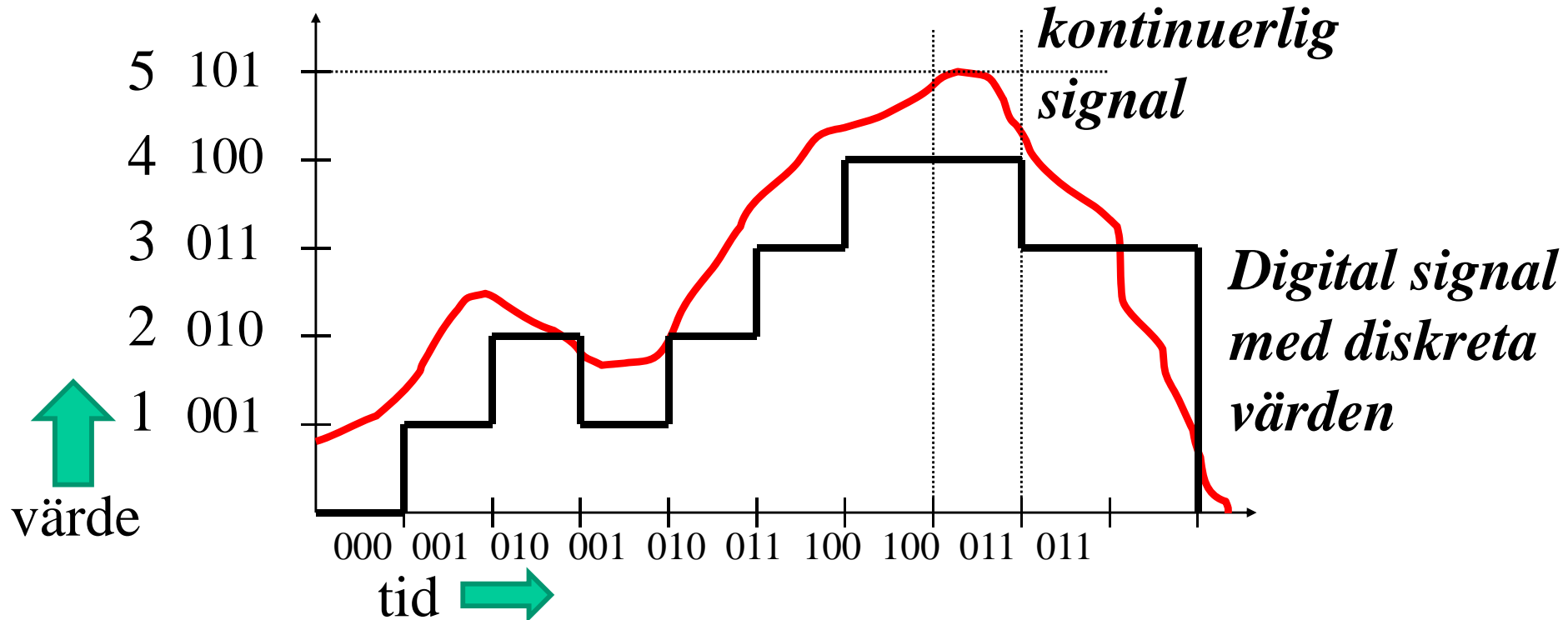


# Enkelhet, störningssäkerhet

- **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 halvledarteknologin
- Effektiva **designmetoder** och **verktyg**

# Analoga eller Digitala signaler?

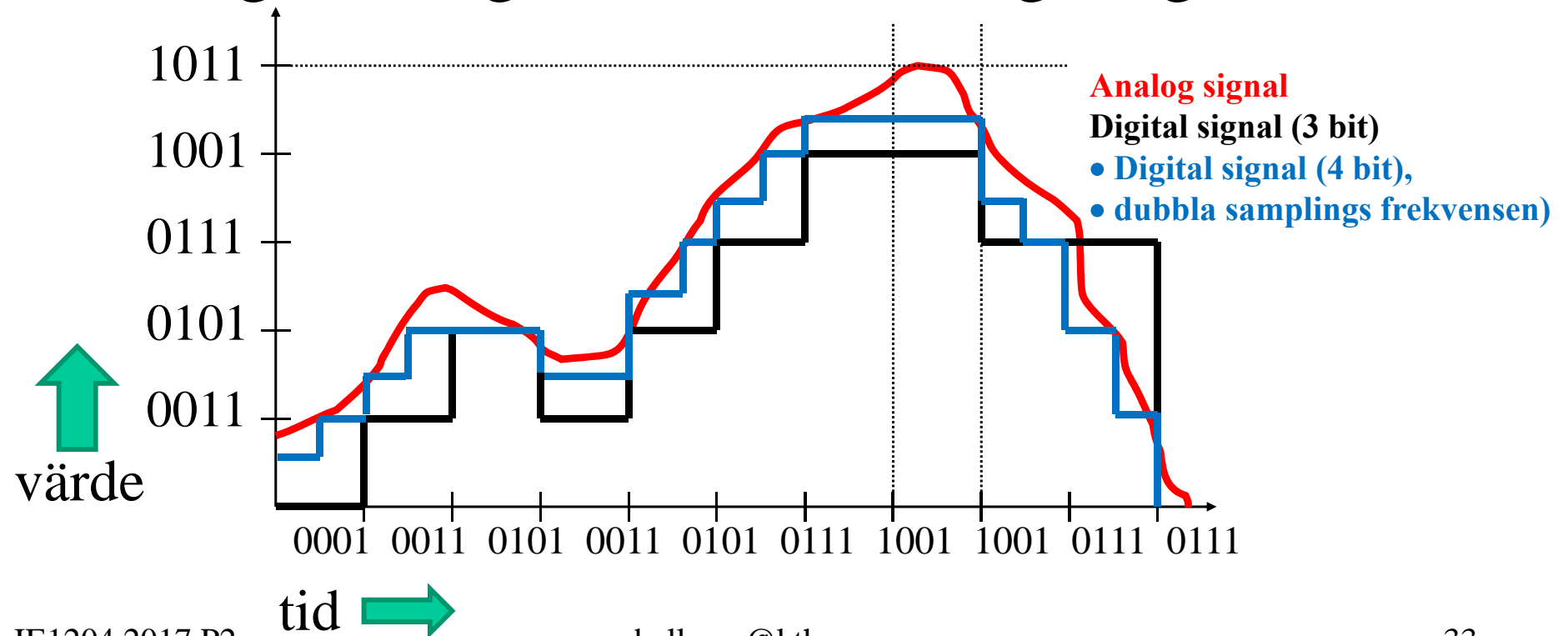
En analog signal kan anta *kontinuerliga* värden, medan en digital signal bara kan anta *diskreta* värden ( här 0...5 )



