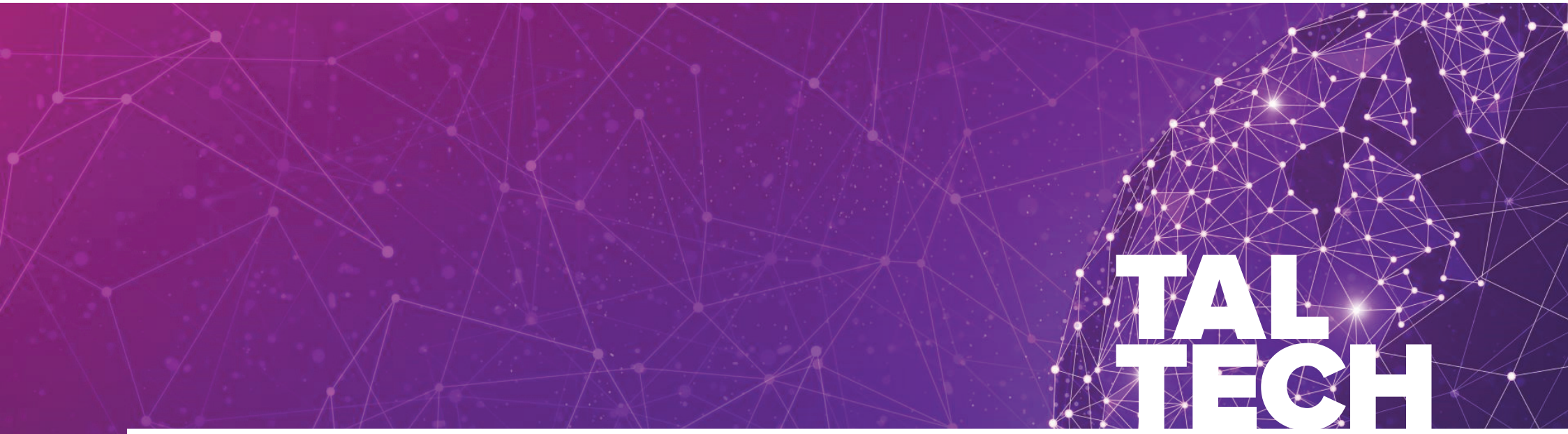


MICROPROCESSOR SYSTEMS (IAS0430)

Peeter Ellervee <peeter.Ellervee@ttu.ee>

Department of Computer Systems, ICT-526, 6202258
Tallinn University of Technology

03.09.2021

INTRO … 2

- **Microprocessor Systems (*IAS0430*)**

  - This course accounts for 6 ECTS (you will have to earn those)
    - 1 ECTS is 26 hours of studying
    - This means we have a total of 162 hours of studying for this course
    - We will have ~10 hours of study each week
      - You will spend 1.5 hours in lecture and 1.5 hours in lab
      - This leaves 7 hours of studying outside class room activities

  - The course is 16 weeks long. Each week we meet in the following times:
    - **Every Friday** – Unless otherwise announced
      - Lecture at **9:45 am – Room U5-103**
      - Lab period 1 at **12:00 – Room ICT-501 (MAHM)**
      - Lab period 2 at **15:00 – Room ICT-501 (IASM)**

INTRO ... 3

- **Topics & schedule (may change)**

Week	Date	Lecture	Labs	Assignments
1	Sept. 3	Introduction		Quiz
2	Sept. 10	CPU	L1 - ALU (deadline **Oct. 14**)	HW1 - Logic gates
3	Sept. 17	ISA	L1 - ALU	(deadline Sept. 23)
4	Sept. 24	MIPS	L1 - ALU	HW2 - Dummy CPU
5	Oct. 1	Pipeline	L1 - ALU	(deadline Oct. 7)
6	Oct. 8	Kernel/User Modes	[catching up]	HW3 - Dummy RISC & CISC
7	Oct. 15	OS	L2 - CPU & ISA (deadline Nov. 11)	(deadline Oct. 21)
8	Oct. 22	Memory Hierarchy (I)	L2 - CPU & ISA	
9	Oct. 29	Memory Hierarchy (II)	L2 - CPU & ISA	Midterm Exam (15 p)
10	Nov. 5	Memory Hierarchy (III)	[catching up]	
11	Nov. 12	Memory Management (I)	L3 - Memory (deadline Dec. 2)	
12	Nov. 19	Memory Management (II)	L3 - Memory	
13	Nov. 26	Process Scheduling	L3 - Memory	
14	Dec. 3	Computer Arithmetic	L4 - Arithmetic	
15	Dec. 10	Arithmetic in Hardware	L4 - Arithmetic	
16	Dec. 17	*Backup time*	L4 - Arithmetic	

