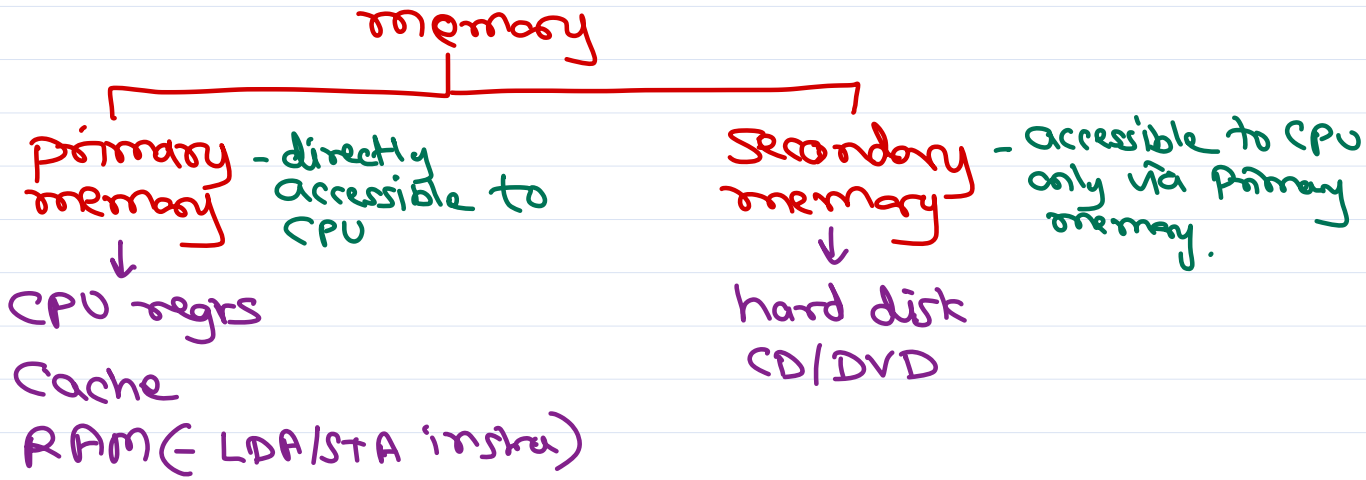
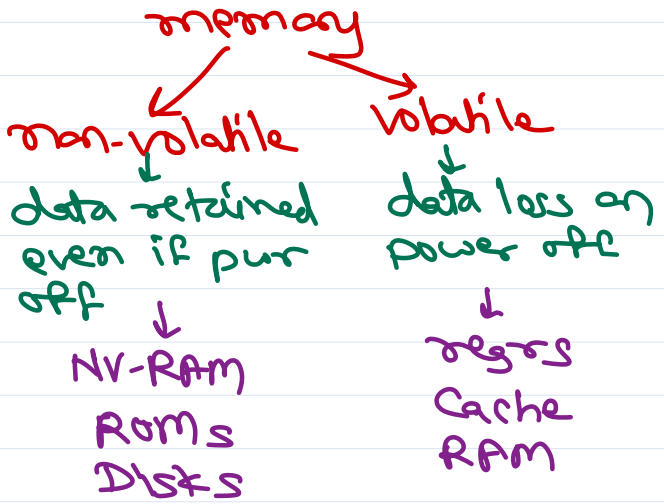