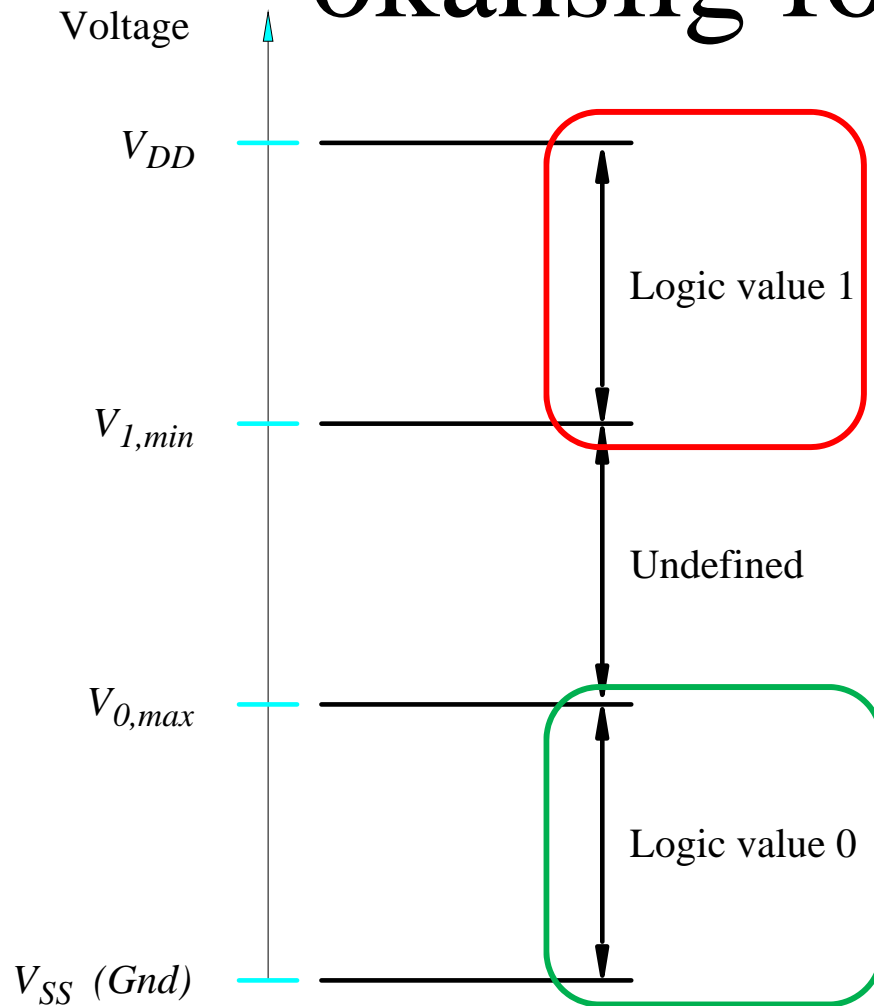


# Digitalt borde vara sämre?

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

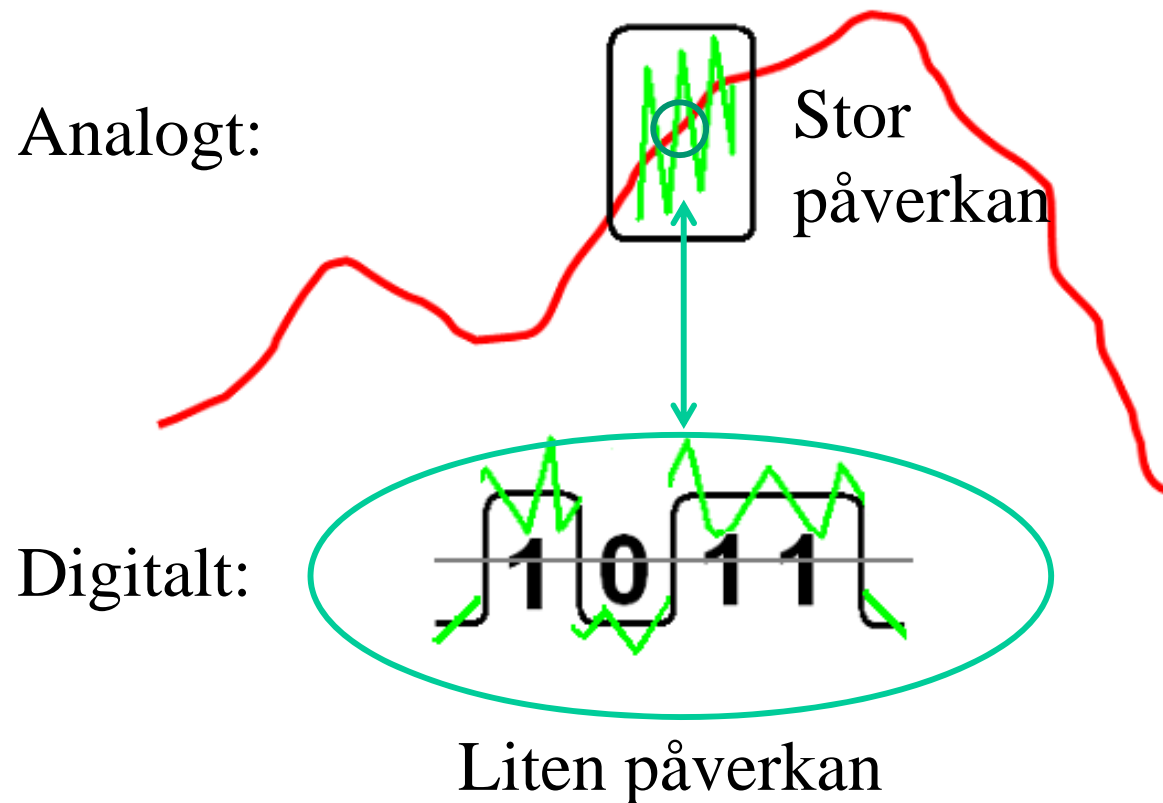


# Digitaltekniken är mycket okänslig för störningar



- 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

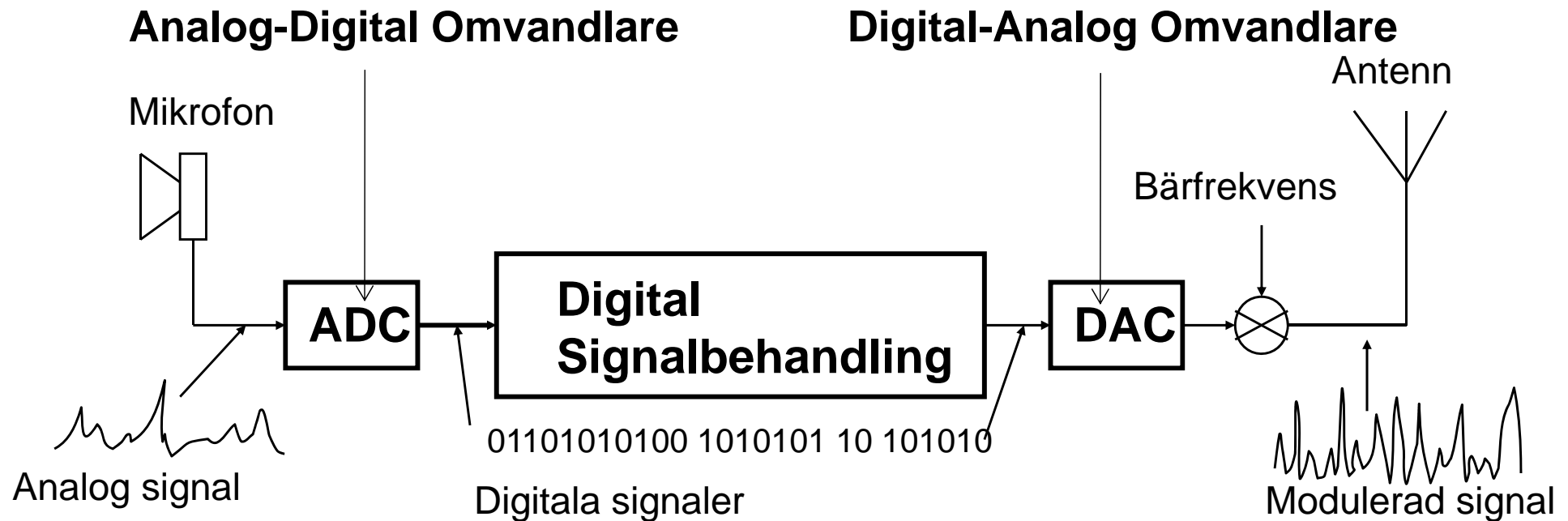
# Störningsokänslighet



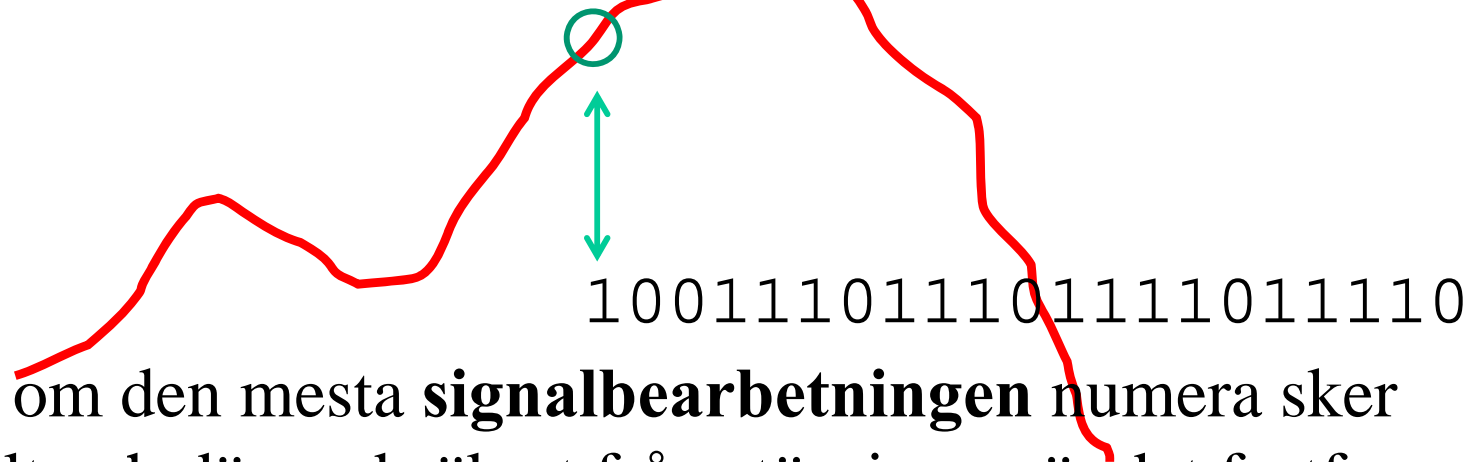
- Digital **databearbetning** kan ske störningsokänsligt!

