

* Cache memory \Rightarrow SRAM

وسيط بين ال RAM وال CPU

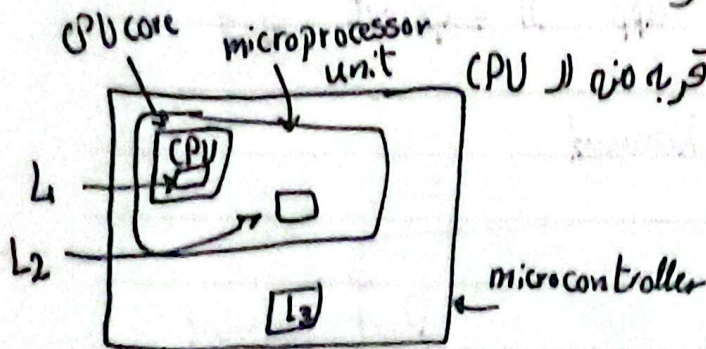
ال CPU أسرع منه ال RAM كانه لازم حاجة بينهم (Cache)

ال Cache يتاويل يوفر وقت بحيث انه لو ال CPU طلب داتا في عنوانه

معيث يجيب الداتا اللي بعده مع الداتا المطلوبة

if CPU needed it \rightarrow we call it (hit)
 didn't \rightarrow it is called (miss)
 hit Ratio = $\frac{\# \text{ hits}}{\# \text{ total}}$

Cache = cache memory + processor



ال Cache يبقى levels على حسب قربه من ال CPU

$L_1 > L_2 > L_3$

extra according to speed

لو 2 caches فيهم نفس ال address لداتا و cache فيهم عديل عليها

الناني فيبقى عنده داتا out of date الحل؟ Cache coherence

\rightarrow Updating all the another caches if one of them

is editing some data

⇒ FPU ⇒ * بتساعد في عمل operations على الكسور

* ممكن تكون جوا ال MP وممكن برا (على حسب الشركة المصنعة)

⇒ MPU ⇒ * بين ال cache & ال main ram

لو في كذا OS على ال MC بييجي دور ال memory protection unit

عشانه تنظم الداتا الى جاية منه ال RAM لا OS المقصود

⇒ MMU ⇒ memory management unit

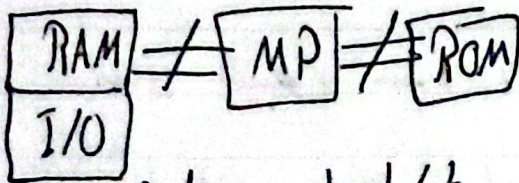
لو ال app memory اكبر من ال RAM له فيفرد ال app على HDD virtual addresses

وي support ال app بال RAM المتاح

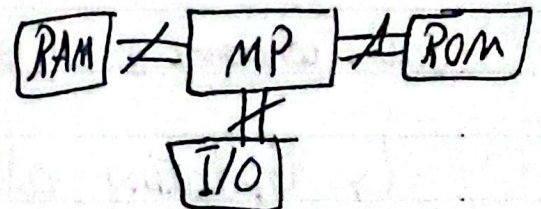
(I/O) hardware

memory mapped

Port mapped



C lang ⇒ load/store



PC \rightarrow Von Neumann Why?

\Leftarrow قلة ال buses : يتصل مع RAM & ROM & I/O بنفس ال buses

\Leftarrow السرعة : يتصل ال Von Neumann بنفس ال RAM ذا بنية أسرع من Harvard

Pipelining \rightarrow Von Neumann can't support / CISC
 \rightarrow Harvard support / RISC



Clock : square wave

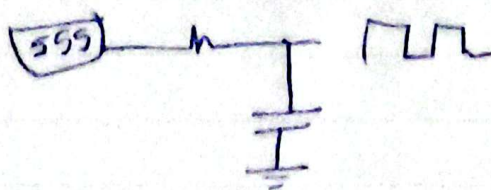
ساعة \rightarrow 8 MHz \rightarrow 8 million cycles per second

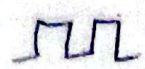
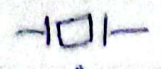
RISC \rightarrow 1 instruction \rightarrow 1 cycle

