SUBJECT: ______ TITLE: ______ SAT SUN MON TUE WED THU FRI DATE: __/__/__

Hybird

Max from Ram and Rom

Read and write ← → un volatile

E²PROM	flash	NVRAM
Electrical		- SRAM + Battary
بيتمسح بالكهرباء		- ERAM + E²PROM + Battary
Endurance = 100.000	Endurance = 10.000	High cost per unit
Byte Access	Block Access	مكيره Battery.
high cost per bit	sector by sector	اول ما الـ Power بيفصل
low density	low cost per bit, High density	بتخزن الداتا الموجوده المهمه على SRAM
internal / External	disadvantage: معرفش اكتب في الـ byte، لازم في sector كله	خلال few millsecond

Cache memory

SRAM → levels

CPU ↔ Cache ← RAM

CPU بيتكلم مع Cache — Cache بيتكلم مع RAM

سرعته عاليا — كاينين حاجه في النص — سرعته اقل

CPU → cache ≠ RAM

cache → cache memory + controller

لو هيجيب من لو كاير address 26

الـ RAM 26 وكمان نتويه تانيين Range

فمن المتوقع ان الـ MP يحتاجها

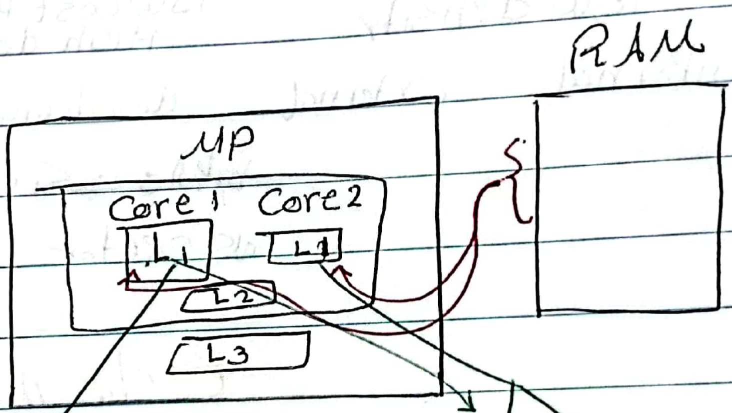
فلو CPU احتاج منها ومرنا وقت

لو لقى العنوان ← Hit

ملقهوش ← miss

cache → levels

$L_3 < L_2 < L_1$



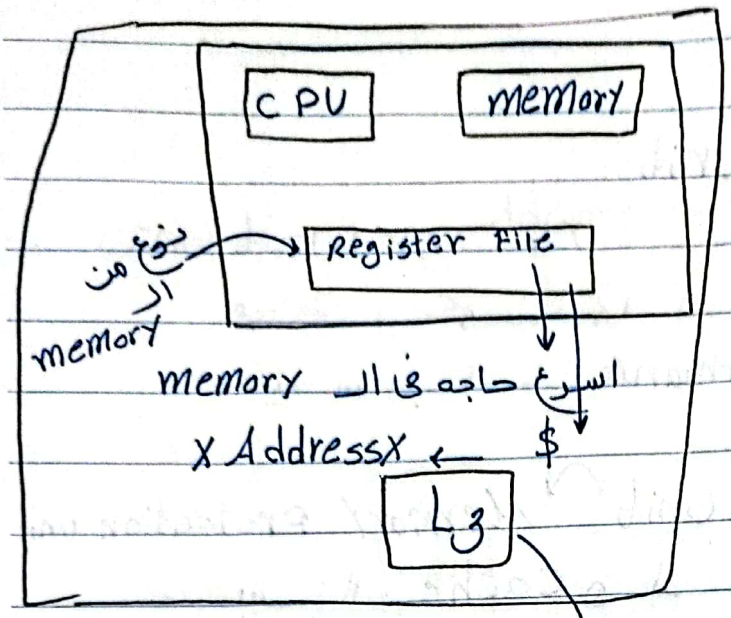
لو هياخدوا نفس الـ Range من الـ adresses