# 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** ( $\bar{x}$ )
- 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 annorlunda 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

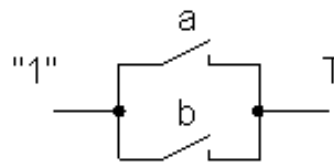


Brytande kontakt, "icke"

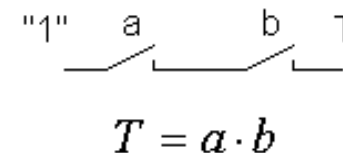


• för seriekoppling  
"och"-funktion

+ för parallellkoppling  
"eller"-funktion



$$T = a + b$$

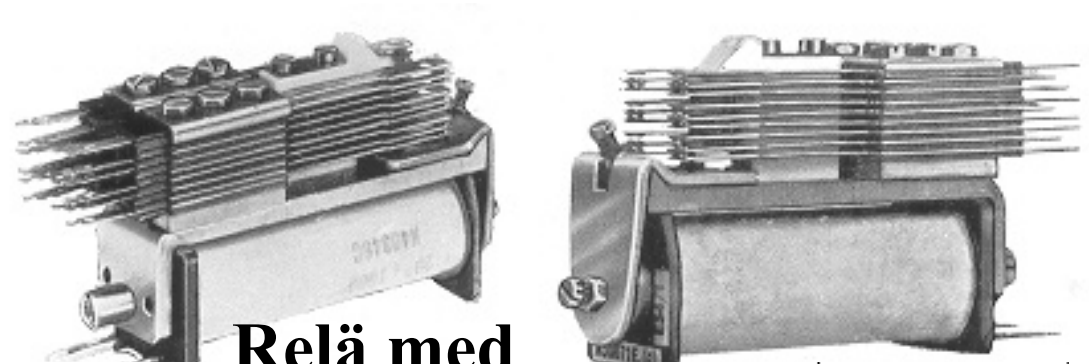




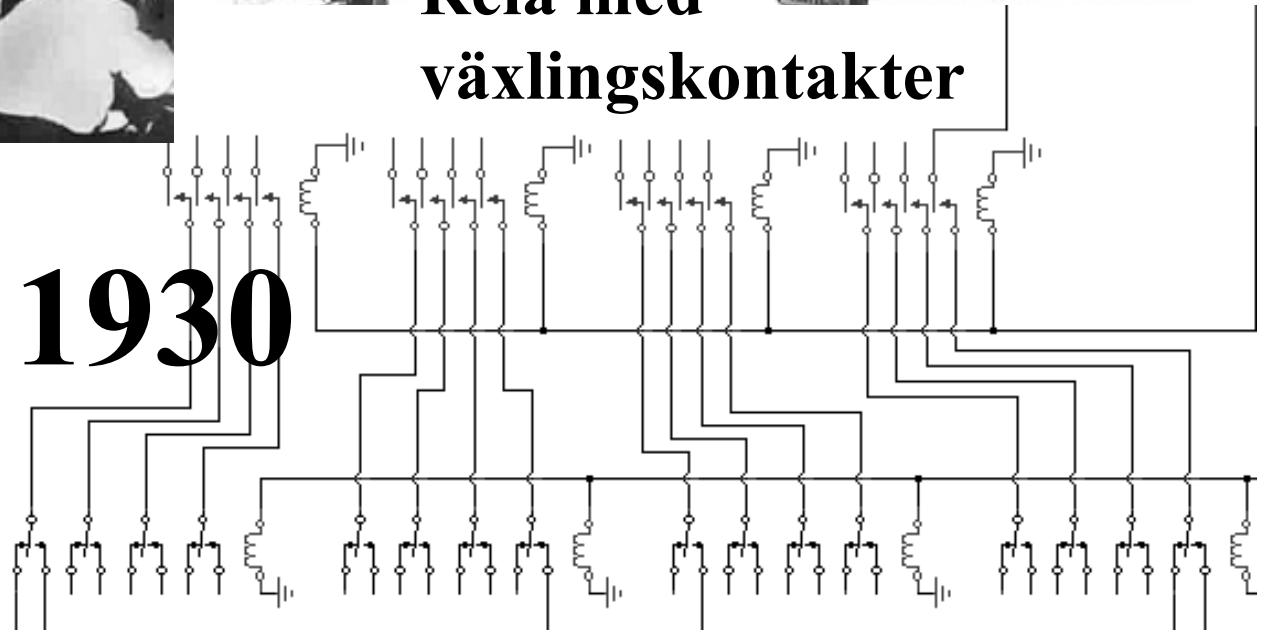
## Manuell telefonväxel



## Automatisk telefonväxel

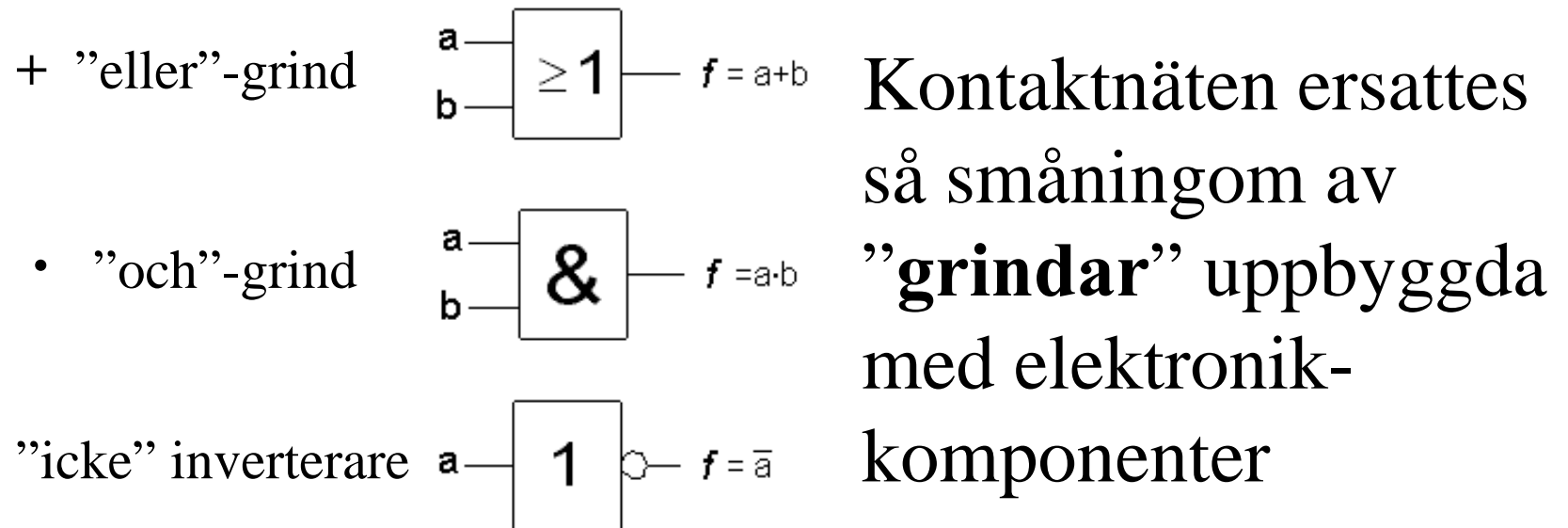


Relä med  
växlingskontakter



Vid automatisering av telefontätens växlar användes elektromekaniskt styrda **kontakter**, **reläer**, till att utföra olika logiska funktioner.

