IE1204 Digital Design

L1 : Course Overview. Introduction to Digital Technology. Binary Numbers

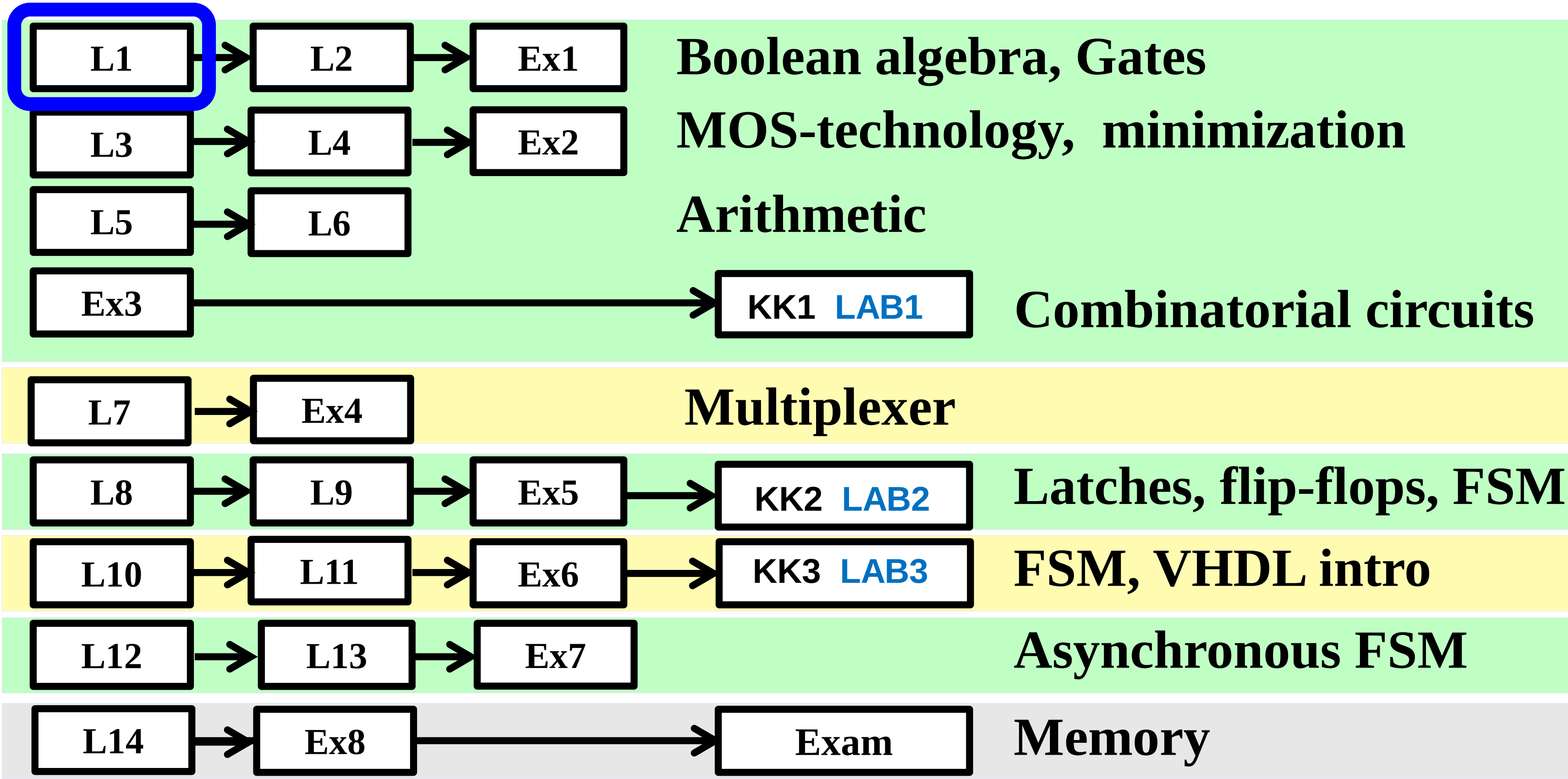
KTH Informations- och kommunikationsteknik

Masoumeh (Azin) Ebrahimi

KTH/ICT

mebr@kth.se

IE1204 Digital Design



Course Group- Responsible

William Sandqvist, University Teacher

School of Information and Communication Technology (ICT)

william@kth.se

Course Group- Lectures

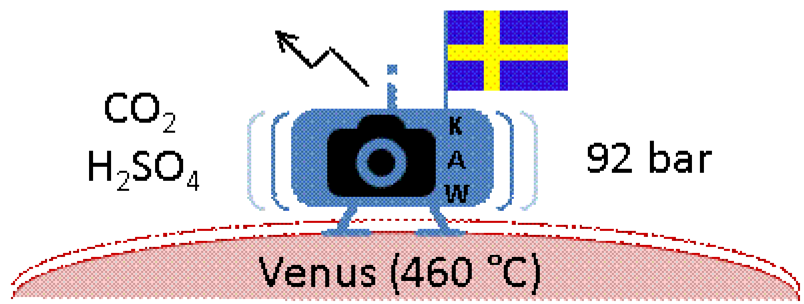
Masoumeh (Azin) Ebrahimi, Senior Researcher
School of Information and Communication Technology (ICT)
http://people.kth.se/~mebr, mebr@kth.se

My research area:

- Fault-Tolerant and Reliable Many-Core Systems
- eHealth IoT Applications
- Machine Learning and Data Analytics in Wearable Devices

Course Group- Exercises

- Muhammad Shakir, mshakir@kth.se
  - His research is about design and high temperature characterization of a Central Processing Unit (CPU) in Silicon Carbide (SiC) for Venus Landers applications.



- First exercise session is on 30/9 10-12 in sal 210.

Studying at university …

- Is very different than in high school!
- That the tempo is much higher at KTH than in a high school!
- Means that you take responsibility for your studies!
- An engineering degree provides excellent opportunities for a well-paid, fun and stimulating job, but first you have to get there!

Contact with KTH-Personal

- The lecturer is available during the break and for some time after the lecture
  - Take advantage of this time
- Email
  - KTH staff is flooded by too many emails
  - If necessary, send a plain and clear message to the right person
  - Use your KTH mail address!
  - Put "IE1204: ...” in the subject of your message

KTH Informations- och kommunikationsteknik

This course...

