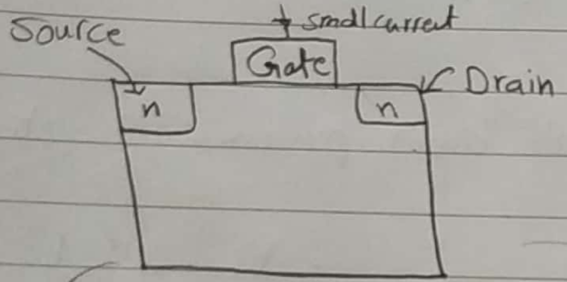
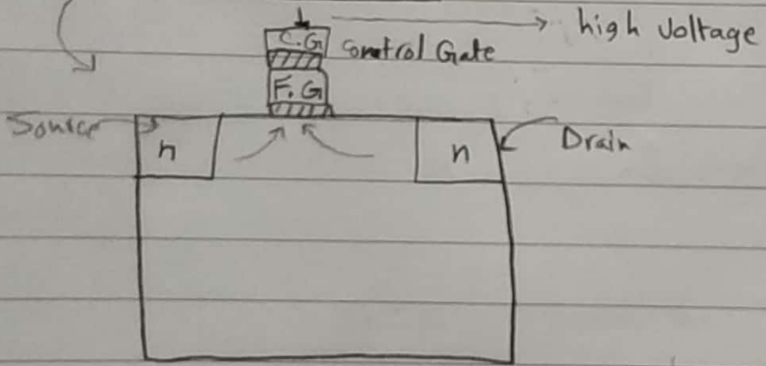
Eng. Ahmed Assaf — Embedded Systems — 17.11.2018

* F.G MosFET:

- يحتاج high power من خلال flash driver علشان نعمل Access عليه

- الـ Access time بتاعه عالي علشان كده بطيء





علشان أشحن

- Erasing State بتاعت الـ ROM بتكون عليها one's

- علشان أتغلب على الجزء العازل النزول بعمل Apply high Voltage فيسحب الشحنات من عند (الـ Drain)

