ES Concepts

→ **Computing system** → General Purpose نظام شبه مُتوقع

Processor ⌐
Memory ⌐ ← Consists of :
I/O ⌐

↳ Special Purpose (Specific) قابل للتخصيص

↳ مقصود الـ ES

, has Constraints 80
① Power ② Cost ③ Speed / Time
④ size

↳ Two ways to instruct 80

[1] JoB	[2] SoC
"system on Board"	"system on chip"
لوحة كبيرة نسيت اولها	IC صغير
ووضّفتها .	بياخذ كل العناصر ⟸

size	larger ↑	smaller ↓ ✓
Cost	more expensive ↑	cheaper ↓ ✓
	[نسبة لإضافة العناصر كلها component]	[نسبة لقلة العناصر اسوا]
Power Consump.	more ↑	less ↓ ✓
	[losses نتيجة سخونة فراغات]	
Performance	"The same"	
, But, Configurability	✓ ✓	✗
	system applicable	ما الـ chip
	اوضح انه مقدور تطوير اقدر	خلاص مقدوره
	أغير حوايل الـ memory	
	و اسوي .	

Microprocessor, Microcontroller, RAM, RoM, SoC
$\xrightarrow{CSI}$ MP   MC

★ IC : integrated circuit → single chip

★ VLSI : very large - scale integrated circuit
يعني عدد اكبر بكثير من الـ gates في المنطقة
الواحدة من الـ chip
∵ **Higher Functionality**

oooo **Moore's law :** بيعبر عن الـ Transistors بتتضاعف و duplication
[expectation]   كل سنتين في مساحة الـ chip

★ MPU : Micro Processor Unit
                                                Central Processing Unit

Processor	Microprocessor	CPU
اي بلوك عبارة عن سباكات يقدر	جديد new Processor يستخدم	بيشتغل
مجموعة كل واحدة	الـ Transistors	Processor
بمثل Bit	يعتبر احدث	بيكون هو الـ Master
Uncharged charged		"Primary"
		يعتبر processor التاني
		slave
		"Secondary"
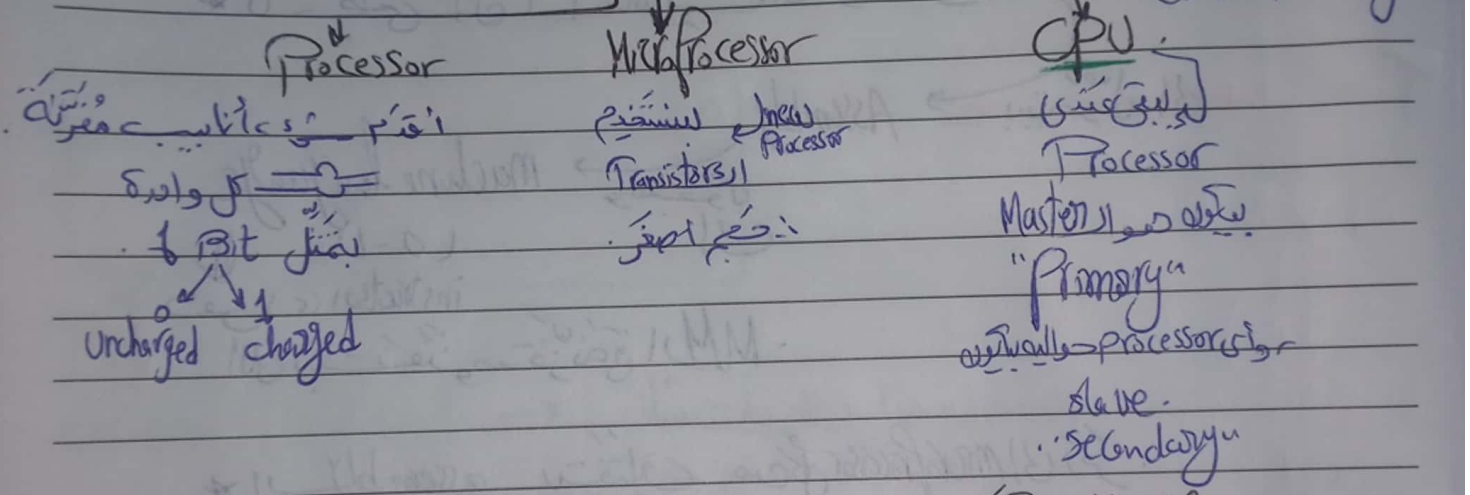
★ 4 Micro Processors بيتكون من حاجات زي طالبة : ALU c Registers c CU...