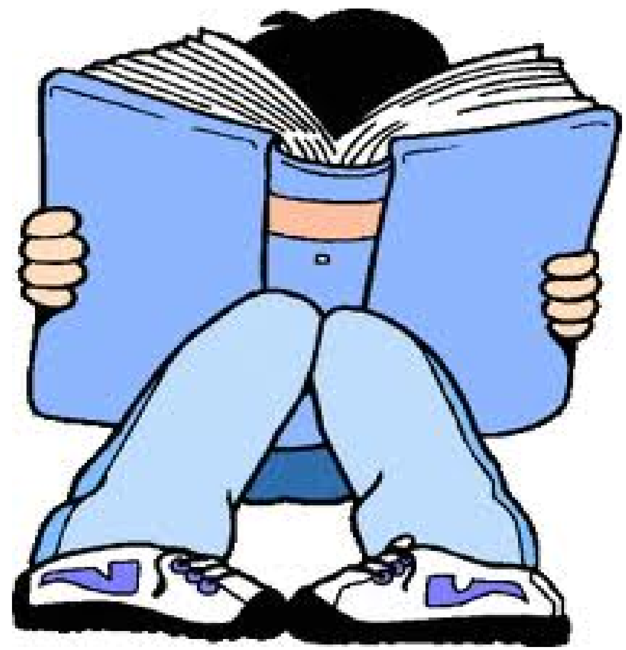
Course aims

- To teach the *theoretical foundations* for the analysis and synthesis of combinational and sequential digital circuits
- Through *practical problem solving*, provide an understanding of various design phases
  - to enable the students *designing* simple combinational and sequential digital systems

There is a course book…

- Brown/Vranesic, Fundamentals of Digital Logic with VHDL Design (3rd edition), Mc-Graw-Hill, 2009. (abbreviated as BV at webpage)

Read it!



Course book

- From 2014-2016
  - More materials are available on course website

    https://www.kth.se/social/course/IE1204/
  - Lecture slides from previous years are on this website
- From 2017,
  - The couese will be maintained in Canvas

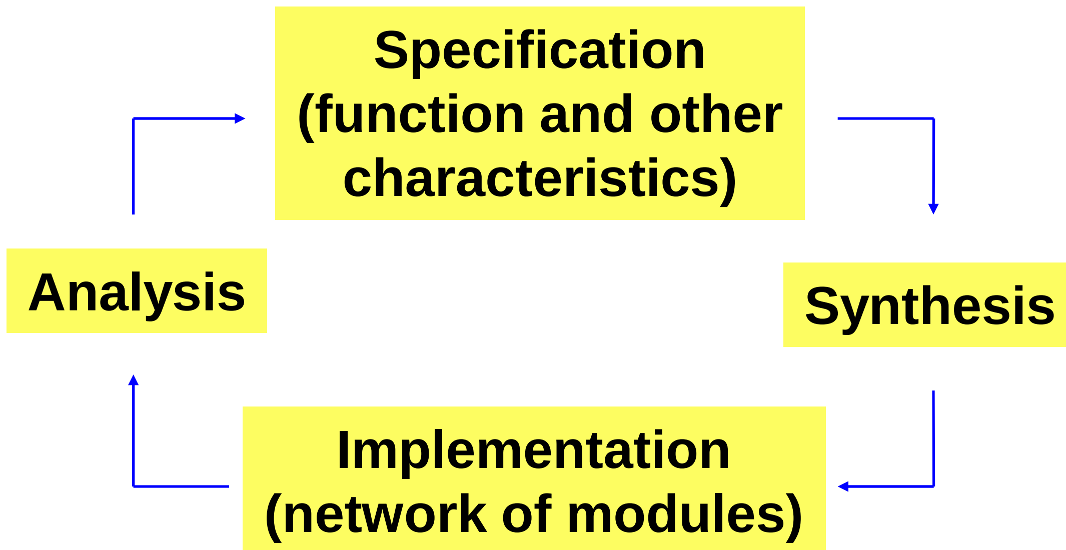
    https://www.kth.se/social/course/IE1204/page/folj-kursen-pa-canvas/
  - Lecture slides will be updated in Canvas

This lecture

- BV pp. 2-4, 8-12, 17-20

Course aims (cont.)

- To teach the students a design methodology



- To confirm that this methodology enables the design of simple digital systems

Course aims (cont.)

- Introduce students to *English textbooks*
  - *All good literature* on the subject is in English
  - English is the *working language* in all major Swedish international companies
  - Speaking English (reasonably) fluent is a prerequisite for a successful career as a civil engineer
- Therefore, we have chosen an English course book!

Examination and structure

- Examination
  - LABA, 3.0 hp
    - Grade: G/U
  - TENA, 4.5 hp
    - Grade A-E/F
- Lectures - 28 h (14 x 2h)
- Exercises - 16 h (8 x 2h)
- Labs - 12 h (3 x 4h)

Lab-times in Canvas

- Located in *room 305*
- *Each student is randomly given a lab time* in Canvas. At these times you are guaranteed your place.
- If the time is inappropriate then send an email to William Sandqvist (william@kth.se) to rebook.
  - NOTE! We do not make late rebooking during the lab weeks, so inform well in advance before the lab.
  - After the lab has begun we know how many students have come - there may be some vacancies. (We can not guarantee this)

Knowledge control

- Labs start sharp at the given time
- To get a lab you must first make a mandatory knowledge control (Web-based)
- You will find your "number" in Canvas
- The same "number" is then used for knowledge control for all three labs

Knowledge control

- Note that knowledge control and preparation for labs are very time consuming
- You are therefore advised to start working on these as soon as possible!
- Right after each lecture find the relevant questions and try to answer them...

Final exam

- Look at the schedule…
  - https://www.kth.se/social/course/IE1204/calendar/
- Registration is required
  - No later than two weeks before the exam!

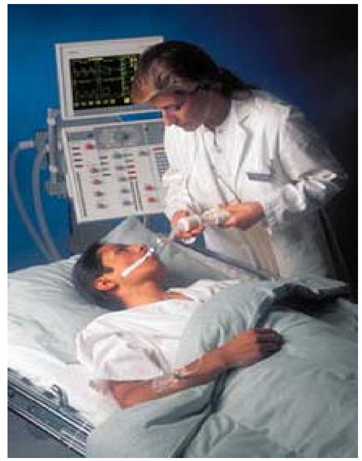
Course content overview

- Specification of the digital functions and systems
- Digital building blocks
- Digital Arithmetic
- Synchronous circuits and state machines
- Asynchronous circuits and state machines
- Larger digital systems: processors and computers
- We will not go through VHDL to any great extent - it is a complete course by itself

KTH Informations- och kommunikationsteknik

Why is digital technology important?

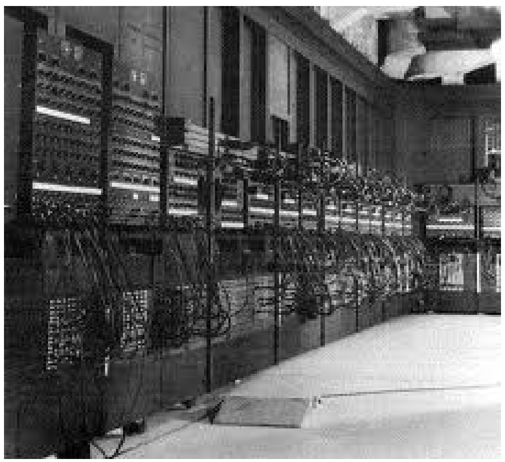
Our daily life is affected by digital technology



- Computers are everywhere and in all kinds of products
  - Did you know that there are about 40 to 100 microprocessors in a new car?