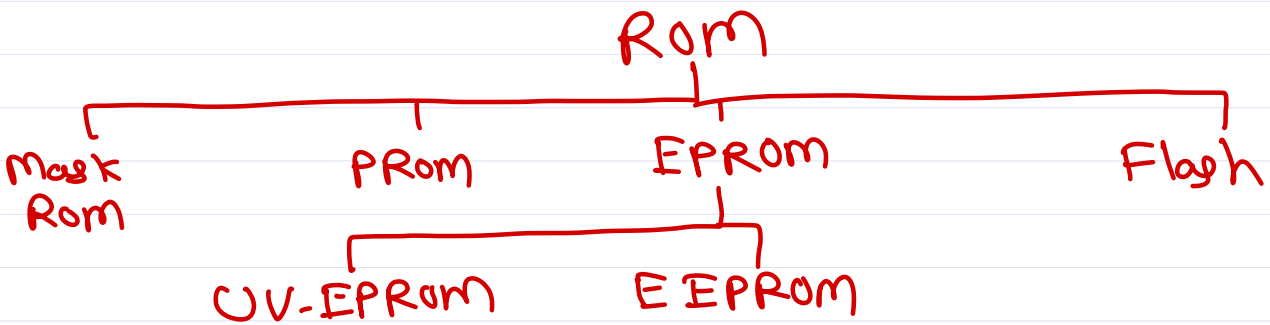


# Advanced Micro-controllers - ARM

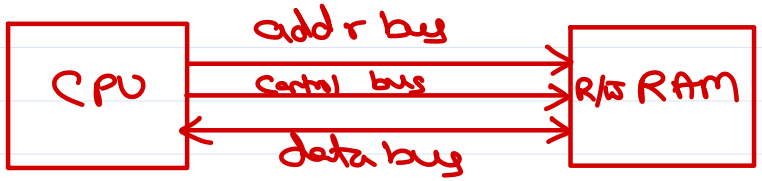
*DESD @ Sunbeam Infotech*

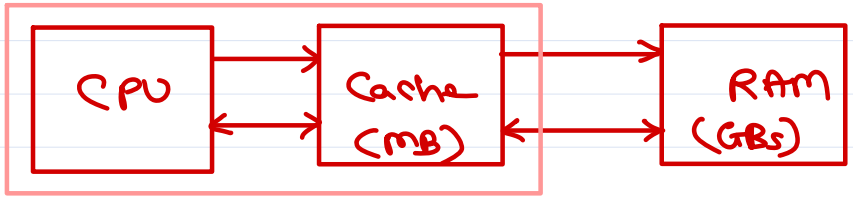

memory
primary memory
- directly accessible to CPU
CPU regrs
Cache
RAM (- LDA/STA instrs)
Secondary memory
- accessible to CPU only via Primary memory.
hard disk
CD/DVD

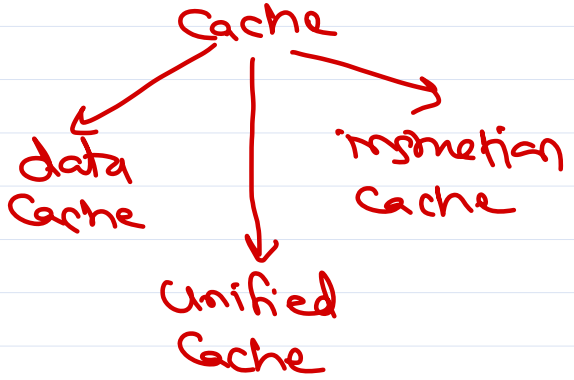

memory
non-volatile
data retained even if pwr off
NV-RAM
ROMs
Disks
volatile
data loss on power off
regrs
Cache
RAM

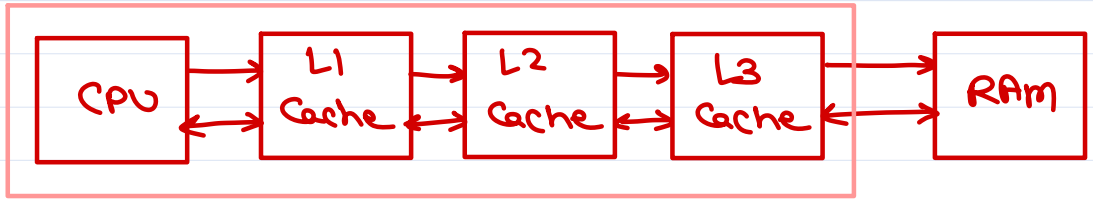

ROM
Mask Rom
PRom
EPROM
Flash
UV-EPROM
EEPROM


RAM
SRAM (Static)
Flipflop
DRAM (dynamic)
Capacitor holding charge
Refresh
NV-RAM




Cache is high-speed associative memory.

key = address, value = data.

