MICROPROCESSOR SYSTEMS (IAS0430)

Department of Computer Systems
Tallinn University of Technology

10.09.2021

LAB 1

- Lab 1 will focus on the different combinational circuits that make up the different components of the CPU. The following circuits will be discussed:
  - The register

  - ALU components:
    - adder
    - subtractor
    - comparator

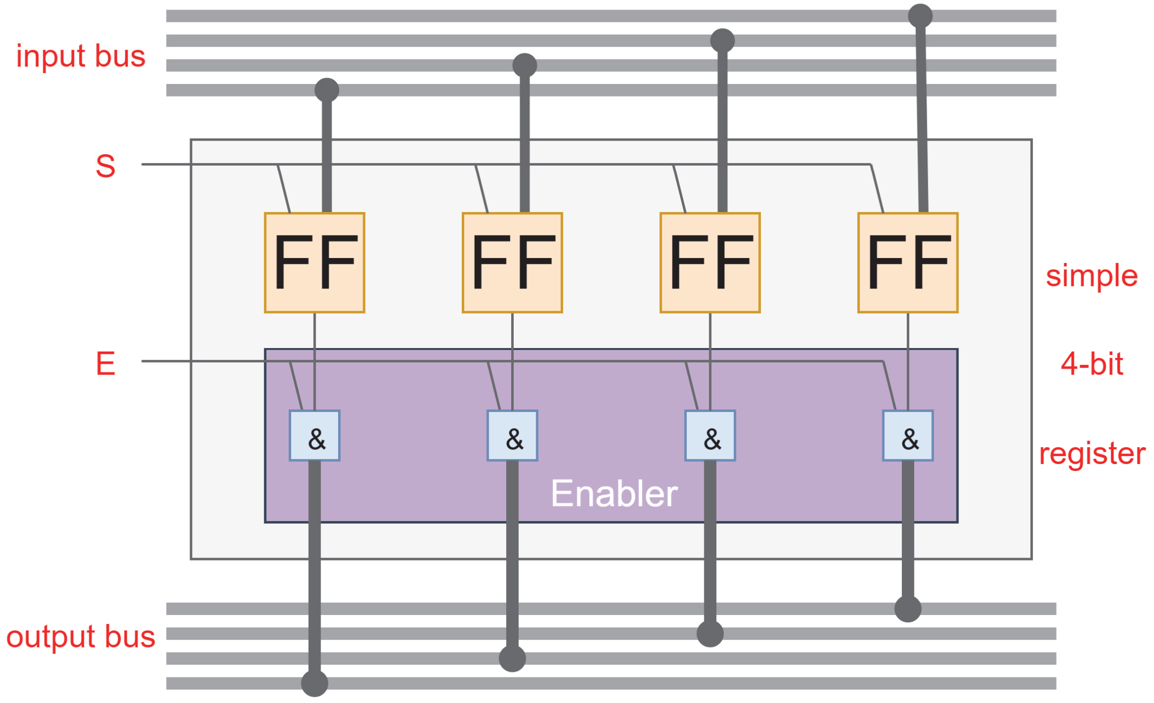
  - Control unit components:
    - decoder
    - encoder
    - mulitplexer

