Sunbeam Institute of Information Technology
Pune and Karad

Module - Micro controller Programming and Interfacing

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Micro processor vs Micro controller

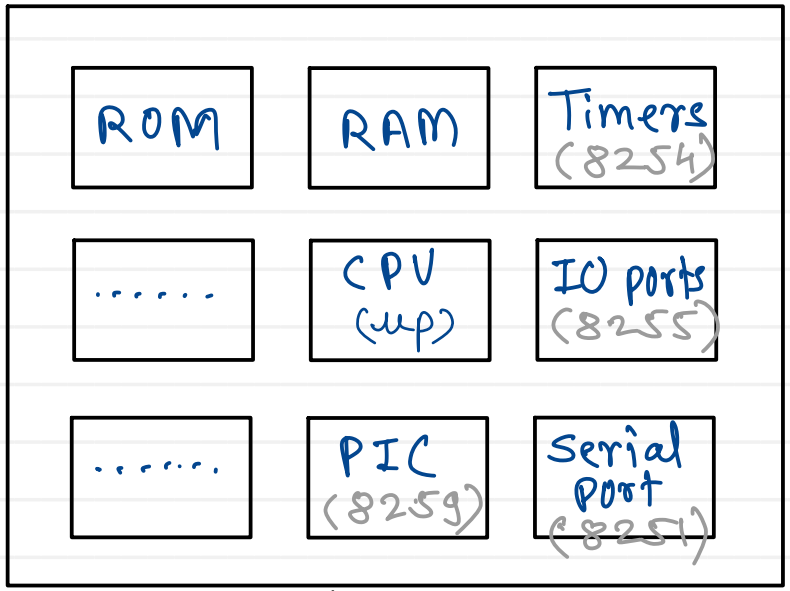
CPU + RAM + ROM + ...