Development of electronics

Technological progress allows more and more functionality to be integrated on a single chip



ENIAC

(1946)

Apple II

(1977)

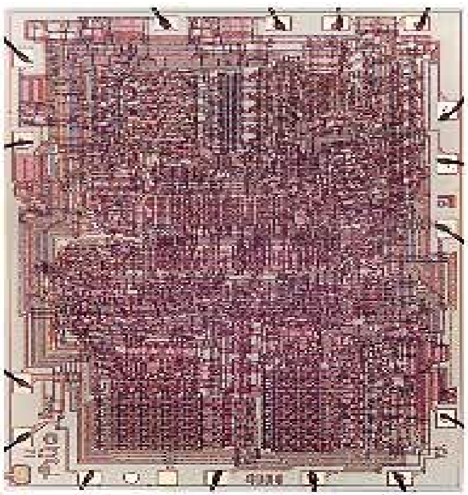
iPhone 7

(2016)

The trend continious!

Development of electronics

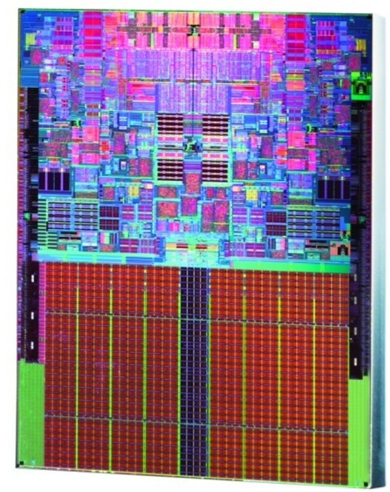
Intel 4004 (1971)



108 KHz
2,300 transistors

Intel Xeon 5400 (2008)

3.0 GHz
820 millions of transistors



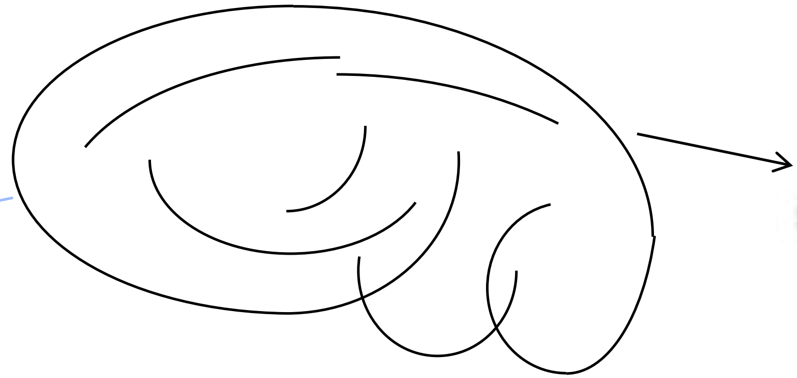
If we had a corresponding development of cars, we would now be able to drive from San Francisco to New York in about 13 seconds (Intel).

Advances in technology

https://en.wikipedia.org/wiki/Transistor_count

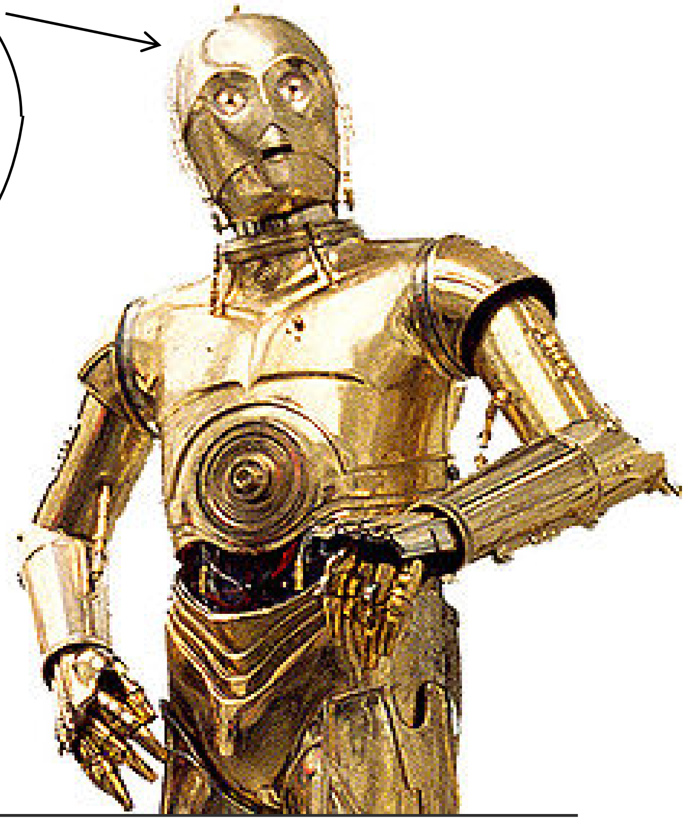
Processor	Transistor count	Year	Designer	Process	Area
Intel 4004	2,300	1971	Intel	10,000 nm	12 mm^2
Motorola 68020	190,000	1984	Motorola	2,000 nm	85 mm^2
Pentium	>3 Million	1993	Intel	800 nm	294 mm^2
Pentium II Klamath	>7 Million	1997	Intel	350 nm	195 mm^2
Itanium 2 McKinley	>200 Million	2002	Intel	180 nm	421 mm^2
AMD K8	>100 Million	2003	AMD	130 nm	193 mm^2
Cell	>200 Million	2006	Sony/IBM/Toshiba	90 nm	221 mm^2
POWER6	>700 Million	2007	IBM	65 nm	341 mm^2
Six-core Opteron 2400	>900 Million	2009	AMD	45 nm	346 mm^2
8core POWER7	>2 Billion	2012	IBM	32 nm	567 mm^2
Apple A10 (mobile SoC)	>3 Billion	2016	Apple	16 nm	125 mm^2
24-core AMD EPYC	>19 Billion	2017	AMD	14 nm	195 mm^2

Examples of digital systems (2022)



Robotic Brain

10x10x10 Chips containing a total of ~6350 Brain Processing Units (BPUs) & ~100 TB DRAM memory



Digital technology has created the basis for this development

- Simple mathematical model
  - Just 1s and 0s as values
  - Boolean algebra (George Boole 1815-1864, English)
  - Efficient implementation of the mathematical model