INTRO … 4

- **Class Design and flow**
  - We will have a **home task** every second week in the first half of the course.
    - Tasks will be related to the topics of the past weeks.
    - They are due in two weeks. Any delay equals point loss.
    - Home tasks are strictly **individual**. No working in **groups**. These are designed for you to figure out on your own. You lose all points if you work with someone else.
    - If you need to know something, go back to the lecture notes or research on your own.
  - We will have **Laboratory assignments**
    - Labs will start on the second week. The schedule is approximate.
    - Requirements for each lab are different. Labs are to be completed **individually**. Labs might be done in a group, but this will also be clear in the lab assignment page on Moodle.

INTRO … 5

- **Class Design and flow**
  - There will be a final exam and midterm which are also **Mandatory**.
    - The midterm will only include the material covered before the 7th week.
    - The final exam will include all the material from the course.
    - There will be a list of topics that are included in the exam.
    - The exam will be work based, not answer based.
    - ~~To pass the class, you must get more than 60% in the exam.~~

  - Almost everything regarding this class will be on Moodle:
    - Look Up: **IAS0430 Microprocessor Systems**
    - No key

INTRO … 6

- **Grading:**
  - Grading will be done according to the following table

Criteria	Points	Priority
Home Tasks	15 pts	Mandatory
Lab Assignments	40 pts	Mandatory
Midterm	15 pts	Mandatory
Final Exam	30 pts	Mandatory
Total	100 pts	

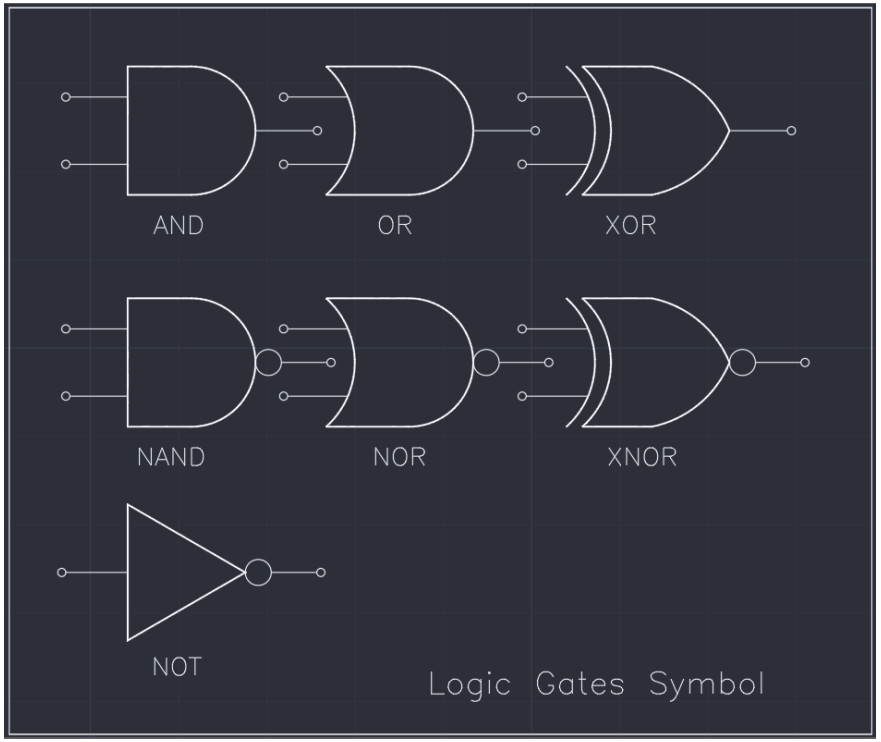
  - Extra points possible depending on the quality of work done

RECAP OF LOGIC AND LOGICAL OPERATIONS

- **Terminology:**
  - The bit: symbol (b)
    - short for **binary digit**, is the smallest unit of data in computing.
    - Values of 0 and 1 – binary
  - The Byte: symbol (B)
    - A series of 8 bits is called a byte.
    - Also called Octet
    - A half-byte is called a **nibble**
  - The word: symbol (w)
    - Is a series of 4-bytes is called a word. Used as standard for instruction length.
    - How many bits there is in a word?
  - Computer Logic:
    - The fundamental operations that all computer systems are built upon.