↓

up = ALU + Registers + EU + BIU

+

Cache + MMU

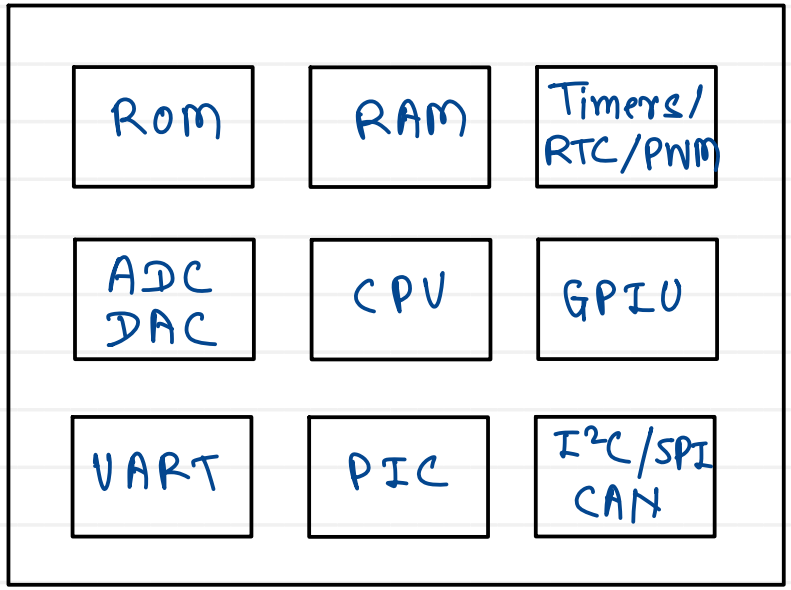


Motherboard

uc = CPU + RAM + ROM + Peripherals

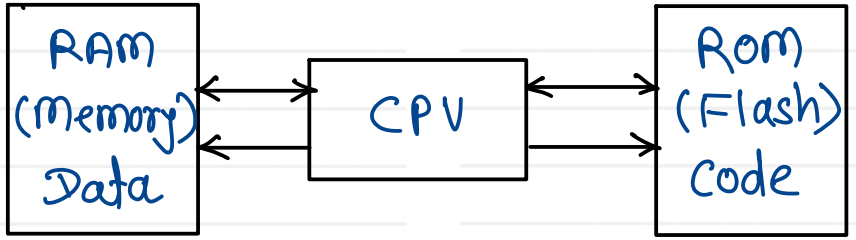
↓

ALU + Registers

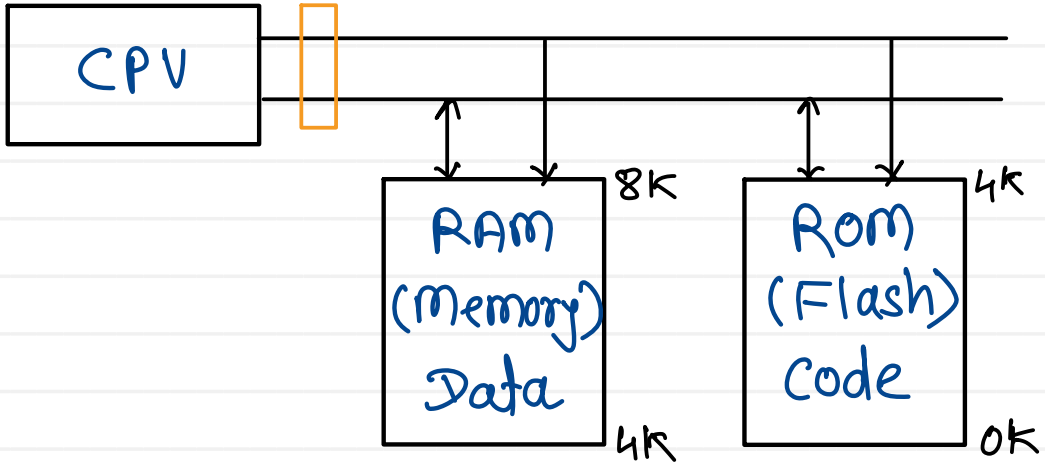


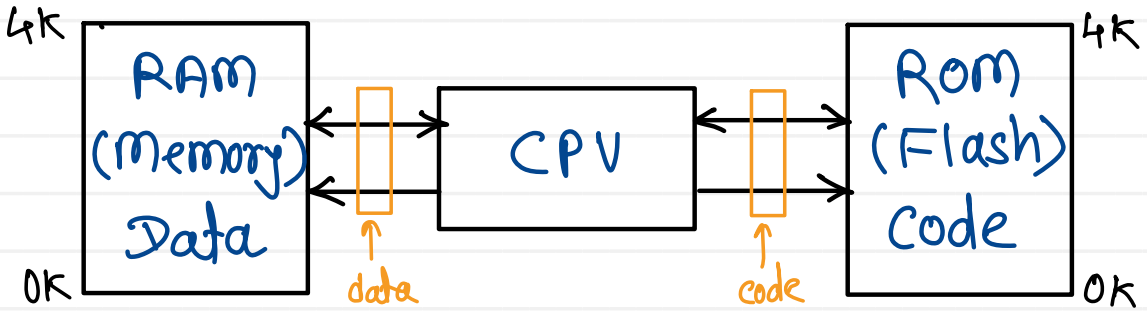
Single Chip (SoC)

Von Neumann vs Harvard



(code+data) Unified cache

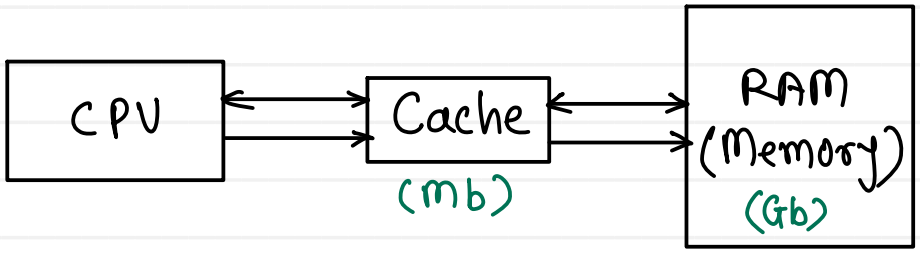




Super harvard : read only data is kept in ROM/ flash along with code.

Modified harvard : every thing is same as harvard only address space is single like von neumann.

