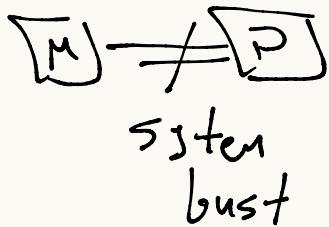


System bus



↳ It's way to communicate system parts together

↳ +IPC → wire → white board
track → PCB
funnel → chip

↳ System bus Content →

- ↳ ① Data bus
- ↳ ② Address bus
- ↳ ③ Control bus

↳ System bus → $\neq \Rightarrow$

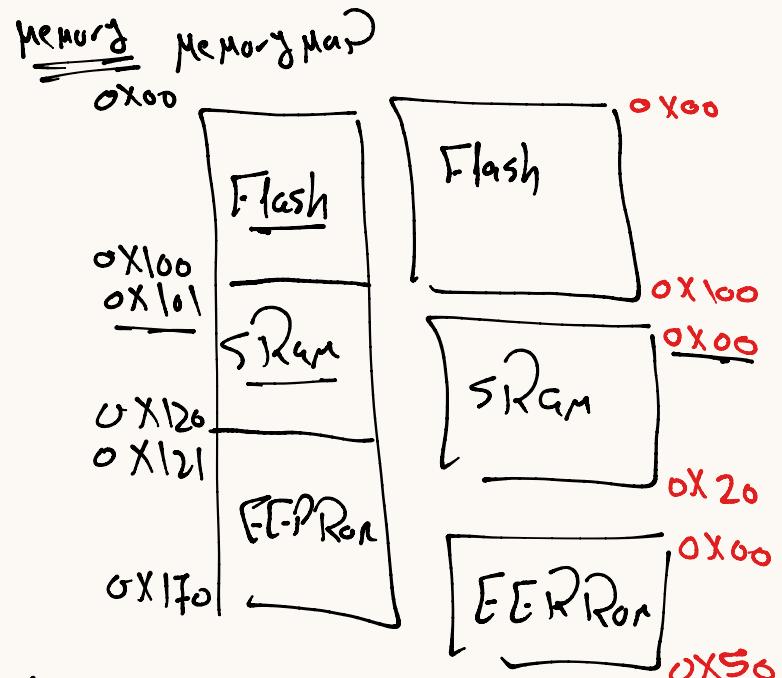
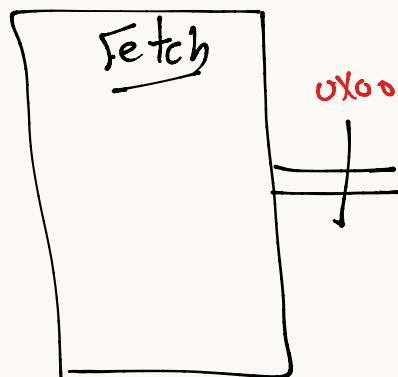
If you have System bus = 32 bit