RECAP OF LOGIC AND LOGICAL OPERATIONS - 2

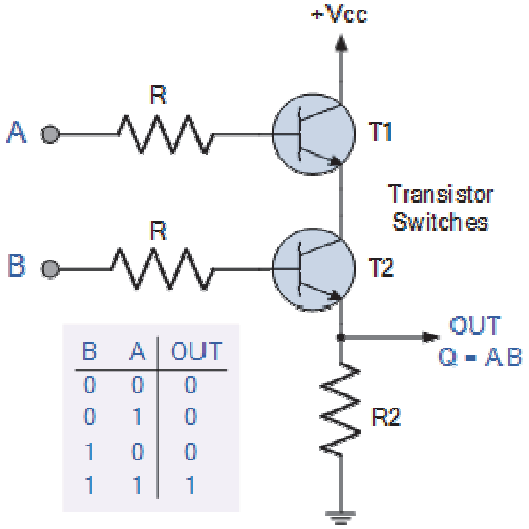
- **Logic Gates:**
  - What is a logic gate?
    - A fundamental building block of the integrated circuit.
    - Allows us to perform functions on single bits (unary gates) or on two bits (binary gates).
    - **Simple logic gates**:
      - Are made out of transistors:
        - AND
        - OR
        - NOT
    - **Complex logic gates**:
      - Mode out of other logic gates:
        - NAND
        - NOR
        - XOR
        - XNOR



https://i1.wp.com/www.linecad.com/wp-content/uploads/2017/04/Logic-Gates-Symbol.png

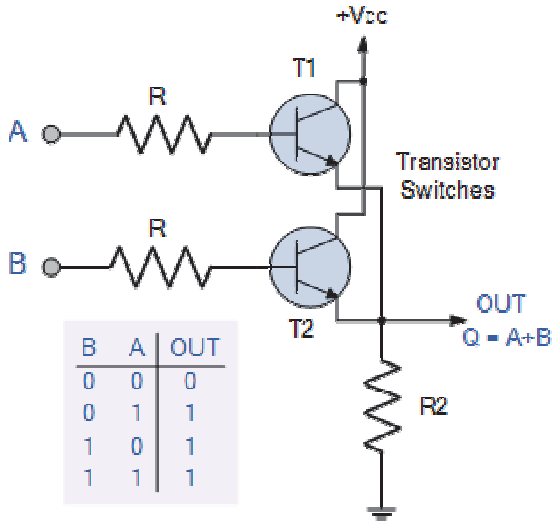
RECAP OF LOGIC AND LOGICAL OPERATIONS - 3

- **Simple logic gates:**
  - **AND GATE:**



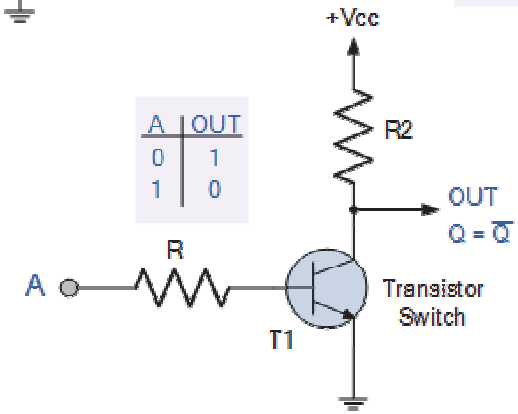
B	A	OUT
0	0	0
0	1	0
1	0	0
1	1	1

  - **OR GATE:**



B	A	OUT
0	0	0
0	1	1
1	0	1
1	1	1

  - **NOT GATE:**



A	OUT
0	1
1	0

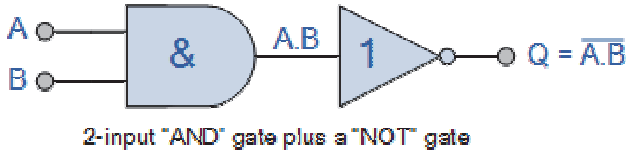
https://www.electronics-tutorials.ws/wp-content/uploads/2018/05/logic-log43.gif