★ MCU : Micro Controller Unit
في بعض من وظائف Computing system
في الـ MPU معه جزء من الـ MCU

let:
SoC ← MCU | MPU | MM | I/O |
MCU with | GPU | DSP |
high Performance)
                  CPU ———→ Secondary
                  "Primary"   Processors

GPU : بيقوم التحكم initial
graphics بالـ بقية الـ

DSP : في الاشارات التحليلية
بقية و functions و
مساعدة للـ ... المعالج ...
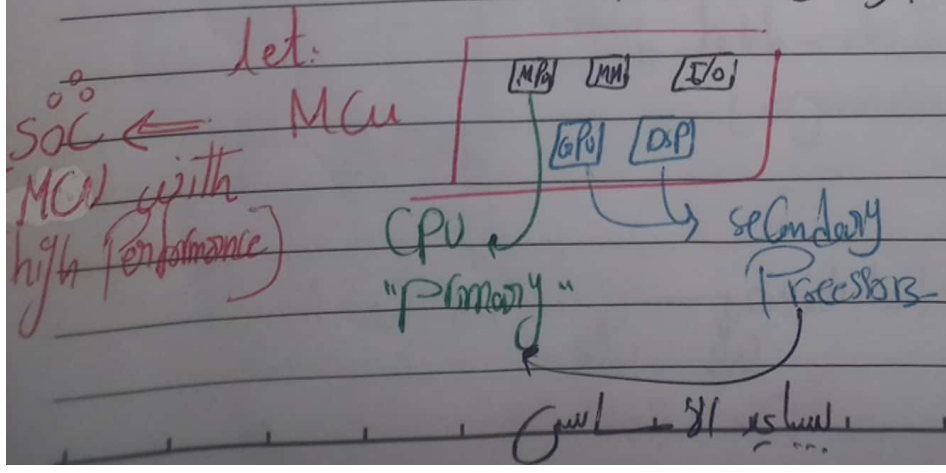
microControllers
with lower
Performance

oo There're Two Types of applications → SoC
زي Rusberry Pi

في امبيديد → لينكس

Bare Metal
→ Software → الي بتشتغل معاه

OS
Application.

APP layer (بنكتب هنا الي احنا بنتعامل)
↓
Drivers (الي بيتعامل HW)
↓
HW

APP كل واحد
↓ بيتحكم الي
Terminal تحته
↓ بيعرف
Driver
↓
HW

سينسرز وبلاتينيم

[SoC]

pins

كله حتى هو ده الك ... SoC :

kit
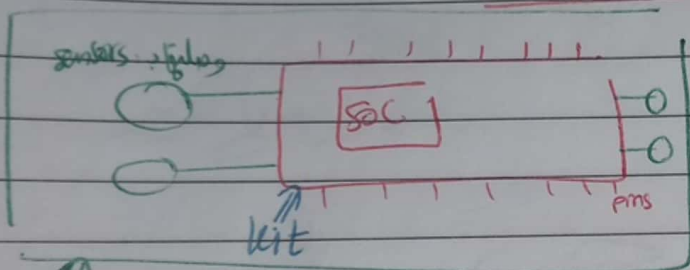
ECU
Electrical Control Unit → بستخدم في العربيات لأنها
(العربية أكتر من 200 ECU)

"BUS المواصلات التي الي سيت انا شايل ل MM c CPU] c E...
set" ونختصر ب ... data c addresses

Processor
[V C.U]
[AlU]
[Reg.]
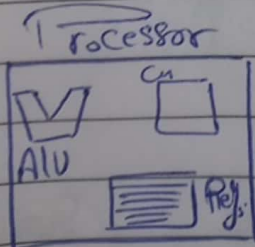
Performs Fetch - deCode - ExCute
(organisation اسم)

→ مفهوم الـ insTr انا طالبين عليه
HW SW
Inst.DeCoder Compiler
format الي بيعمل الي (بتفهمه)
الي بيعطي ال micro بعرف بحل المسائل

⋆ ISA ⋆
"Instruction set Architecture"

RISC	**CISC**

RISC
* Reduced instruction set Computing

Compiler] ⋆⋆ معناها زيادة SW في compiler [فال
شكل الكلام بيكون اسهل عملية قليلة
since it's reduced ↓ Then

⋆ ISA ⇒ Hardwired
(decoder)
IP [1.6] كله doble
reduced bof وهو logic اكبر
gates
فلسفته بيزيد اكثر كثير

CISC
* Complex instruction set Computing
⋆⋆ معناها بيقلل SW كمان
عنده already Complex
فال compiler بيكون متعب كتير
بس بتكون كتير اوامر complex then

⋆ ISA ⇒ Memory Mapped
بيتم تخزين جوه الـ memory الاوامر في inst
وبتبقى
Grates بيتم تشغيلها خارجية بس
كبيرة