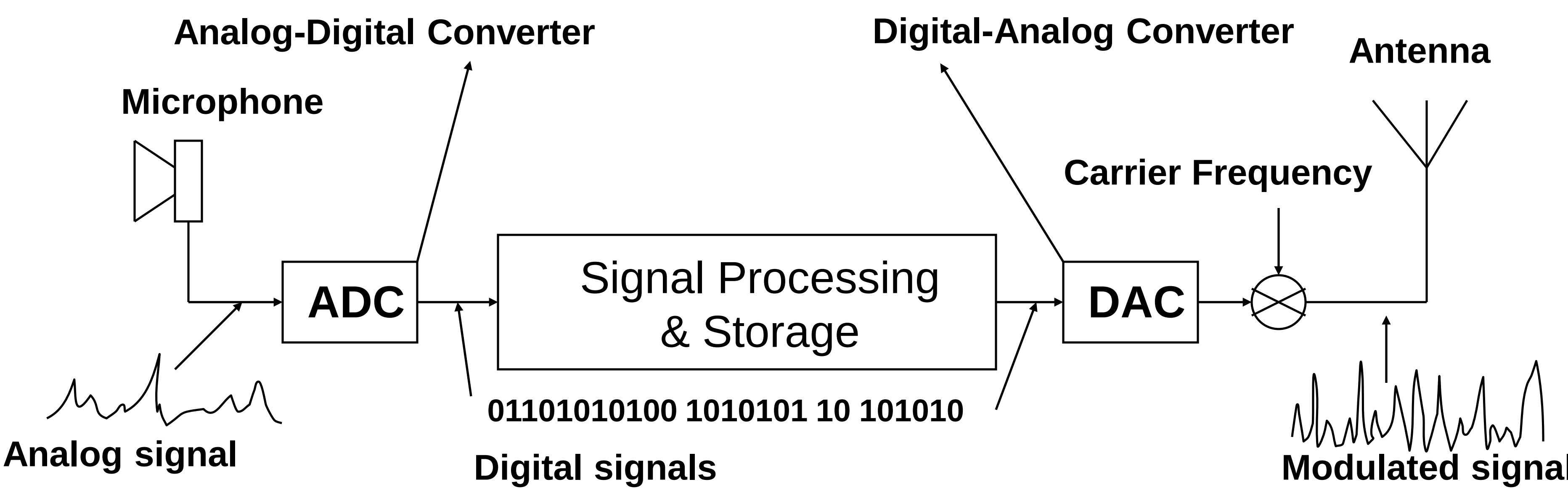
- Transistor-based integrated circuits
  - Progress in semiconductor technology
  - Effective design methods and tools

KTH Informations- och kommunikationsteknik

Why digital?

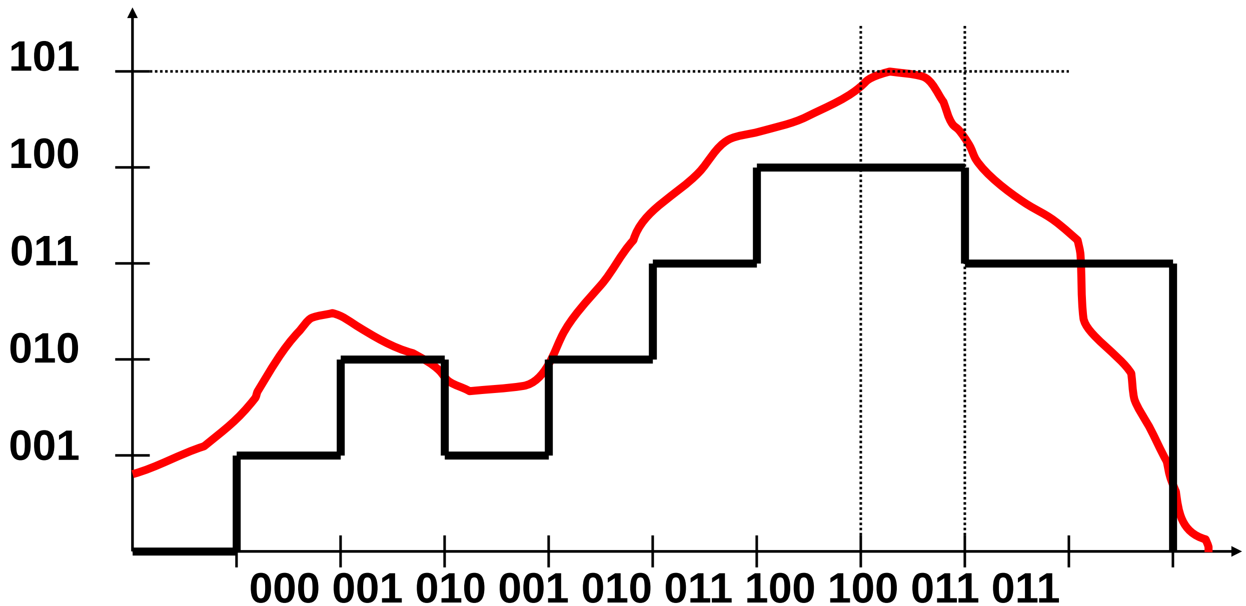
Analog & digital signals

- Most of the signal processing today is done digitally



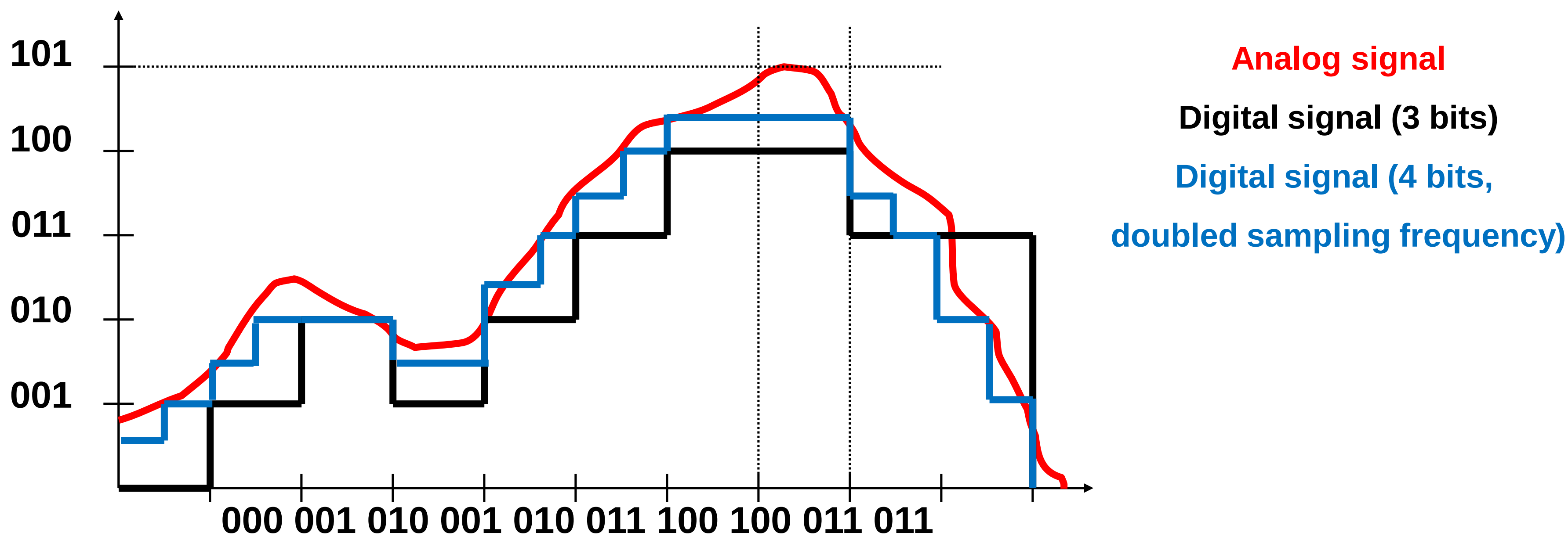
Analog vs. digital

- Instead of an analog signal that can assume *continuous* values, a digital signal only assumes *discrete* values



