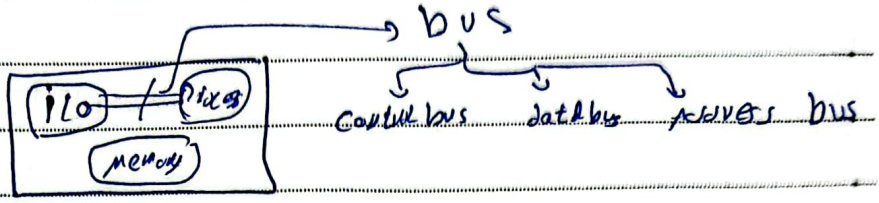
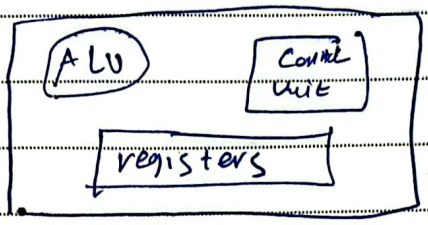
an : f

what is embedded system



what is processor

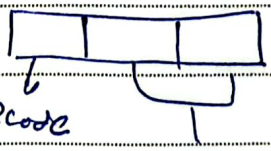


what it does ? fetch → decode → exute

instruction Life Cycle

fetch → PC refers to the next instruction (address)
then the CU take that address from memory to IR register → ID to decode it

decode ← Inst - set
Inst - format



after decoding comes the exection

each MP has it's machine Language

ISA → instuction set Archtivure

- Risc — Low instuction
- Cisc → High level more instructions

ID differes based on ISA

two ways for implemeing Risc and Cisc

- Hard wired logic gate → Risc #
- Memory Mapped — memory → IS — use searching for knowing the instruction → Cisc #

	Risc	Cisc
Size	ALU ↓ ~~ID ↑~~ ID ↑	ALU ↑ ID ↓
Cost	SW ↑ HW ↓	SW ↓ HW ↑
Performance	logic gates	search
Power	ALU ↓ ID ↑	ALU ↑ ID ↓

They are the Same

What is Register files ?

GPR ——————————— SPR

data in temporary

SPR

① PC → Points to next ins!

② SP → Stack Pointer → Points to the head of the Stack

③ ACC → Result from ALU

④ IR → Carries the instruction to be fetched

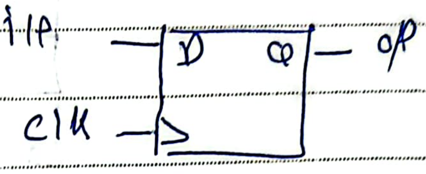
⑤ PSW → Some flags

what is memory ?

Access time (R/W) — Acc time write > Read

Access time ↓ ⟶ Performance ↑

*Basic Mem element ⟶ D flip flop



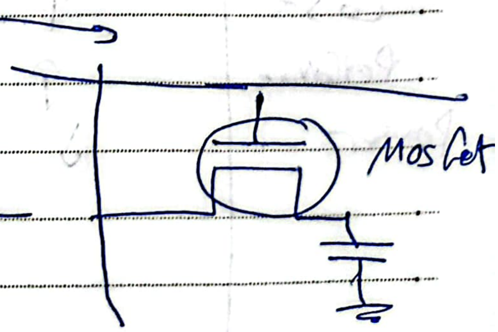
1- Capacity ⟶ Size
2- speed ⟶ Access time
3- organization ⟶ can be merged

memory

- Volatile (ram)
- NON-Volatile (rom)
- Hybrid (mix)

Ram ⟶ Random Access Memory (RWM)
working

Ram
- SRam ⟶ Static
- DRam ⟶ Dynamic



Based on capacitor ⟵

1 ⟵ مشحون
0 ⟵ فاضي

Need Refersh circuit to recharge the Cap

→ it has higher Priority over CPU to access the memory → higher access time So Lower performance and speed.

it's advantages
- Simple
- Low Cost Per bit.
- high denisity ↑ size
- Low Power Consuption