∘∘ Comparison Between
them Both acc.to
ES challenges ∘∘

	RISC	CISC
size	ID Higher↑, Alu↓ lower	ID Lower↓, Alu↑ higher
∘∘ Almost Both have same size		
Cost	↑SW, HW↓	SW↓, HW↑
∘∘ // //		
Power Consump.	ID Higher↑, Alu↓ lower	ID Lower↓, Alu↑ high
∘∘ // //		
Performance	الـ inst بتتنفذ بتبقى بشكل متسلسل وقليلة	بتبقى في searching
∘∘ // //		

$$\boxed{\text{Register files}}$$

GPR	SPIFR
(General Purpose)	(special Purpose I function)
store data (any data for any operation)	① PC [Program Counter]
*Temporarily	② stack Pointer
	له أنواع كثيرة على حسب هو
	بيشاور عين
	③ AC [Accumulator]
	حالياً مفيش موجود
	④ IR [instruction register]
	⑤ PSW [Process status word]

signfty ⟶ بشكل في اجباره اخو operating الـ ALU عملها
بس ممكن في حاجات من نوعي اصلاً
الـ ALU
(يعني Inc ليه) ← وظيفة الـ flags من هيستي
Bit
Flag
وزارة هيعمل العديد ولغير والالغا هيخزن
مجموعة من الـ flags
كل Bit بتشكل على حاجة
دي من الترم الثاني
لو موضعها الـ flags دي كلها بتغير
أخر شي حاصل في الـ ALU
(HM) تحفظتي الـ (MM)

Capacity   Speed ✓   → organization.

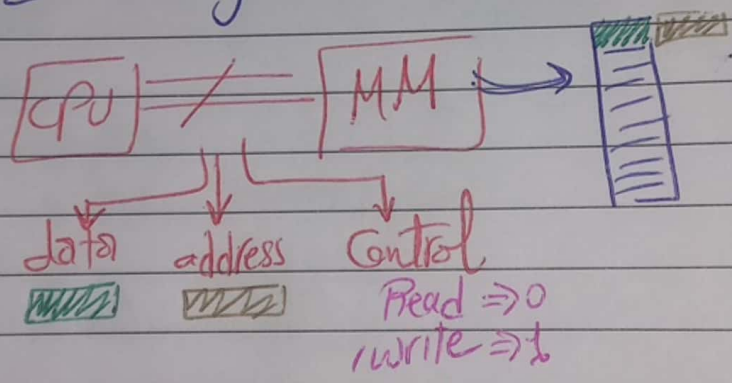العنوان address   (**Memory**)   اختيارات
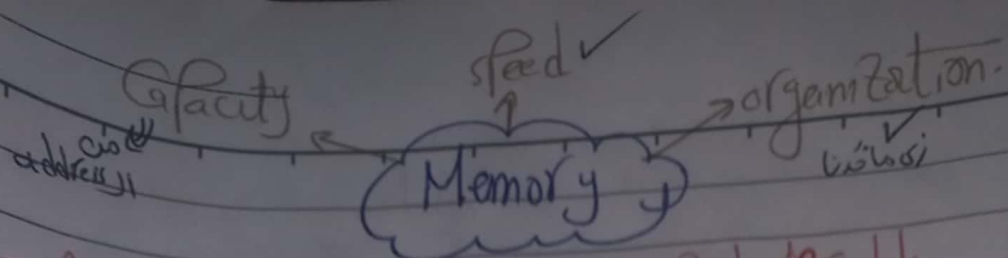
⟹ Access Time ↑↑ inc.   ∴ speed dec ↓↓

    and vise versa.

    ↳ access Time of write > that's of read

✶✶ Basic memory element ⟹ D-Flip flop.

∵ Using 8-D flip flops ⟹ 8 Bit Register.

[الـRegisters من different نوعية على حسب تركيب الـflip flops وطبعاً فرق وتكون]

[note that]   n : number of address lines

     ↳ 2^n memory locations for addresses.

data   address   Control

Read ⟹ 0
/write ⟹ 1

∵ Read : ① Control ⟹ 0 , ② address ⟹ ✓

    Then I 'll receive data on " data " lines

∵ write : ① Control ⟹ 1 , ② address ⟹ ✓ ③ data ⟹ ✓

    ∵ وهيتم store
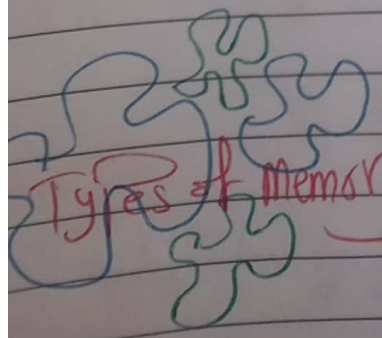
Types of memory :∘ ① Volatile ② Non-Volatile ③ Hybrid

     ↳ RAM   ↳ RoM   مزيج بينهم

                                      الاتنين مع بعض

١) Volatile ⇒ RAM /RWM   R/w < RomJl عكس أي

random   Randm
access   working
memory   memory.

DRAM   SRAM   [كبير جدا]
"Dynamic"   "static" → الCache   ابو
[→ mosfet + capacitor   في Transistors ال
  → needs refreshment
    circuit "DRAM Controller"   ✓ Faster
    (has priority over CPU)   xx Higher Cost.
(تفقد data في المتصل و تعمل)
(تحسن الaccess و %85)   ⇓
       Time
                                we Used it in Es →
ADV. ✓ simple Hw   with some Constraints.
     ✓ High density
     ✓ No Power Consumption
     ✓ Low cost per Bit.   ولكن مكلفة ١
DisADV xx dec. speed   كبير : ال bus
       (inc. access time).   higher
                               power Consump. [الوجود ال Cap.]
                               Consump.