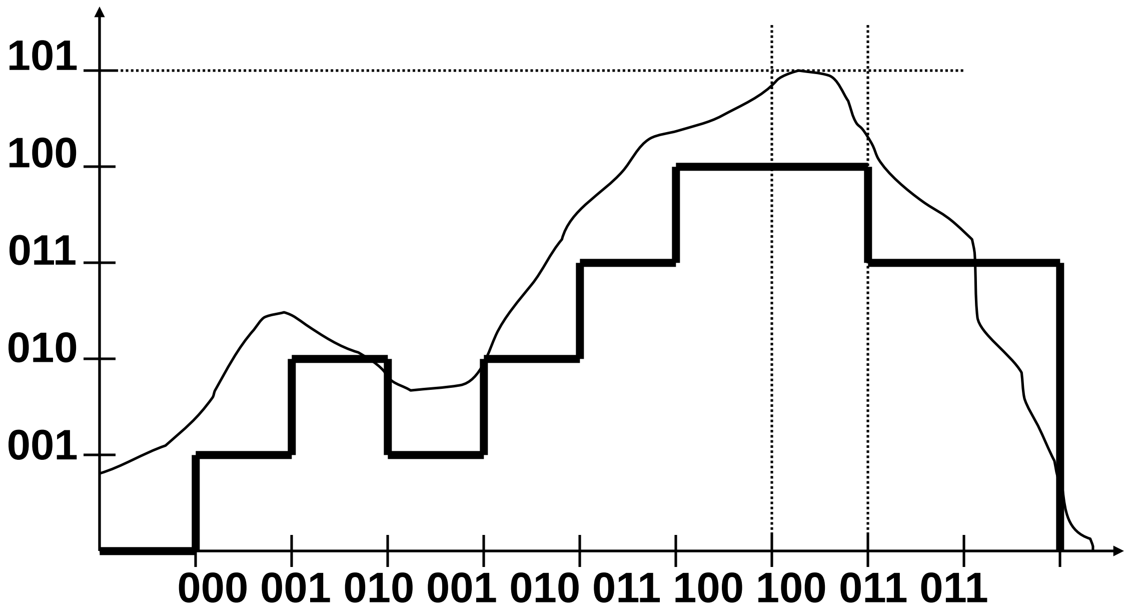
More bits and higher sampling increase signal quality

- If you have enough bits and sufficiently high sampling frequency, a digital signal can efficiently mimic the analog signal