SRam → Static Ram → based on transistor

bit → Flip flop → 6 transistors

SRam → DRam as speed
DRam → SRam as low cost

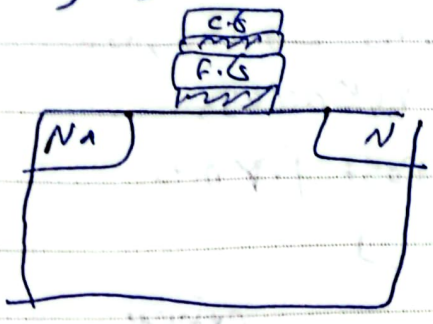
embedded → SRam

	SRam	DRam
Size	↓	↑
Cost	↑	↓
Performance	↑	↓
Power	↓	↑

Non-Volatile → Rom → it has the code

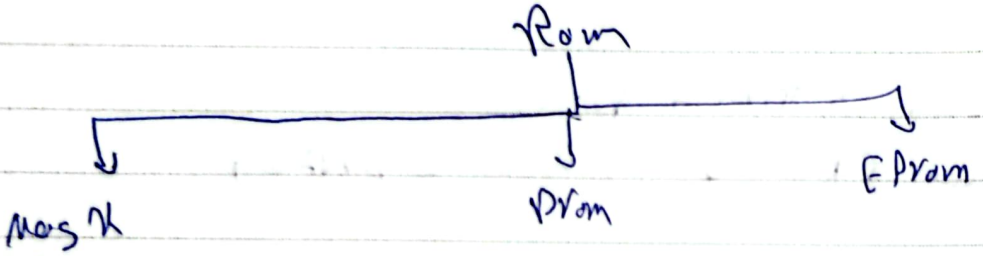
Access time ↑ Ram AT > Rom AT

Rom → based on floating gate mosfet



erasing state (1)
programing state (0)

Read only memory → Processor



Mask Programable → OTP

① OTP

② Prom → Programable Rom → OTP

③ EPRom → Erasable Rom

Advantage → erasable
↳ 10 years Non Volatile

disAd → Noise / Radiation

Hybrid → Can Read and Written

EEPROM — Flash — NV Ram

Electrical

- Endurance → 100,000
- ~~Access~~ byte access
- high cost per bit

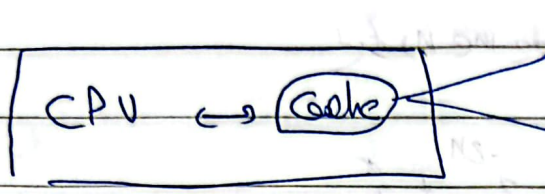
internal — external

electrical → EEPROM
sector by sector
Endurance : 10_000
Low Cost per bit

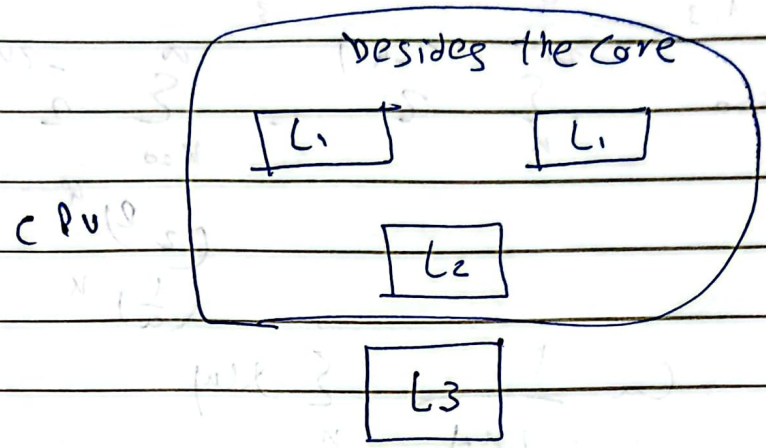
- NV Ram → SRam + battery
  - SRam + EEPROM + battery

High Cost per bit

Date: Page:



Cashe has levels

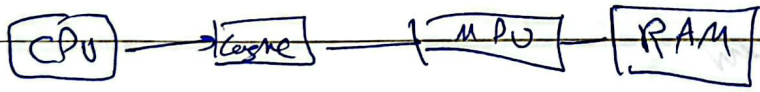


* → cashe Coherence → to make sure that any change in one of the caches reaches the other.