فـ الـ F.G تكون فيها الشحنات المسحوبة و الـ Drain فاضي وده الـ Erase state

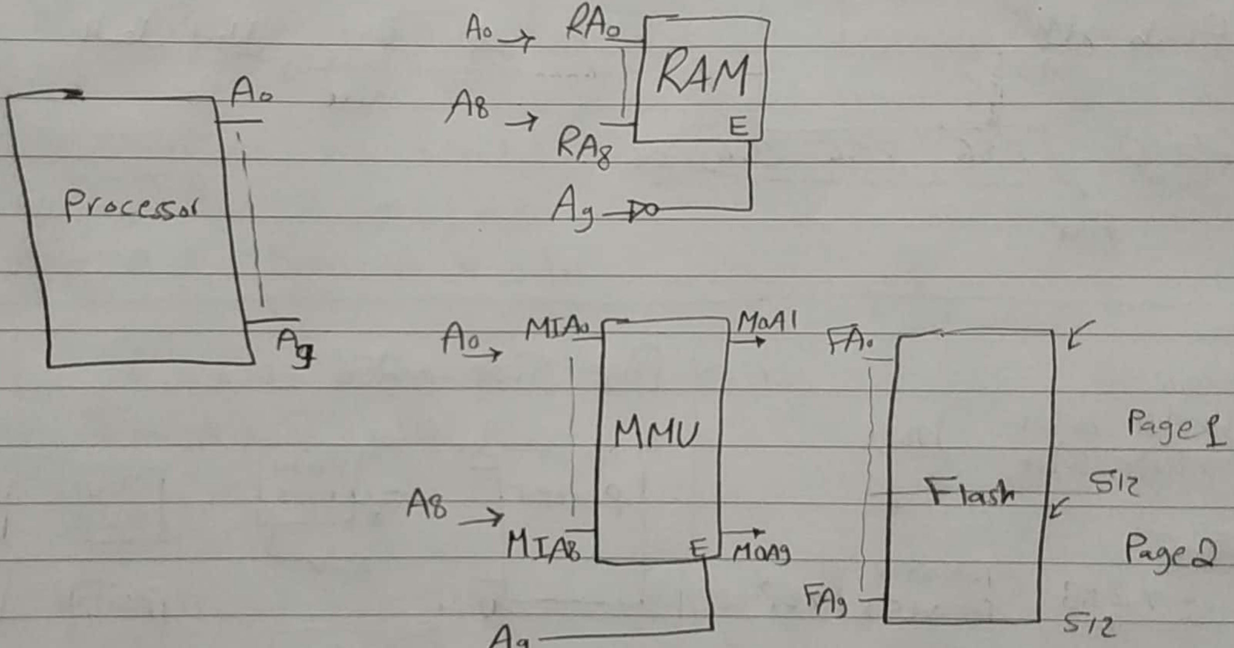
وعلشان كده بياخد وقت في الـ Access

* Memory Management Unit

Processor → Add 10bit → 1024 B
→ Data 8bit

RAM → 512B
→ WZ 8

Flash → 1 KB
→ WZ 8



Physical Address ← الـ Address اللى طالع بعد من الـ MMU

Logical Add. أو Virtual Add. ← الـ Address اللي طالع من الـ Processor واللي بيكون افتراضي لأنه ~~بعد~~ لسه هيدخل MMU ويعمله تصحيح ،

Address Map

	A9	A8 ——— A0
RAM	0	x ——— x
Flash	1	x ——— x

Normal Case
↳ Access time → t_A

After MMU
$$\text{Access time} = t_A + MMU_{delay} + \text{Page switching time}$$
(Page switching time) neglected if Access happens in the same page several times

Flash → 1KB
w28

الحل لتقليل Page switching time هو

near → (256) [256 256 256] → Far

System
Processor
- Add 10 bit) 1024
- Data 8 bit

Flash
↳ 2KB (Near → 512, Far → 1.5 K) → 6 Pages
w28

SRAM
↳ 1KB (Near → 256, Far → 0.75 K) → 3 Page
w28

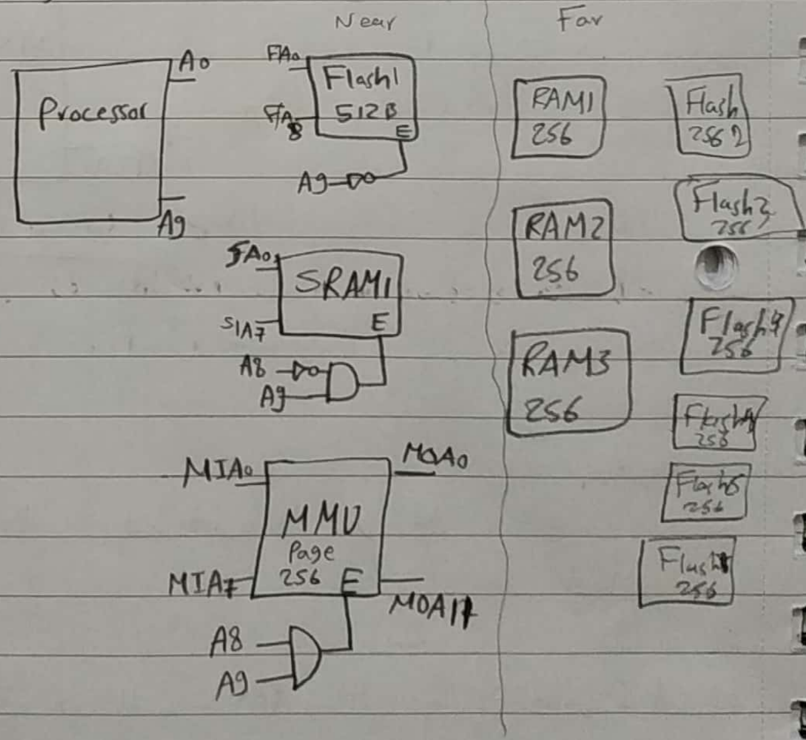
How to cal the no. of output in MMU?
Ans. I have 9 Pages

So I have 9×256 = 2.25 KB

10 bit → 1024
11 bit → 2048
12 bit → 4096 ✓
↳ page 16 كده عندي ، أنا

Page 7 = 16-9 فاضل من ←

Page Size → 256



ملاحظة: لـ رقم الـ Page size يكون من مضاعفات الـ 2

	MOA_{11}	MOA_{10}	MOA_9	MOA_8	MOA_7 ——— MOA_0
Flash1	0	0	0	0	x ——— x
Flash2	0	0	0	1	x ——— x
Flash3	0	0	1	0	x ——— x
Flash4	0	0	1	1	x ——— x
Flash5					