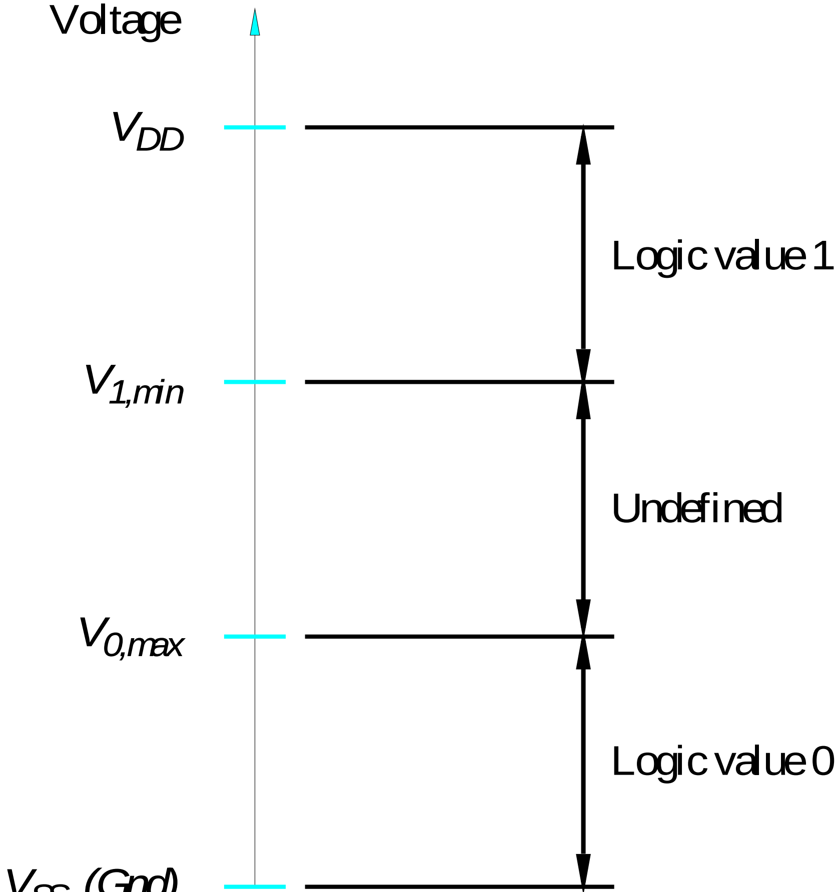
Quantization and digitalization

Quantization Levels



Digital signal sequence

Digital technology is very insensitive to noise

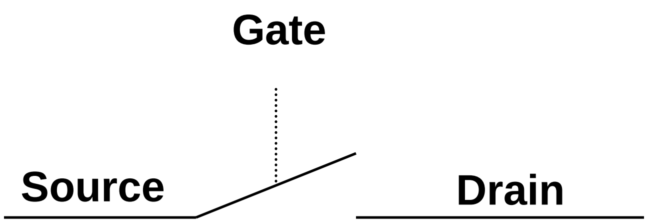


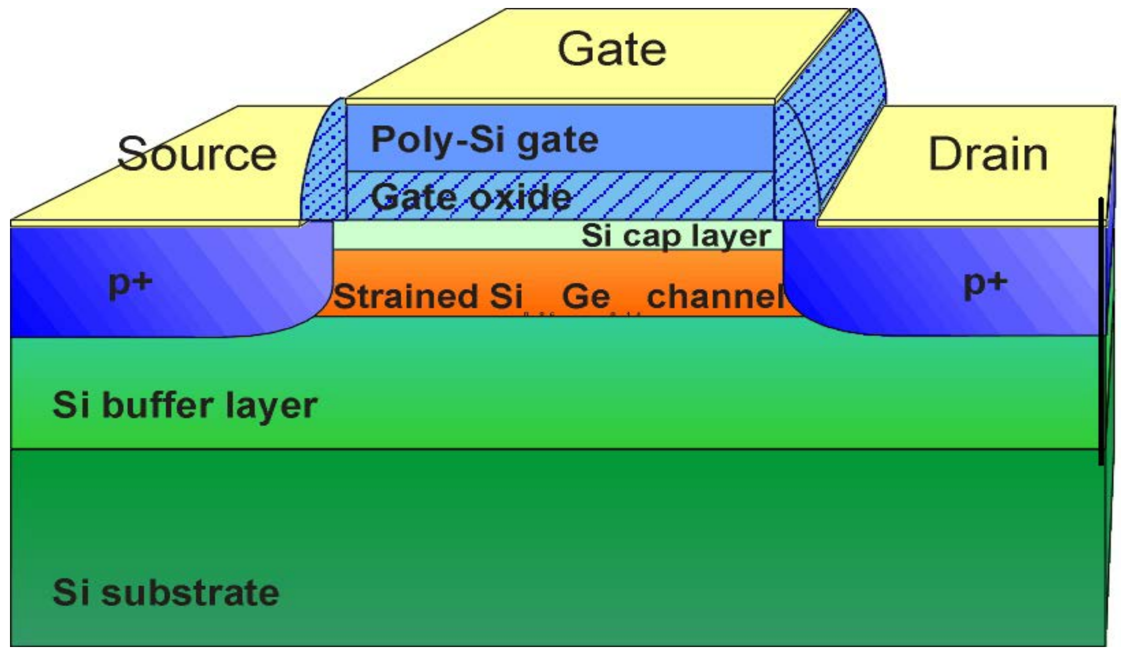
- Not only *one* voltage value is interpreted as 1 or 0, but an *interval* of voltage values
- A deviation of a few mV can be very distracting in analog technology, but makes no difference in digital technology

Why binary?

- CMOS transistors are
  - Cheap (made from ordinary sand)
  - Reliable
  - Efficient
- It is easy to make a transistor work as a switch

Transistor - switch with no moving parts





Schematic diagram of the SiGe transistor (KTH)

Mathematical model - Boolean algebra: axioms

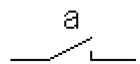
- In Boolean algebra, there is only 1 (True) and 0 (false) as values
- The following operations are defined: AND, OR, NOT
- The following axiom defines the Boolean algebra

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 $\overline{x} = 1$	(4b) If $x = 1$, then $\overline{x} = 0$

The technical background

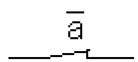
Boolean algebra was used as a tool to calculate how the networks could be simplified.

Making contact,



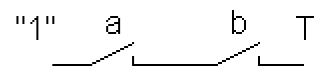
$T = a$

Breaking contact, ”not”



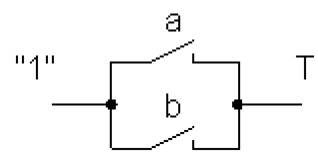
$T = \overline{a}$

$\cdot$ for serial connection, the ”and” -function



$T = a \cdot b$

$+$ for parallell connection the ”or”-function



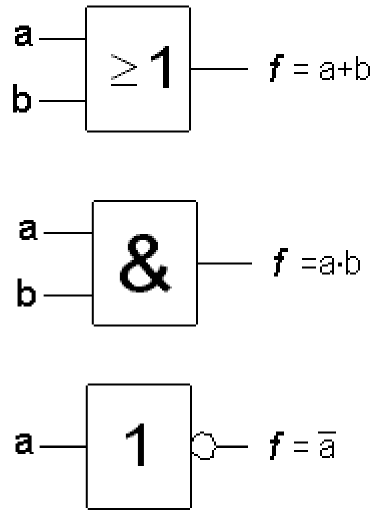
$T = a + b$

Gates instead of switches

\+ ”or”-gate

• ”and”-gate

”not” inverter

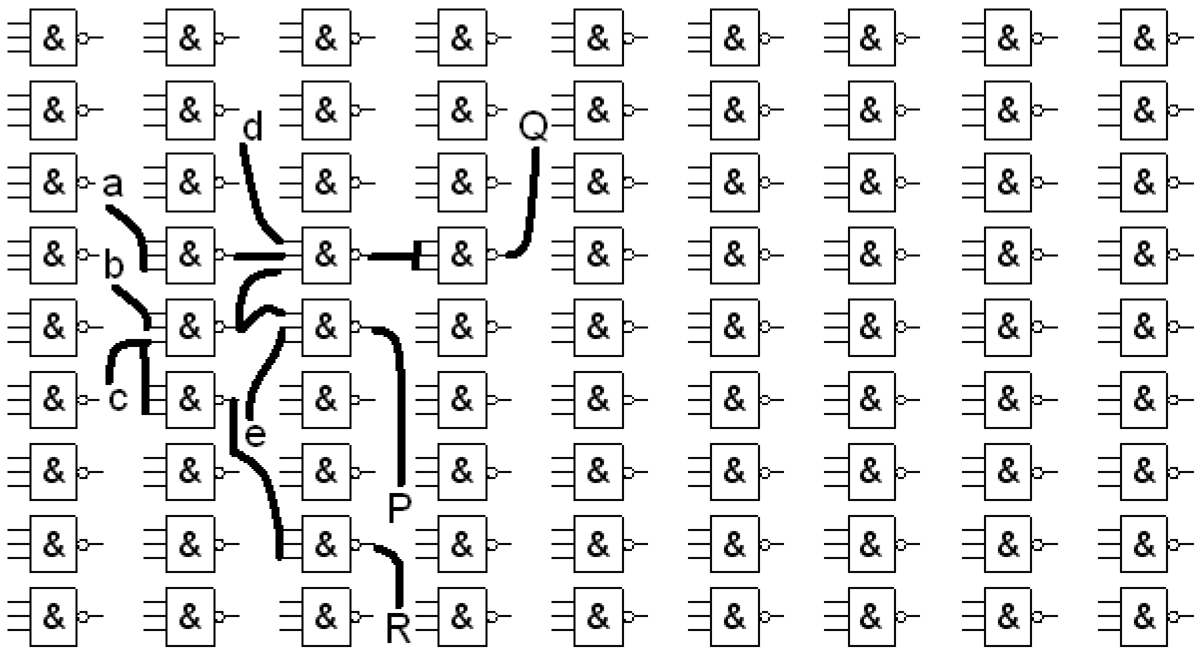


The switches were replaced by "gates" made of electronic components

With these three basic gate types: OR AND NOT can all logical functions be performed.

• *More soon in the course ...*

See-of-Gates



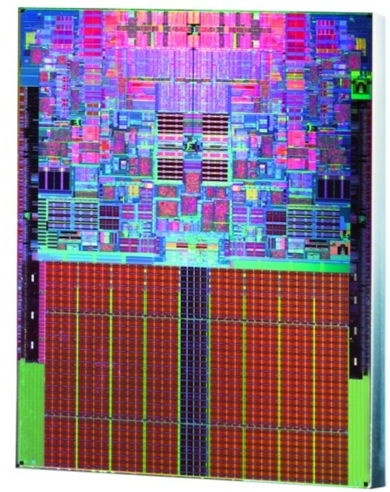
With a single type of gate (e.g. NAND), you can do anything with a "sea of gates".