Cache memory



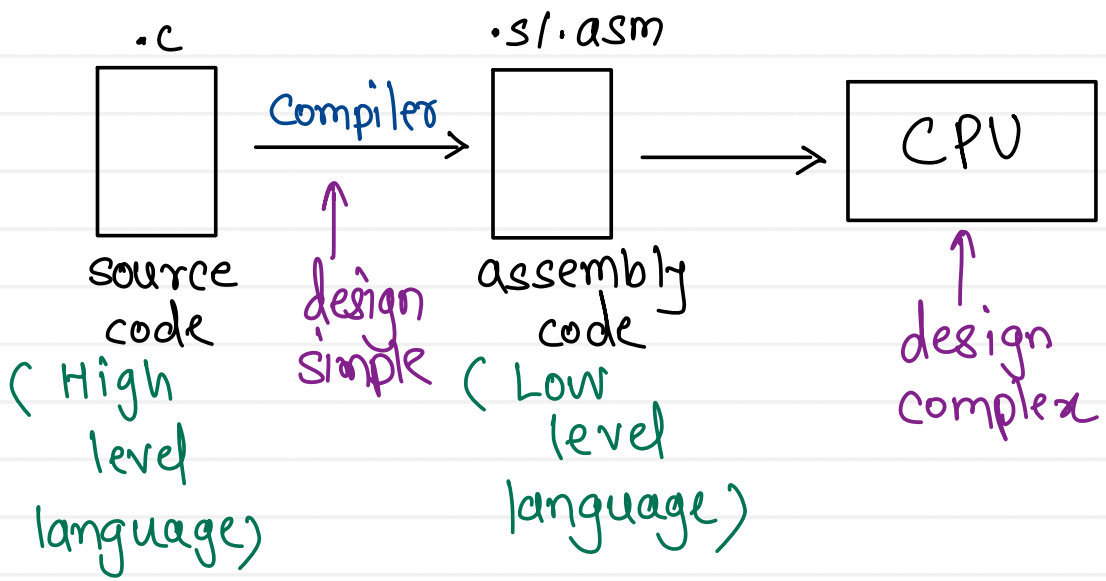
- to avoid speed mismatch of memory & CPU, cache memories are used in between them.
- Cache memories are always faster than RAM.
- Associative data access
  key - address
  value - data

- older data is over written by new data when cache memory will get full.
- it will hold recently accessed data.

Cache hit : (faster)
data is found in cache memory

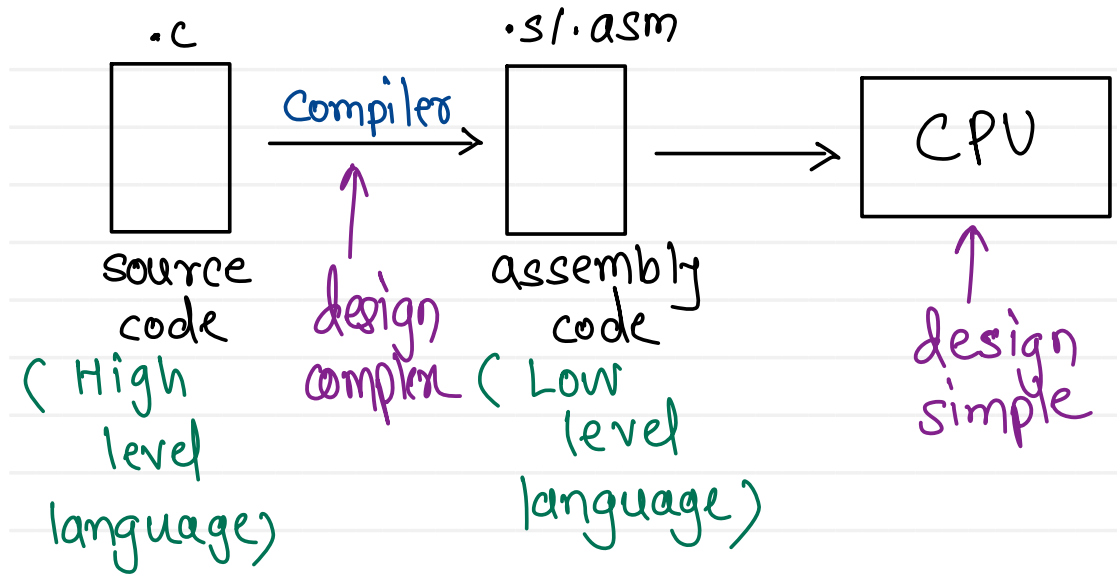
Cache miss : (slower)
data is not found in cache memory