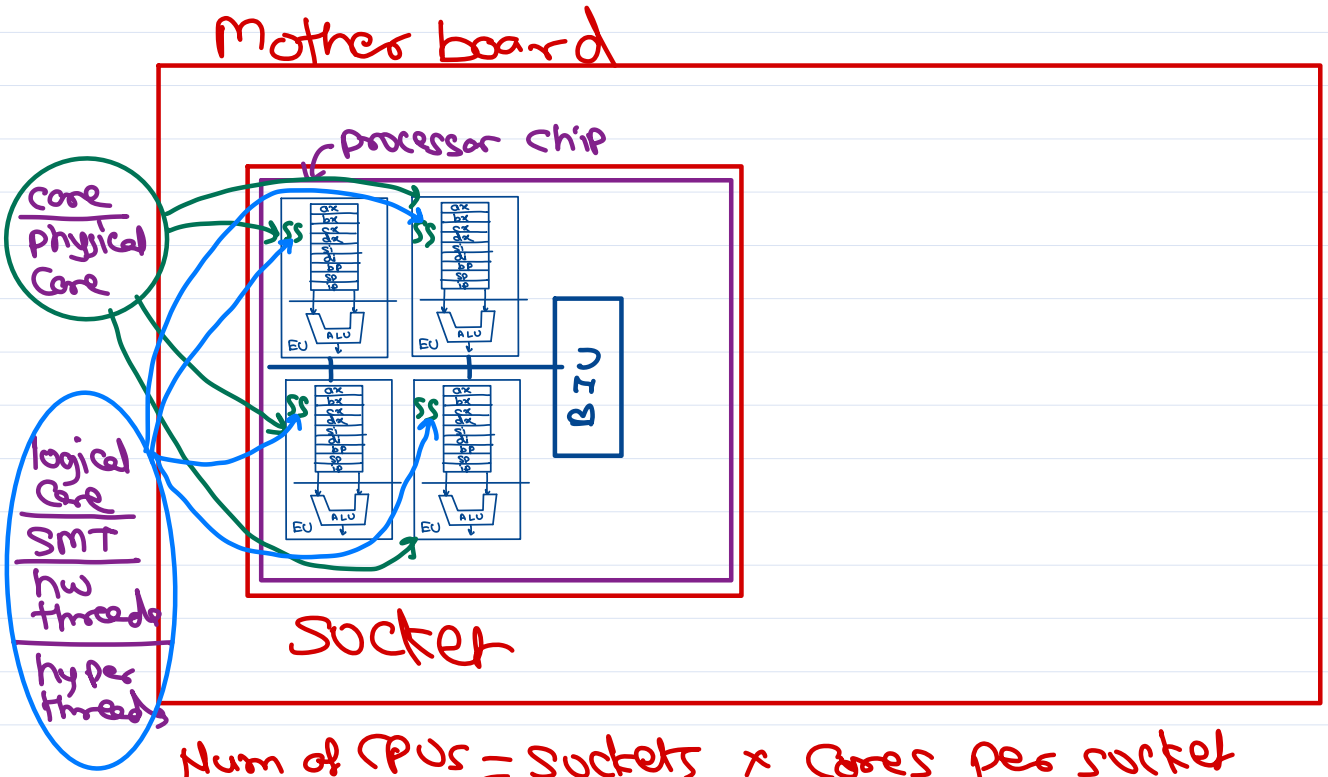
① Cache memory is limited (less than RAM).

② When cache is full oldest data is overwritten. (LRU)