           Comparison المقارنات:

size   ↑ Higher [higher density]   Lower ↓

cost.   ↓ Lower   Higher ↑

Power   ↑ Higher [بسبب]   Lower ↓
Consmp.   [ال Cap.]
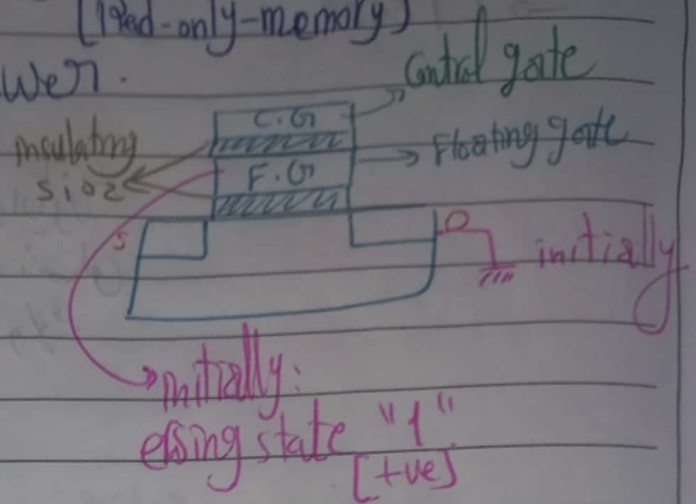
Performance ↓ lower   ↑ Higher.
(speed)

2) Non-Volatile → Program memory "RoM"
[Read-only-memory]

★ has higher access Time , slower.

★ Based un FGM

Floating Gate Mosfet.

دي ترانزستور عادي

القوة اللي ←———— واللي دي ي قوة

ال Bit نفسه

Cntral gate

insulating SiO2

Floating gate

initially

→ initially:
ering state "1"
[+ve]

ج اما الكابة الوصية اللي تقدر تضيفه عليها دوار الساعة الحالي
وتكتب عليه من الاساس

← Flasher

Types of RoM 80

Maske Programmable	PROM	EPROM.
بترق الكوندت واحدة عجل الأشياء	Programmable RoM	Erasable PROM
معرفش اجل ال Bugs	من جوانب يضبس	↓ By ultraviolet
└→ OTP	هو ال user	فوقت ما
one Time Program	└→ OTP	✓ Erasable for thousand Times.
البرنامج هيشتغل لوحده	بتجي Blank وانا	✓ Non Volatile For many years
★ Toys , Bioschip	بيرتضم وضعه	×× data affected By noise
└→ cheap		& radiaton
		, leading to corruptions
		بتخطا عوبي
		وطبعا حوار ال uu نكون
		من ممكن يدخله

(from RAM → R/w
(from ROM) → Non-volatile

3 Hybrid

(E²PROM) EEPROM	Flash	NVRAM → Cache
[Electrical EPROM]	↳Electrically	[Non-Volatile RAM]
↳ تقسيح بالكهربائ	بركهربا	أو نوعين

SRAM   SRAM

High ↳Endurance 100 k   ↳Block access   ⊕Battery   E²PROM
[أقصر إذا كان update]   [Sector-by sector]   Battery
↳Byte access   Bytes بس بتروح   ⇓Faster
أكتب في كل Byte كده   ↳نقدر Faster   than RoM   xx High cost
↳ expensive   per Bit
[High Cost perBit]   lower↳Endurance
مقترنش   10 k only   ⇓
أحسن سنقل الأزال كبير   ↳ lower   data ال
                           Cost per   متعنى
internal   external   Bit   Battery لو
MCU الجوا   بره MCU ال   متقى بالقوة
                  [اشتري chip]   مواقر أجيب   Power أول ما
                  [لو أحطها برا]   كبيرة أجبه   يفصل
                                  كده.
                  Comm. ال متصل
                  protocol   منتشبه أحول
* للجهاز MCU ال من رو أقيل كمية   ال sector ة
* مكونات على كده   Auto من يتعامل ⇓
                                  بس سنقلوا مع
                                  ال Bytes
                                  Flash ال تكوم
                                  E²PROM بت
                                  more complex واكبر
                                  ماتحمى.