CISC vs RISC



if else
loops
switch
functions

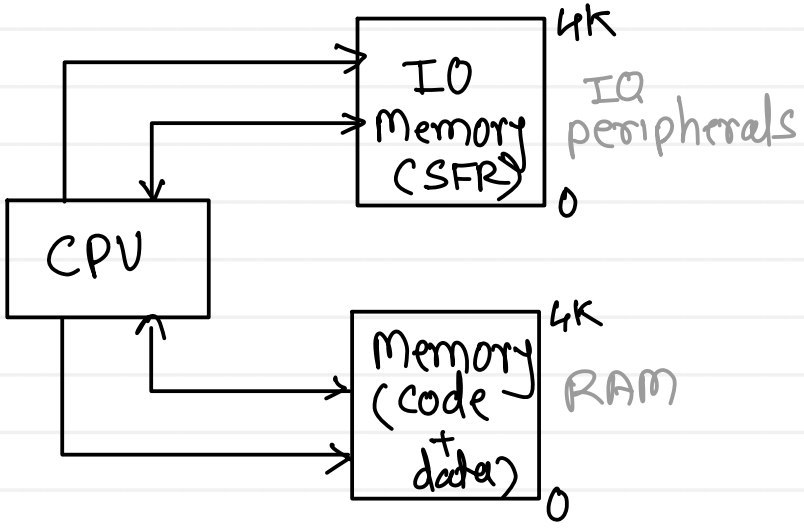
MOV, CMP
B, BL ....
JUMP
LD/ST
CALL, MUL
(macro instruction) need multiple CPU cycles
(micro instruction) need single CPU cycle



