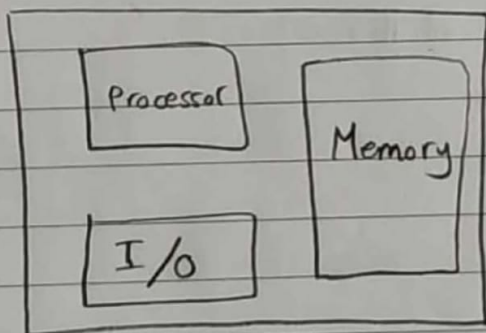


- \* ES Concepts: How to read data sheets (2-3 weeks)
- \* C Programming: (4 weeks)
- \* Interfacing or Device Driver

AVR Atmega32      ARM Cortex M3

## ES Concepts

### Computing System



IC

↳ Integrated Circuit

Electronic chip or Silicon chip which has an implemented circuit.

فيسر Pins لول chip

Embedded Systems is a computing system with limited resources to perform specific tasks.

System Board

"SB" Search on "IC"

System on Chip

"Soc"

Embedded Sys. Challenges

① Performance

② Size


③ Cost

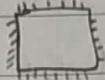
④ Power Consumption

Configurability

Note considering the IC on board

- it is made of Silicon & ذهب or فضة
- Two types: DIP, SMT

Silicon ↑  
2 \$  


↓  
70-80 cent  


Development

SB

↳ because of Configurability

Production

SOC

ex. Microcontroller

Processor Vs Microcontroller

↓  
one element  
from the three  
elements to  
build ES

↓  
Complete System  
On chip

Processor vs microprocessor

old vacuum tube  
تubes  
كانت الـ vacuum tubes

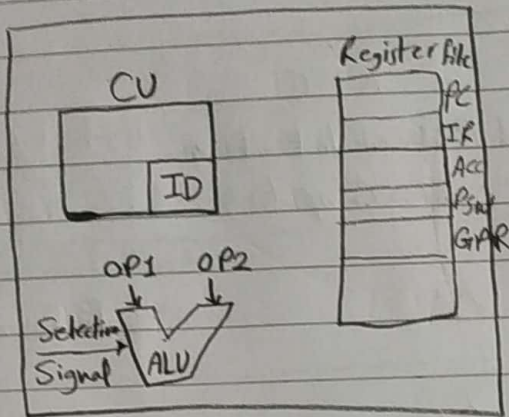
Silicon  
based on Silicon

but Now Processor = microprocessor

CPU: processor which control several Processors.



## ① Processor

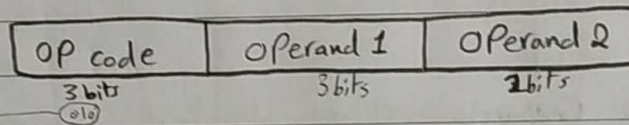


Memory (بنية الذاكرة)

CU: Control unit → Fetch (take the instruction from memory to execute)

● ID: Instruction Decoder → Decode (understand the meaning of instruction *يفهم معنى instr.*)

### ① Instruction Format



### ② Instruction Set

| Instruction | OP code |
|-------------|---------|
| Add         | 000     |
| And         | 001     |
| OR          | 010     |
|             |         |
|             |         |

ALU: Arithmetic Logic unit → Execute (selection signal → اختيار بيشوف)

الـ OP code و بيطع Analog موازي  
(للـ OP code ده واصلنا بتكون Analog)  
يسمى OP code و على اساس  
Signal

