EE1D1: Digital Systems A

BSc. EE, year 1, 2025-2026, lecture 1

Introduction to Digital Systems

Computer Engineering Lab

Faculty of Electrical Engineering, Mathematics & Computer Science

TU Delft

Delft University of Technology

Outline

Introduction

- Instructors
- Computer Engineering
- Importance of Digital Systems
- Course Overview

Digital Systems

- Digital versus analog
- Binary Systems and Boolean Algebra
- Data representation
- Summary

Instructors: Arjan van Genderen

- Education
  - 1985 MSc. Electrical Engineering, TU Delft
  - 1991 PhD, TU Delft
- Lecturer for
  - Digital Systems A
  - Digital Systems B
  - IP-2 Smart Robot Challenge
  - Chip Design
- MSc. Coordinator for
  - Master Computer and Embedded Systems Engineering

A.J.vanGenderen@tudelft.nl
Room HB 09.080
Computer Engineering,
Faculty EEMCS, TU Delft

Instructors: Mansureh Shahraki Moghaddam

- Education
  - 2015: PhD from Indian Institute of Technology Delhi

- Lecturer:
  - Digital Systems A
  - Digital Systems B
  - IP-2 Smart Robot Challenge
  - Chip Design
  - Computer Architecture and Organization

M.ShahrakiMoghaddam@tudelft.nl
Room HB 09.080
Computer Engineering,
Faculty EEMCS, TU Delft

Instructors: Ton Slats

- EE Lab staff member:
  - Digital Systems A
  - Digital Systems B
  - IP-2 Smart Robot Challenge
  - Computer Architecture and Organization

A.M.J.Slats@TUDelft.nl

LB 01.260

Electrical Eng. Education
Faculty EEMCS, TU Delft

Importance of Digital Systems

- Digital Systems are everywhere.
- Examples
  - Smartphone
  - PC, game console
  - Communication
  - Industry
  - Household
  - Transport
  - Health
  - ...

Digitals systems = Hardware (HW) + Software (SW)

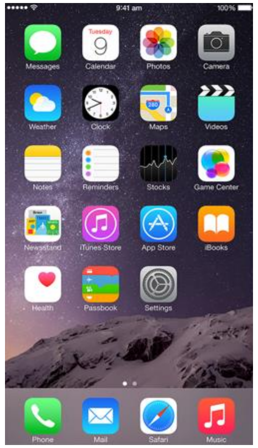
Computer scientists: SW (+ HW) (e.g. databases, compilers)
Electrical engineers: HW+SW (e.g. instruments, chips)