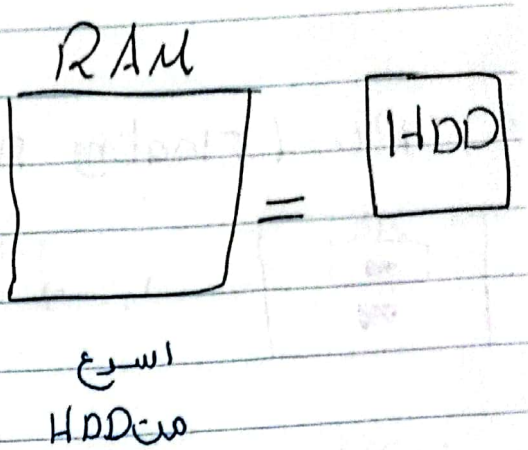
ممكن تعدل على var معين ← مش هيعرف

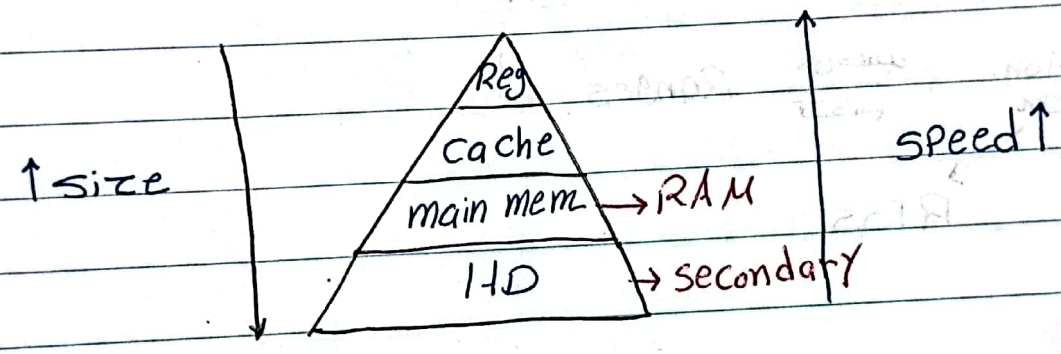
solution → cache coherance →

لو حاجه اتغيرت بتسمع باقي الـ controller

SUBJECT: TITLE: SAT SUN MON TUE WED THU FRI DATE: / /







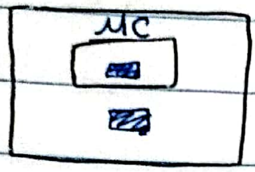
Cache → SRAM

$$\text{Hit cache} = \frac{\#\text{ hits}}{\#\text{ Total}}$$

SUBJECT: ______ TITLE: ______ SAT SUN MON TUE WED THU FRI DATE: __/__/__

[redacted]

FPU / Floating Point unit



ارقام float كاينو * / + -

ممكن يكون جوا ال MC او برا

بحسب ال Performance

MPU → Micro Processor Unit → Memory protection unit

بين ال cache و ال RAM

CPU ↔ cache ↔ MPU ↔ main Ram → hardware circuit

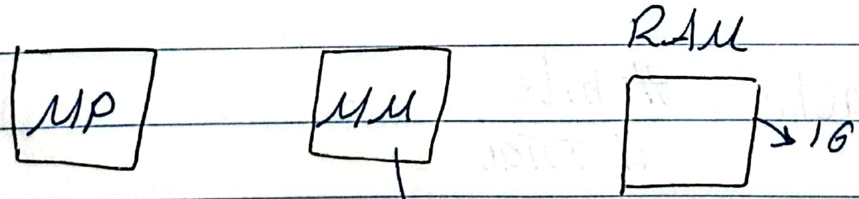
main Ram → Ranges ← تتقسم على حسب ← Operation System

