Digital Systems L2 – Computer Systems

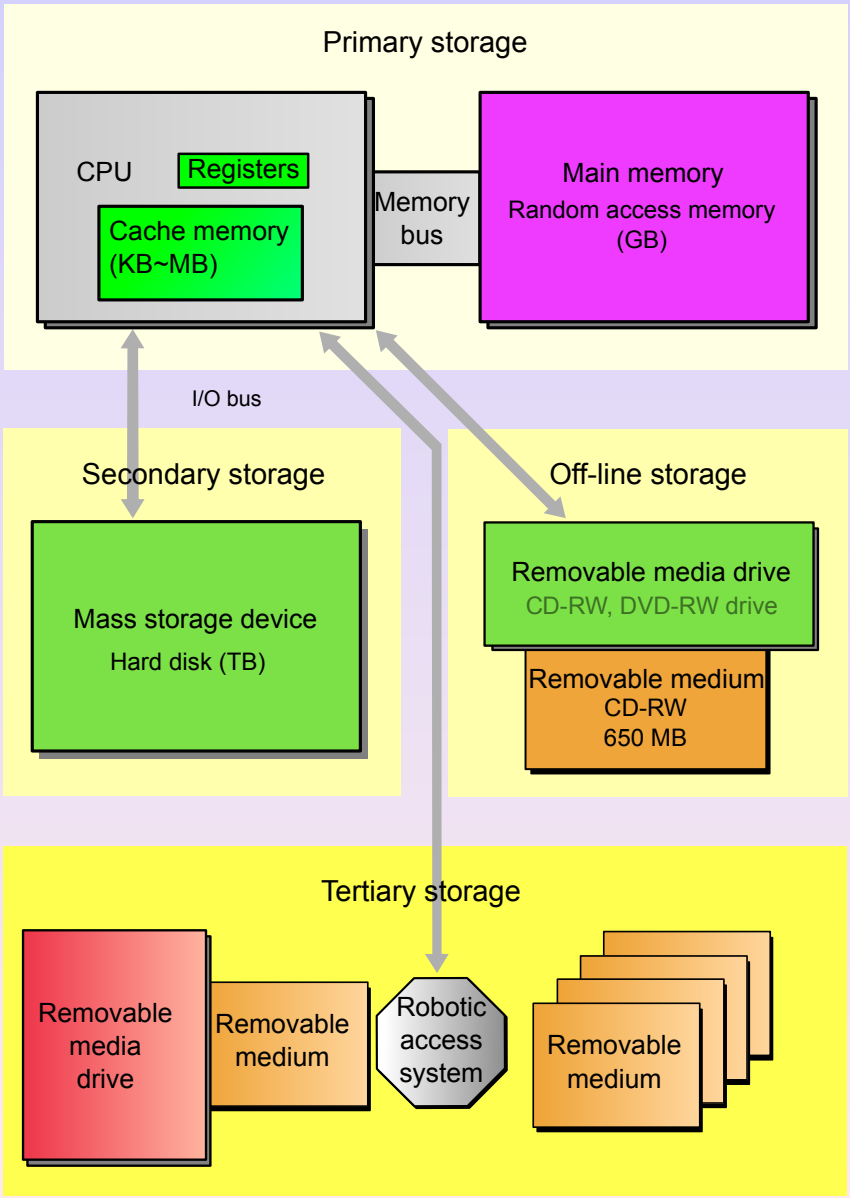
Fang Yuan

School of Electronics Science and Engineering
Nanjing University
Nanjing 210046

2022

Outline

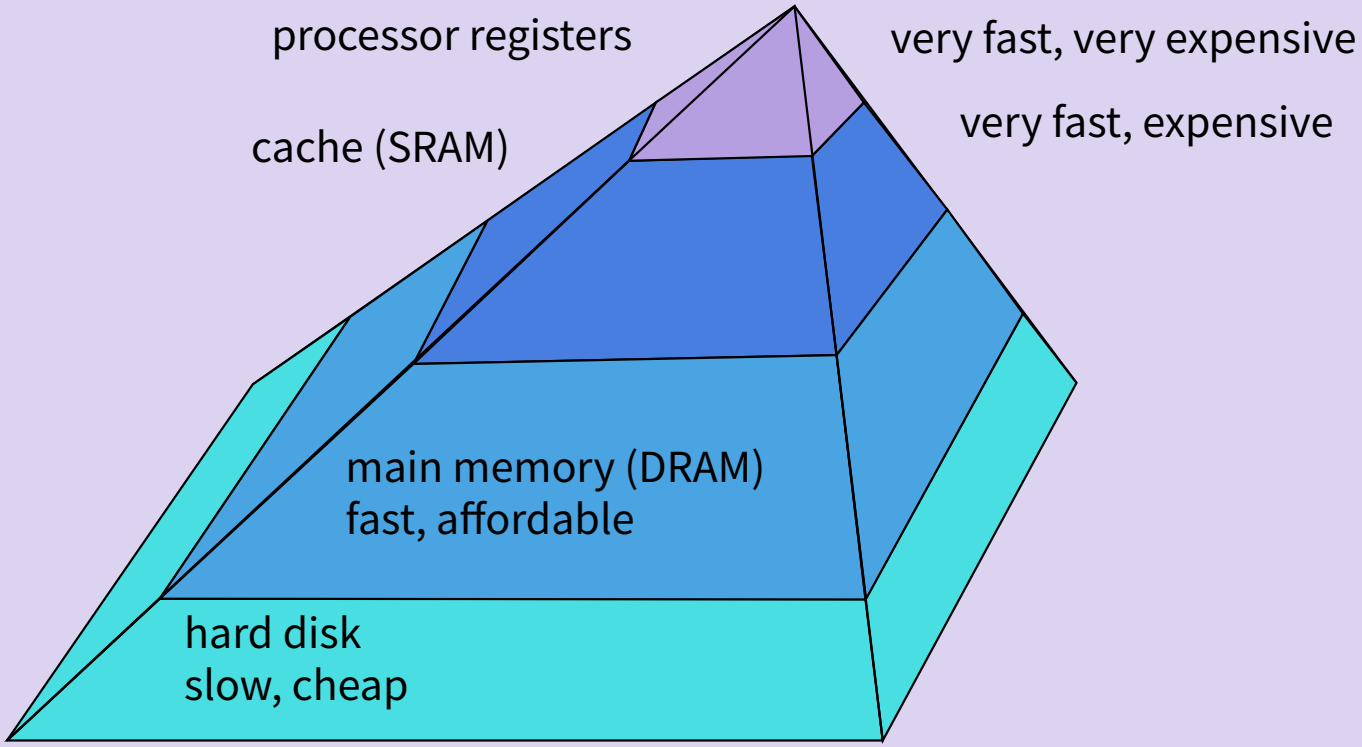
Types of Memory