# *Grindar i stället för kontakter*



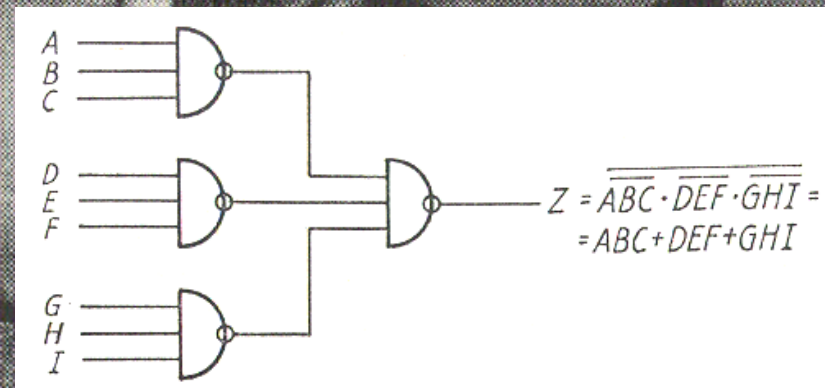
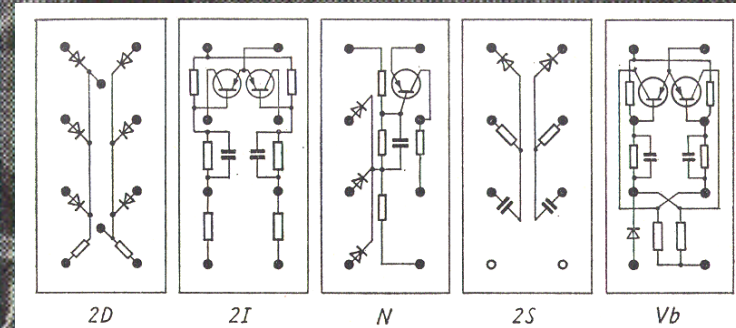
Med dessa tre grundläggande grindtyper: **OR**  
**AND NOT** kan **alla** logiska funktioner utföras.

• *Mer snart i kursen ...*



# 1960

Reläerna ersattes senare  
av elektroniska logiska  
funktionsblock, grindar



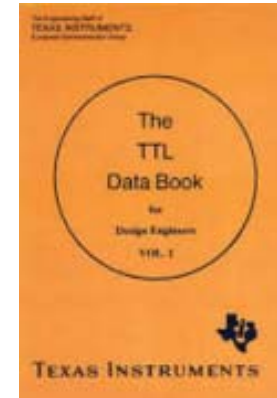
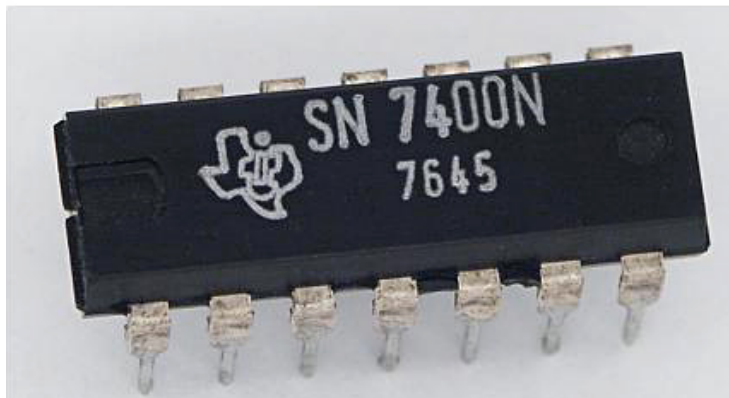
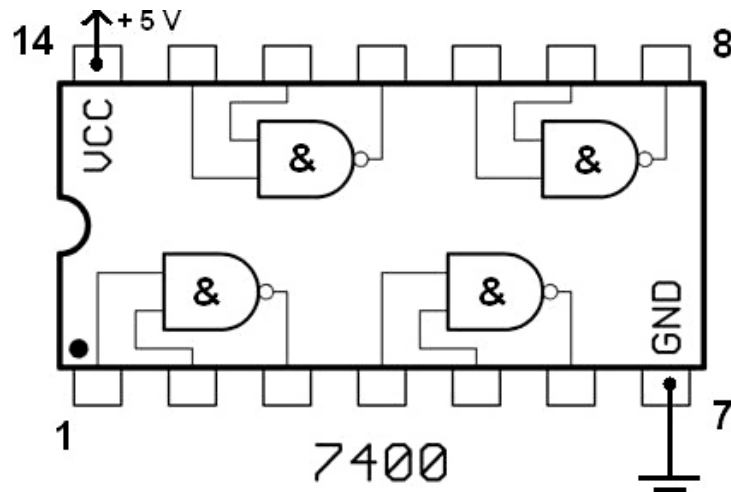
KTH Transistorgruppens  
elektronikgrindar 1963



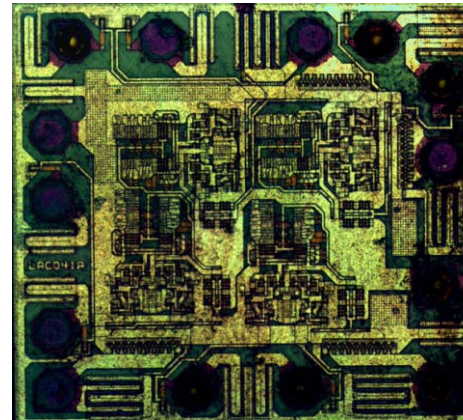


# 1970

- Integrerade logikkretsar  
74-serien (600 olika kretsar)

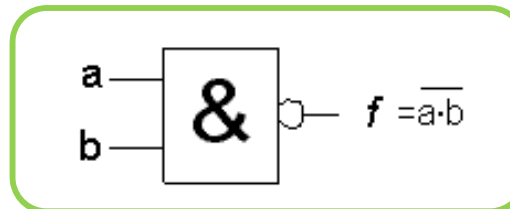


## 1973 Texas Instruments "The TTL Data Book"

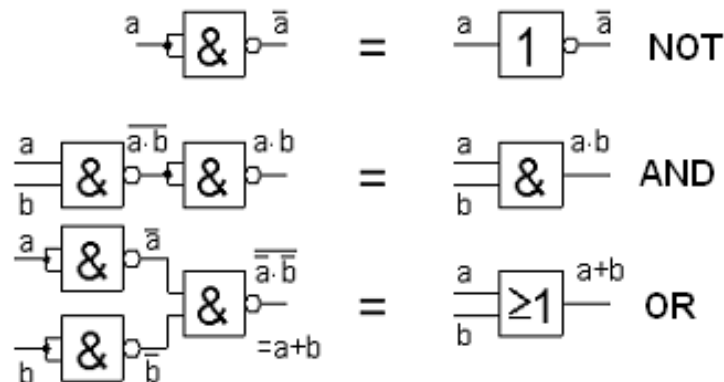


- DIL Dual In Line kapsel

# *Bara en enda grindtyp!*



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

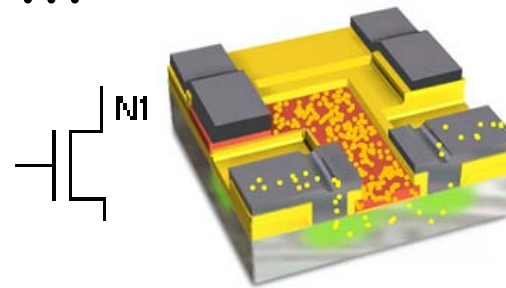
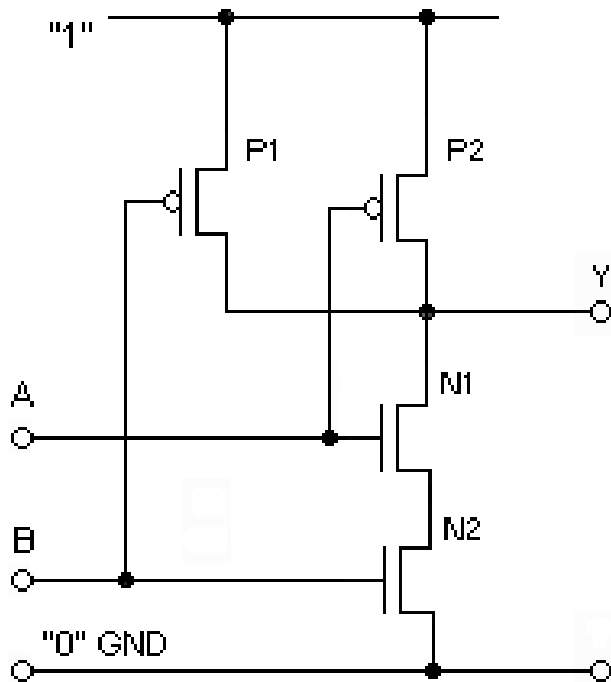