Operation System → linux

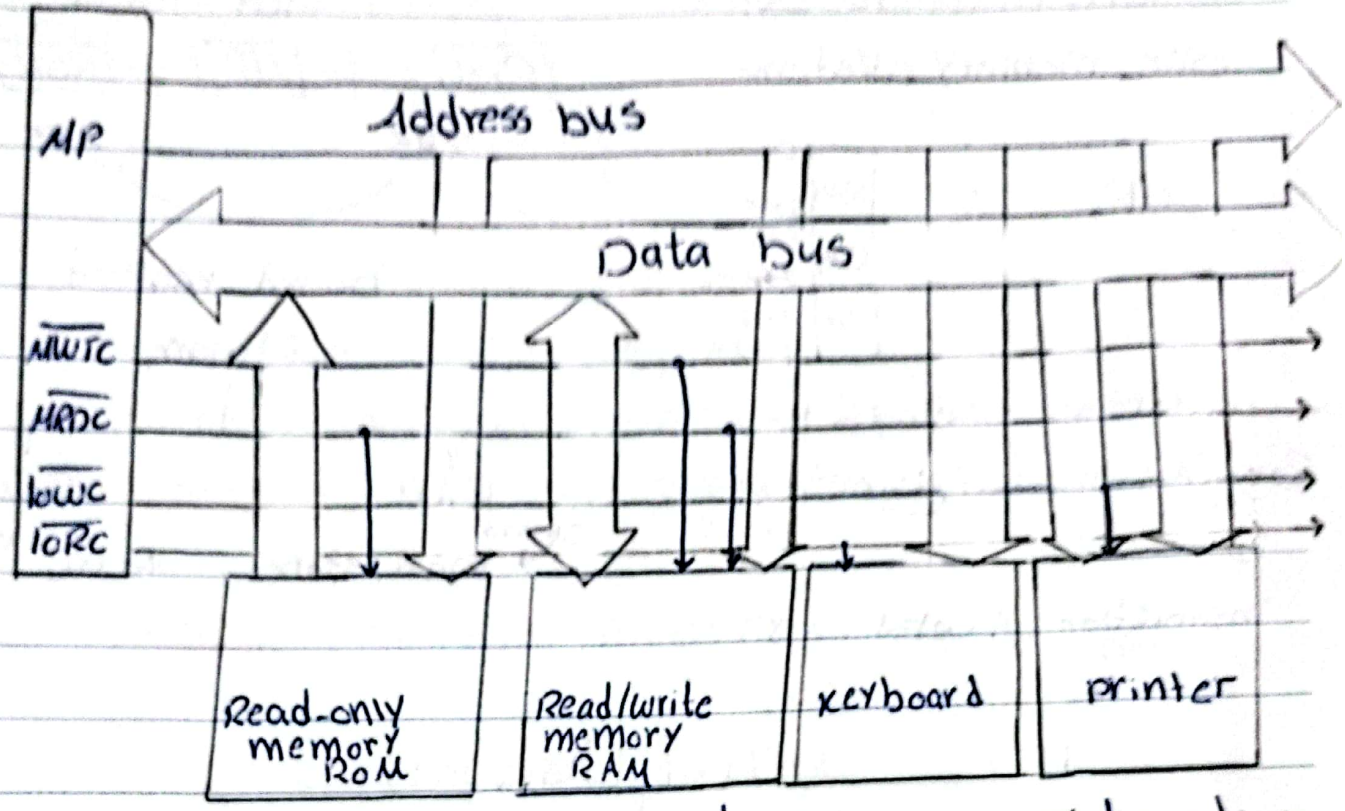
Operation System → RTos

MMU

Memory management unit



لو كايزه انزل APP ← 8G وانا عندي 16 ← هقتنع انه في 8 بس هو فيه 1



Data bus ⟶ Bidirection

address bus ⟶ unidirect

SUBJECT: ______ TITLE: ______ SAT SUN MON TUE WED THU FRI DATE: __/__/__

Arc

Von-neumen → PC

one memory system

MP ≠ | RAM | 0 – 255 | ROM | 256 – 511 | I/o | 512 – 1023 |

address كل حاجه ليها

مش هينفع اكلم الـ ROM, RAM, I/o سوا

لازم اخلص الـ instruction الأول

Harvard → MC

RAM ≠ MP ≠ ROM

255 — 255

board maped

software

- RAM → load/store (ASS)
- ROM → R/W (ASS)