↳ Data bus = 32 bit = 32 wire
↳ var \Rightarrow 4 byte

↳ Address bus = 32 bit \Rightarrow 32 wire
↳ Address → 32 bit

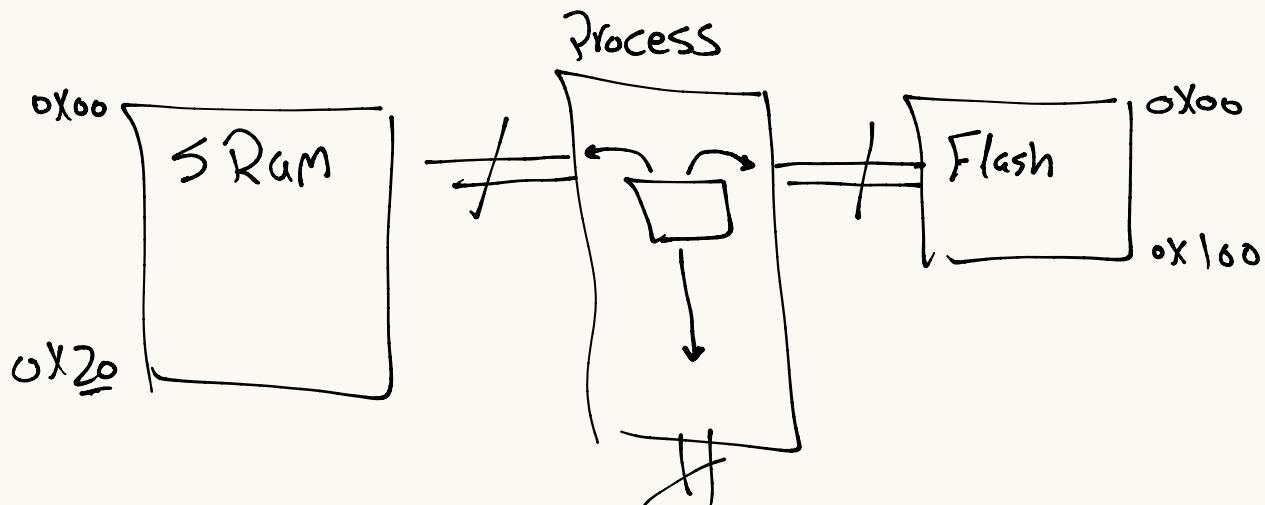
System t-1 Pe Arch

① Von Neumann



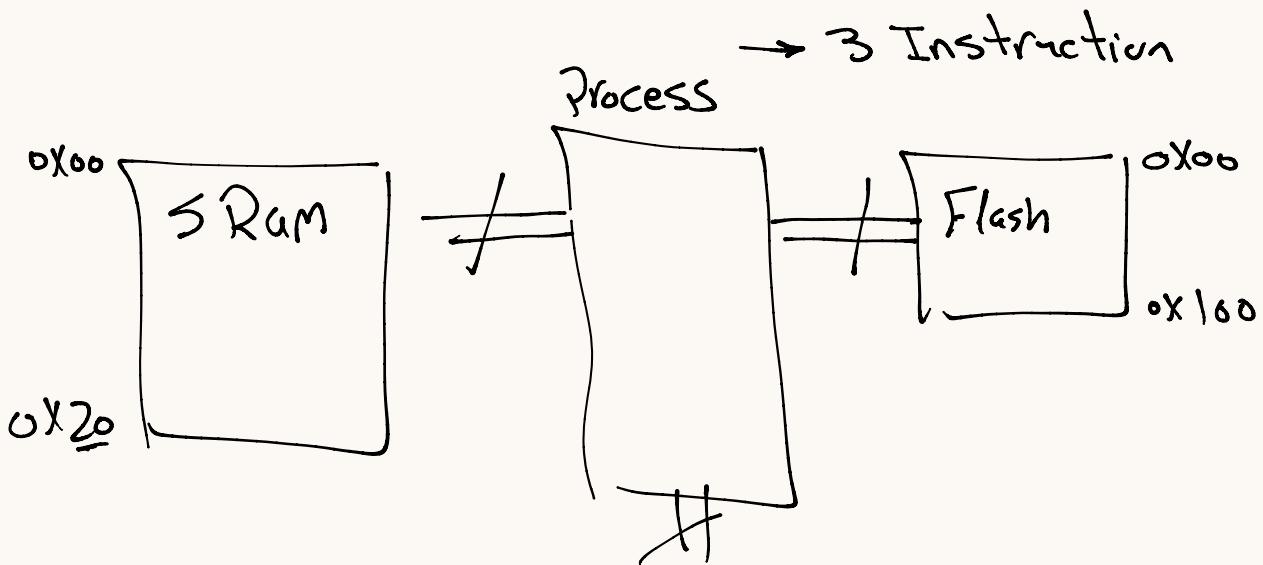
↳ low Cost | Simple
↳ take long time

② Harvard Arch : For each memory → System Bus



↳ ① High Cost
↳ ② High Complex

$$\underline{\text{ex:-}} \quad *(\underbrace{(\text{char} *)}_{0x20}) = 20;$$



→ For Flash

↳ Assembly Instruction ⇒ store / load

→ For SRam

↳ Assembly Instruction ⇒ Read / write

→ for EEPROM

↳ Assembly Instruction ⇒ In / out

if we write Code C → Compiler Convert For
which Instruction