Importance of Digital Systems

- iPhone 16

Hardware

+



Software
- apps
- ios
- drivers

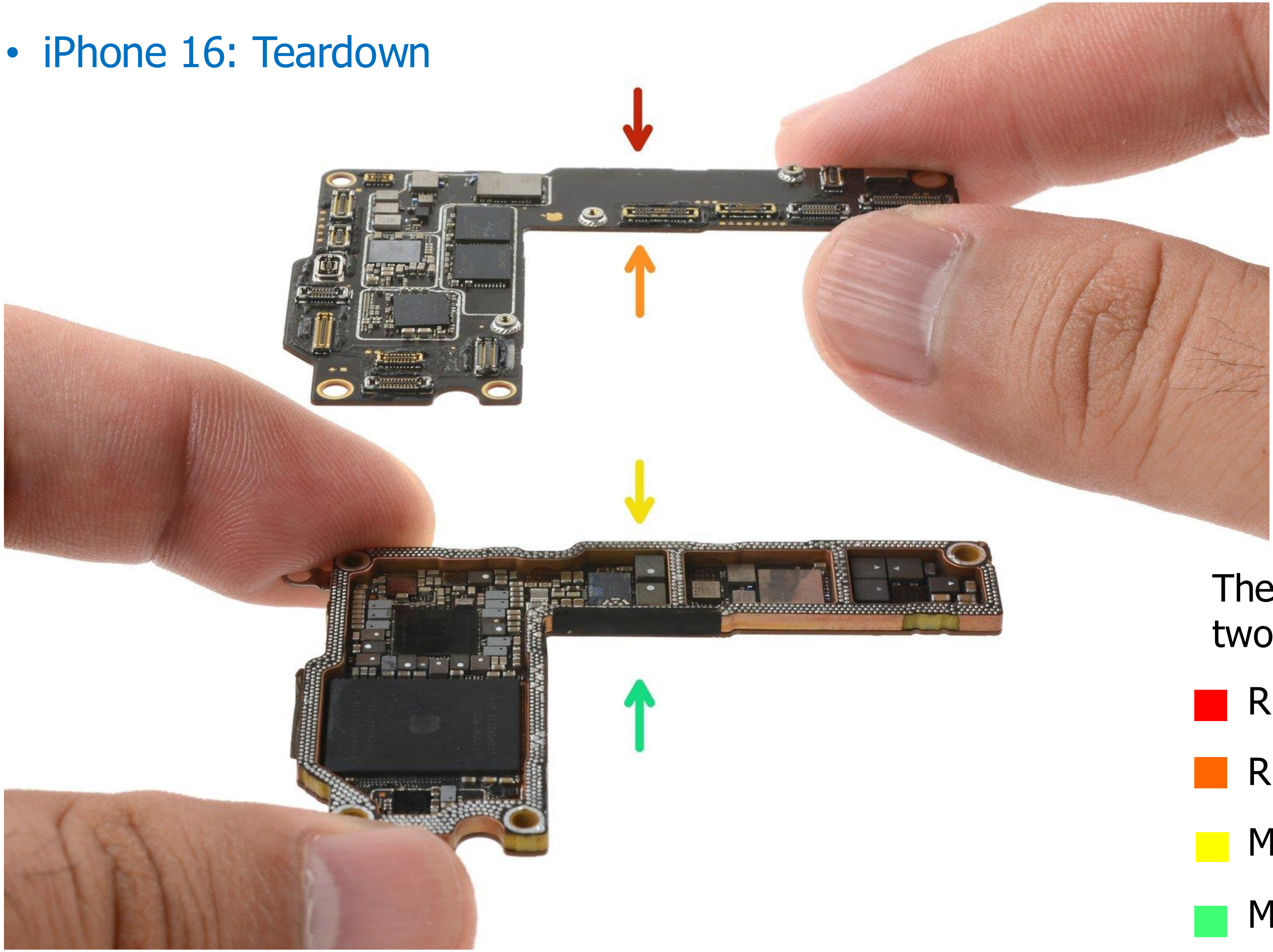
Importance of Digital Systems

- iPhone 16 Teardown

source: www.how-fixit.com

Importance of Digital Systems

- iPhone 16: Teardown



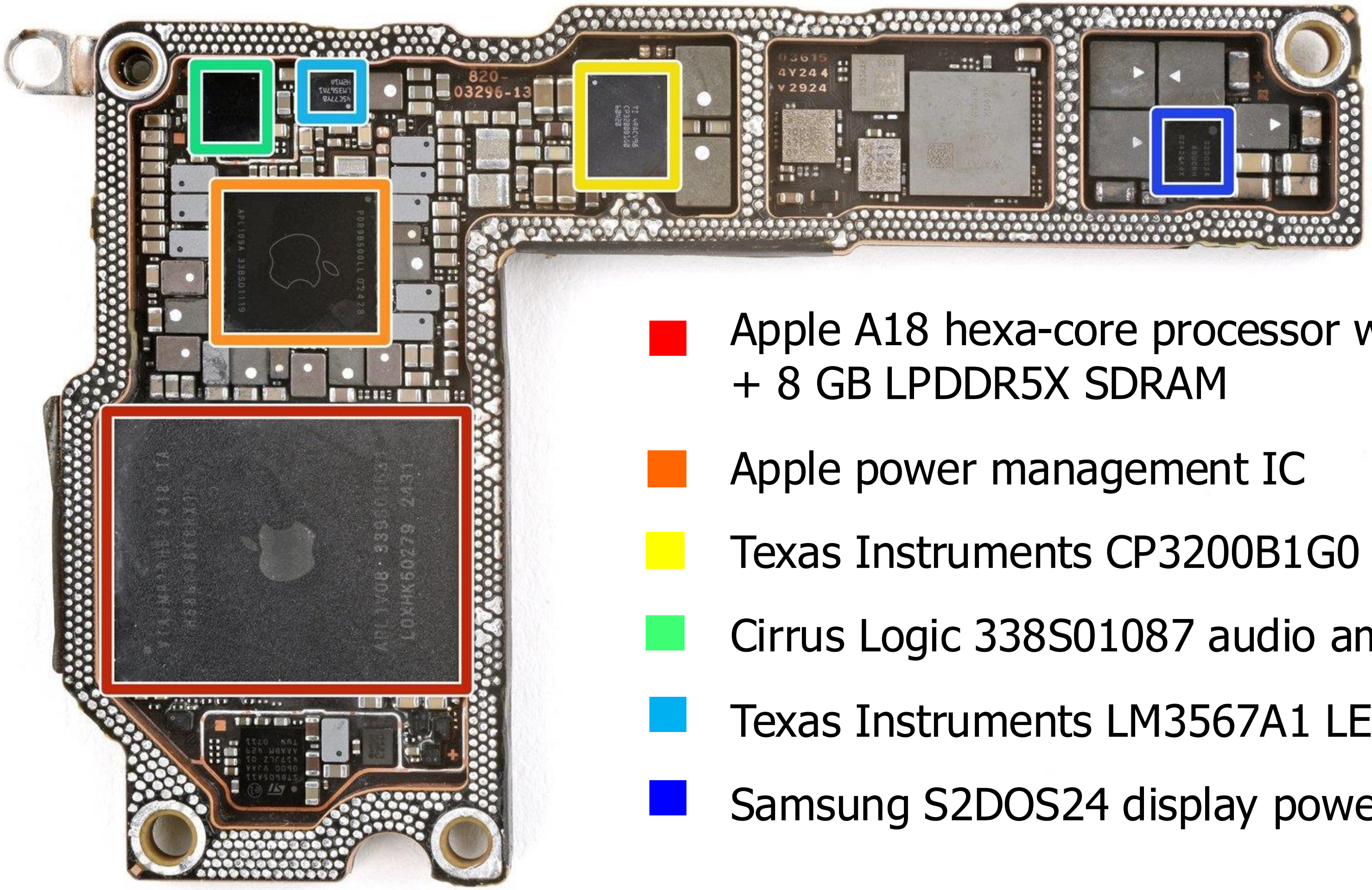
The iPhone 16's logic board contains two double-sided PCBs:

- RF board top
- RF board bottom
- Main board top
- Main board bottom

source: nl.ifixit.com

Importance of Digital Systems

- iPhone 16 Teardown: main board top

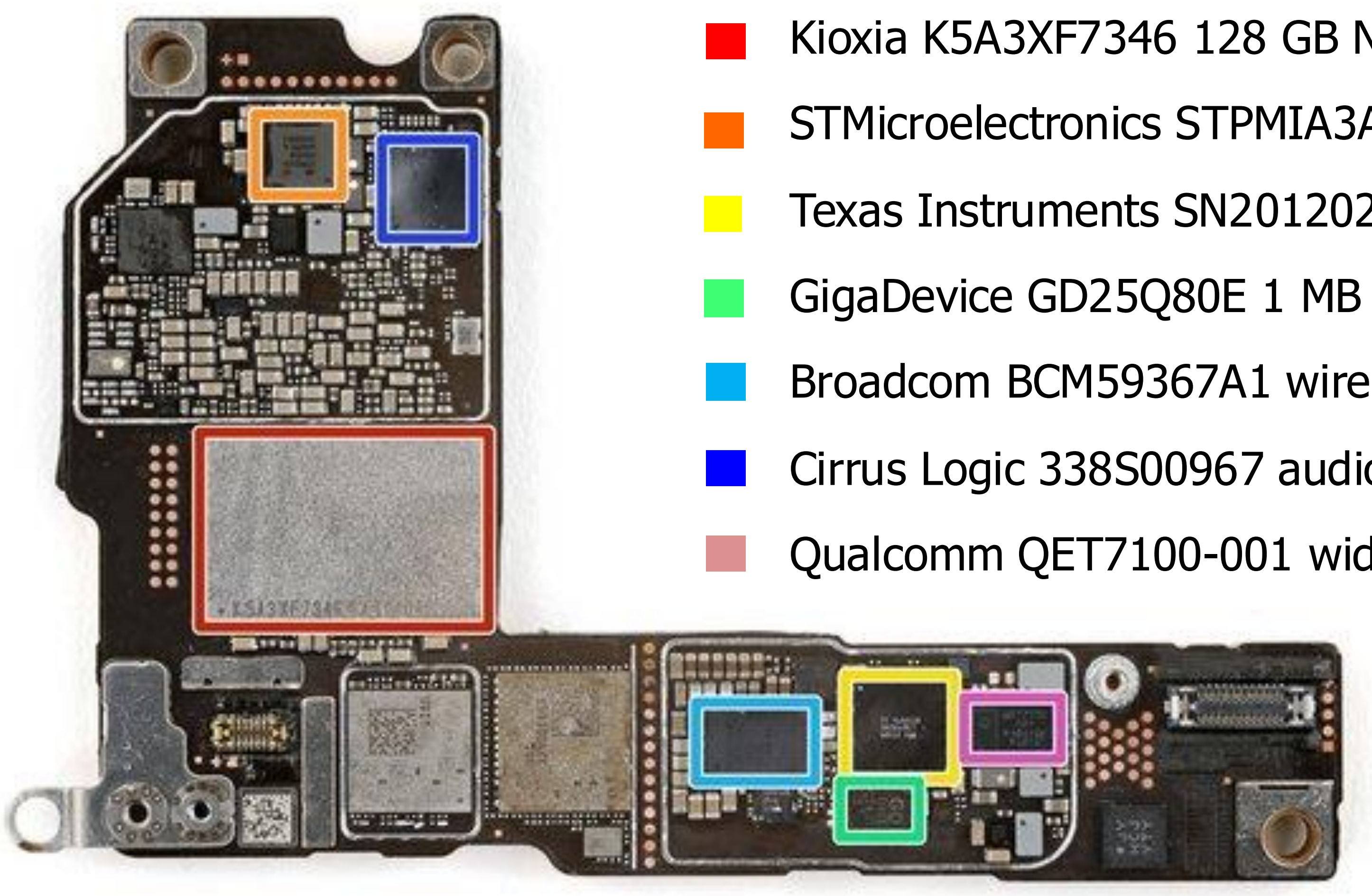


- Apple A18 hexa-core processor w/ GPU & Neural Engine ST + 8 GB LPDDR5X SDRAM
- Apple power management IC
- Texas Instruments CP3200B1G0 battery charger
- Cirrus Logic 338S01087 audio amplifier
- Texas Instruments LM3567A1 LED flash driver
- Samsung S2DOS24 display power supply