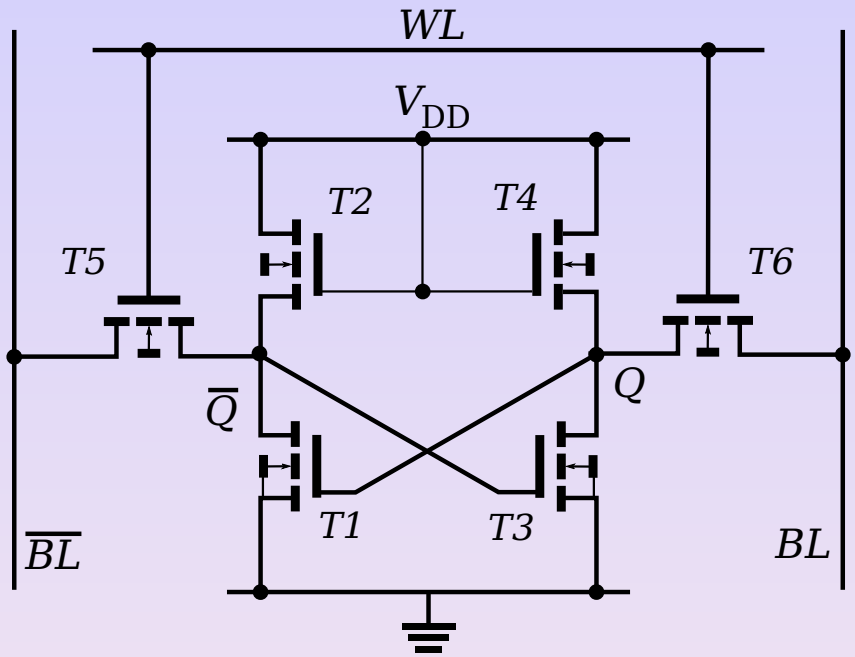
- Primary storage vs. secondary storage.
- Volatile vs. Non-volatile.
- Static RAM vs. dynamic RAM.
- Types of ROM, from Masked-ROM through Flash ROM.
- Physical memory vs. virtual memory.