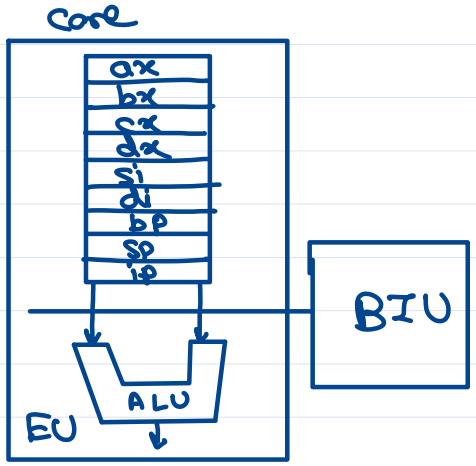
③ Cache always stores recent data

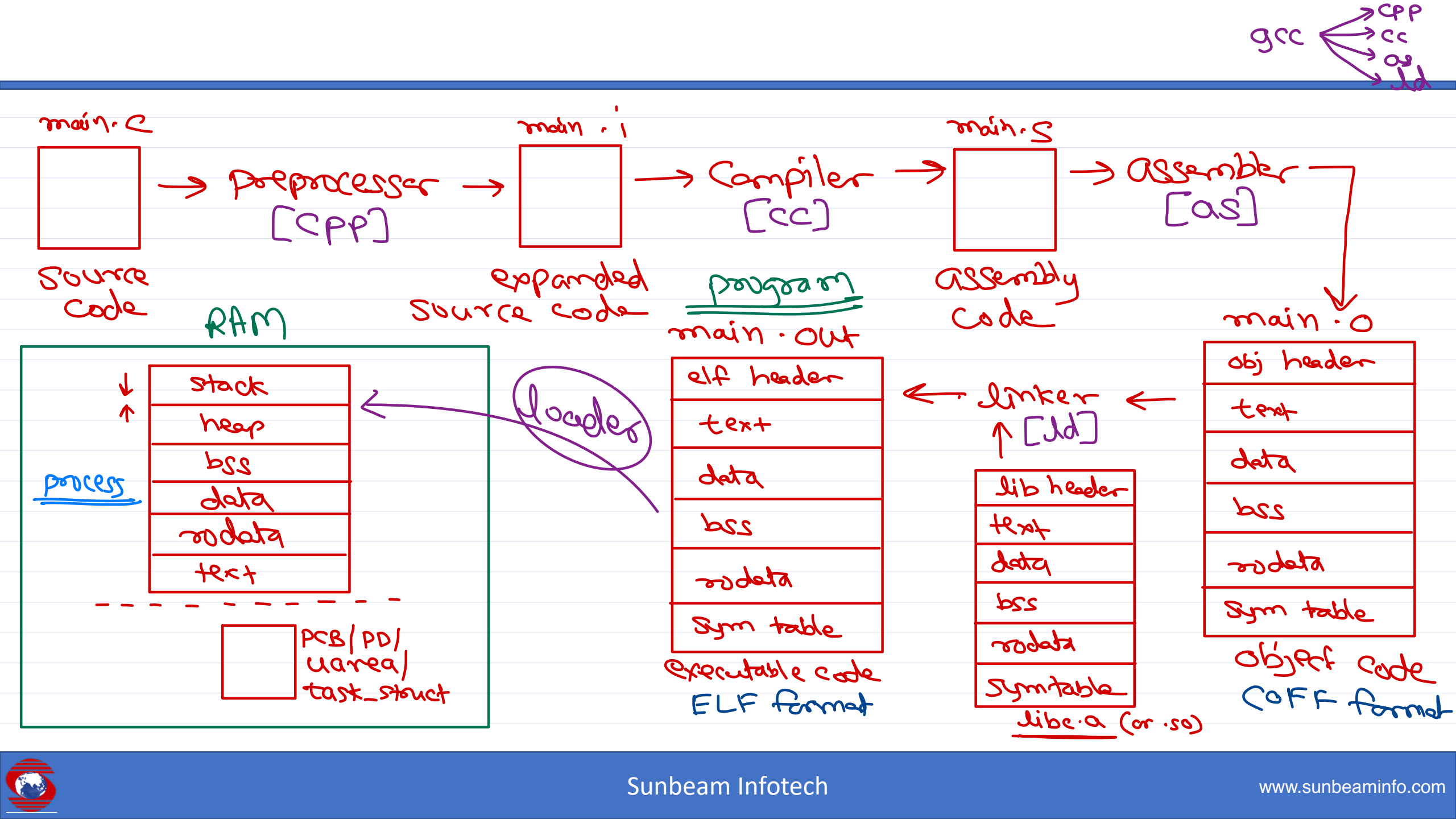
Cache hit : CPU ⇄ Cache

Cache miss : CPU ⇄ Cache ⇄ RAM




Num of CPUs = Sockets * Cores per socket
* threads per Core.




gcc
cpp
cc
as
ld
main.c
Source Code
preprocessor [cpp]
main.i
expanded Source code
Compiler [cc]
main.s
Assembly Code
Assembler [as]
main.o
obj header
text
data
bss
rodata
Sym table
object code COFF format
linker [ld]
lib header
text
data
bss
rodata
Symtable
libc.a (or .so)
program
main.out
elf header
text
data
bss
rodata
Sym table
executable code ELF format
loader
RAM
process
stack
heap
bss
data
rodata
text
PCB/PD/ uarea/ task_struct


*Thank you!*

Nilesh Ghule <nilesh@sunbeaminfo.com>