لحدور ال Battery أول ما ال Power تفصل وهيخلاص بسنقل ال (E²PROM) (data) ال كل
كابل و مفتش مشنه.
↳ (كمان) EPROM ال SRAM من تاني سينقل تاني > الآلة > تاني يشتغل تاني ك

بنستخدمه... هذا

Cache
memory ←

Cache ⟵ level 1 يعني ⟵
بنستخدم الـ MM و الـ CPU لتشوف الدنيا ⟵
بتأخذ الحاجات من memory ⟵ range CP to memory → one Block ← Controller

+
Controller

addresses
العناصر اللي بنشتغل حول addresses ⟵

[... كل] cache دي في ⟵ | L1 | Gre1 | MPU
                         | L2 |
نشتغل عليها [بشكل سريع] → | L2 | Gre2

| L3 | قريبة CPU
            (الأبطأ)

inc speed ↑ | Reg
                 | Cache
inc size | MM (RAM)
         ↓ | Secondary ثانوية

⟵ **Cache coherency** بتتكلم على أثر كبير في تصميم الـ Caches ⟵
⟵ التفسير يوصل لكل الناس ⟵

★ **Hit ratio** $= \dfrac{\#\text{Hits}}{\#\text{Total}}$ ⟵ هو عدد مرات اللي في الـ hit جت في الذاكرة ⟵
Cache الـ ⟵ Miss ← X التي فشلت

قوة النص
أنشطة

FPU	MPU	MMU
Floating Point Unit	Memory Protection Unit	Memory Management Unit
لمعالجة الحسابات الطائفية العائمة	موجودة في الـ MM	إدارة ذاكرة حوار
improve performance	وفي Cache نفسها	الـ physical و virtual addresses
موجودة ↓	الدنيا نفسها ↓	++ أو الـ program
interrupt معالجة أو تتركز process الـ unit	تنقسم الـ MM إلى	استخدام أكبر لـ RAM
على تحسين الشركة لها عازلة	ranges > كل ست لأمر	virtual نفترض أنه موجود add.
	range في كل MPU بتكون هناك	[التفاصيل تعمل بعيد]
	نعيّن حوار الـ MM لو انكسر	
	Bus معينة تسبب في fault	
	يوقف الدنيا كلها	

I/o Peripherals ⟶ interface واحدة جزء

They have the same range of addresses.

* وَصِيل الحاجات — ١ و *
عند كتابة Achitecture

** Data Bus
** Address Bus
** Control Bus.

Von
—neuman
↳ one
memory system

1st way
to connect
I/o
by memory mapped

Havllord

RAM ─ MP ─ RoM

I/o │ I/o

2nd way
I/o port
mapped

Mapped

2 address
Buses

MP

RAM ?
RoM ?
I/o ?
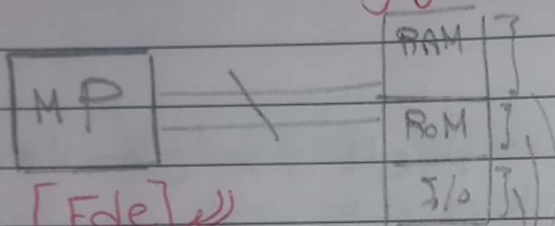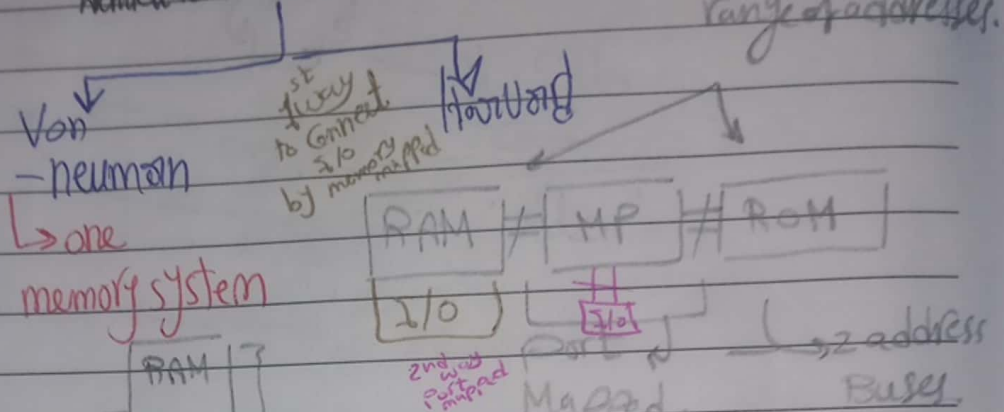
[Fde]
Should finish
the inst. Before
accessing new address.

"مبيقدرش أكلّم اتنين في
نفس الوقت"

Moving
mapped

كل واحد ليه range

لأنه أقدر أكلّم الاتنين في نفس
الوقت واحدة
طب هكون ازاي بكلّم سيس
ازاي؟
BY software