source: nl.ifixit.com

Importance of Digital Systems

- iPhone 16 Teardown: main board bottom

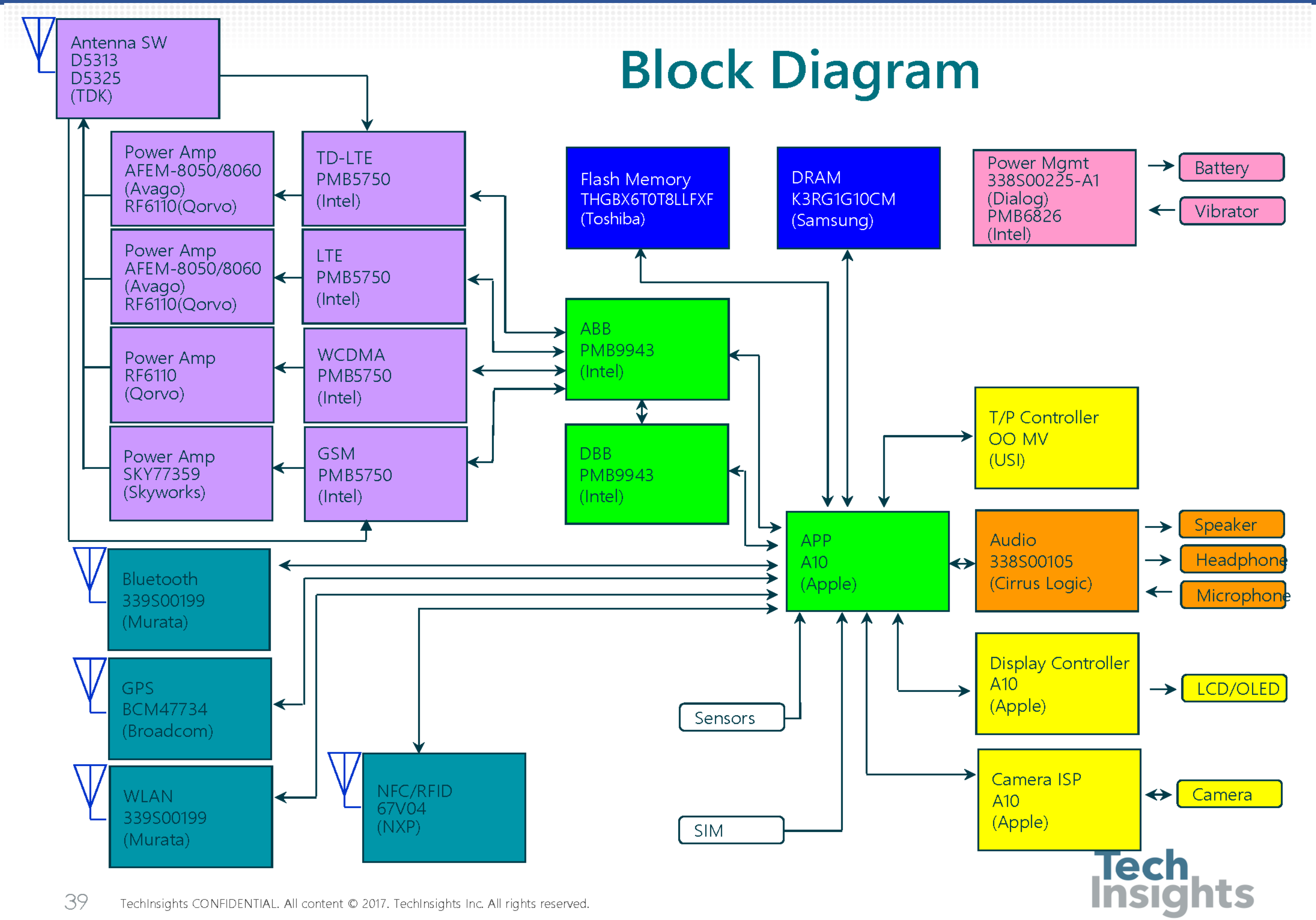


- Kioxia K5A3XF7346 128 GB NAND flash memory
- STMicroelectronics STPMIA3A426AEQ power management
- Texas Instruments SN2012027 USB type-C controller
- GigaDevice GD25Q80E 1 MB serial NOR flash memory
- Broadcom BCM59367A1 wireless charging controller
- Cirrus Logic 338S00967 audio codec
- Qualcomm QET7100-001 wideband envelope tracker

source: nl.ifixit.com

Importance of Digital Systems

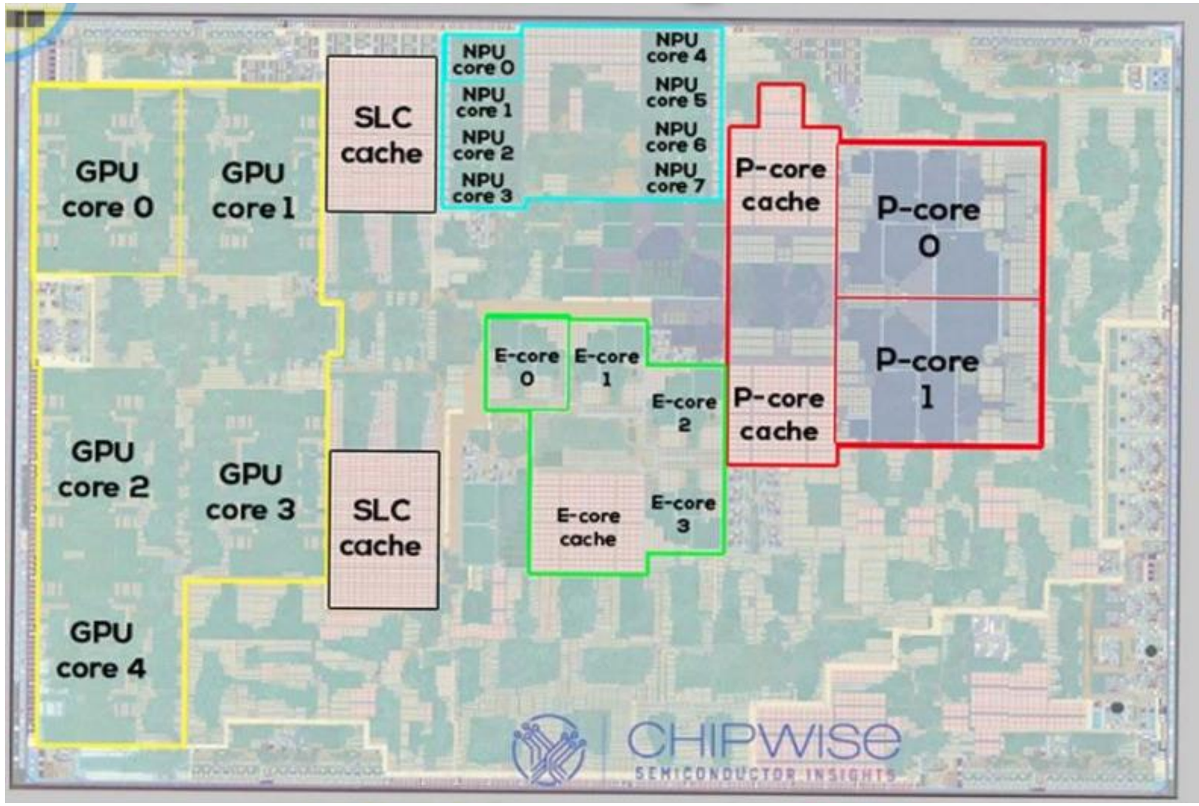
- iPhone 7



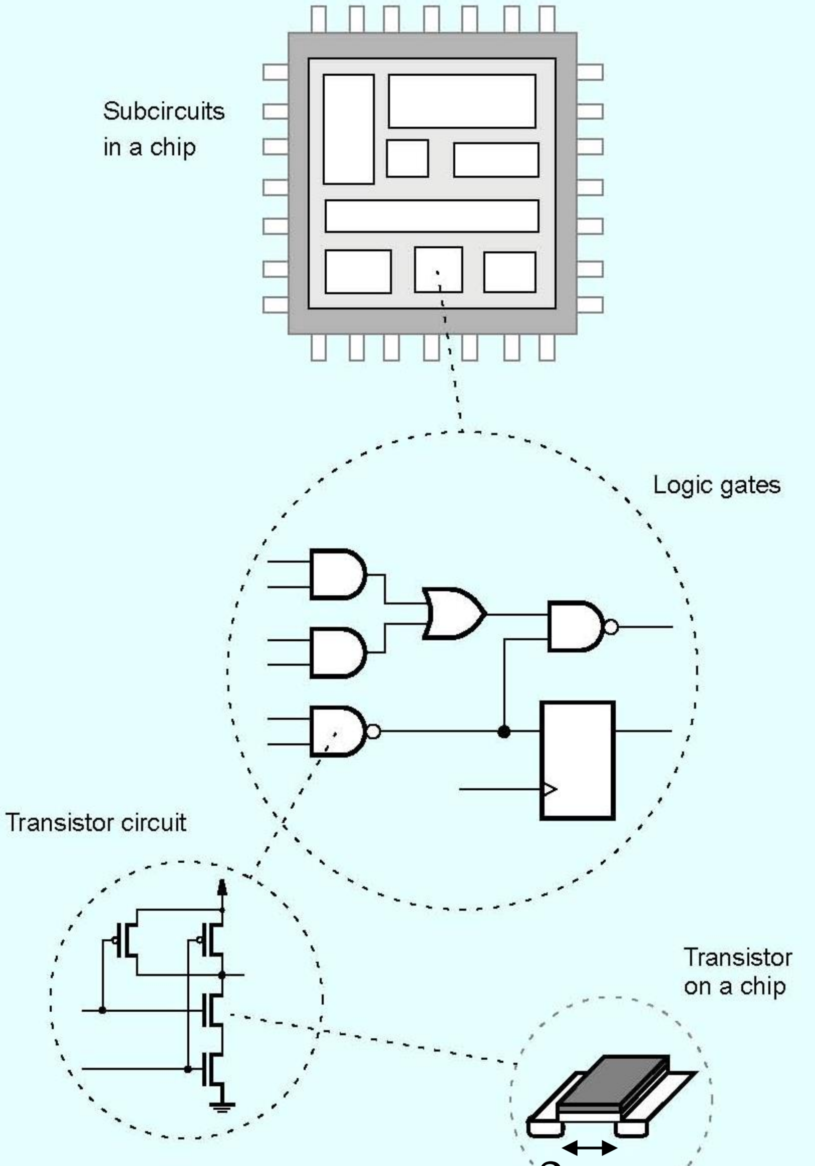
Importance of Digital Systems

- iPhone 16 A18 Chip

source: https://www.chipwise.com



- 3 nm technology
- 90 mm²
- 15.2 x10⁹ transistors
- 2+4 64-bit ARM proc. cores (4.05 GHz and 2.42 GHz)
- 5 GPU cores
- image signal proc.
- 16 core neural network engine