ADD كده بمرح ع مثلاً بيشغل الـ  
circuit  
ALU

# Instruction Set Architecture

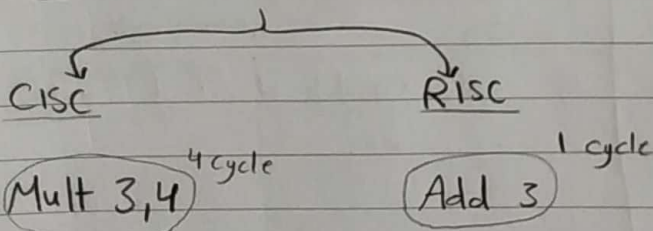
ISA

→ ZISC  
→ NISC  
→ OISC

| CISC                              | RISC                              |                   |
|-----------------------------------|-----------------------------------|-------------------|
| Complex Instruction Set Computing | Reduced Instruction Set Computing |                   |
| ~                                 | ~                                 | Performance       |
| ~                                 | ~                                 | Cost              |
| ~                                 | ~                                 | Size              |
| ~                                 | ~                                 | Power Consumption |

عشان جعل consume لنفس ال time فيعمل consume لنفس ال energy  
" بجعل Enable لدايرة واحدة At a time

Ex.  $x = 3 * 4$



" 3  
" 3  
" 3

ملحوظة: ال Compiler يغير ال ins وبيحولها  
لسطور ins. عشان ينفذها (لو ins من مجموعة كبيرة)  
لو موجودة هينفذها مباشرة "عشان كذا compiler  
ال RISC معقد ولكن ال compiler ال CISC.

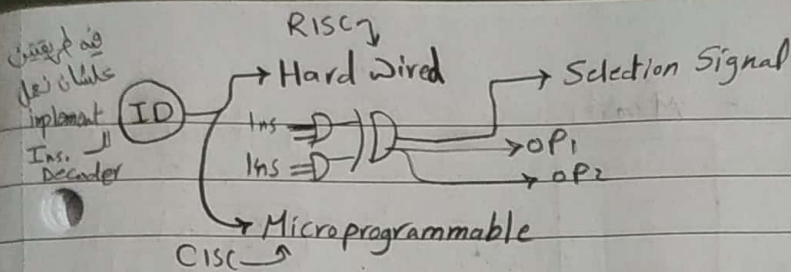
Note: - The decoding time in RISC is much smaller than CISC  
- In Average the instruction in RISC takes 1 Cycle  
" " " " " CISC " 4 Cycle

So, the performance is equal but the line of instruction is faster in RISC than CISC

الواحد بيتنفذ في نفس الوقت في ال RISC وال CISC

لكن في ال RISC بيتغير لأكثر من طرف ال الواحد أسرع.  
ال CISC عنده ALU معقد لكن عنده compiler بسيط وال RISC العكس وال واحد لنفس





Decoding phase done through software program like if condition etc to produce selection signal, OP1 & OP2.

أبطال Hardwired أوسع من Software لكن ال Hardwired أسرع بكثير من ال Software

ال ALU بتاعت ال CISC بتكون 1000 inst. فيكون حجم كبير ويكون معقد عشان فيكون أقل " " RISC " " 100 inst.

ال ID بتاع ال RISC بتكون Hardwired فيكون حجم كبير " " ال CISC بتكون Programmable فيكون أقل

ال CISC مش منطقياً انه يكون Hardwired لأنه عنده ins. كثير وبالتالي circuits كثير فيكون معقد جداً ومشكلاً وبما انه Soft وهو بطيء عشان كده بيأخذ 4 cycles للسطر الواحد أما ال RISC بتكون Hardwired لأن عنده ins. قليلة وبالتالي circuits قليلة فيكون بسيط وتكلفته معقولة وبما انه Hard فهو سريع جداً عشان كده السطر الواحد بيأخذ 1 cycle.

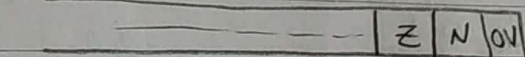
Register File: memory location for specific use. أي memory مخصصة For specific Register

PC: Program Counter: hold the address of the next instruction. (initial value is zero & increment every fetch after)

IR: Instruction Register:

ACC: Accumulator:

PSW: Processor Status Word



OV overflow Flag  
N Negative Flag  
Z Zero Flag

لنواتج العملية رقم بالمالب  
لنواتج العملية ب Zero

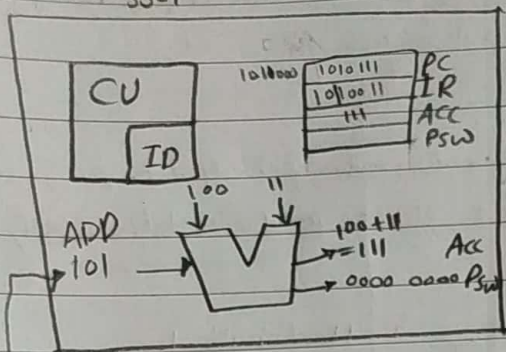
GPR: General Purpose Register

أكد أوسع من ال Flash و ال Cache (هناك رأي نكتب عليه في ال Cache)

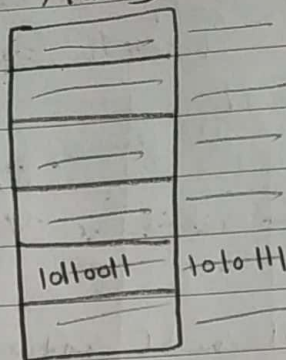
Note two Outputs From ALU to PSW & ACC.



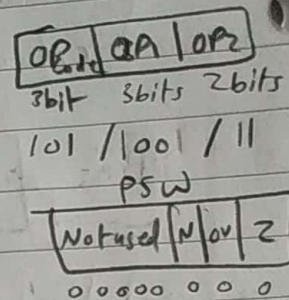
## Processor



## Memory



## Instruction Format



## ISA

|     |     |
|-----|-----|
| AND | 110 |
| OR  | 000 |
| XOR | 001 |
| ADD | 101 |
| SUB | 111 |

ما يجب تعريفه نوع الـ Processor، الـ RISC

لنأخذ الـ instructions كأمثلة

One Cycle

بأن الـ instruction بتأخذ

## \* Memory

Volatile

متطايير

Non-volatile

RAM

RAM

ROM

DRAM

SRAM

Based on Capacitor  
لوشحنون  
بتي 1  
ولو العكس  
يبي 2

Based on transistor

cut-off mode  
Saturation  
Short circuit

needs Refreshing Cycle

↑ (w)  
↓ (w)  
↑  
↓ (w)

① Performance  
② Size  
③ Cost  
④ Power consumption

Note

Based on capacitor needs "Refreshing circuit" to recharge the capacitor

RAM قبل ما نغيره علينا ان نغذيها  
Dynamic



- The main reason to choose SRAM in  $\mu C$  is because it consumes less Power.  
 Rule: It's forbidden to see microcontroller using DRAM