https://www.electronics-tutorials.ws/wp-content/uploads/2018/05/logic-log44.gif

https://www.electronics-tutorials.ws/wp-content/uploads/2018/05/logic-log47.gif

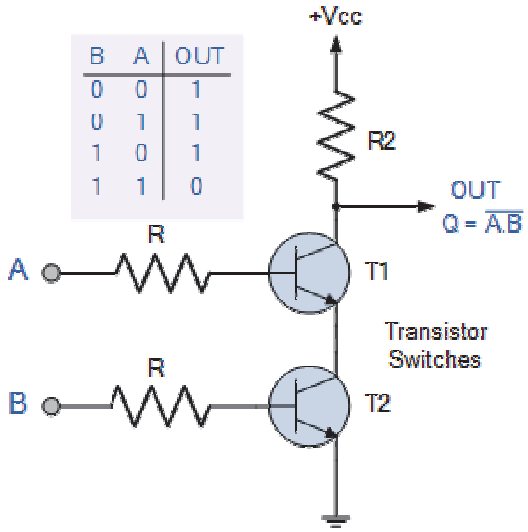
RECAP OF LOGIC AND LOGICAL OPERATIONS - 4

- **Complex logic gates:**
  - **NAND:**

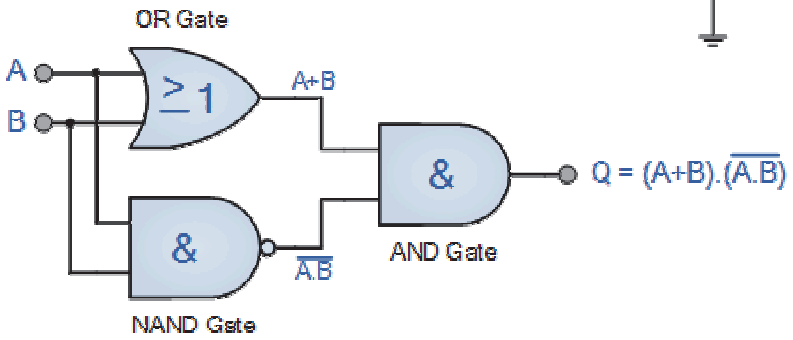


2-input "AND" gate plus a "NOT" gate

B	A	OUT
0	0	1
0	1	1
1	0	1
1	1	0



  - **XOR**



https://www.electronics-tutorials.ws/wp-content/uploads/2018/05/logic-log45.gif

https://www.electronics-tutorials.ws/wp-content/uploads/2018/05/logic-log20.gif

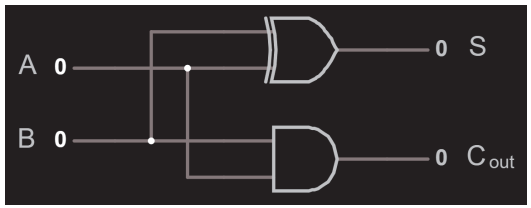
https://www.electronics-tutorials.ws/wp-content/uploads/2018/05/logic-log29.gif

RECAP OF LOGIC AND LOGICAL OPERATIONS - 5

- **Uses of logic gates:**
  - What can we do with Logic gates?
    - Create more complex integrated circuits that can perform:
      - Logic and arithmetic functions:
        - ADDERS
        - SUBTRACTORS
        - COMPARTORS
      - Date Transmission:
        - ENCODERS
        - DECODERS
        - MULTIPLEXERS
      - Code Converters:
        - 7-Segment led
        - 16-Segment led
        - Led Arrays