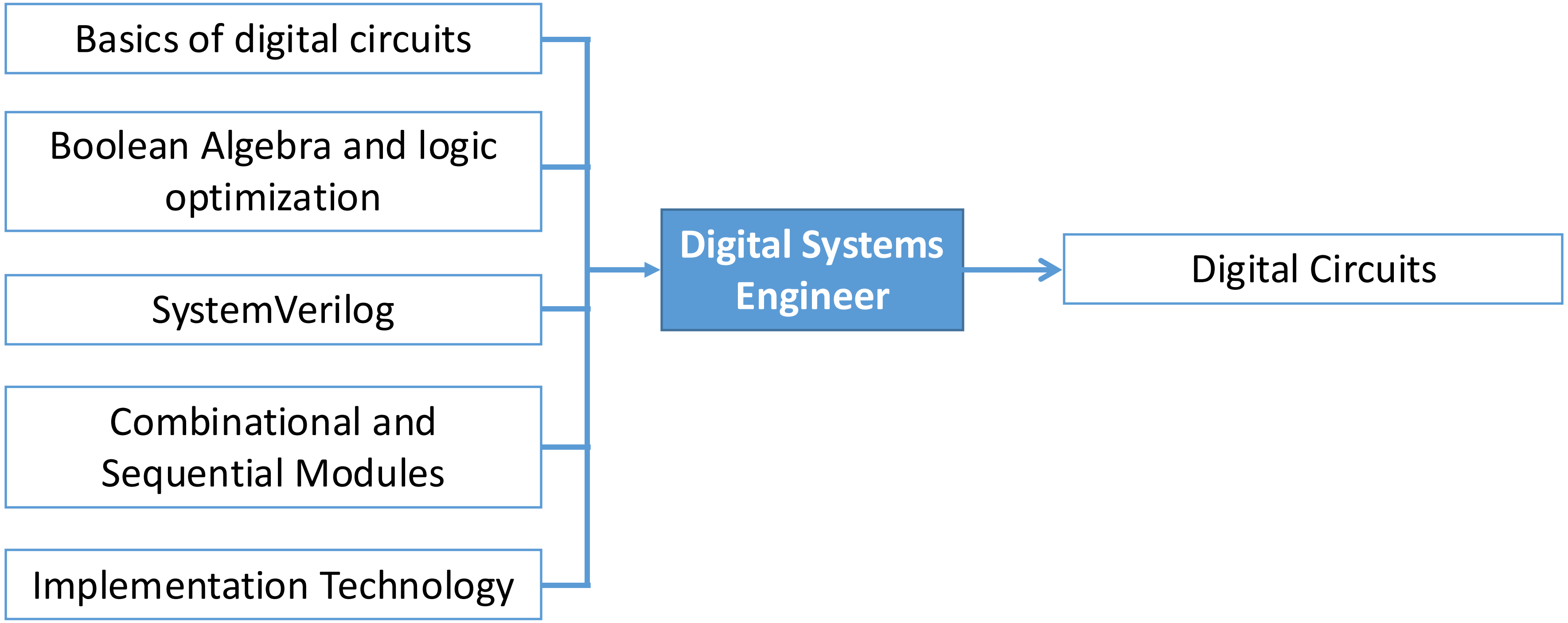


Si atom is 0.2 nm!

Course Overview

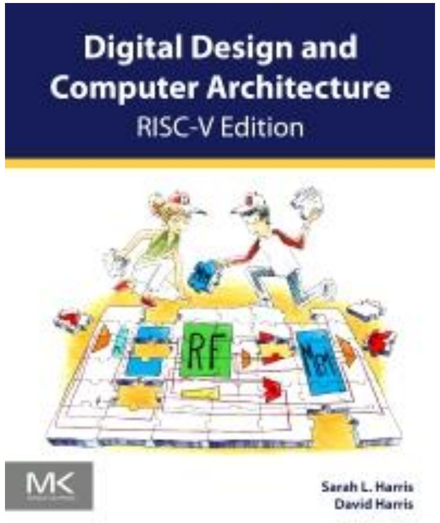
Understand

Design/Implement



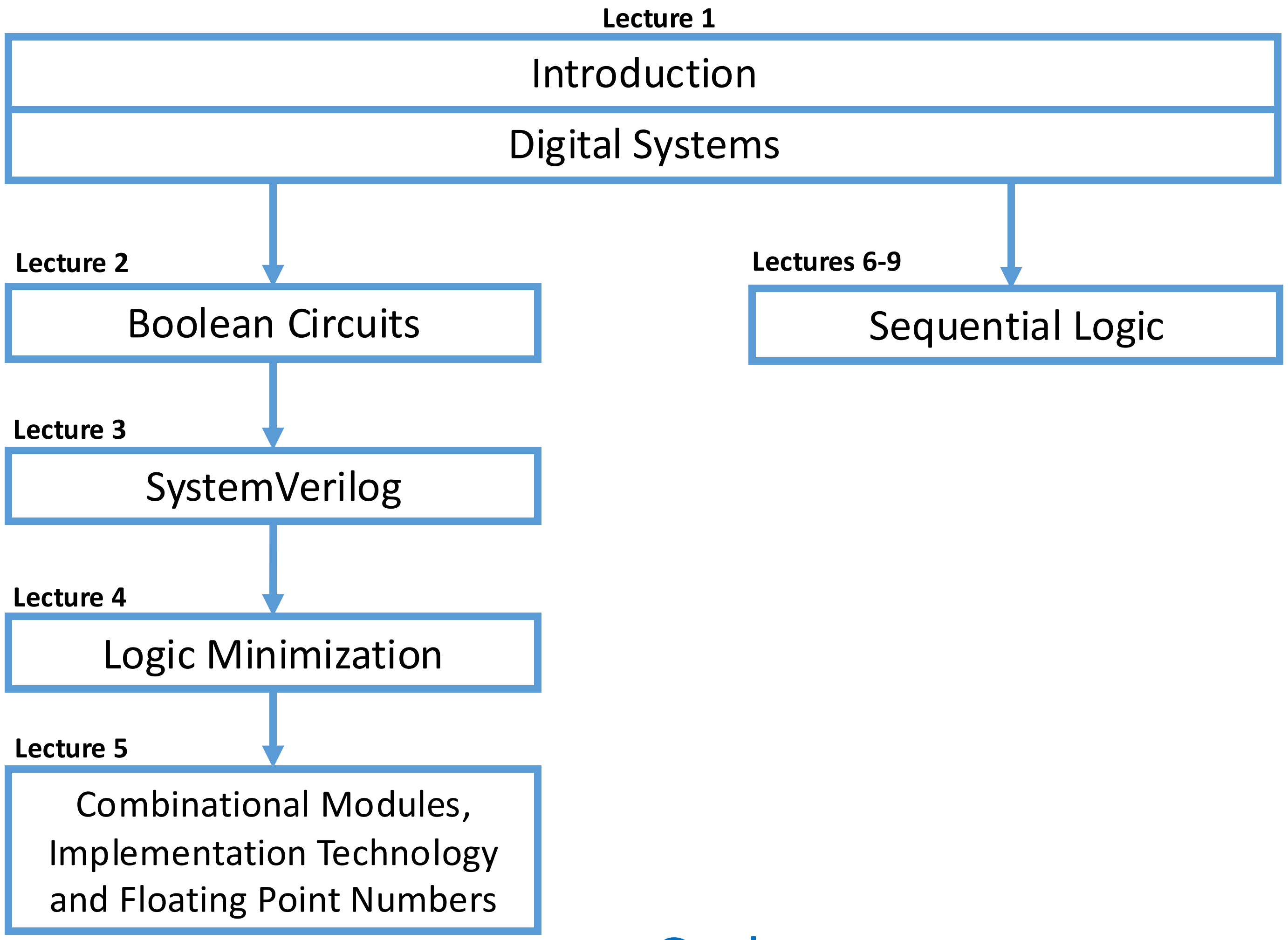
Course Overview

- Study Material
  - Lecture-slides (on Brightspace)
  - "Digital Design and Computer Architecture RISC-V edition", 2nd Edition, Sarah L. Harris, David Harris, Morgan Kaufmann, 2022, ISBN 978-0-12-820064-3