\therefore 8 MIPS \leftarrow million instr per second

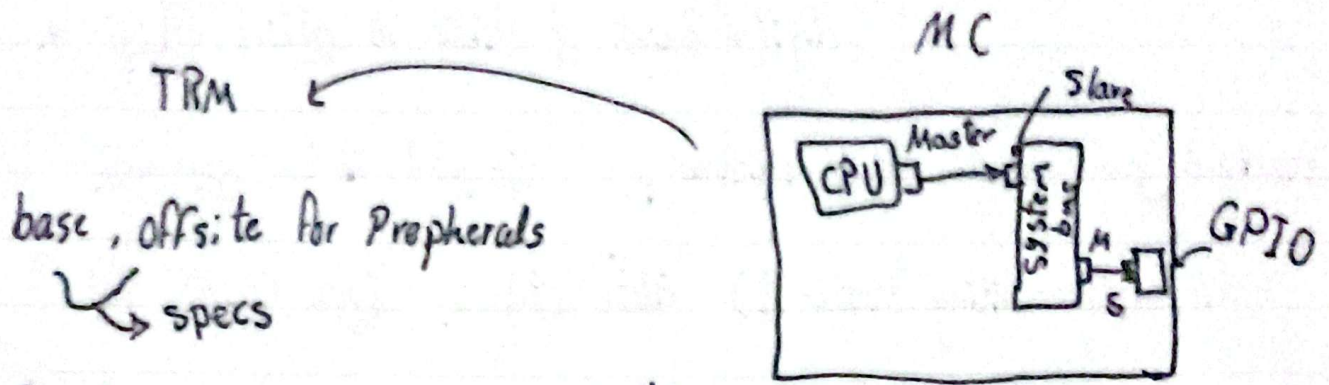
Clock systems

Electrical (RC-oscillator) Mechanical



material + power = 

 Ceramic crystal

	RC	Ceramic	Crystal
cost	↓	~	↑
Accuracy	↓	~	↑
Settling time	↑	~	↓
الوقت حتى الاستقرار			
Noise imm			
temp	↓	↑	↑
EMI	↓	↑	↑
Vibration	↑	↓	↓



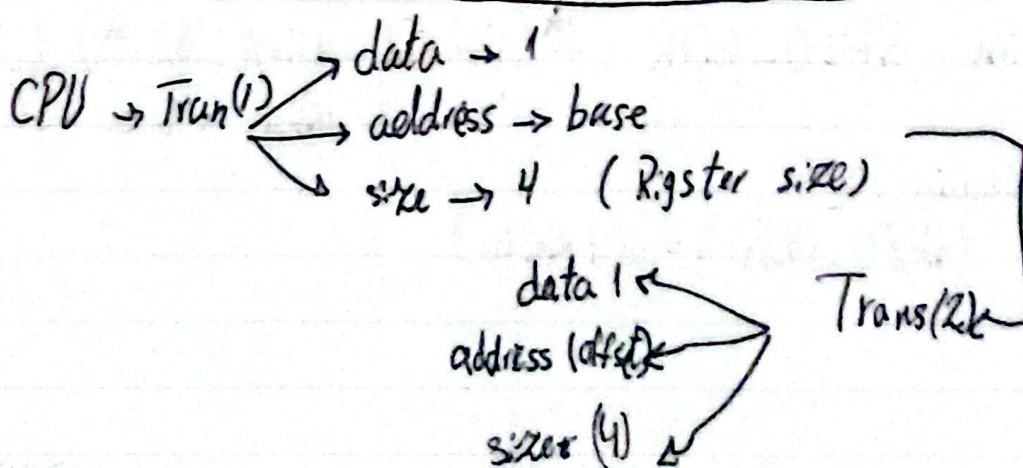
- ① Memory map
- memory size
- size + base
- Each prepheral has base address, end

- ② Prepherals has offsites for registers to access them

* Master & slave must match the system bus type ex. AXT

For turning led on

- ① Pointer \rightarrow (base + offsite) \rightarrow Data Register \rightarrow ①
- ② Pointer \rightarrow Out \rightarrow DR \rightarrow 1 (High)



* تجنبنا مشاكل ال size في digital design

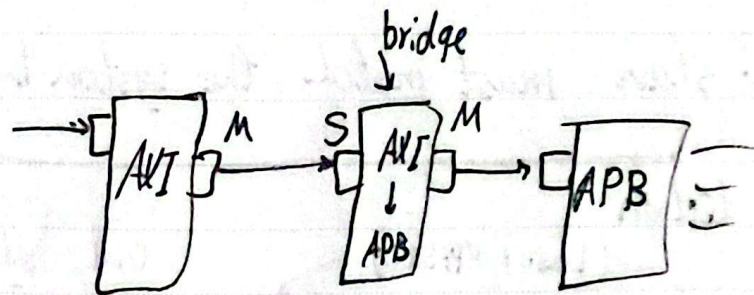
يتم حجز مساحة أكبر من حجم ال registers في ال master تسمى reserved

لو كتبت عليها متعاملش مشكلة (Overwriting Master 2) ex.

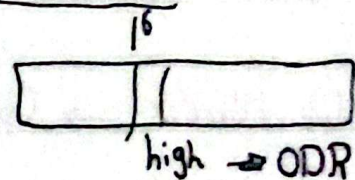
أو مشانه ال enhanced development

لو نوع ال master وال system bus اختلف بنستخدم ال bridge

أو في كذا نوع مشانه قلل ال rate الى



Direct access



#define GPIO_ODR *(volatile uint_32 *) ☐
offset + base ←

```
void main() {
    GPIO_ODR 1 = (1 << 16)
```