- الفرق بين SRAM و DRAM هو

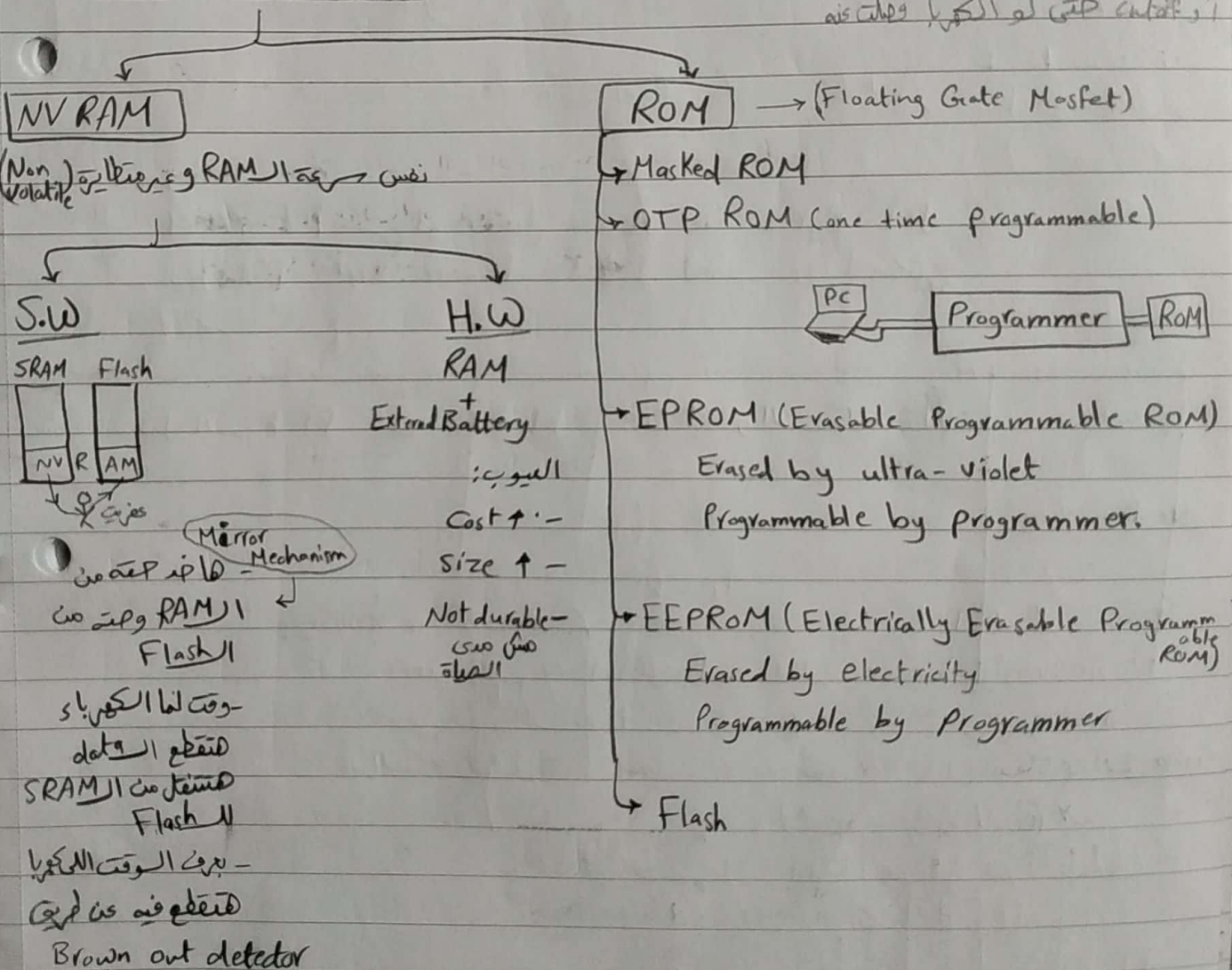
$$\text{Price SRAM} = 60 * \text{Price DRAM}$$

→ double data buffer dynamic RAM في عالم الـ 8 بت

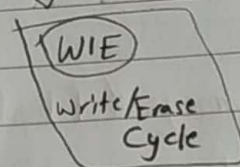
## Non-Volatile Memory: (Floating Gate Mosfet)

هو الـ Floating Gate Mosfet

أو الـ EPROM في عالم الكبريا



| Flash        | EEPROM      | Access means               |
|--------------|-------------|----------------------------|
| Block access | Byte Access | wright or read             |
| Cost ↓       | Cost ↑      | example from real life:    |
| life time    | 100,000     | EEPROM Access time → 1μsec |
| Endurance    | 10,000      | Flash Access time → 10μsec |



$$\text{Price EEPROM} = 60 \times \text{Price Flash}$$

ال EEPROM أسرع لو سجل Access لعدد Bytes أقل  
ال Flash " " " " " " " " " " " "

°° يستعمل ال Flash لأنه أسرع في الكتابة والكتابة قليلة

\* RAM why it has a name of Random Access memory?

يعني ال time التي طافه على ال access أو ال memory

هو نفس ال time " " " " " " " " " " " "

°° ال time ثابت " " " " " " " " " " " "

\* ROM (Assignment)

له حقوق Read only memory مع دايما بقدر أكتب عليها

ال RAM أسرع بكثير من ال ROM لكن ال Volatile RAM عكس ال ROM