- Study material can be used during exams

Course Overview



- Lectures 1-5
  - 5 Gated Practice assignments
  - 2 course labs
  - 1 mock-up exam (on-campus) (week 4, Tuesday)
  - 4 instructions including Q&A (Friday)
  - 1 partial exam (week 5, Friday)
- Lectures 6-9
  - 3 Gated Practice assignments
  - 2 course labs
  - 1 mock-up exam (online) (week 9)
  - 4 instructions including Q&A
  - 1 partial exam (week 10, Tuesday)
- Re-exam (17 Jan 2026)

Grade:

When R is average of the mock-up exams and T is average of the partial exams, then final grade $C = T + R \times (10 - T) / 100$; course lab sessions must be completed

Course Overview – Schedule

Week	Lecture 1 (Mon)	Lecture 2 (Tue)	Assignments	Mock-Up/Exam
1.1	Intro	Boolean Circuits	GP-lec1, GP-lec2	
1.2	SystemVerilog	Logic Minimization	GP-lec3, GP-lec4	
1.3	Combinational Modules		GP-lec5 Course Lab Part 1	
1.4			Course Lab Part 2	Mock Exam (Tuesday) Discuss Mock Exam (Friday)
1.5				Partial Exam 1 (Friday)

Weekly seminars are on Friday

Course Overview: Reading Material

Week	Date	Lecture	Topics	Material
1.1	04/09	Lec 1	Introduction to Digital System	Slides Sections 1 – 1.5.4
	05/09	Lec 2	Boolean Circuits	Slides Sections 1.5.5, 1.5.6, 2.1, 2.3 (not yet 2.3.5) and 2.4
1.2	11/09	Lec 3	SystemVerilog	Slides Sections 4.1, 4.2 (not 4.2.3, 4.2.6, 4.2.9), 4.3 and 4.9
	12/09	Lec 4	Logic Minimization	Slides, Sections 2.2, 2.3.5, 2.5, 2.7 and 2.9
1.3	18/09	Lec 5	Combinational Modules, Implementation Technology and Floating Point Numbers	Slides Sections 1.6, 1.7.4-1.7.7, 2.6.2, 2.8, and 5.3.1, 5.3.2 (no rounding and addition)

Course Overview

- Assignments
  - Gated practise assignments are available on Brightspace
    - Try to finish them before the next lecture
  - Info about 2 course lab sessions also on Brightspace
    - Homework part has to be completed before entering the lab
    - Finishing lab assignments is compulsory

Course Overview

- Weekly Tasks
  - On Brightspace you can find the "Weekly Tasks".
  - Per lecture the tasks are:
    a. Study the slides
    b. Read the relevant text in the book
    c. Do the exercises/assignments
  - Make sure that a and b are done before c.

Course Overview: Fraud during the Lab

- The lab is an integral part of the course! Committing fraud can have serious consequences:
  - Your will be immediately expelled from the lab
  - You will have to answer to the exam committee

- For more information, see:
  - https://www.tudelft.nl/en/student/my-study-me/rules-guidelines-and-participation/fraud-plagiarism

- In case you (expect to) run into problems: contact the responsible staff immediately!

Course Overview: Learn by Practice

- You learn by looking at examples and by trying it yourself!

- Although study material may be used at the exam: practice and learn important things by heart! Otherwise:
  - You will not be able to solve the questions.
  - You will not have enough time.

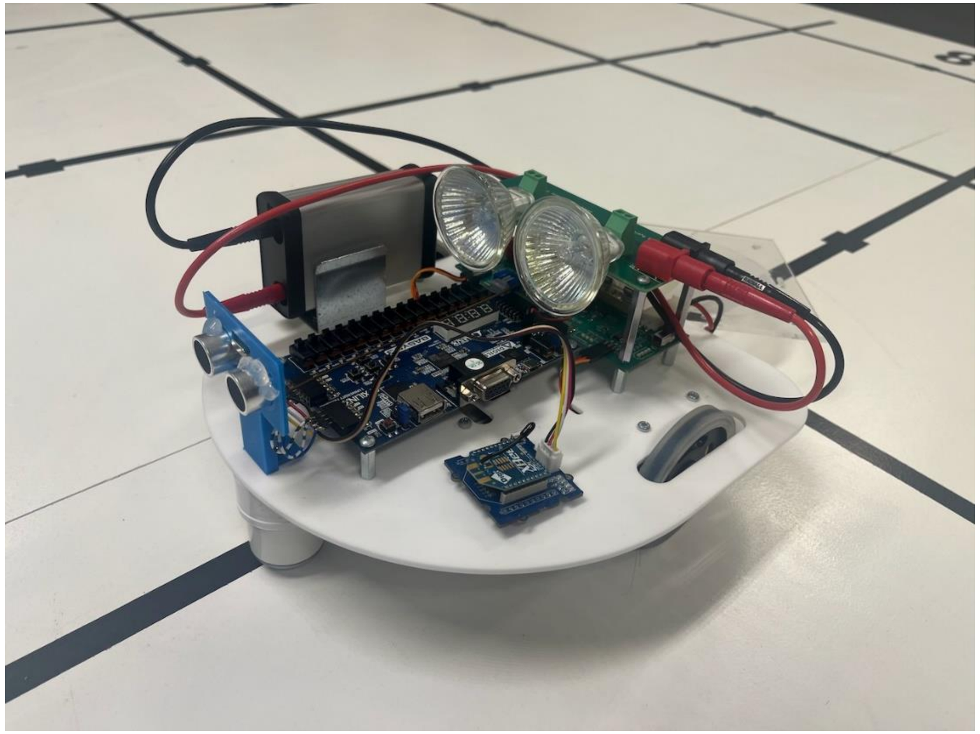
- Note that there are old example exams available on Brightspace for practising.

Course Overview: Next Courses

- Year 1
  - EE1D2 Digital Systems B (Q3)
  - IP-2 Smart Robot Challenge (Q4)
- Year 2
  - EE2C2 Mixed-Signal Circuits and Systems
  - EEX05 Chip Design (elective)
- Year 3
  - EE3D1 Computer Architecture and Organization

Course Overview

IP-2 Project “Smart Robot Challenge



Robot searching its way in a maze

- local control on FPGA: SystemVerilog
- global control on PC: C
- wireless communication via ZigBee

EE1D1: Digital Systems A

BSc. EE, year 1, 2025-2026, lecture 1

Introduction to Digital Systems

Computer Engineering Lab

Faculty of Electrical Engineering, Mathematics & Computer Science

Delft University of Technology

Outline

- Learning Objectives
- Digital Systems
- Digital versus Analog
- Binary Systems and Boolean Algebra
- Data Representation

Learning Objectives

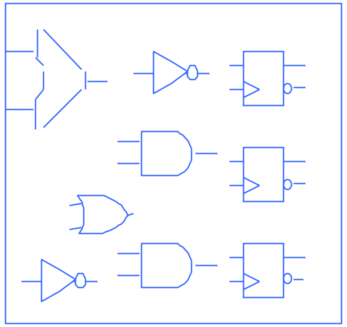
As student you should be able to:

- Explain the importance of digital systems
- Describe the benefits of digital systems over analog systems
- Get familiar with binary systems and basic Boolean algebra
- Convert numbers between different base systems (e.g., from decimal to binary and vice versa)
- Compute with negative and positive numbers

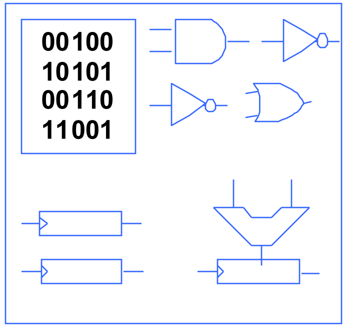
Digital Systems

DS

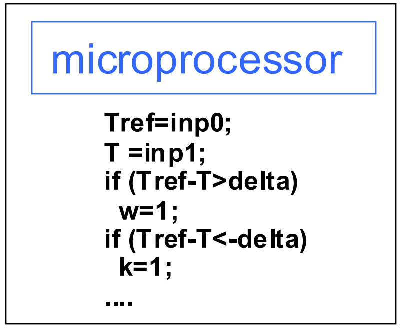
Design freedom:



100% HW design



50% HW, 50% SW design

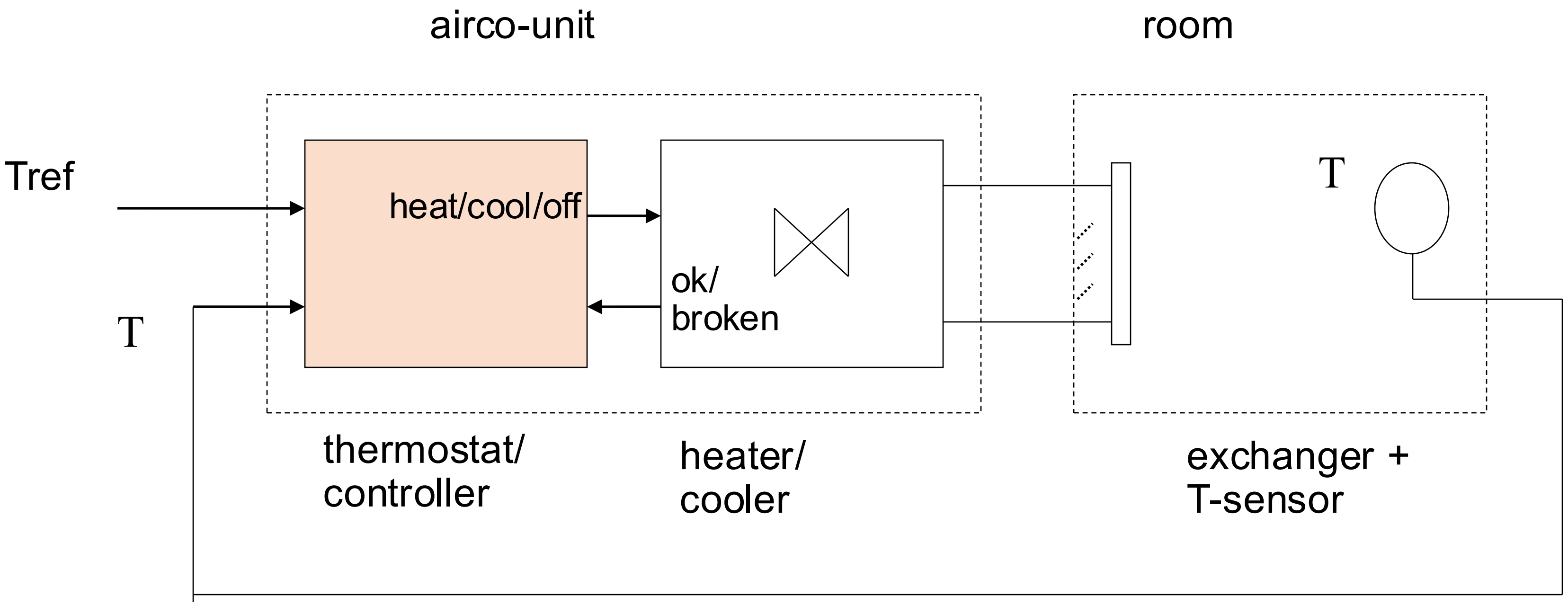


100% SW design

advantages: speed
parallelism

flexibility
complexity

Simple example: Airco thermostat/controller



= computer system, specification =

Tref-T > delta AND ok --> heat
Tref-T < -delta AND ok --> cool
|Tref-T| <= delta OR broken --> off

compute and decide

Digital Systems

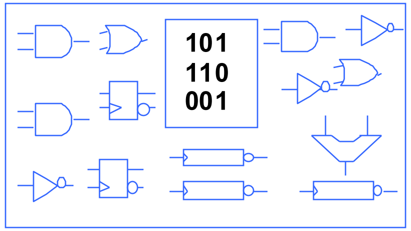
Digital circuit design in practice

desired behavior: **“...,
heater on when too cold and airco/heater ok,
...”**

Boolean algebra:

...
heat = too_cold . ok
....

circuit:



SystemVerilog prog:

```
...
assign heat = too_cold & ok;
...
```

(simulation)

desired behavior?