* hit ratio = $\frac{\#hits}{\#total}$

* FPU → floating point unit → used for operations on floating numbers.

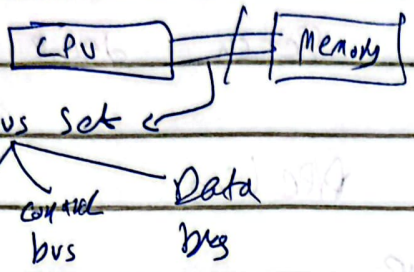
* MPU → memory protection unit → between cashe and ram

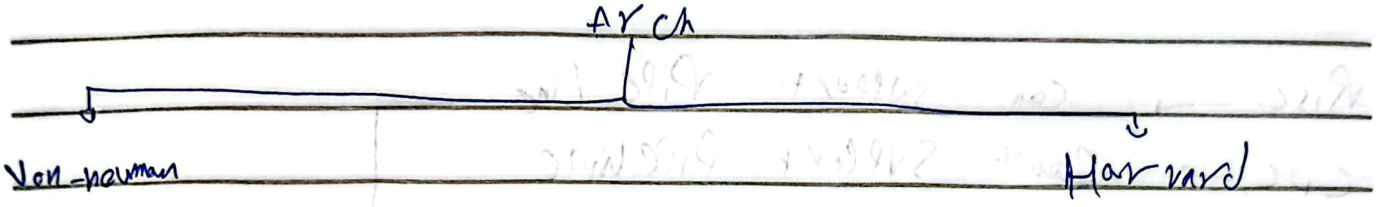


* MMU → (memory managment unit.

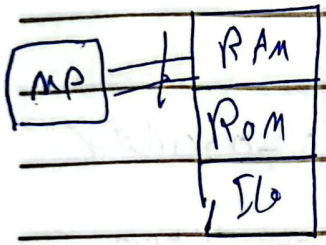
Date: Page:

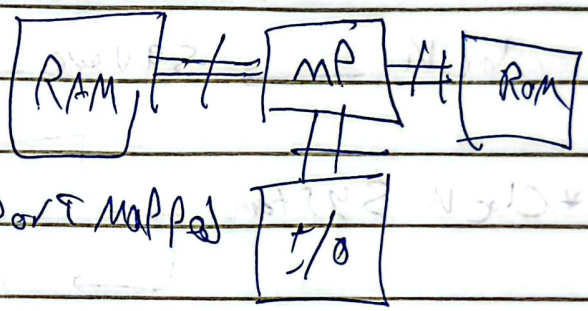
~~I/O techniques~~ :-





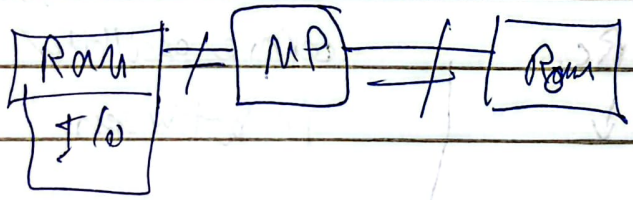
one memory system





I/O → Port Mapped

I/O (Harvard)



I/O → memory mapped

- To talk to ram we can use C programming to convert into Load/store assembly instructions
- To access ROM we use ASS directly

Date: Page:

→ Pipelining — Fetch decode exu
Fetch decode exu
Fetch decode exu

→ Von neuman → Can't support Pipe Line
→ Harvad → support Pipe Line

Risc → Can support Pipe Line
Cisc → Can't Support Pipeline
Finish in 1 clock

clock → square wave

*Clock System → electrical | Rc-oscillator
→ Mechanical | Material → wave
power

	Rc	ceramic oscillator	crystall oscillator
cost	↓	in between	↑
~~accuracy~~ accuricy	↓	~	↑
setting time	↑	~	↓
Noise Immunity			