RECAP OF LOGIC AND LOGICAL OPERATIONS - 6

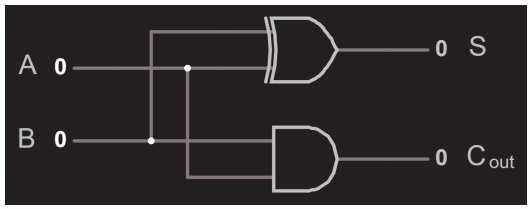
- The **ADDER:**
  - The adder does what its name suggests, it performs the operation of addition:
    - It is made out of two components:
      - The first is the **half adder**:
        - Takes two inputs (A and B)
        - Produces two outputs (S and C_{out})
        - The half adder is composed of an XOR gate to produce the S (sum) and an AND gate that produces the C_{out} (Carry out).
        - Performs the logical expression as follows:
          - S = A **XOR** B
          - C_{out} = A **AND** B
        - Create the Truth Table – Anyone?



- Circuit simulator – http://www.falstad.com/circuit/

RECAP OF LOGIC AND LOGICAL OPERATIONS - 6

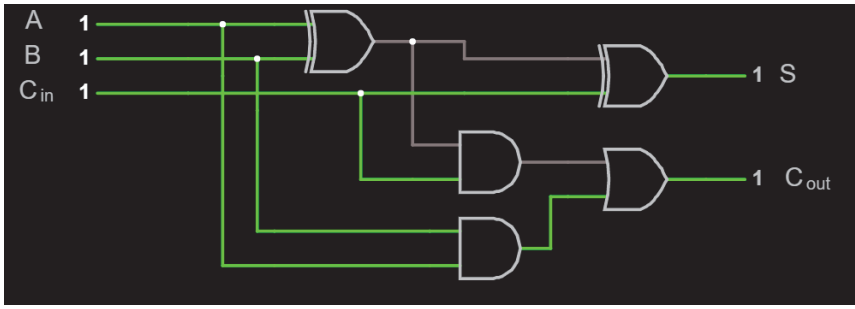
- The **ADDER:**
  - The adder does what its name suggests, it performs the operation of addition:
    - It is made out of two components:
      - The first is the **half adder**:
        - Takes two inputs (A and B)
        - Produces two outputs (S and C_{out})
        - The half adder is composed of an XOR gate to produce the S (sum) and an AND gate that produces the C_{out} (Carry out).
        - Performs the logical expression as follows:
          - S = A **XOR** B
          - C_{out} = A **AND** B
        - Truth Table:



A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

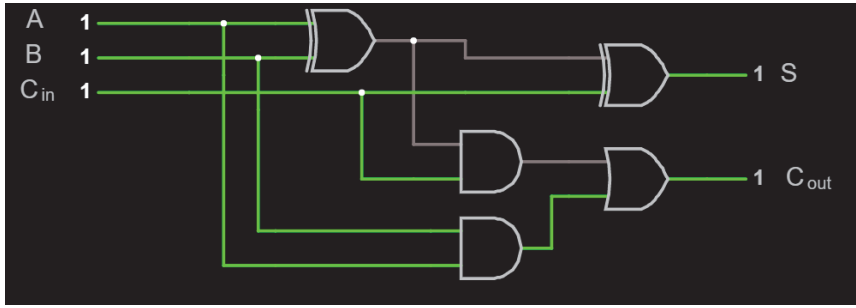
RECAP OF LOGIC AND LOGICAL OPERATIONS - 7

- The **ADDER:**
  - The adder does what its name suggests, it performs the operation of addition:
    - It is made out of two components:
      - The first is the **full adder**:
        - Takes three inputs (A, B, and C_{in})
        - Produces two outputs (S and C_{out})
        - Create the logical expression as follows:
          - S = ??
          - C_{out} = ??
        - Create the Truth Table – Anyone?



RECAP OF LOGIC AND LOGICAL OPERATIONS - 7

- The **ADDER:**
  - The adder does what its name suggests, it performs the operation of addition:
    - It is made out of two components:
      - The first is the **full adder**:
        - Takes three inputs (A, B, and C_{in})
        - Produces two outputs (S and C_{out})
        - Create the logical expression as follows:
          - S = (A **XOR** B) **XOR** Cin
          - C_{out} = ((A **XOR** B) **AND** Cin) **OR** (A **AND** B)
        - Truth Table:



A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

HOME TASK

- **Home Task of the Week:**
  - Finish the **Level Quiz** in Moodle