Memory Hierarchy

Memory Hierarchy



SRAM

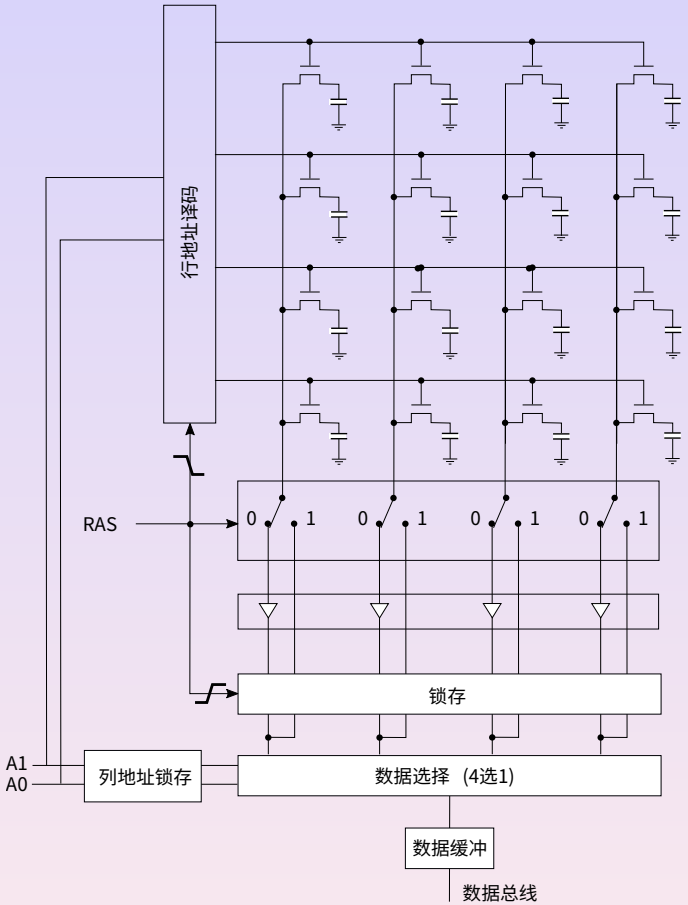


SRAM cell

READ BL and $\overline{\text{BL}}$ set to HIGH, $\text{WL}_{\text{HIGH}} \rightarrow \text{BL} = \text{Q}$

WRITE set BL and $\overline{\text{BL}}$ different level, $\text{WL}_{\text{HIGH}} \rightarrow \text{Q} = \text{BL}$