RAM   RoM
load /store   R/w
BY
Assembly

ونتشير اى الحاجات
Arch ↱ تبعتة تتغير
ال اتنين مالتنين

1st way
بيكتب أوكواد كود I/o
load store بيحط
اكبر RAM

In /out
جاية و كتبت فيها مثلا of
↓ مع ↓
Input   output
devices devices

↳ Must
لازم نعمل Compiler مخصص لينا هيوليلنا

وظيفة ال RoM
اننا نسيب read
وسيب write تتمودجودش
نستخدم Flasher لنرمجة ال RoM

2nd way
هكتب I/out
وهقدر أكلّم
كل وقت في نفس الوقت
بقدر واصلة ال Port ببايتن

RoM ↳ لنفس يتوى
دا ال I/o في نفس الوقت

** Pipeline

قادرة [F d e] تجزيء

في كل cycle تنفيذ

مختلف inst
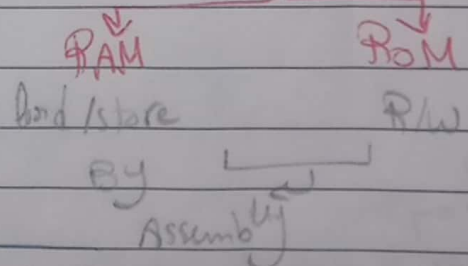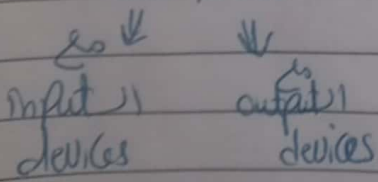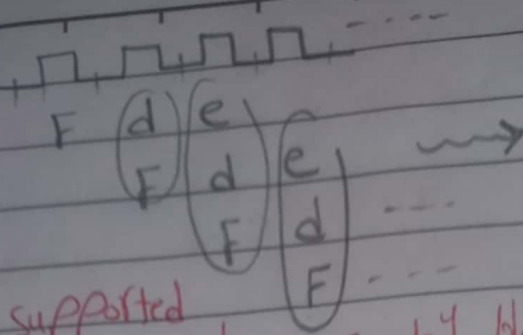
, which is not supported
by Von-neuman, while is so by Harvard Arch...

lower
clock cycles
than

∴ faster
than

∴ Higher Performance than.

* RISC ⟹ Support Pipelining قادرة على تنفيذ أكثر من inst او اكثر
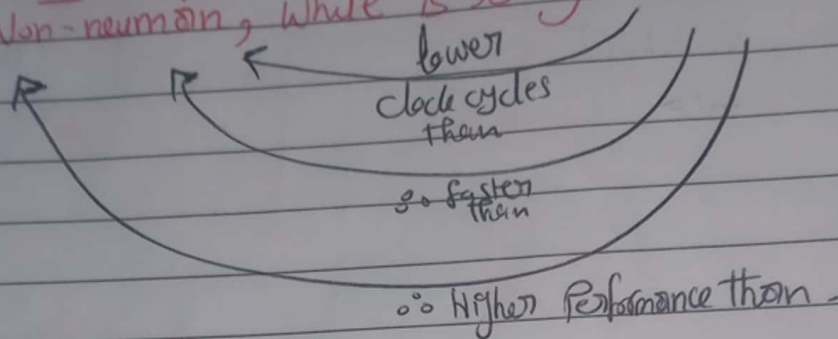
* CISC ⟹ Does Not support pipelining تنفذ عدد واحد فقط inst في كل cycle

** clock ⟹ square wave

$$T = \frac{1}{F}$$

تنفيذ inst. في RISC اسرع لانه في إنشاء أوامر
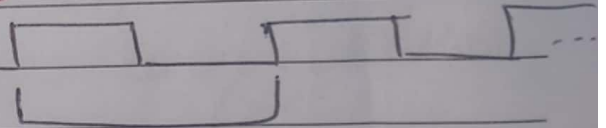
1 cycle → 1 inst.   Per sec.

** clock systems
→ Electrical ⟹ Pic oscillator Timersss+ R+cap
→ mechanical ⟹ Material M/as [material] →
① Ceramic
② Resonator
③ crystal oscillator Electricity

لك بتأثر رسم الدائرة على ← تناسب في اي applications بتاخد منها قوة اكبر

[يبقى ② و ③ او ② دون ③ بس clock بتأثر في Electrical والصوتي vibration clk خالص بفتش استخدمهم في جانب معين Noise]