*Extrem förenkling,  
en enda grindtyp  
räcker till **allt** !*

• *Mer snart i kursen ...*

# CMOS NAND - grind

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



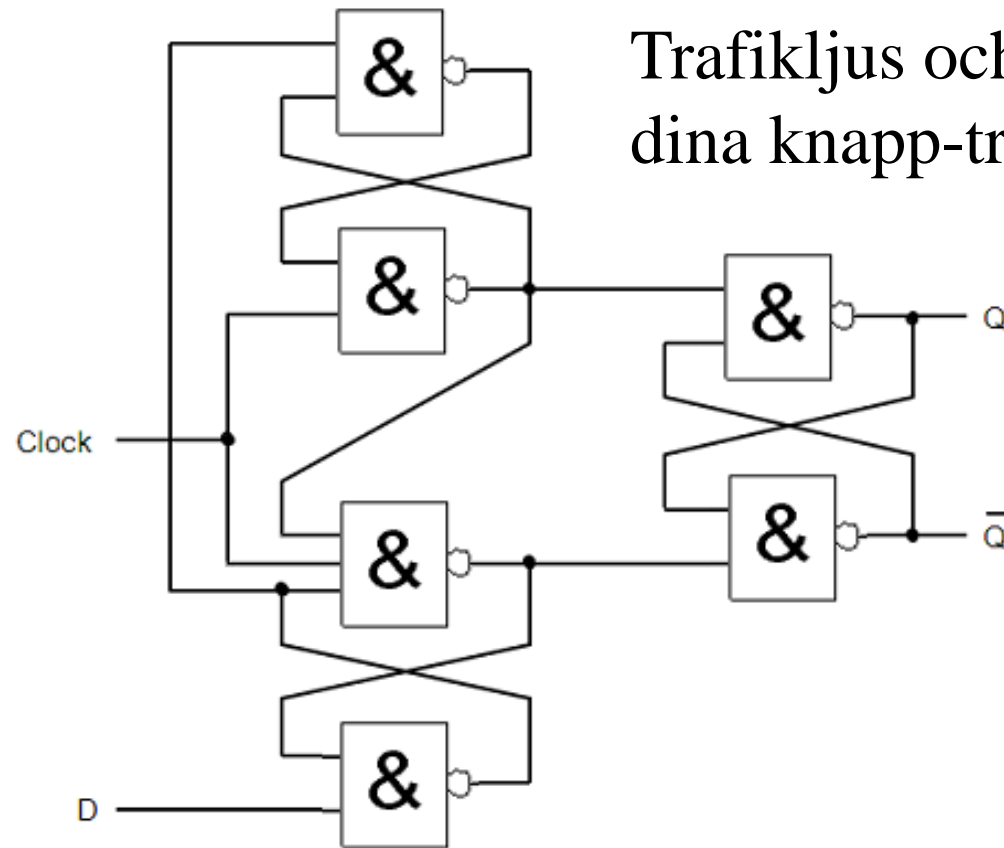
En N-MOS transistor.

Finess: CMOS-grindar förbrukar bara effekt vid omslag.

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

• *Mer snart i kursen ...*

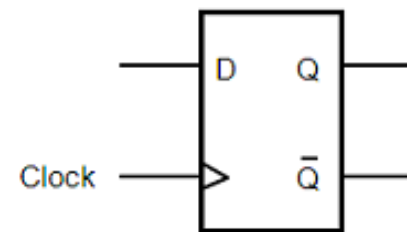
# Minnesfunktion?



(a) Circuit

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

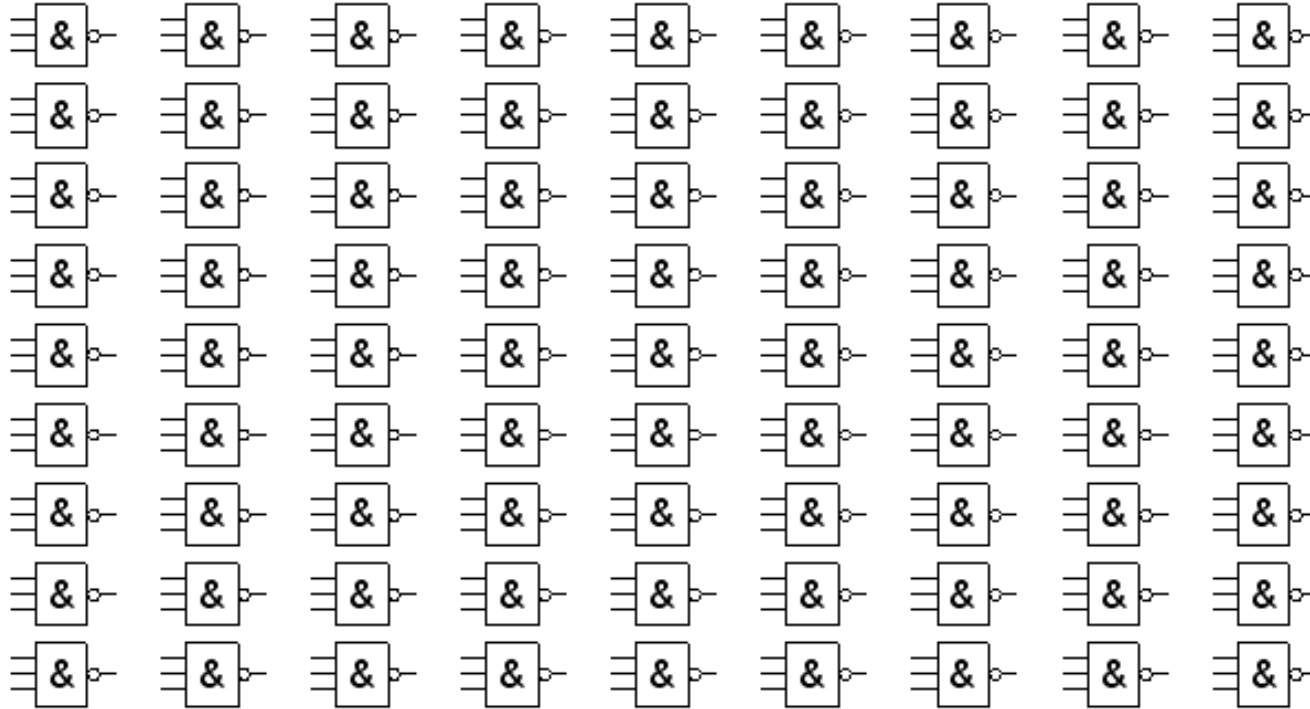
Detta utförs av ”vippor” eller ”låskretsar” – som man också kan göra med NAND-grindar.