KTH Informations- och kommunikationsteknik

Have we can handle millions of transistors in a design?

A processor contains many gates ...

- Very inefficient to describe it by drawing a network of gates
- We need other methods to describe a system!



Digital hardware in a computer

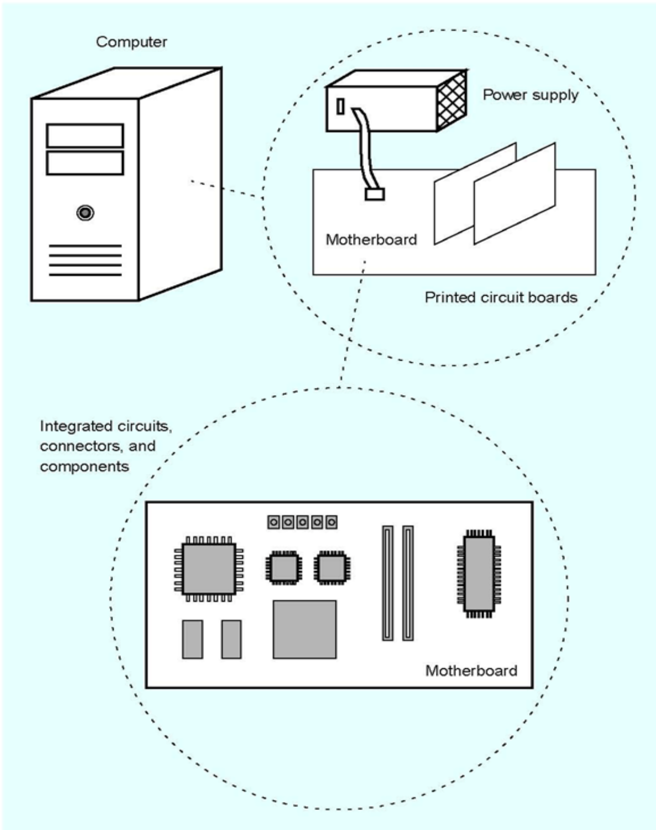


Figure 1.5. A digital hardware system (Part *a*).

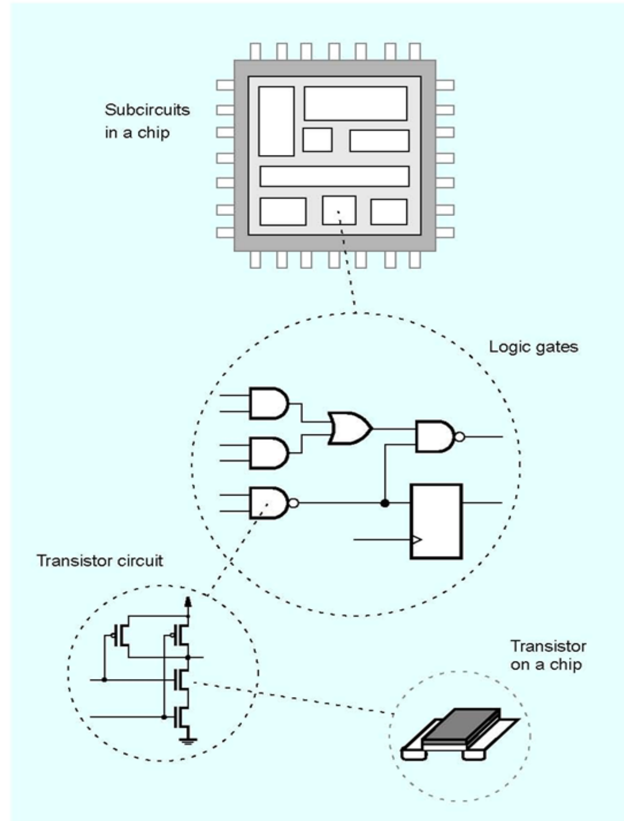
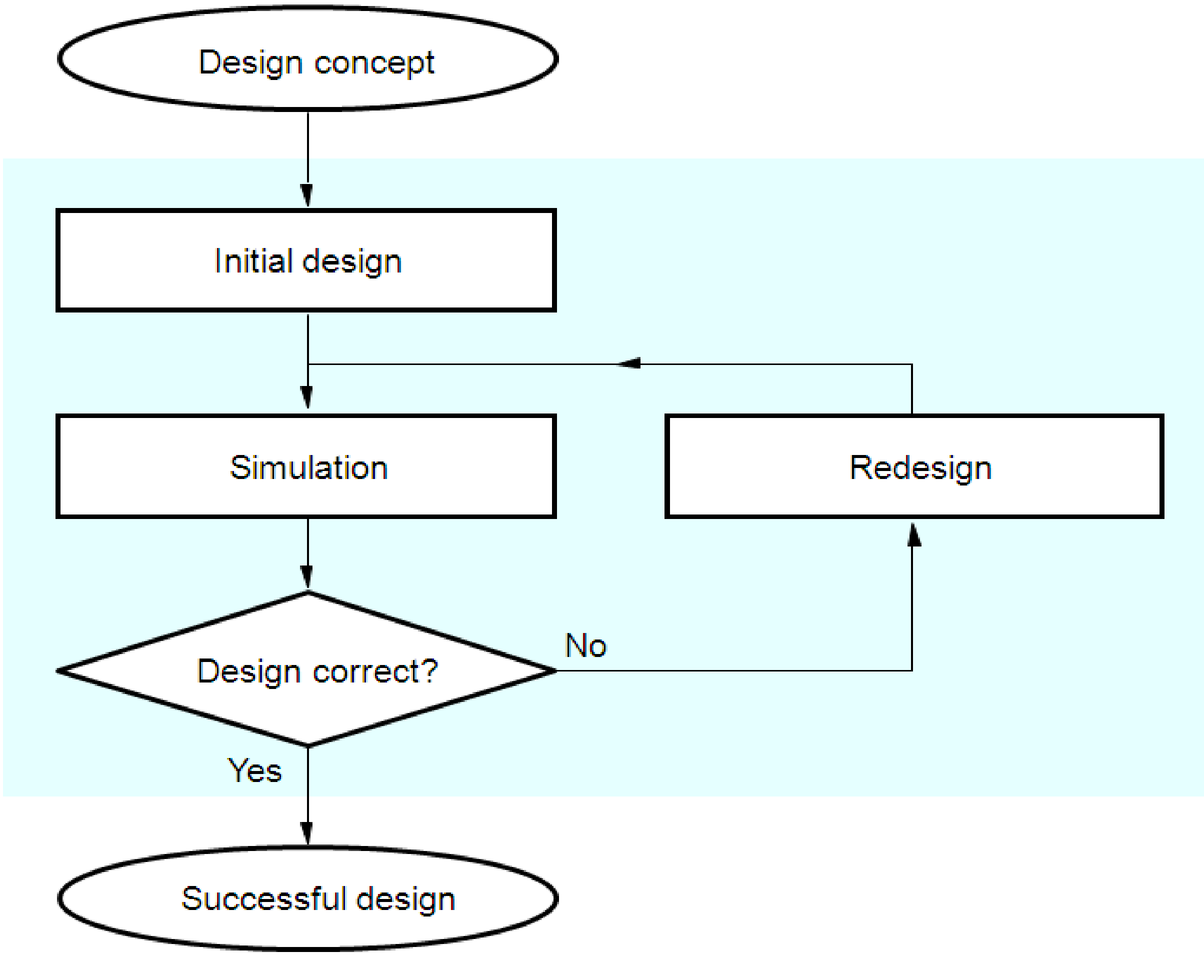


Figure 1.5. A digital hardware system (Part *b*).