In/out (harvard)

- RAM ≠ MP ≠ ROM, I/O → load/store — memory mapped
- RAM ≠ mP ≠ Rom (L/S, R/W), I/O ≠ in/out — address Bus — Port mapped

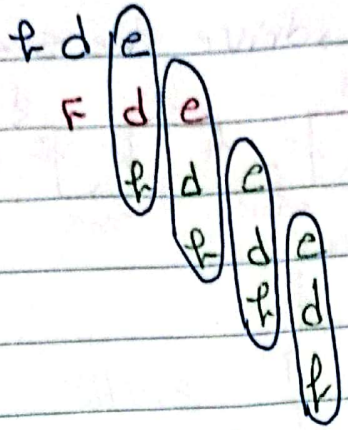
Port mapped:

- Bus set1 (مختلف عن التاني — بيفرق بينهم) → C → L/S — عن طريق الـ instruction
- Bus set2 → I/o → Assembly

مينفعش اتكلم مع ROM

SUBJECT: _____ TITLE: _____

Pipeline



بدل ما الستنى بكمل كذا واحده

Von-neumen → Can't support Pipeline

Harverd → support pipeline

RISC → support pipeline

CISC → Can't support Pipeline

MIPS → Milion instruction per second

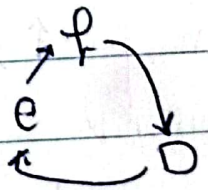
ex 16 MIPS

MC → n bit ↘ 8bit

↓8bit ↓8bit

Inst → Assembly instruction

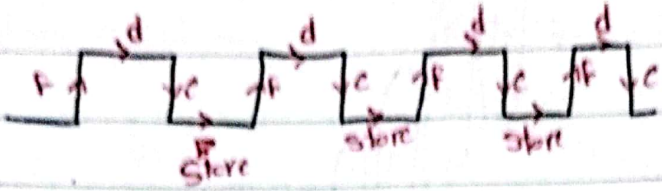
depend clock ←



SUBJECT: TITLE: SAT SUN MON TUE WED THU FRI DATE: /_/_

clock

square wave → drive the system



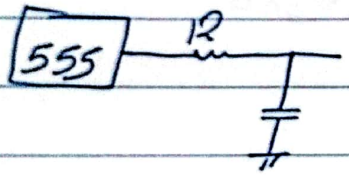
$f = \frac{1}{T} = 8\,MHz$

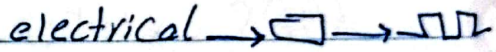
clock ↓ cycle ↓

RISC → 1 inst → 1 cycle → 8 MHz → 8 MIPS
cycles

clock system

- electrical
  - RC oscillator



- mechanical
  - material
    - electrical → 
    - 
    - ceramic ↓ ↓ crystal

	RC	ceramic	crystal
cost	↓ ✓	~	↑
Acc	↓ x	~	↑ ✓
Setting time	↑ x		↓ ✓
Noise Immunity			
↳ T	↓ x		↑ ✓
↳ EMI	↓ x		↑ ✓
↳ vibration	↑ ✓		↑↓ ✓