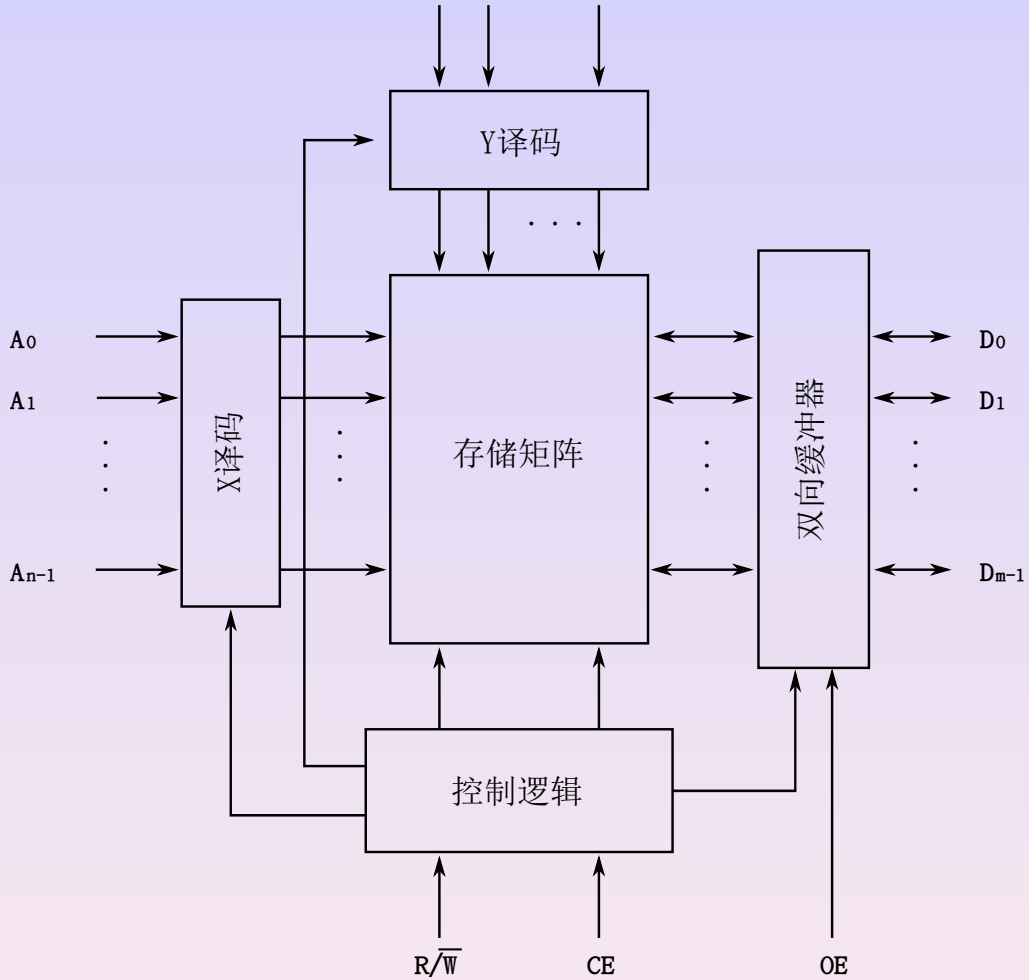
DRAM



4×4bit DRAM array

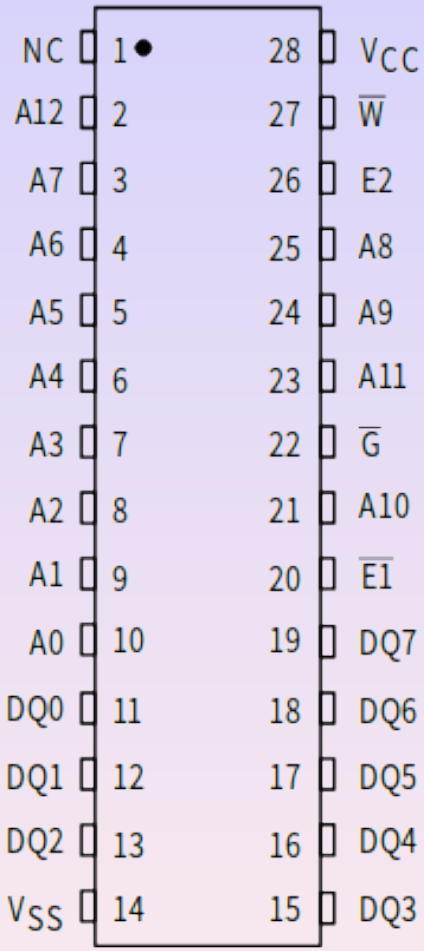
- higher density (lower cost per bit)
- capacitor needs recharge (refresh, more complex circuitry)