	① RC oscillator	② Ceramic Resonator	③ Crystal oscillator
Cost	↓ ✓	Between (intermediate)	↑
Accuracy	↓	← " →	↑ ✓
Setting Time	↑	← " →	↓ ✓
Noise Immunity			
Temperature Temp → "RT"ال ↓ بسبب		↑ ✓	↑ ✓
Electro magnetic Interference [EM → "RC"ال ↓ بسبب		↑ ✓	↑ ✓
Vibration [Uib	↑ ✓	↓ ← عالية أو عالية تقليل → ↓ بالvibration	↓
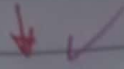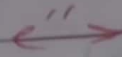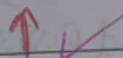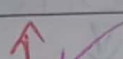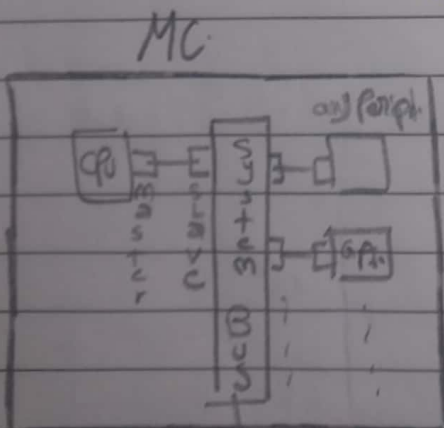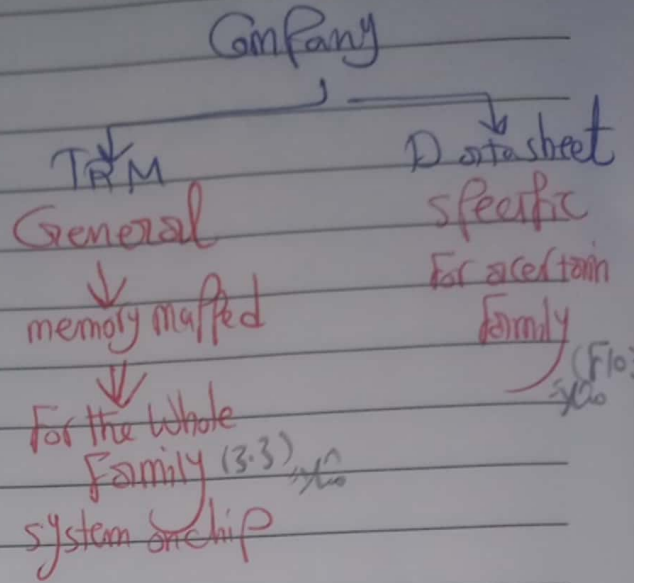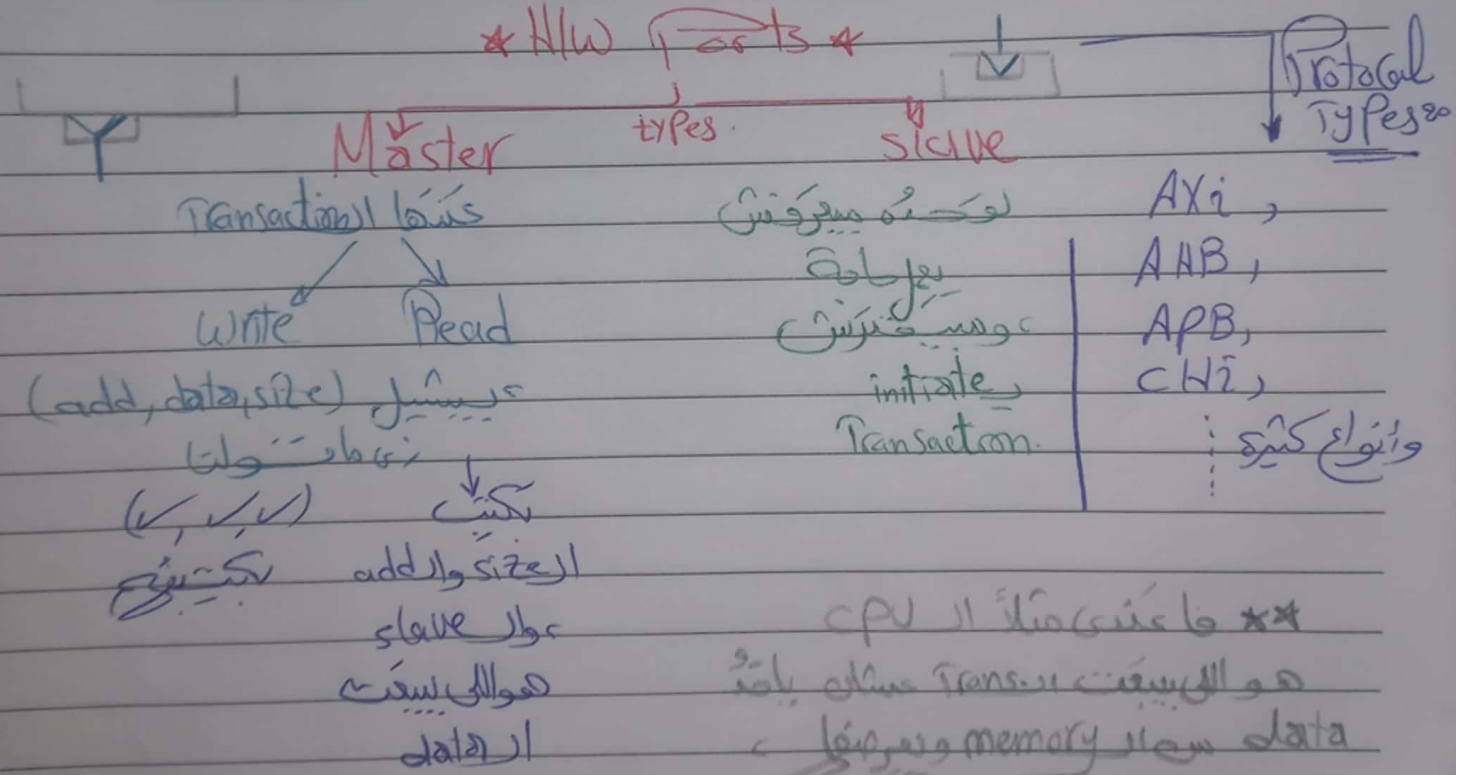
~~~~~~~~~~~~

66 نرجع لموضوع ومنصوع memory 80 mapping

device كل مع range كل تشيل addresses ال

Range = Base + offset
ما تشيل تشغلها من العنوان تبدأ من المنطقة
[عنوان البداية] بعنوان تعريف RG

MC.

→ Peripheral 8   Hw circuit + Registors.

كل البيراف لمعنى
(1 c o) موصلات
عليها تعريف أو فتحة الحالة

★
و   ↳ TRM : Technical Reference Manual
memory map ← specs ال أوصف بتاعت
peripherals ← [design/features الحاجات ال وظيفة]

APB, AHB,   Base + size   [Per.Reg.إسم البيراف مع عنوان]

every Transaction ← data
has. → address
→ size.

البكل لز BUS كيفية تعامل مع hardware الدي
(ال address عندي يبقى offset) وليه بيعمل فيول

الـ BUS ← قبل الـ BUS
الـ address عندي يبقى هو الـ Base
(بزود عليا أساسك) Peripheral الـ BUS

(ال address عندي يبقى offset)
(بزود على أساس على أي) Peripheral الـ Register

** أماكن reserved الـ Peripheral زكاسك أكبر جوه من الـ (Reg.) علشان يبقى بقسمك من الـ overlapping علشان يبقى بعدين لوحب يزود Features أو كده.

⭐ Bus Bridges ⭐

ممكن نوصل كذا Bus بعض بألوان مختلفة،
أنظم الـ Master والـ slave هيختلفوا
ه لستخدم Bridge مآسنقي

AXI ← من ارخص → AHB ⭐
↓ ↓
نوصل نوصل
Masters الـ Peripherals
دا ه

✱ ✱ ReUsability

- data/addresses
- reused to serve
  various needs
- كأن ال GP وعملي مشتاك
- registers نفس تخدم حوائج

Company

TRM
General
↓
memory mapped
↓
For the whole
Family (3.3) مثلا
System on chip

Datasheet
specific
For a certain
Family
(Flo:
100