THE REGISTER

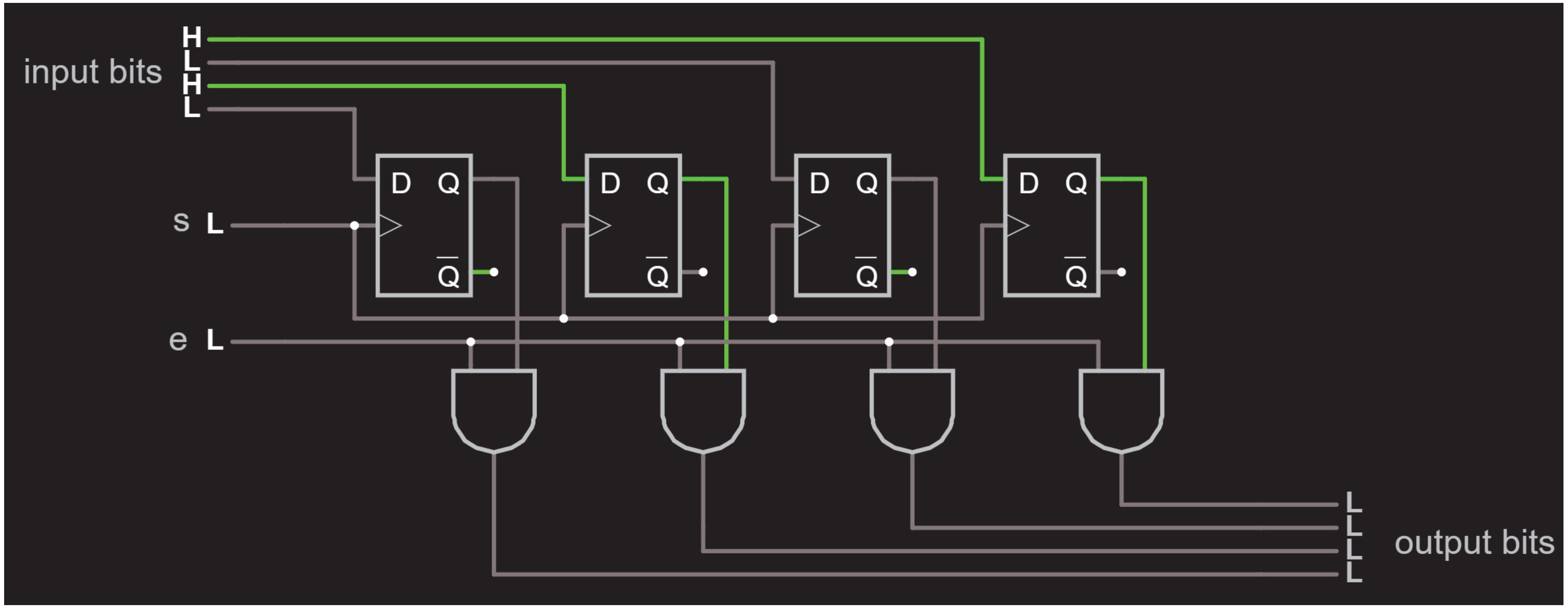
- The register is a simple component that is used to store information.
- It is a **<u>one dimensional series of flip flops</u> that are connected to the register input on one side, and an AND-gate-based <u>enabler</u> on the other side. The enabler is connected to the register output.**
- The register is a fundamental component of everything computerized.
- It is found in the simplest machines from mice and keyboards to dense and complicated circuits used to auto pilot an air plane. It is everywhere.
- The register has three inputs:
  - **Data input (i):** usually a bus that contains the data we want to store in the register.
  - **A set bit (s):** used to signal the register to store the data from the data input bus, also called a write bit.
  - **An enable bit (e):** used to signal the register to release the data stored in the register to the output data bus, also called a read bit.
- The register has one output which is the output from the enabler circuit.

THE REGISTER

- The register is made up of two main circuits:
  - **The flip flop series:**
    - Is a collection of flip flop. Each flip flop is connected to one of the input bits. The flip flops are connected to the set bit to signal them to store each of their respective input bits.
  - **The enabler:**
    - The enabler is what makes the register useful. The enabler is a series of **AND** gates. Each **AND** gate is connected to one flip flop on one input of the **AND** gate and the enable bit on the other input.
    - When the enable bit is at HIGH (1), the **AND** gates release the bits stored in the flip flops to the data output bus.