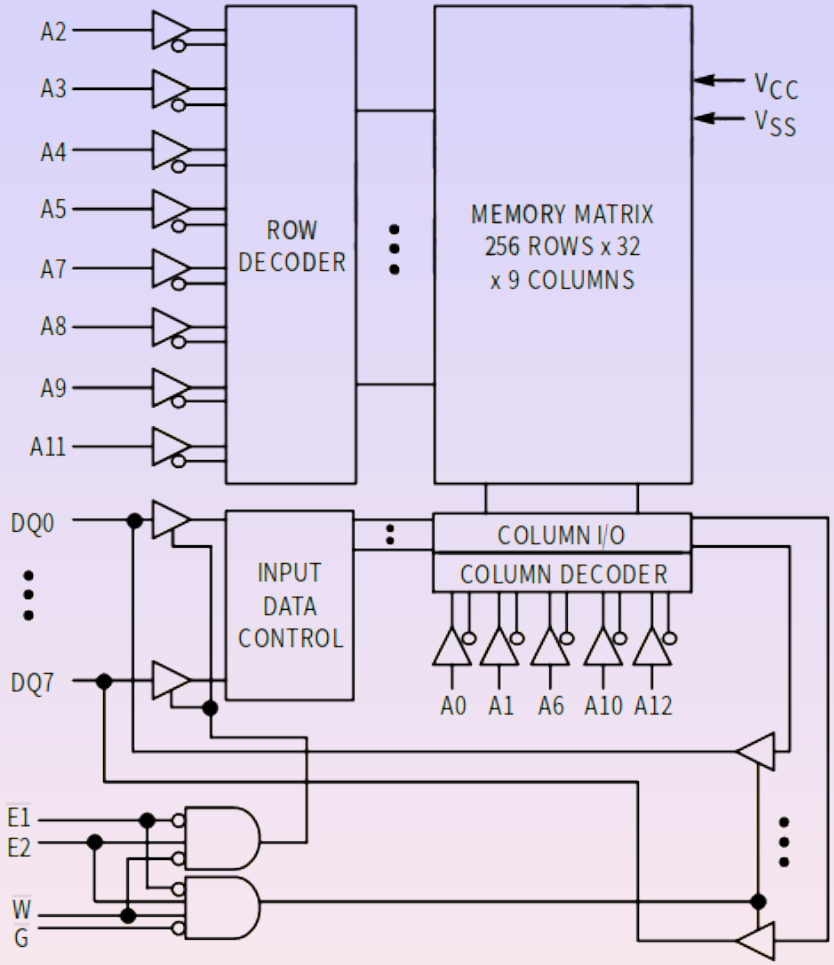
Memory array



Memory diagram

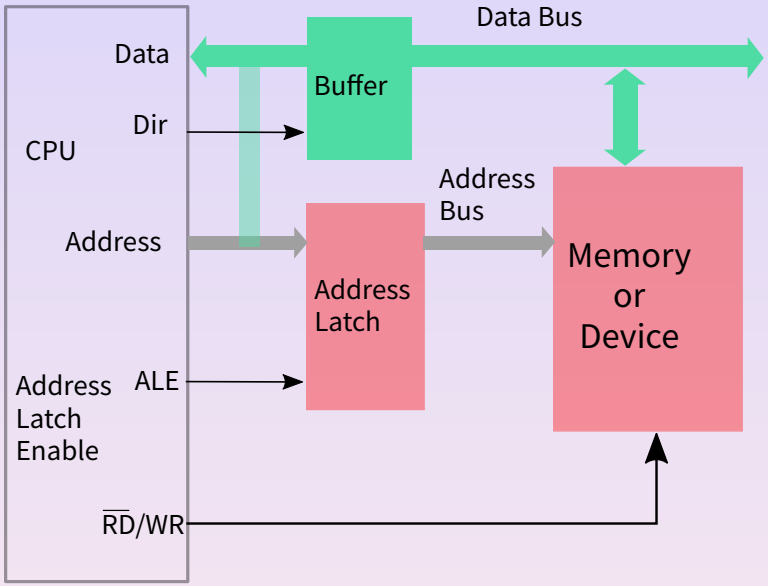
Static RAM

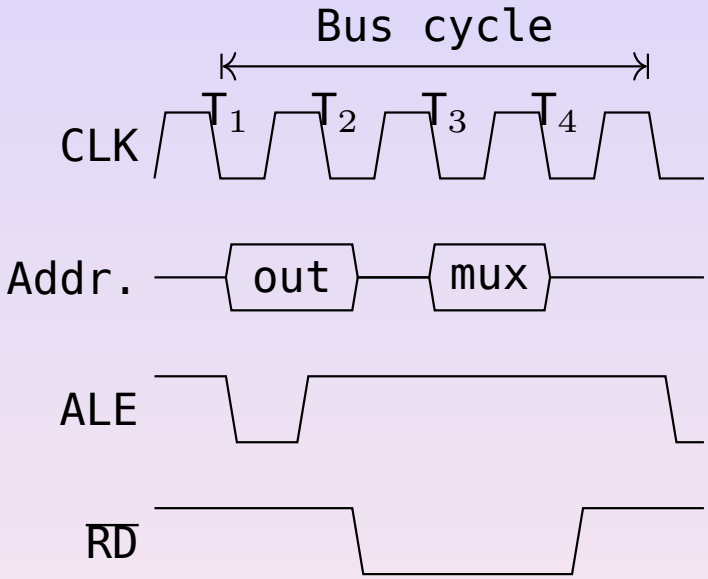

NC 1
A12 2
A7 3
A6 4
A5 5
A4 6
A3 7
A2 8
A1 9
A0 10
DQ0 11
DQ1 12
DQ2 13
VSS 14
28 VCC
27 W
26 E2
25 A8
24 A9
23 A11
22 G
21 A10
20 E1
19 DQ7
18 DQ6
17 DQ5
16 DQ4
15 DQ3


A2
A3
A4