Simplified Design Process



CAD tools

- CAD tools are capable of handling the complexity (in term of the total number of transistors)
- They help us to *design highly efficient integrated circuits* for a broad spectrum of applications
  - Computers
  - Telecommunications (switches, routers, mobile)
  - Transport industry (aerospace, automotive)
  - The entertainment industry
  - …

CAD tools (cont.)

- A CAD tool is *a program that helps an engineer to design (e.g. an integrated circuit)*
- CAD tools can *be fully automated or interactive*
- CAD tools are based on *algorithms that define the order of steps* to be applied

CAD tools (cont.)

- CAD tools are not only used for *synthesis* of circuits, but also for the *analysis* of circuits
  - E.g. to simulate a circuit description in order to *analyze its time delays, how much power it consumes*, etc.

Hardware description language (HDL)

- Hardware Description Languages (HDLs) is a programming language that makes it possible to *describe the hardware* features in a natural way
- In contrast with normal programming languages, like C, where a code is executed sequentially, HDL languages *execute a code in parallel*
- HDL descriptions *can be used as an input to a "synthesizer"*, a type of compiler which *produces "executable code" for hardware*

KTH Informations- och kommunikationsteknik

But now we take it from the beginning ...

Binary numbers

- Digital technology uses only two numerical symbols: 0 and 1
  - Easy to implement - each value corresponds to a voltage level, e.g.
    - 0 volts correspond to 0
    - 3 volts correspond to 1

How can we represent ordinary decimal numbers?

Decimal number system

- Decimal number system has 10 different numeric symbols: 0-9
- A decimal number is represented with a sequence of numeric symbols
  - The *position in the sequence gives the digit weight* which is multiplied by a power of 10 (*base* in the decimal system is 10)

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

Decimal number system

- Representation of an integer in decimal

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

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

- Representation of a number with ”comma”

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

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

Binary system

- The binary system works in the same way as the decimal system, but uses base 2 (symbols 0-1) instead of 10 (symbols 0-9)

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

2^2 2^1 2^0

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

2^1 2^0 2^{-1} 2^{-2}

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

Binary system

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

2	10
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16

Binary to Decimal

$$(1011010)_2 = (?)_{10}$$

$$(1011010)_2 = 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0 = (90)_{10}$$

Octal number system

- The octal number system uses base 8 and numerical symbols 0-7

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

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

Octal number system

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

2	8	10
1000	10	8
1001	11	9
1010	12	10
1011	13	11
1100	14	12
1101	15	13
1110	16	14
1111	17	15
10000	20	16

Octal to Decimal

$$(567)_8 = (?)_{10}$$

$$(567)_8 = 5 \cdot 8^2 + 6 \cdot 8^1 + 7 \cdot 8^0 = (375)_{10}$$

Binary to Octal

$$(1011010)_2 = (?)_8$$

$$(1011010)_2 = (1\ \ 011\ \ 010) = (132)_8$$

Hexadecimal number system

- The hexadecimal number system uses base 16 and numerical symbols 0-9 and letters A to F

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

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

Hexadecimal number system

2	8	10	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

2	8	10	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

Hexadecimal to Decimal

$$(1AE)_{16} = (?)_{10}$$

$$(1AE)_{16} = 1 \cdot 16^2 + 10 \cdot 16^1 + 14 \cdot 16^0 = (430)_{10}$$

Binary to Hexadecimal

$$(1011010)_2 = (?)_{16}$$

$$(1011010)_2 = (101\ \ 1010) = (5A)_{16}$$

Number systems with base *b*

- A general formulation can be obtained for the base *b*

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

Integers in different number systems

2	8	10	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

2	8	10	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

Conversion between decimal and binary numbers

- Conversion from binary to decimal is trivial
- Conversion from decimal to binary can done by repeatedly dividing by 2
  - The remainder gives us the numerical value
  - The numbers are in reverse order - Least Significant Bit (LSB) comes first

Quotient Remainder

$$
\begin{array}{ccccccr}
53 \div 2 & = & 26*2+1 & \Rightarrow & x_0 = 1\,(\text{LSB}) & & \mathbf{1} \\
26 \div 2 & = & 13*2+0 & \Rightarrow & x_1 = 0 & & \mathbf{01} \\
13 \div 2 & = & 6*2+1 & \Rightarrow & x_2 = 1 & & \mathbf{101} \\
6 \div 2 & = & 3*2+0 & \Rightarrow & x_3 = 0 & & \mathbf{0101} \\
3 \div 2 & = & 1*2+1 & \Rightarrow & x_4 = 1 & & \mathbf{10101} \\
1 \div 2 & = & 0*2+1 & \Rightarrow & x_5 = 1\,(\text{MSB}) & & \mathbf{110101}
\end{array}
$$

Decimal to Binary

$$(35)_{10} = (?)_2$$

$$\begin{aligned}
35 \div 2 &= 17*2+1 &\Rightarrow\quad& x_0 = 1\,(\mathrm{LSB}) \\
17 \div 2 &= 8*2+1 &\Rightarrow\quad& x_1 = 1 \\
8 \div 2 &= 4*2+0 &\Rightarrow\quad& x_2 = 0 \\
4 \div 2 &= 2*2+0 &\Rightarrow\quad& x_3 = 0 \\
2 \div 2 &= 1*2+0 &\Rightarrow\quad& x_4 = 0 \\
1 \div 2 &= 0*2+1 &\Rightarrow\quad& x_5 = 1\,(\mathrm{MSB})
\end{aligned}$$

$$(35)_{10} = (100011)_2$$

Summary

- **Design Methodology**
  - **Digital Technology**
  - **CAD tools**
    - Necessary to manage the complexity of large designs
  - **Hardware description language (HDL)**
    - It takes too long to construct a gate-level circuit ”by hand”
    - By describing a circuit at a higher level, we increase the productivity level (gates/hour) of a designer.

Summary

- There are many different number systems
- Digital technology uses the binary number system
- It is possible to convert numbers between different number systems