(b) Graphical symbol

• *Senare i kursen ...*

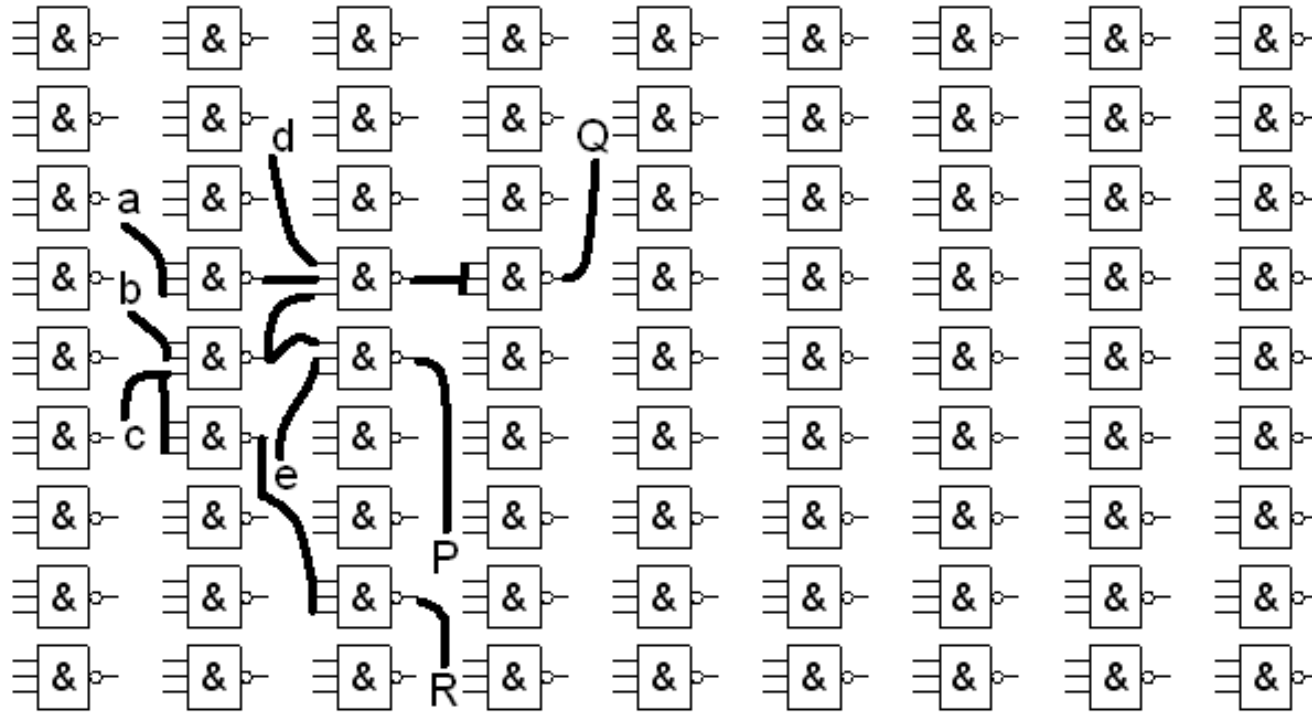
# Sea-of-Gates



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.



# Sea-of-Gates

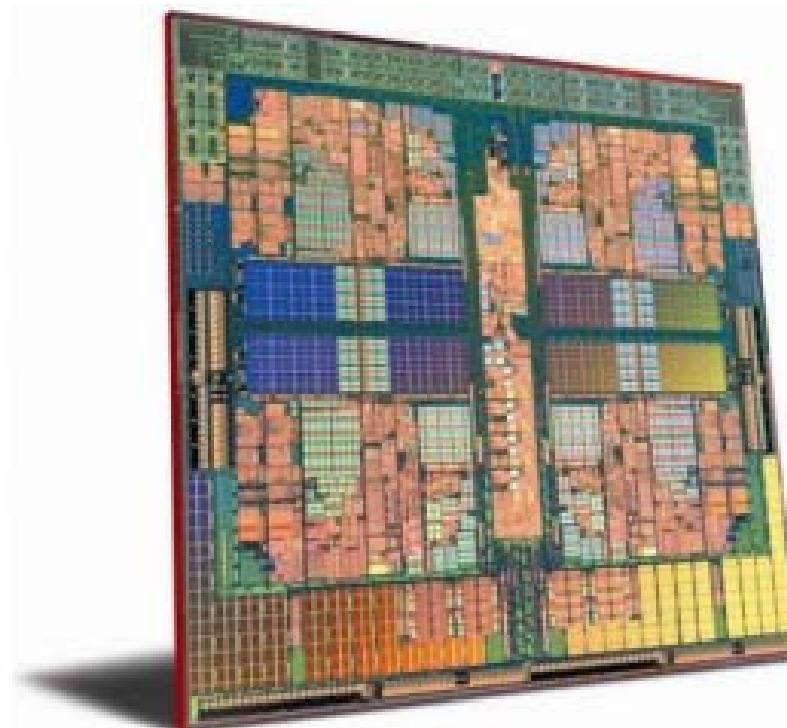


Eftersom bara en enda typ av grindar behövs, kan man göra vad som helst med "ett hav av grindar" som får ett lednings-mönster utformat för en egen specifik funktion.

# Idag



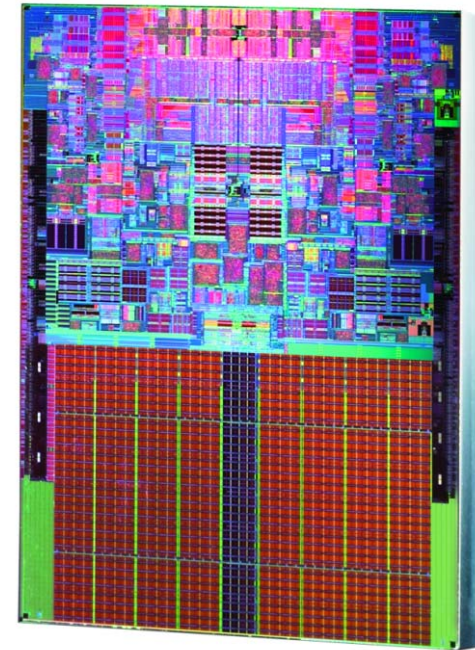
Idag,  
högintegrerade  
processorer med  
**miljarder**  
logikfunktioner ...



# En miljard grindar?

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

*Det behövs andra metoder för att beskriva ett sådant system!*

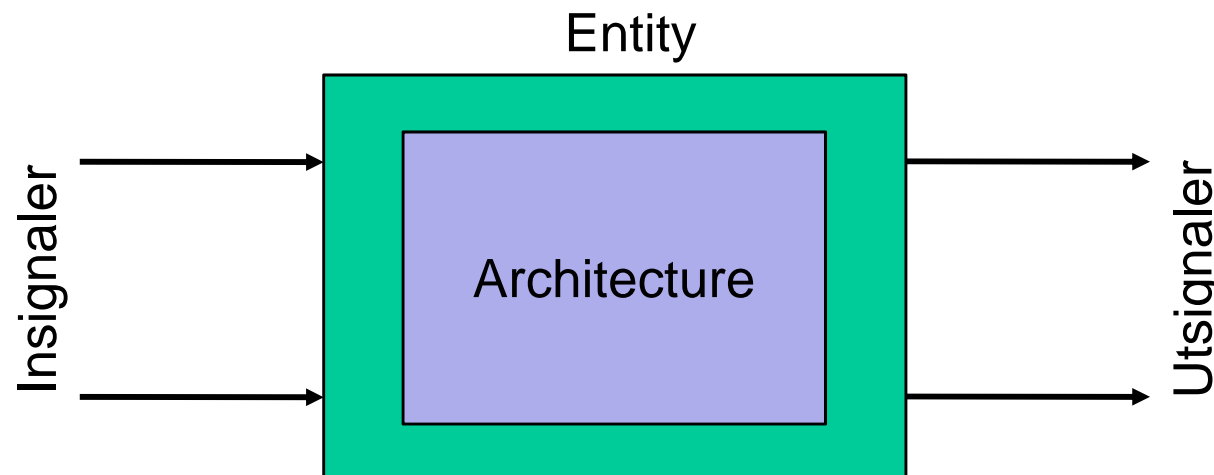


# CAD-verktyg

- 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

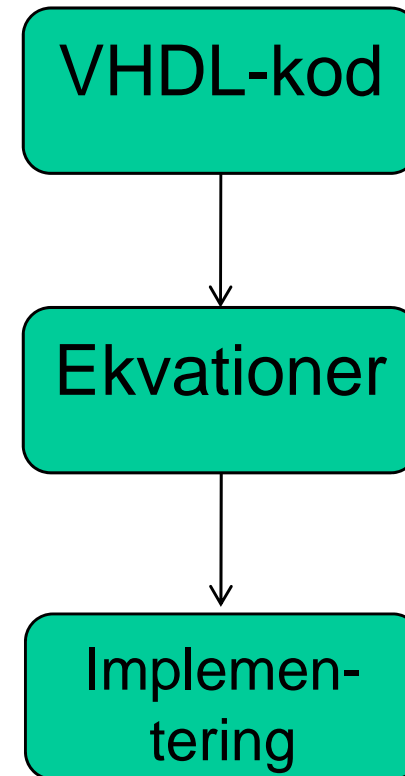
Designen beskrivs med *entity* (black box) och *architecture* (innehållet i boxen)