GOLDEN LINE

SUBJECT: TITLE: SAT SUN MON TUE WED THU FRI DATE: / /

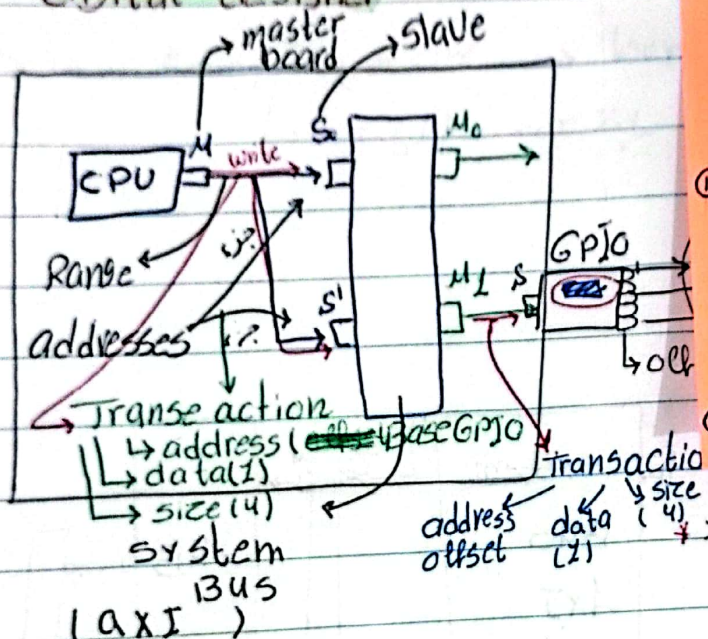
In/out (Harvard)

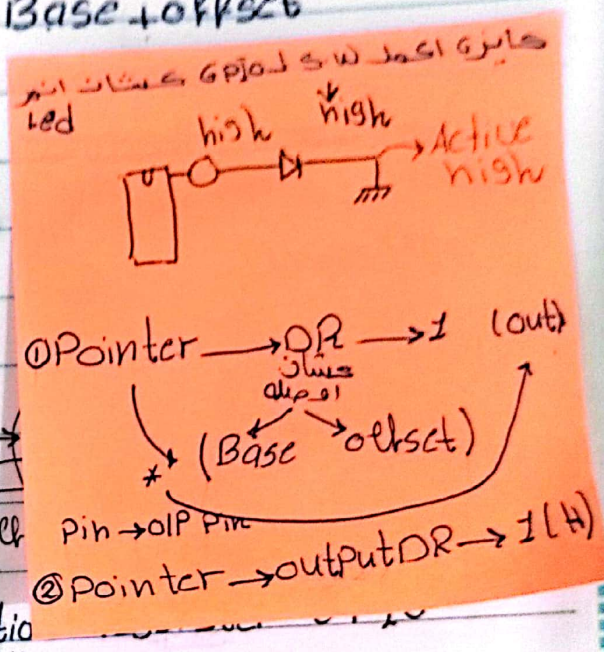
Memory Mapped — Port mapped

ex

device (GPIO)
↳ Range addresses
↳ Base + offset

digital designer





لو اناكايزه اكتب كمال

Periphere → Hardware + Register

لو اناكايزه اكتب

embedded system

Tech Ref manual

TRM

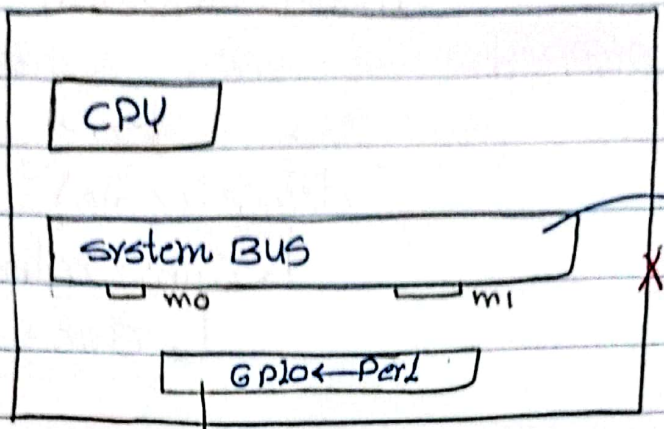
specs → TRM
↳ memory map
↳ GPIO

GPIO → Base 0x0
↳ END → Size + B

DR 0x0 → +Base
ODR 0x4 → +Base
IR 0x8 → +Base

GOLDEN LINE

SUBJECT: _____ TITLE: _____ SAT SUN MON TUE WED THU FRI DATE: __/__/__



Reg1 → 0x0
Reg2 → 0x4
Reg3 → 0x8

Reserved ← لحط الحجم اكبر من ١٢
↳ لسببها المستقبل

Bus Bridge