~~~~~~~

✱ H/W Ports ✱

```
┌─┐                    ┌─┐              Protocol
│Y│        Master   types.   Slave    │▽│  Types
└─┘                                    └─┘

Transaction انشأ   سيرقب لوكسة     AXi,
   ↙    ↓          بطاطة           AHB,
Write   Read      يرسن وسة         APB,
(add, data, size) يشيل   initiate بـ  CHi,
تولت باتش نذت     Transaction.   وأنواع كثيرة
(√, √, √) يكتب
تتكيب   add والsize ال
        سلاس دوال
        سيده لوالي هوال
        data ال
```

[we use DMA]

✱✱ فانت متنا ال CPU
هو الي سيعت transct باتت
data سهاد memory ويرفعل،
ونت اهدار سيكلس cycles،
هو ليشتغل مع ال MM

هو ال CPU يشتغل حنك بابي شغلة
Where, DMA يا M&S
نتقسل ↙   ↓ تتقسم
MM ال   CPU ال
عماد CPU)

★ **AMBA** 8 The **ARM** Advanced MicroController
Bus Architecture

↳ improve reusability , و Flexibility

★ بنستخدم أنواع من standards في كتابة Bus،

أقدر أستخدم الـ standard دي نكتب في عدة مكونات (modules)

اللي اشتريتُ مع الـ BUS بتاعي .

[حقول الشبكة بتختلف لوجُم أنواع Buses مختلفة .]

Bandwidth
Bus ⟶ BW ⟶ fmax

بتتحكم أو في
↘ latency ⟶ بتتحكم في & Performance الـ

الوقت اللي بياخده ما الكومبيوتر.

set this
But to 1 → W
16

Reg
ODR

| Examples |

set this Bit to 1

∵ ODR| = (1 << 16)

و مكتب الـ ODR

و يحتاج منه يشتغل ∴ Absolute = GP{o Base + ODR
Address Address offset

Code: (GP{o _ODR)| = (1 << 16);

طيب علشان يعرف ته هو قوقعة

address ⟹ # define GP{o_oDR Base + offset

void main() {

مكتبة 1 (volatile uint 32_t *) GP{o ODR |= (1 << 16);
software الـ
يشتغل ه قبل أن

Reg } write الـ تكتب ه في الـ بتتغير لأن ا اذن Reg

** في الـ يعني اسمـه volatile و دي الـ data الخاص اشتغل مني
من تتغير و أنا دائماً يقرأها (read)