Compiler by default will → Convert to SRam
Instruction

if you need to write EEPROM → must be
write Assembly

extern .asm &



→ 3 Instruction

Von Neumann

- Code - C
- take long time

Memory Map

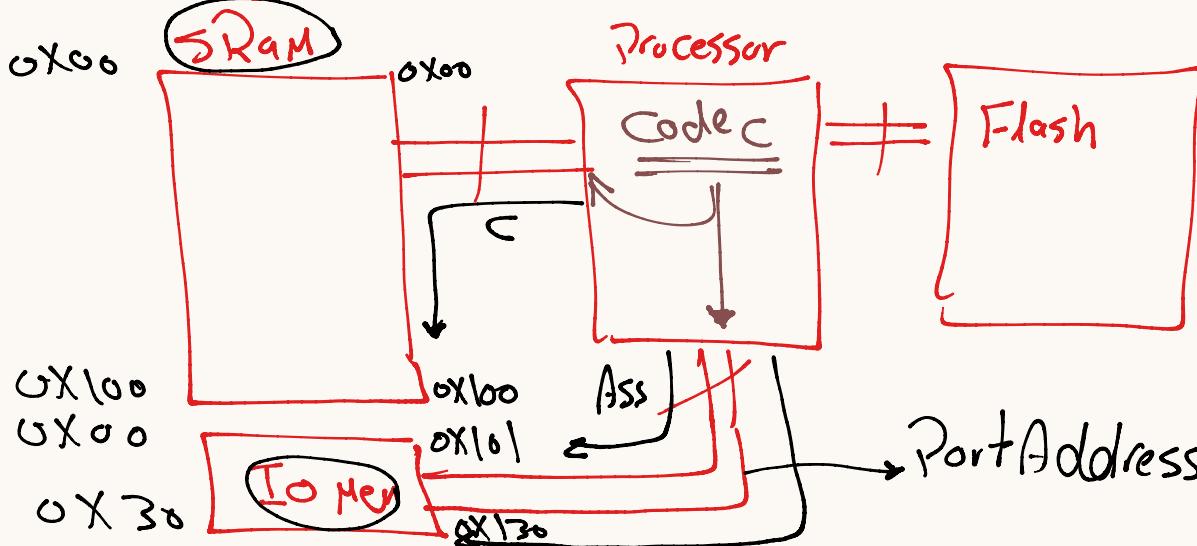
Harvard

- Code C For SRAM
- otherwise → write Assembly
- Fast time

Port Map

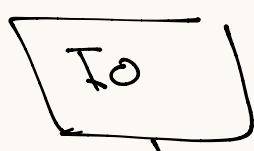
↳ For each system bus have instruction.

* Hybrid → AVR



write C-Code to Io-Memory

Io Memory
Port Map
Assembly



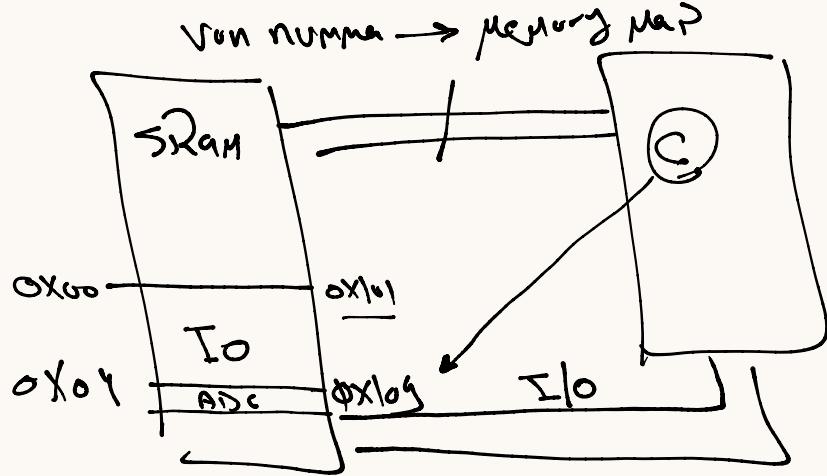
Memory Map

Port Map

* Assembly
* Correct Address
(0x10)

Code C

```
[*(char*)0x100] = 30;
```



Von nummern
✓
* Memory Map
* C-Clock

Harvard
✓
* Port Map → 0x09
* Assembly