# 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!*

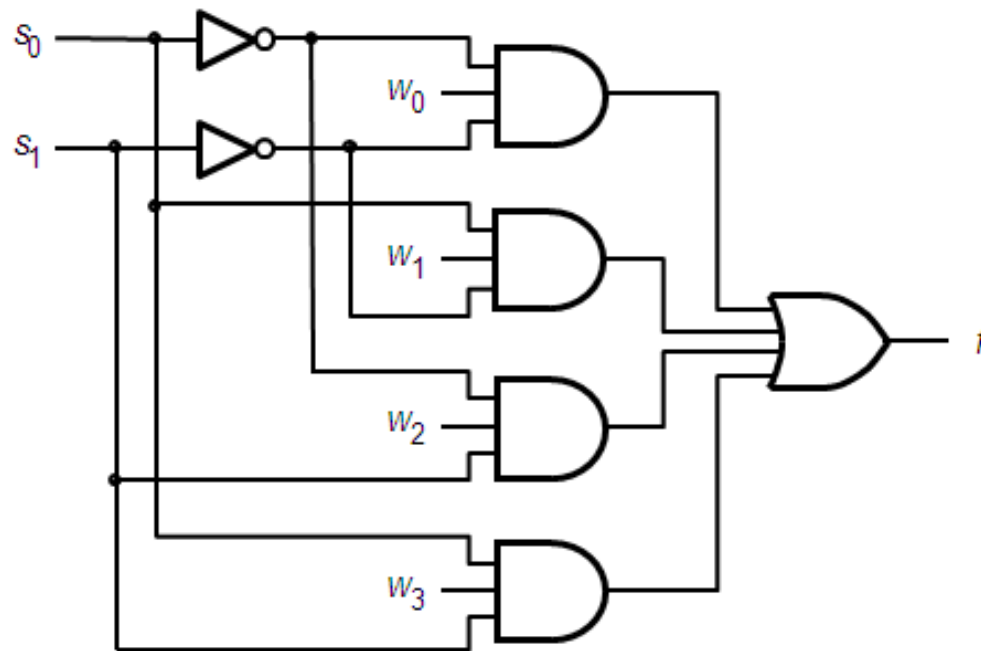
# Boolesk ekvation

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

• *Mer snart i kursen ...*

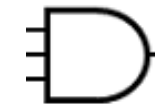


# Ett grindnät som implementerar funktionen



*Läroboken använder  
Amerikanska  
symboler:* ↓

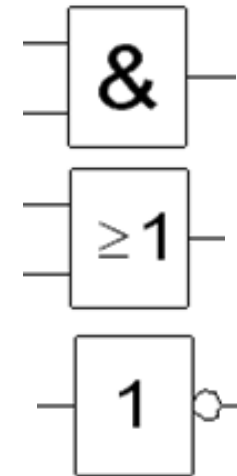
**AND**



**OR**



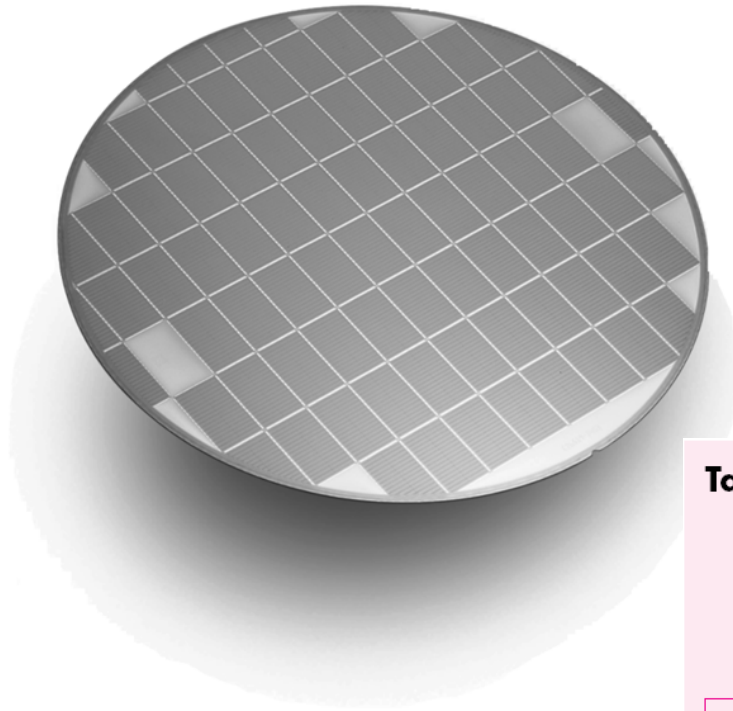
**NOT**



• *Mer snart i kursen ...*

↑  
Annat utseende på den  
internationella standarden:

# 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 <sup>2</sup>	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

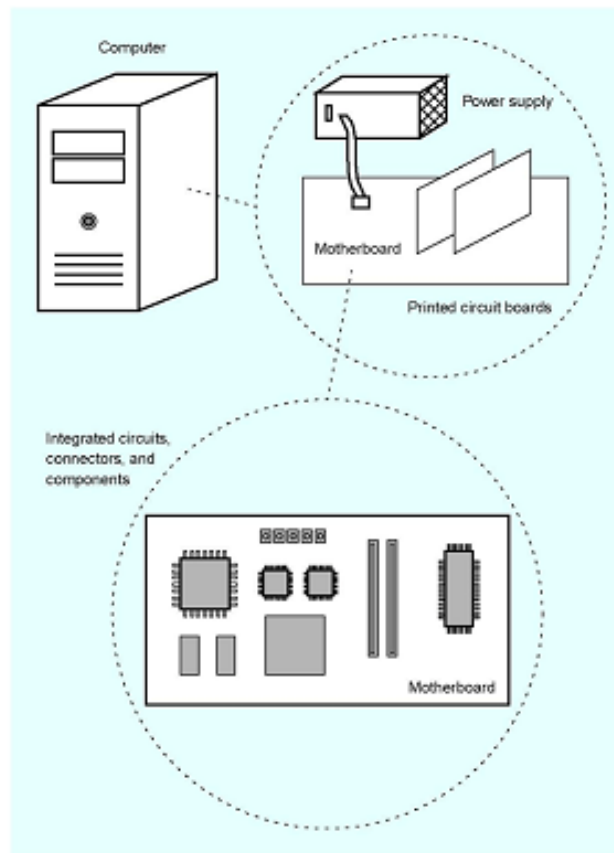
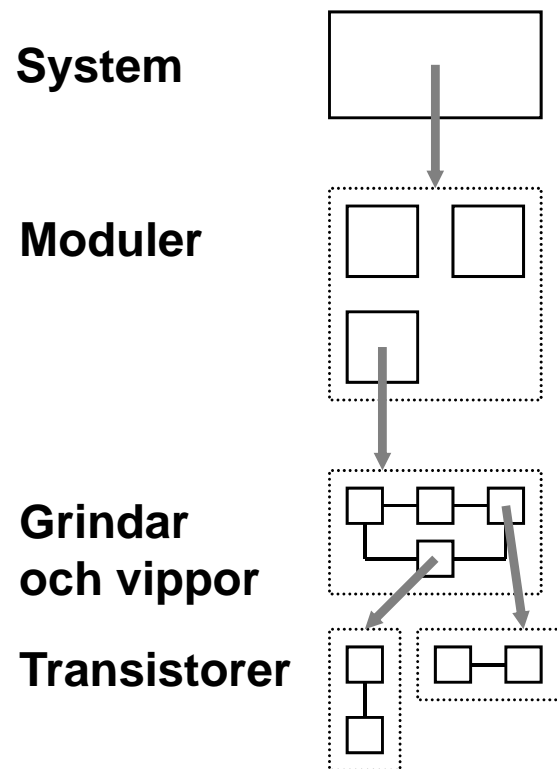


Figure 1.5. A digital hardware system (Part a).