A5
A7
A8
A9
A11
ROW DECODER
MEMORY MATRIX
256 ROWS x 32
x 9 COLUMNS
VCC
VSS
DQ0
DQ7
INPUT DATA CONTROL
COLUMN I/O
COLUMN DECODER
A0 A1 A6 A10 A12
E1
E2
W
G

Bus Cycle

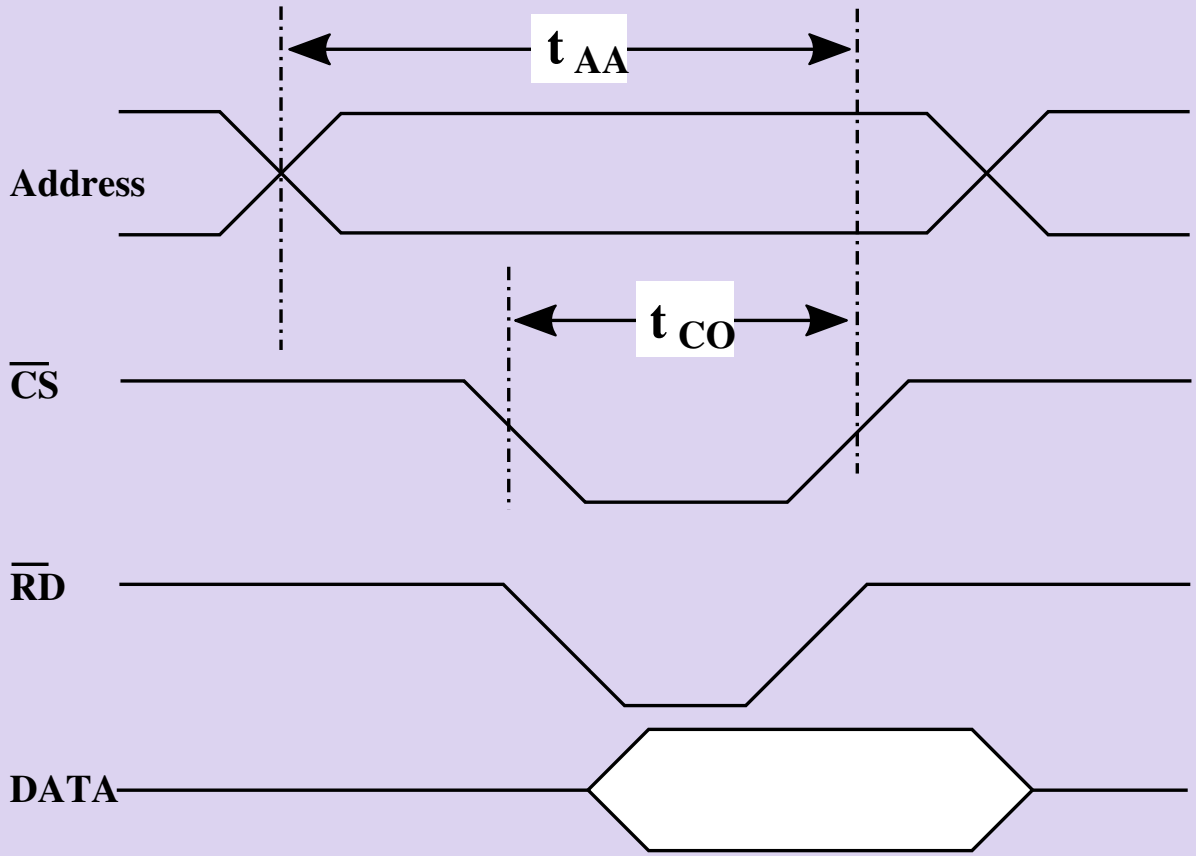
Bus Cycle is the cycle or time required to make a single read or write transaction between CPU and an external device (including memory).