Page Register 11 1111 1111

مرحلة ① - أول حاجة بعت على الـ Add Bus 11 1111 1111 أقول أنا هغير الـ Page و يكون رقم الـ Page باعت على الـ Data Bus

مرحلة ② - تاني حاجة بعت الـ Address بتاع الـ Loc. جوه الـ Page و بعت على الـ Data Bus

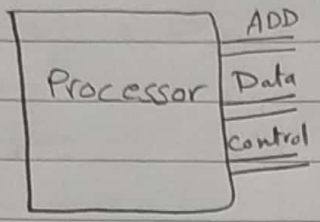
وده علشان كده مش بعرف أكتب في أي مكان في كل Page لأنه 11 1111 1111 دايما بيشير إنه indication أنا عايز أغير الـ Page,

Example to select MMU

1. Add: 11 1111 1111 عايز أغير الـ Page → كده يكون بكلم الـ Page Register
   data: 0000 0100 → رقم الـ Page
2. Add: 11 0000 1110 to select MMU
   data: ——— الرقم اللي هكون عايز أكتبه

Von-Neumann (The Father of Digital Design) VS Harvard Architecture

Von-Neumann



RAM

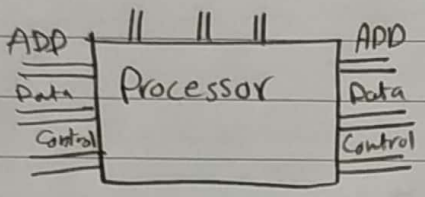
Flash

Pros
- Simple in design
- Cheap

Cons
- Slower

Harvard

I/O



Flash

RAM

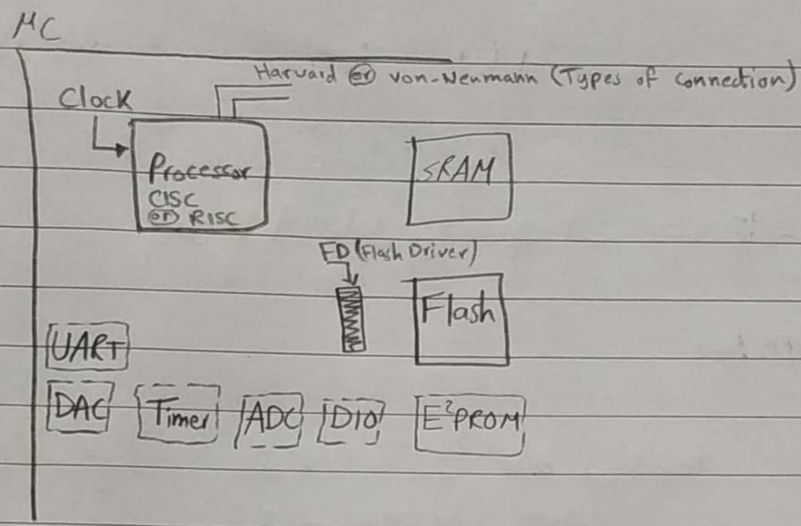
Pros
- Complex in design
- expensive

Cons
- faster
- Pipeline is capable.

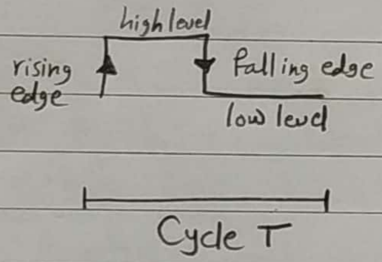
Assignment

؟ Von-Neumann و Harvard و الـ RISC و الـ CISC هل في أي علاقة بين الـ

μC



Clock



Periodic time T

Freq يعني اتكرر كام مرة في الثانية

HZ يعني كام Cycle في الثانية

$$Freq = \frac{1}{T} \text{ HZ}$$

MIPS (Million Instruction Per Second)

Pipelining ⇐ عدد MIPS أكبر من عدد الـ Mega hertz

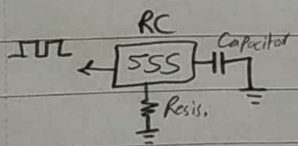
Clock System

EMI → Electric magnetic field

Clock System → Electrical / Mechanical

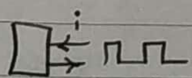
Electrical

RC



Mechanical

- مادة طبيعية بتاخد كهربا فتتهز (osillate) بين الـ Low Level Voltage و high level Voltage