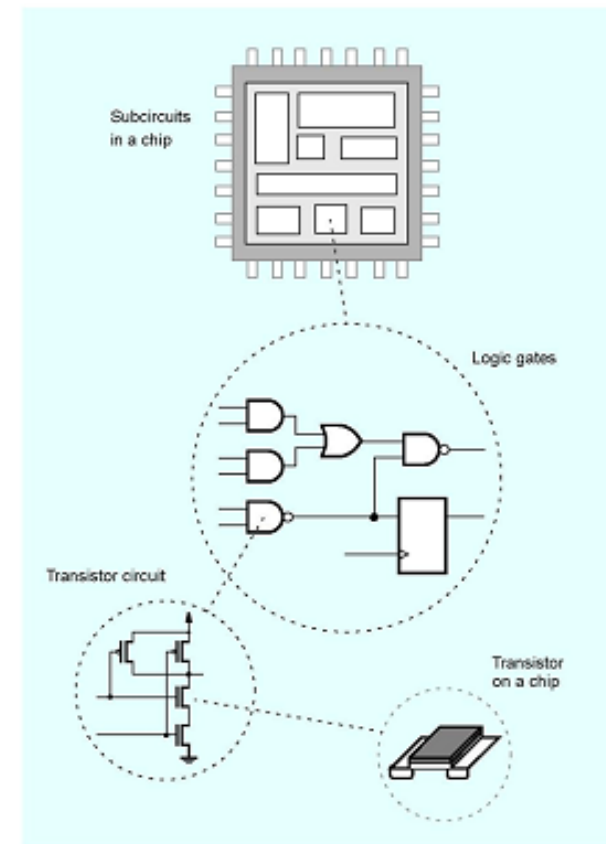
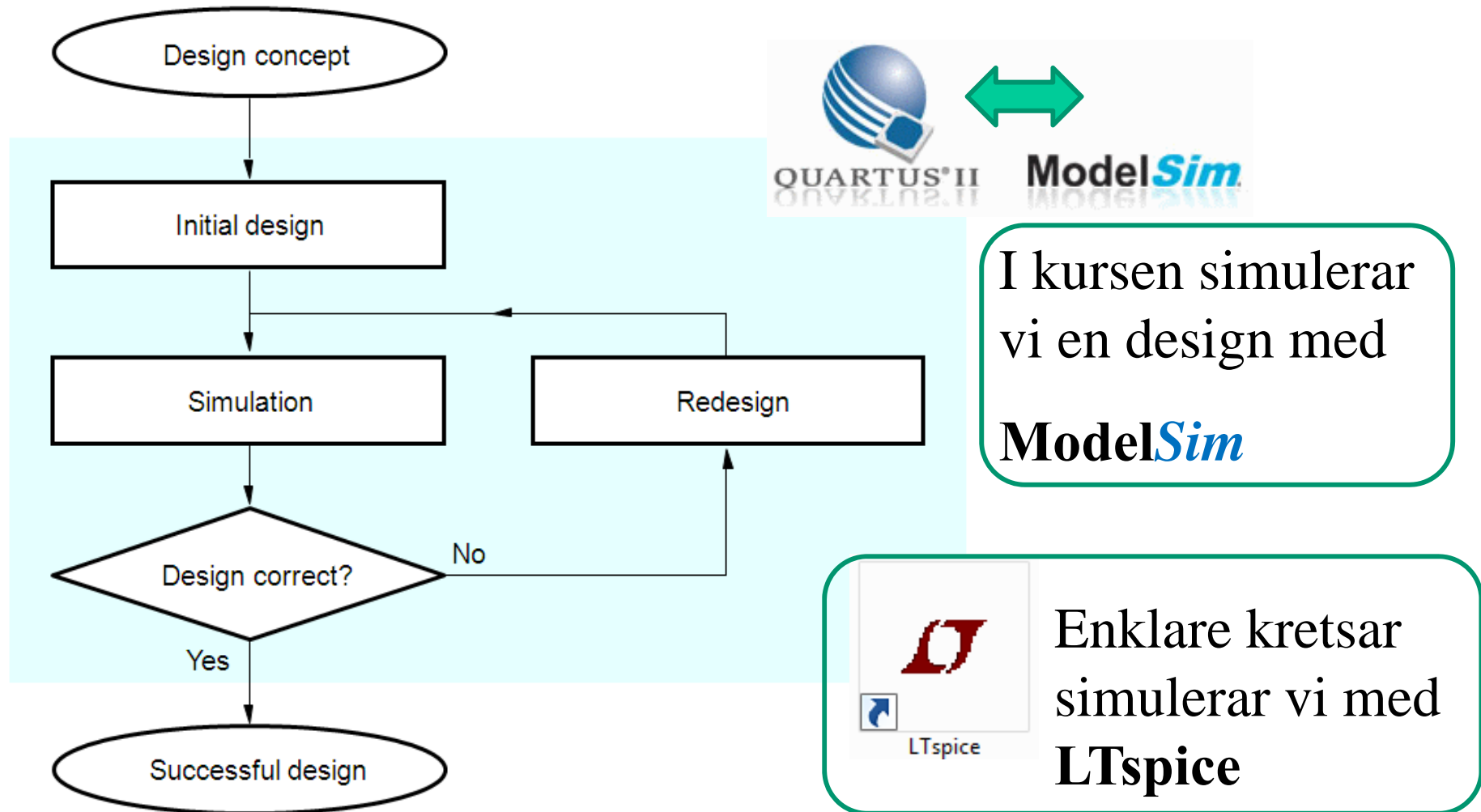
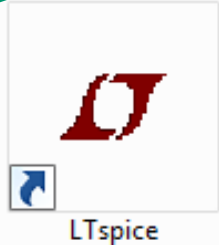


Figure 1.5. A digital hardware system (Part b).

# Designprocessen



# Simulering Lab1 LTspice



Enklare kretsar  
simulerar vi med  
**LTspice**

Ladda hem, och installera  
**LTspice** – behövs inför  
Lab1 och Lab2.

**LTspice** finns installerat i skolans datorsal Ka-309.

# Designprocessen Lab3



I kursen simulerar  
vi en design med  
**ModelSim**

Installera *inte* den version av **Quartus** som medföljer engelska läroboken.

-  Vi lånar ut USB-minnen med installationsfiler **senare i kursen.**

# 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  
 $m$  antal heltals siffror

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

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

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**!

$m$  antal binära heltalssiffror.

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

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

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

binärpunkt

# Oktala talsystemet

I det oktala talsystemet är basen **8** och därmed används siffersymbolerna **0** till **7**  $m$  antal oktala heltalssiffror

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

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



Siffer-  
displayer

# Hexadecimala talsystemet

I det hexadecimala talsystemet är basen **16** och därmed används siffersymbolerna **0** till **9** och **A** till **F**  $m$  antal hexadecimala hetalssiffror.

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

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



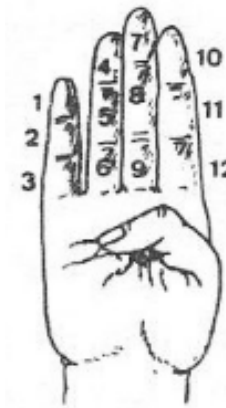
Sjusegment-  
displayer

# Talsystem med basen $b$

En allmän formulering kan erhållas för basen  $b$

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

- Karl den 12:e ville införa basen **12**!
- Sexagesimala talsystemet med basen **60** används fortfarande för tidmätning och till GPS:en – men inget för den här kursen!



# 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 bänkgranne!*



# 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 division med 2 (Bekvämare omvandlingssätt visas på övningen!)
  - Resten ger siffervärdet
  - Siffrorna kommer i omvänd ordning – Least Significant Bit (LSB) kommer först

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

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

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

$$6 \div 2 = 3 \text{ Rest } 0 \Rightarrow x_3 = 0$$

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

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

Klar!

$$x_5 x_4 x_3 x_2 x_1 x_0$$
$$53_{10} = 110101_2$$

# ( Omvandling mellan decimalbråk och binära tal )

Omvandlingen från decimalbråk till binärtal görs genom upprepad multiplikation med 2

- Heltalsdelen ger siffervärdet
- Siffrorna kommer i rätt ordning – Most Significant Bit (MSB) kommer först

$0,6875 \times 2 = 1,3750$	$1 + 0,3750$	$x_{-1} = 1$	$0.x_{-1}x_{-2}x_{-3}x_{-4}x_{-5}x_{-6}$ $0,6875_{10} = 0.1011_2$
$0,3750 \times 2 = 0,7500$	$0 + 0,7500$	$x_{-2} = 0$	
$0,7500 \times 2 = 1,5000$	$1 + 0,5000$	$x_{-3} = 1$	
$0,5000 \times 2 = 1,0000$	$1 + \boxed{0,0000}$	$x_{-4} = 1$	

— Klar! —

Inte säkert att det blir  $\boxed{0,0}$  – då fortsätter man tills man uppnått tillräckligt många binärer (bitar)

# Sammanfattning

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

# Nu kan Du tillräckligt för att lösa din första webbuppgift!

## Exercise: 1, question: 1

*Decimala, binära, hexadecimala och oktala tal* [Hjälp?](#)

Omvandla följande hexadecimaltal,  $7A_{16}$  till

- a) binärtal?
- b) oktaltal?
- c) decimaltal (heltal)?

Omvandla följande reella binärtal,  $1110.1100_2$  till

- d) reellt decimaltal (tal med decimaler, systemet kräver decimalpunkt när Du svarar)?

*Du kan skriva ner dina svar i tabellen för att ha det till hands senare.*

a) Svar: [ binärtal ]	b) Svar: [ oktaltal ]	c) Svar: [ decimalt heltal ]	d) Svar: [ reellt decimaltal ]
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

- Gör det!
- Vi kollar vid nästa föreläsning ...

[DD1-001](#) / [Exercise 1](#) / [Question 1](#)

a)

b)

c)

d)

[Submit](#)