if else
loops
switch
functions

MOV, CMP
B, BL ....
JUMP
LD/ST
(micro instructions)
need single CPU cycle

IO mapped IO vs Memory mapped IO



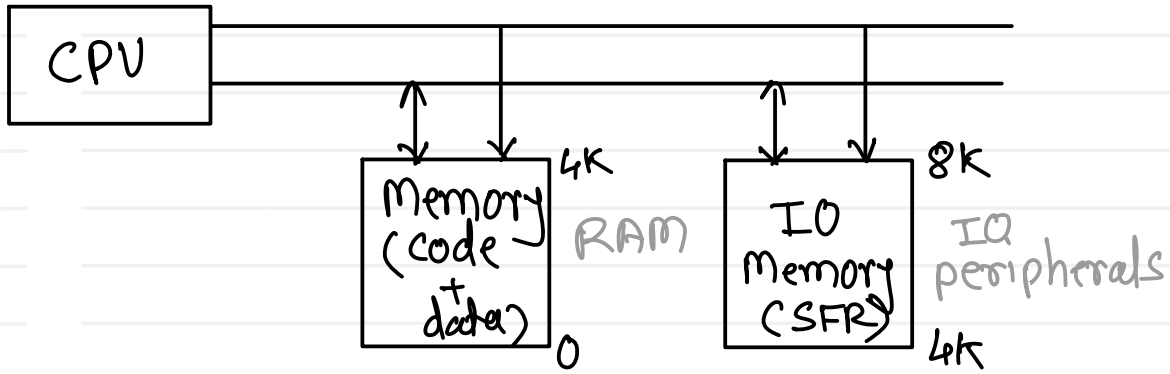
IO/$\overline{M}$

0

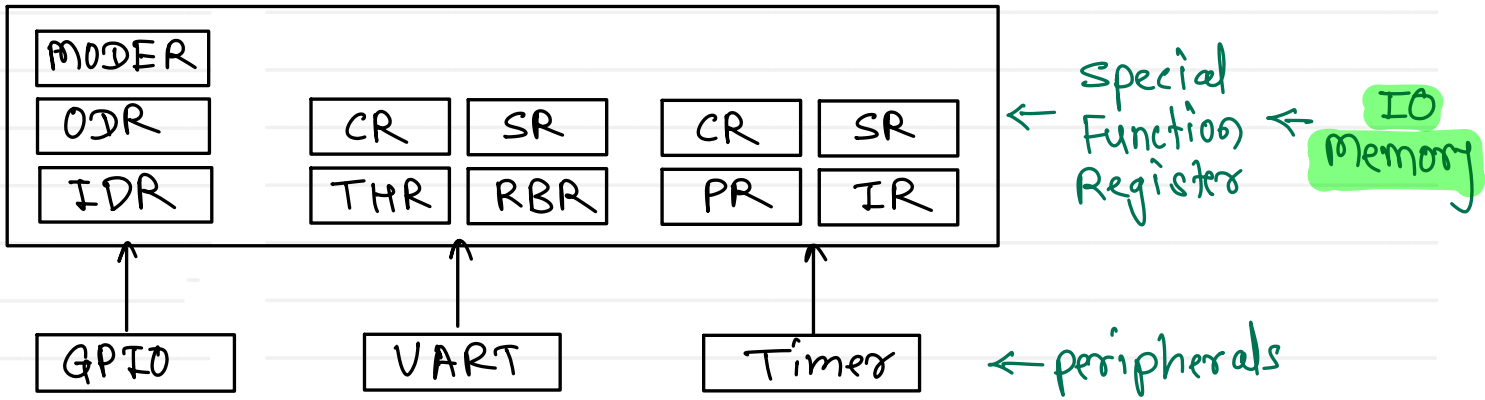
1

Memory : MOV, LD/ST

IO : IN/OUT



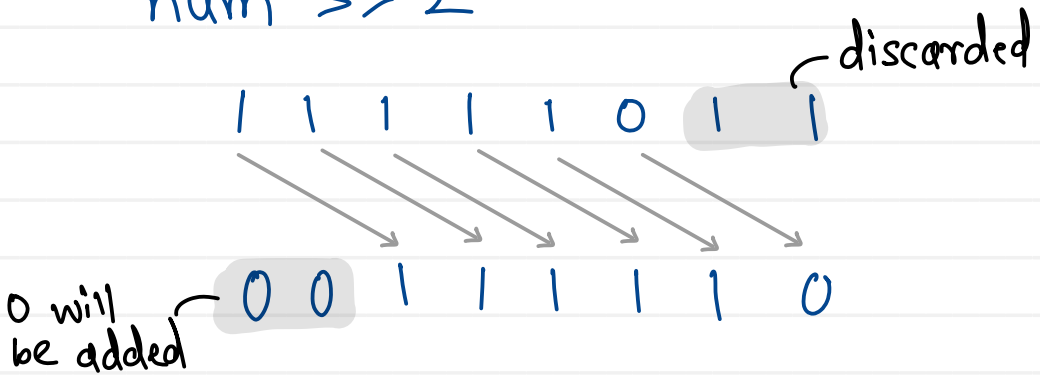
Memory / IO = MOV, LD/ST



unsigned char num = -5;

-5 : 1111 1011

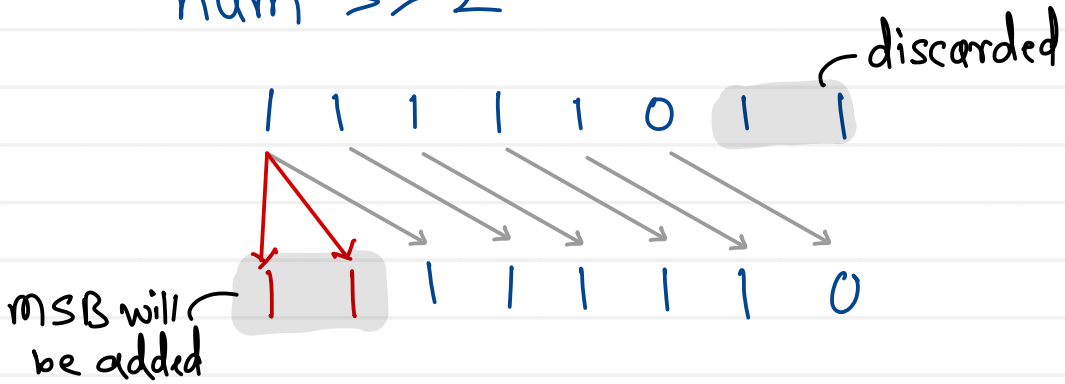
num >> 2



signed char num = -5;

-5 : 1111 1011

num >> 2



num = -5

```
  1111 1011
& 1000 0000
-----------
  1000 0000
```
(nonzero) → 1

```
  1111 1011
& 0100 0000
-----------
  0100 0000
```
(nonzero) → 1

```
  1111 1011
& 0010 0000
-----------
  0010 0000
```
(nonzero) → 1

```
  1111 1011
& 0001 0000
-----------
  0001 0000
```
(nonzero) → 1

```
  1111 1011
& 0000 1000
-----------
  0000 1000
```
(nonzero) → 1

```
  1111 1011
& 0000 0100
-----------
  0000 0000
```
(zero) → 0

```
  1111 1011
& 0000 0010
-----------
  0000 0010
```
(nonzero) → 1

```
  1111 1011
& 0000 0001
-----------
  0000 0001
```
(nonzero) → 1

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com