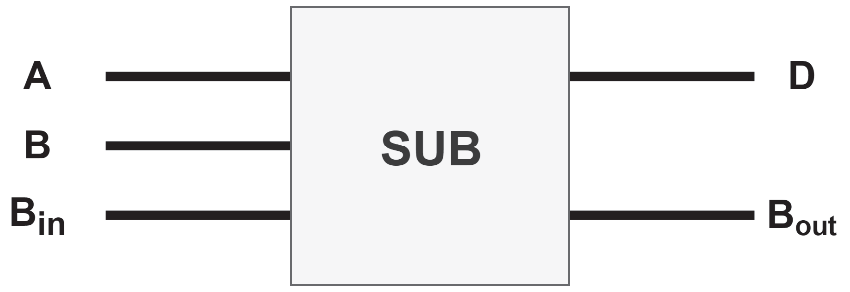


THE REGISTER



THE SUBTRACTOR

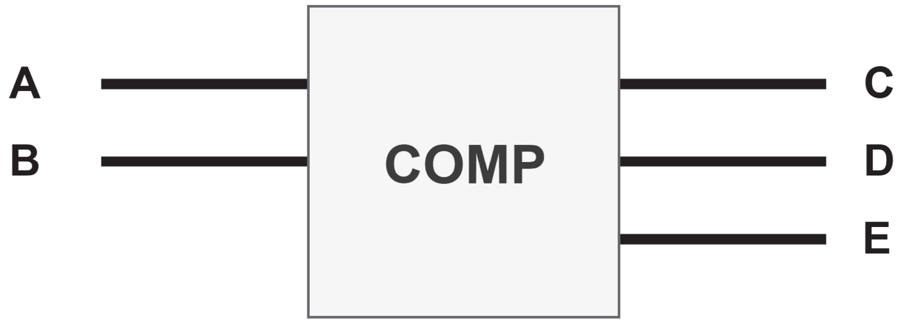
- As the name suggests, **the subtractor subtracts two input bits from one another.**
- It is found in the ALU.
- The half subtractor as is as follows
  - D = A **XOR** B
  - Bout = **NOT** A **AND** B
- The full subtractor has three inputs:
  - **The minuend (A)**: the bit in which B is being subtracted form. **Always bigger than B**
  - **The subtrahend(B):** the bit being subtracted from A. **Always smaller than A**
  - **The borrow in (Bin):** whatever is left from previous subtraction
- The full subtractor circuit has two outputs:
  - **Difference (D):** donates the result of subtraction
  - **Borrow out (Bout):** borrow for next subtraction
- The circuit follows the Logical expressions:
  - D = A **XOR** B **XOR** Bin
  - Bout = (**NOT** A **AND** Bin) **OR** (**NOT** A **AND** B) **OR** (B **AND** Bin)



It implements **A – B = D** and Bout as borrow.

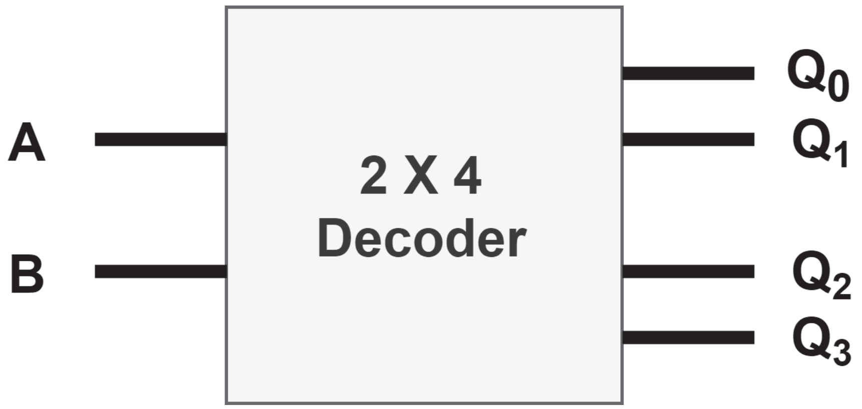
THE COMPARATOR

- As the name suggests, **the comparator combination logic circuit that perform comparison to see if sequences of bits are equal, greater, or less than one another.**
- The circuit has two inputs:
  - **A** and **B** bits to be compared
- The circuit has three outputs
  - **C** donates that A is **less** than B
  - **D** donates that A is **equal** to B
  - **E** donates that A is **greater** than B
- The circuit follows the logical expressions:
  - C = **NOT** A **AND** B
  - D = **NOT** ((**NOT** A **AND** B) **OR** (A **AND NOT** B))
  - E = A **AND NOT** B