	RC	Crystal oscillator	Ceramic oscillator
Accuracy	↓	↑	~
Cost	↓	↑	~
settling time (لحد ما ابدأ)	↑	↓	~
Noise immunity → temperature	↓	↑	↕
→ EMI	↓	↑	↓
→ vibration	↑	↓	

ATMega 32

- n-bit Microcontroller
  ↳ n-bit ممكن تبقى size الـ operand مع ALU بتتعامل يعني

- Advanced RISC Architecture

  - Fully Static operation — يعني كل حاجة فيه transistor based على (Refreshing مفيش circuits)
  - 16 MIPS — أعلى سرعة 16 MIPS في 16 MHz
  - On-Chip 2-cycle Multiplier — يعني الـ Multiply بتاخد 2 cycle

- 32 Kbytes of In-System Self-Programable flash
  يعني فيه Flash Driver جواه

-

2 Assignments

Alldatasheet.com

1) CISC (or) RISC
2) # of instruction
3) Longest time instruction
4) Flash??
5) SRAM or DRAM?
6) EEPROM?

→ PIC 16F877A
→ PIC 18F4550
→ STM32F103C8T6 — ARM Based
  32 ↓ Processor 32 bit

Companies

Analyst يفضل يكون مش مهندس

RFQ : Request for Quotation

① ↘

CRS : علشان RFQ مش مكتوبة بشكل مش فني

↘ Customer require specification

②

System Engineer مهندس شاطر جدا كسوفت وير وهارد وير

→ SRS : Software Requirement specification وده أغلب الشركات في مصر

HSI : Hardware Software Interface

level in Egypt →

Architect ← بيقسم الsoftware الكبير لـ milestones

* Static Architecture

define every modules and their function then their arguments and returns.

③

* Dynamic Architecture

RAM Consumption

ROM Consumption

execution time (Kam cycle 3lashan tetnafez)

→ GDD "HLD"

GDD: Global Design Document

HLD: High Level Design

④

Designer

→ CDD: Component Design Document

UML: Unified Modeling Language

⑤

Developer

→ Code → **Tester**

⑥ unit testing

⑦ Module testing

⑧ integration testing

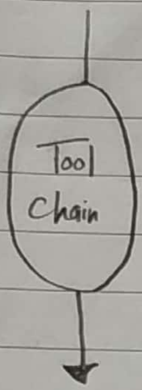
Validator

⑨ Validation testing

Development

"Source file"

File.c



"executable file"

File.hex
- .elf
- .exe
- .Coff

Burner → Flash

HEX file

: | CC (2 digits) | Address (4 digits) | RT (2 digits) | Data (2*CC) | CS (2 digit) |

كل سطر اسمه Record أو Rec و بيبدأ بـ (:)

CC: Character count
مكتوب فيها عدد الـ Bytes

بيعبر عن عدد الـ Bytes جوه الـ Data bytes

RT: Record type
- 00 Data
- 01 EoF "End of file"

CS: check Sum

- طريقة علشان يضمن بيها إن الحاجة اتحرقت صح

مجموع كل الـ Bytes في كل سطر و بعد كده أجيب الـ 2's Complement و بعده أكون جبت الـ check sum للسطر ده.

↑ خد الـ Least Significant Byte و بعدكده

- الـ tool chain و الـ Burner بيحسبوها بنفس الطريقة

- الـ Burner بيقارن الـ check sum ولو فيه فرق (بعد الفرق) هيدي error

instruction byte
→ Address
→ Opcode
→ operands

ins. → 16 bits
2 bytes

8B BA

1000
1001
11 1010 A
00 1011 B

R_{20} 1111 15
0010000 16
0001 17
0010 18
0011 19
0010 0100 20

2nd → Out 11, R_{20}

Opcode

-Instruction set

1) xx xx
2) construct op code
3) write new Ins.

high Low
1011 1001, 0100 1011
B 9 4 B

1011 1001 0100 1011
op code
operand 2
operand 1

94 0C, Hex
1001 0100 0000 1100 Bin

1011 1001 0100 1011 Bin
OUT
00 1011
11
20

OUT 11, R_{20} Assembly