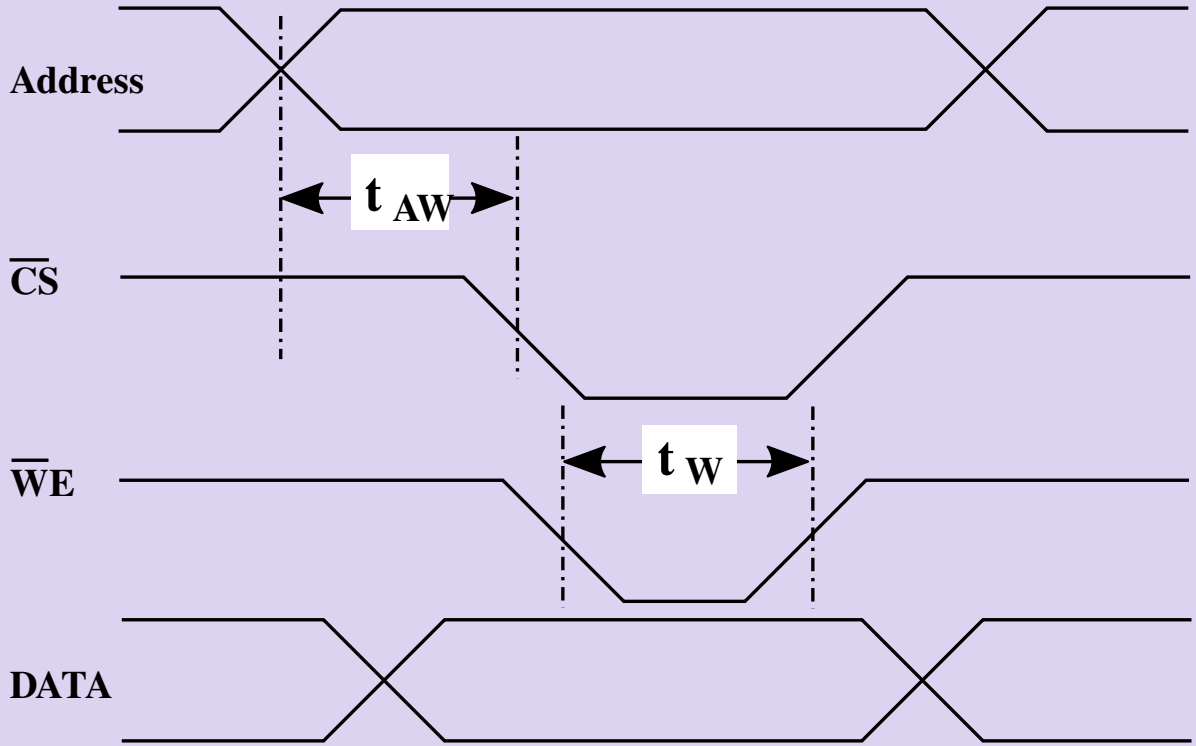
Memory Timing

Read Cycle

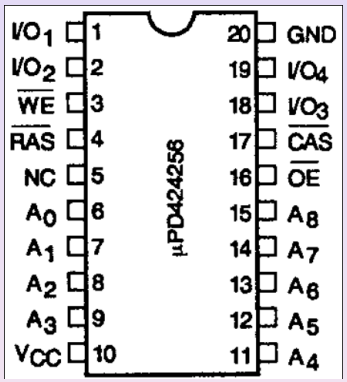