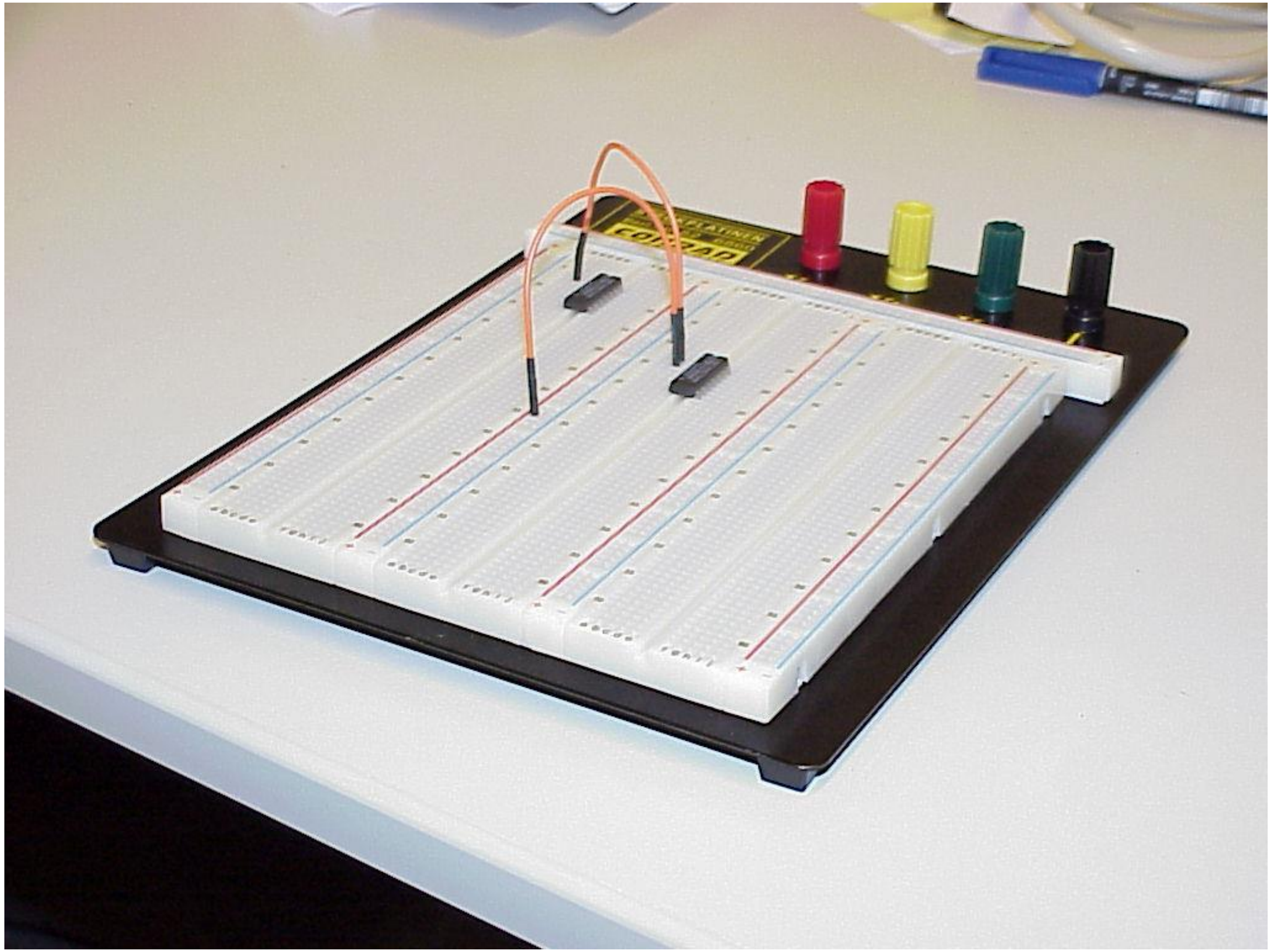
e.g. FPGA chip

desired behavior

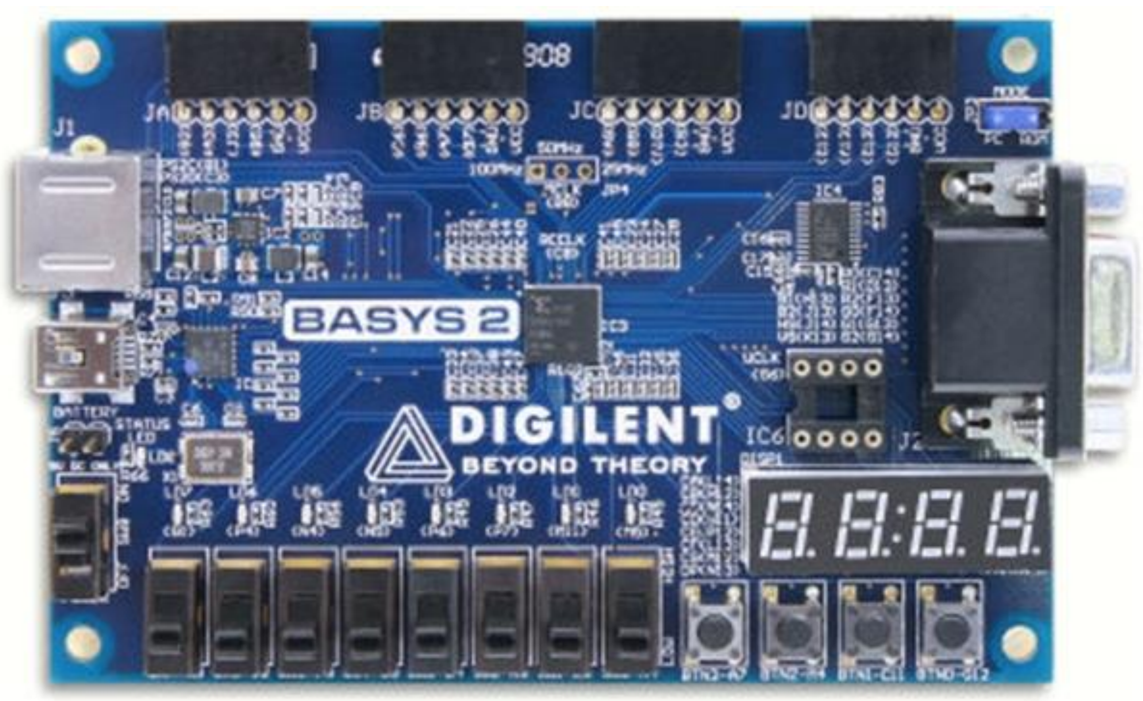
Many tools are available for this

Digital Systems

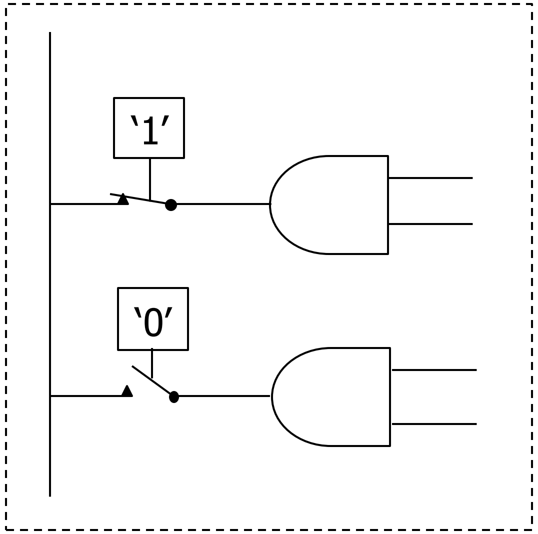
Experiment hardware: Breadboard



BASYS2 Board with Xilinx Spartan 3E FPGA



FPGA contains components that are connected by switches controlled by RAM memory



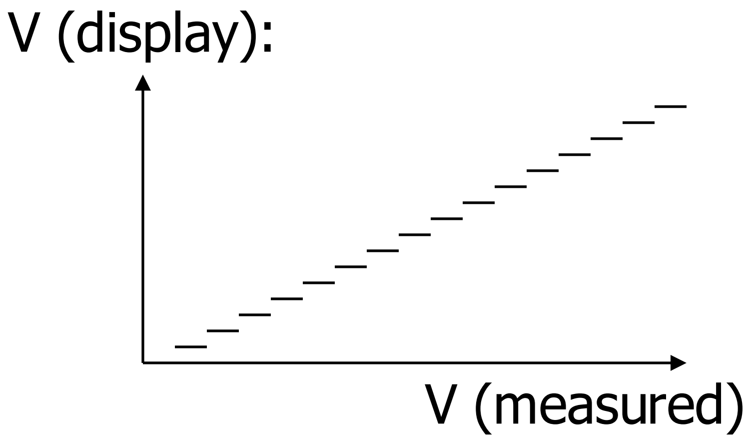
FPGA

Digital Versus Analog

Digital versus Analog

Digital systems: represent analog values (e.g. temp, voltage, natural constants) by a *finite* number of digits

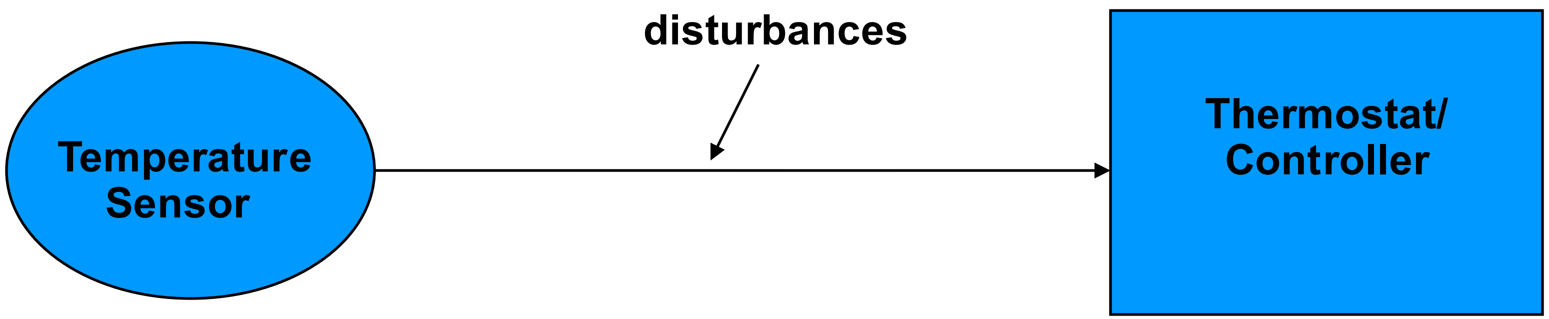
E.g. digital voltage meter:



Other examples:

- voltage meter with pointer: analog
- thermometer met digital display: digital
- mercury thermometer: analog
- π: analog
- 3.1415: digital
- e: analog
- old fashioned clock: analog

Digital versus Analog



How can we best transfer the temperature (25.0°C) ?

- Analog: 1 signal wire, e.g. 0…10V, 0-100°C, 0.1°C accurate

  25.0°C = 2.50V
  accuracy has to be 10/1000=0.01 V

- Digital: e.g. using 3 signal wires, each using 10 different levels: 0V, 1V, 2V, …, 7V, 8V, 9V

  25.0°C ⇒ 2V,5V,0V = 250
  $(2 \bullet 10^2 + 5 \bullet 10^1 + 0 \bullet 10^0) \bullet 0.1°C = 25.0\ °C$

  even deviations of 0.1 - 0.4V are no problem

- Advantages:
  - Less sensitive to noise
  - Simpler circuits
- Disadvantage:
  - more signal wires and components (but plenty are available on a modern chip!)