ROXi

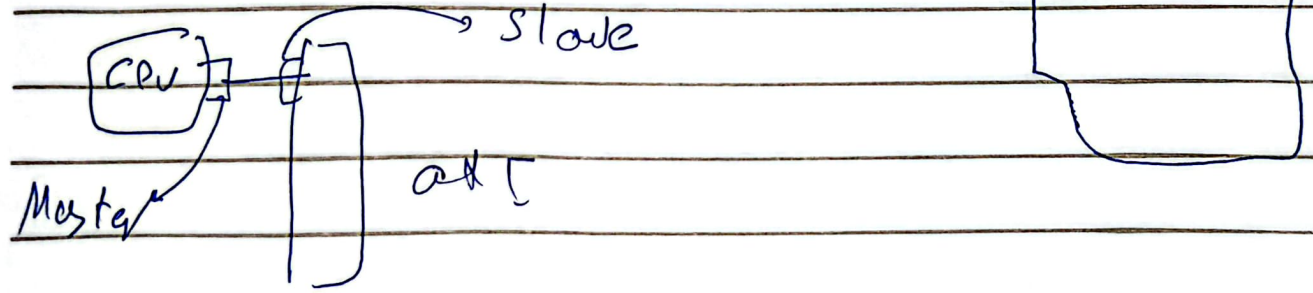
Date: Page:

	RC	ceramic	crystal
temp imm	↓	↑	↑
	↓	↑	↑
Emit	↓	↑	↑
vibrating	↑	↓	↓

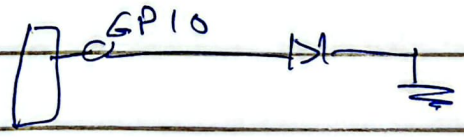
Peripherals → (HW civ + Reg)

* TRM → Tech Ref Manual
→ specs

* specs → TRM
→ memory Map
→ GPIO



Ex SW → Low High

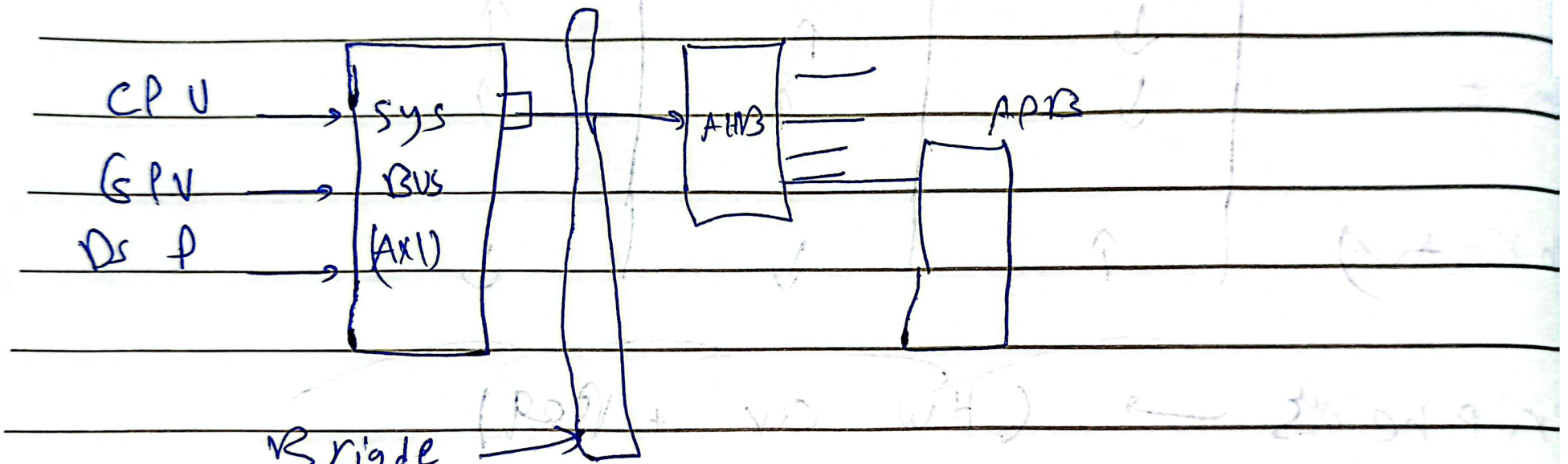


① Pointer → DR → 1 (GPIO)
↓
(Base + offset)

② Pointer → ODR → 1 (High)

Date: Page:

Bus Bridges



AXI → AHB