ZUFS - Task 1

What is an embedded system?

Ans: it is a computing system that has limited resources because it is used for a specific function, an the word “embedded” refers to the system being in most cases a small system embedded in another system for a specific task to control and maintain the system.

EX: air conditioner and other home appliances, projector

For a computing system to be functional => 3 components are required:

1. Processor: the part that executes instructions
2. Memory: it stores the data
3. I/O peripherals

Ways to implement an embedded system:

1. System on board (SOB): **i.e.** when you get the IC for the processor, IC for the memory, IC for I/O peripherals, and put them on a board that is in most cases big enough to fit in a hand
2. System on chip (SOC): **i.e.** to call the manufacturer to implement the whole system on a chip and fabricate it.

Embedded system challenges:

1. Performance
2. Size
3. Cost
4. Power consumption

   For all those points, the SOC wins (except in performance) but what makes the SOB still used to this day is its **Configurability**, as any system must go through an R&D stage before going to mass production.

Microcontroller definition: is a complete system (processor, memory, and i/o peripherals)

Microprocessor definition: is only one of the system components

Difference between Processor, microprocessor, and CPU:

The processor: was based on the old vacuum tubes.

Microprocessor: was based on transistors.

CPU: is more of a communication controller between different microprocessors.

A processor has these crucial 3 parts:

1. CU (control unit)
2. ALU (arithmetic logic unit)
3. Register File

What the processor does when executing code:

1. Fetching: The Control Unit fetches the data from the memory
2. Decoding: the Instruction Decoder (a part inside the CU) decodes instructions based on the instructing format and instruction set for different operations
3. Executing: the ALU executes the code and does arithmetic (add, multiply, etc.) and logic (AND, OR, etc.) operations

ISA (instruction set architecture):

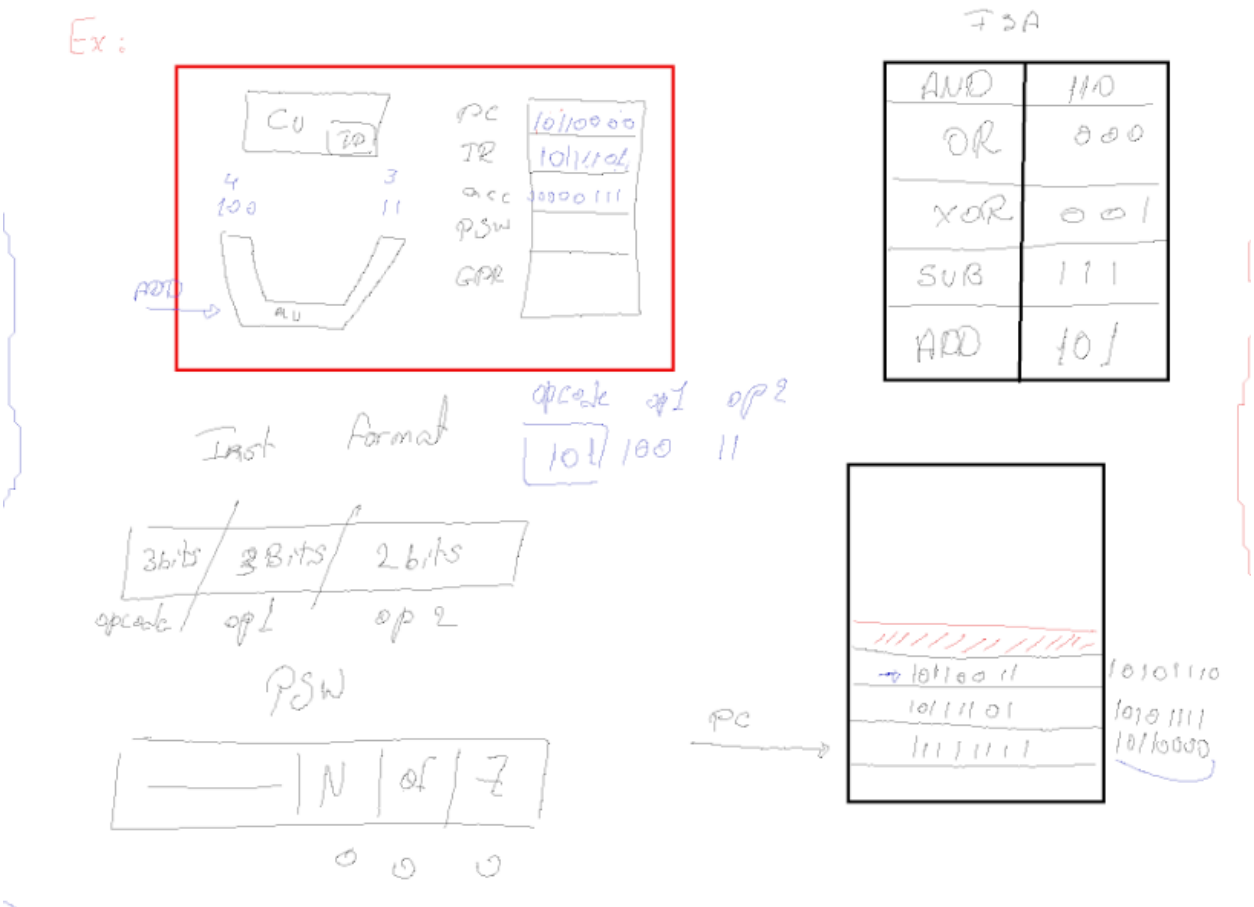
1. RISC (reduced instruction set computing)
2. CISC (complex instruction set computing)

In embedded system challenges, they are on par with each other as in performance as for simplicity the RISC tend to do more of the same operation to do one CISC operation, cost as for cheap hardware an expensive tool chain software is used to decode complex operation and vice versa for the CISC, for size is quite the same as the ALU is in fact smaller in the Risc but the ID is hardwired which makes it bigger, but for the Cisc it is microprogrammed so it is smaller, for power consumption it is pretty similar too.

Register Files:

1. PC (program counter): points to the next instruction to be executed
2. IR (instruction register): points to the instruction to be executed now
3. ACC (accumulator register): hold the result of the ALU
4. GPR (general purpose register)
5. PSW (processor status word): holds the flagis, i.e. zero flag, negative flag, overflow flag
6. SP (stack pointer)

and an example for how instructions are executed in a processor:



Memory:

1. Non volatile
2. Volatile:
    1. DRAM (dynamic RAM) which is based on capacitors
    2. SRAM (static RAM) which is based on transistors

    SRAM is preferred for embedded systems as the capacitors need Refreshing Circuits which is overly consuming for the limited resources the system has, and while the SRAM IS more expensive, mostly minimal amounts of memory are required.

Explaining Video Link