Binary Systems and Boolean Algebra

Binary Systems and Boolean Algebra

Use variables with only two discrete values (binary values):

- 1, true, ... (e.g. 10 V, 25 mA, …)
- 0, false, ... (e.g. 0 V, 0 mA, …)

bit = **bi**nary digi**t**

- To represent all kind of data and perform computations on it:
  - 25.0°C $\Rightarrow$ 10 V, 10 V, 10 V, 10 V, 10 V, 0 V, 10 V, 0 V = 11111010 = 250
  - $(1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0) \cdot 0.1^{o}C = 25.0^{o}C$
  - Other quantities, can be represented by (long) sequences of 0's and 1's such as sound and video

- To make decisions based on **logic**

IF	$(T_{ref} - T > D)$	AND	(airco/heating. unit ok)
THEN*	(turn on heater)		

Both condition have to be true for "heater on" to be true

THEN* = then and only then

Binary Systems and Boolean Algebra

- Algebra defines values and possible operations on values

Boolean algebra:

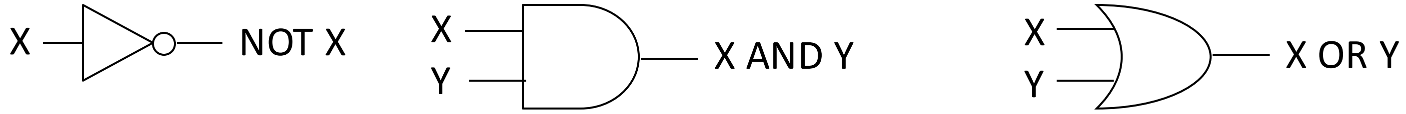
- values:
  - 0 (false), e.g. $x = 0$
  - 1 (true), e.g. $x = 1$
- operations:
  - ’ , ‾ (NOT), e.g. $f = x'$ equals $f = \overline{x}$
  - • (AND), e.g. $f = x \bullet y$
  - + (OR), e.g. $f = x + y$

X	NOT X
0	1
1	0

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

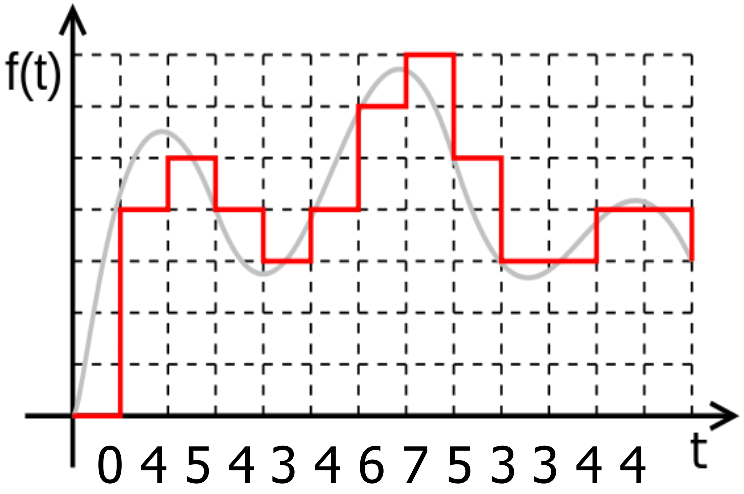
Symbols for use in circuit diagram:



Data Representation

Data Representation – Approximation

- All kind of information can be approximated by sequences of numbers:



- Examples:
  - Text: characters are represented by a number
  - Sound: air pressure amplitude as function of time points t_1, t_2, t_3, ...
  - Image: light intensity for a matrix of red, green and blue (RGB) pixels
  - Video: matrix of RGB pixel values as function of time points t_1, t_2, t_3, ...

- Sometimes Giga bits or Tera bits are required (e.g. for video)

Data Representation – Natural Numbers and Characters

number	binary representation
0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
.	. . . .
254	11111110
255	11111111

with 8 bits $2^8 = 256$ possibilities

8 bits = 1 Byte

character	representation
.	. . . .
'%'	00010101
'&'	00010110
.	. . . .
'A'	01000001
'B'	01000010
'C'	01000011
'D'	01000100
.	. . . .
'a'	01100001
'b'	01100010
'c'	01100011
.	. . . .

ASCII coding

Clearly: More bits per representation, more possible values:
32 bits → 2^{32} = 4.294.967.296 values

Data Representation – Decimal, Binary, Octal and Hexadecimal

Decimal (base 10, digits 0 ... 9)

$26_{10} = 2 \cdot 10^1 + 6 \cdot 10^0$

Binary (base 2, digits 0, 1)

$11010_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$

$= 26_{10}$

Octal (base 8, digits 0 ... 7)

$32_8 = 3 \cdot 8^1 + 2 \cdot 8^0 = 26_{10}$

Hexadecimal (base 16, digits 0 ... 9, A, B, C, D, E, F)

$1A_{16} = 1 \cdot 16^1 + 10 \cdot 16^0 = 26_{10}$

Data Representation – Conversion Between Decimal and Binary

From binary to decimal:

$11010 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$

$= 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$

$= 26$

From decimal to binary:

```
26 / 2 = 13 rest 0
13 / 2 = 6  rest 1
6  / 2 = 3  rest 0
3  / 2 = 1  rest 1
1  / 2 = 0  rest 1
-> 1 1 0 1 0
```

From decimal to binary (trial and error):

32	16	8	4	2	1	Weights
0	1	1	0	1	0	Converted value
26	10	2	2	0	0	Remainder

From decimal to binary (trial and error):

2^5 (32) : $26 = 0 \cdot 2^5 + 26$ => digit 0 (MSB)

2^4 (16) : $26 = 1 \cdot 2^4 + 10$ => digit 1

2^3 (8) : $10 = 1 \cdot 2^3 + 2$ => digit 1

2^2 (4) : $2 = 0 \cdot 2^2 + 2$ => digit 0

2^1 (2) : $2 = 1 \cdot 2^1 + 0$ => digit 1

2^0 (1) : $0 = 0 \cdot 2^0 + 0$ => digit 0 (LSB)

Data Representation – Binary, Octal and Hexadecimal

Simple mutual conversion:

Weight: 2^{15} 2^{12} 2^{9} 2^{6} 2^{3} 2^{0}

$1\ 000\ 010\ 111\ 100\ 111_2$

$1\quad 0\quad 2\quad 7\quad 4\quad 7_8$

Weight: 8^{5} 8^{4} 8^{3} 8^{2} 8^{1} 8^{0}

$1\quad 0\quad 2\quad 7\quad 4\quad 7_8$

$1000\ 0101\ 1110\ 0111_2$

$8\quad 5\quad E\quad 7_{16}$

Ideal for describing values in digital systems

Data Representation – Conversion Question:

What is the binary representation for the octal number 254_8 ?

a. 10101010
b. 10101100
c. 11010110
d. The correct answer is not listed

$254_8 = 2 \cdot 8^2 + 5 \cdot 8^1 + 4 \cdot 8^0 = 128 + 40 + 4 = 172_{10}$

```
172 / 2 = 86 rest 0
86 / 2 = 43 rest 0
43 / 2 = 21 rest 1
21 / 2 = 10 rest 1
10 / 2 = 5 rest 0
5 / 2 = 2 rest 1
2 / 2 = 1 rest 0
1 / 2 = 0 rest 1
```

=> 10101100, hence b

But easier is (in this case) per digit:

	2	5	4	(octal)
=	10	101	100	(binary)

Data Representation – Positive and Negative Numbers

- Sign bit
- Two's complement

Sign bit		Two's complement	
5	000101	5	000101
4	000100	4	000100
3	000011	3	000011
2	000010	2	000010
1	000001	1	000001
0	000000	0	000000
0	100000		
-1	100001	-1	111111
-2	100010	-2	111110
-3	100011	-3	111101
-4	100100	-4	111100
-5	100101	-5	111011

If 0 => positive
If 1 => negative

-2^5 … 2^1 2^0 **Weight**

- With both representations the left-most bit represents the sign. In contrast, *unsigned* numbers don't have a sign bit.
- Two's complement appears to result in much simpler add/subtract circuits
- Trick to change sign for 2's compl: invert all bits and add 1

 3 -> 000011 -> 111100
 -> 111101 -> -3
-3 -> 111101 -> 000010
 -> 000011 -> 3

Data Representation – Binary Computation

Addition

	dec.		bin.	
carry to next column →	1		1111	
	27		0011011	
	14	+	0001110	+
	41		0101001	

Subtraction

dec.		bin.	
22		010110	
14	–	001110	–
8		001000	

- But borrowing from next column is a complex operation.
- Easier is to use a two's complement representation for -14 and add both numbers.

–14 = 110001 + 1
= 110010

dec.		bin.	
		~~1~~1 11	
22		010110	
–14	+	110010	+
8		001000	

outgoing carry can be neglected in this case.

Summary

- Introduction to Digital Systems
  - Hardware
  - Software
- Digital versus Analog
  - Needs more resources
  - Digital requires less accuracy
- Binary Systems and Boolean Algebra
  - Logic/Boolean operations
- Data Representation
  - Positive/negative numbers
  - Binary, decimal, hexadecimal

To do list

- Reading material book “Digital Design”:
  - Section 1 – 1.5.4

- Reading material for next lecture: Boolean Circuits
  - 1.5.5, 1.5.6, 2.1 and 2.3 (not yet 2.3.5) and 2.4

- Gated Practise assignment "Lecture 1"

Thank you