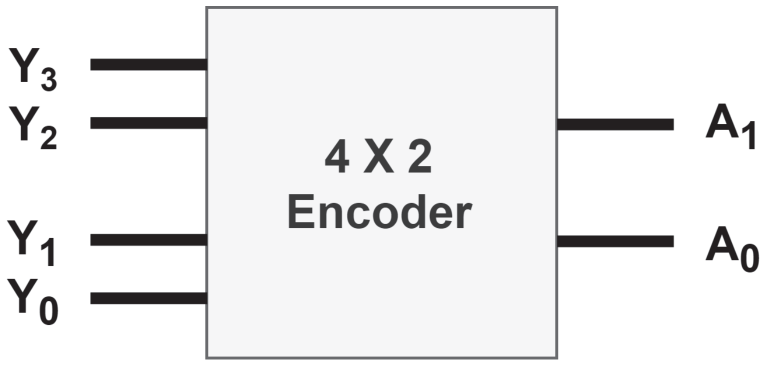
THE DECODER

- **A decoder decodes n-bits input to a 2^n-bits output with only one bit as logical 1**
- Naming a decoder is by the number of inputs to outputs according to **n to 2^n**
  - **2 to 4** decoder
  - **3 to 8** decoder
  - And so on...
- Think of a decoder as a component that tells you the order of the input in the truth table.
- In a 2 to 4 decoder, the two inputs, **A** and **B** are single bits that are converted to 4 bits **Q_0**, **Q_2**,**Q_3**, and **Q_4** output.
- The combinational logic as follows:
  - Q_0 = **NOT** A **AND NOT** B
  - Q_1 = **NOT** A **AND** B
  - Q_2 = A **AND NOT** B
  - Q_3 = A **AND** B



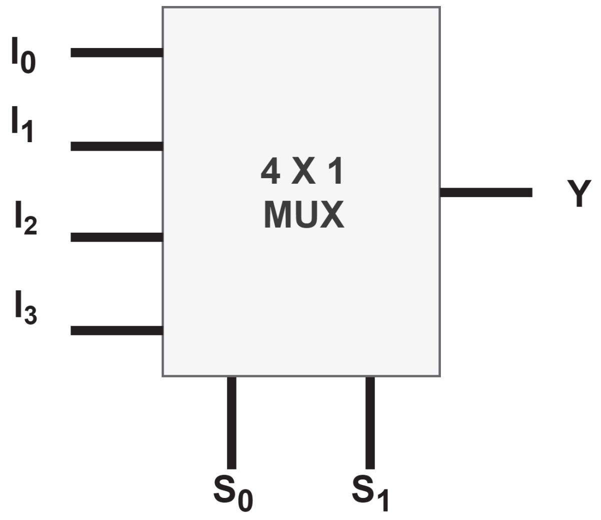
THE ENCODER

- They reverse the behavior of the decoder.
- **Encoders take 2^n-bits and encodes those bits into an n-bits output**
- The naming of the encoder is reverse to the naming of the decoder.
  - **4 to 2** encoder
  - **8 to 3** encoder
  - And so on..
- Think of the encoder as a component that gives you the possible combination of the outputs at a given input
- In 4 to 2 encoder, the 4 inputs $\mathbf{Y_3}$, $\mathbf{Y_2}$, $\mathbf{Y_1}$ and $\mathbf{Y_0}$ are 4 single bits converted into $\mathbf{A_1}$ and $\mathbf{A_0}$ outputs.
- The combination logic as follows:
  - A_1 = Y_3 **OR** Y_2
  - A_0 = Y_3 **OR** Y_1



THE MULTIPLEXER

- **Is combination logic that returns one of multiple inputs as output using control inputs.**
- The number of inputs, donated **2^n** requires **n** number of control inputs **S** to produce output **Y**.
- Naming of multiplexer follows the number of the Inputs and output
  - **4 X 1** MUX
  - **8 X 1** MUX
  - and so on…
- Think of the multiplexer as selector of which input to allow to pass.
- In a 4 X 1 multiplexer, 4 inputs, **I_3** , **I_2** , **I_1** and **I_0** 2 control inputs **S_1** and **S_0** will be used to select one input as output **Y**
- The combinational logic as follows:
  - Y = (**NOT** S_1 **AND NOT** S_0 **AND** I_0) **OR**
    (S_1 **AND NOT** S_0 **AND** I_1) **OR**
    (**NOT** S_1 **AND** S_0 **AND** I_3) **OR**
    (S_1 **AND** S_0 **AND** I_0)



LAB 1

- **Lab 1: (find details in Moodle Lab 1)**