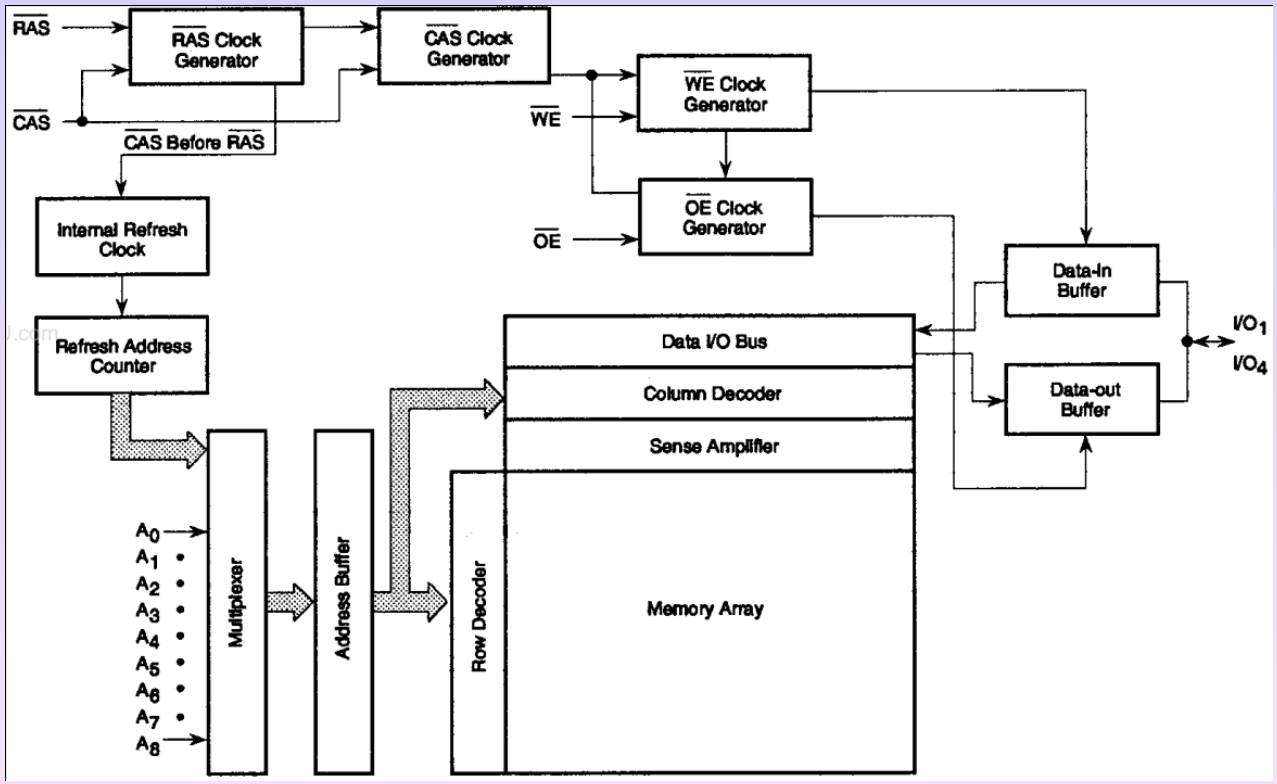
Memory Timing

Write Cycle



Dynamic RAM



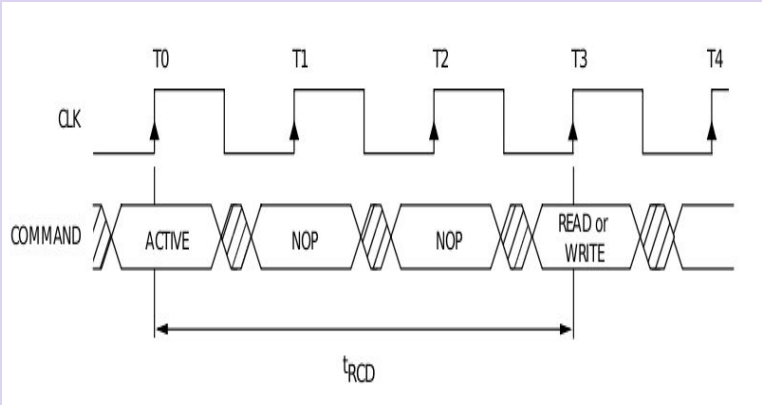


DRAM (256K×4bit)

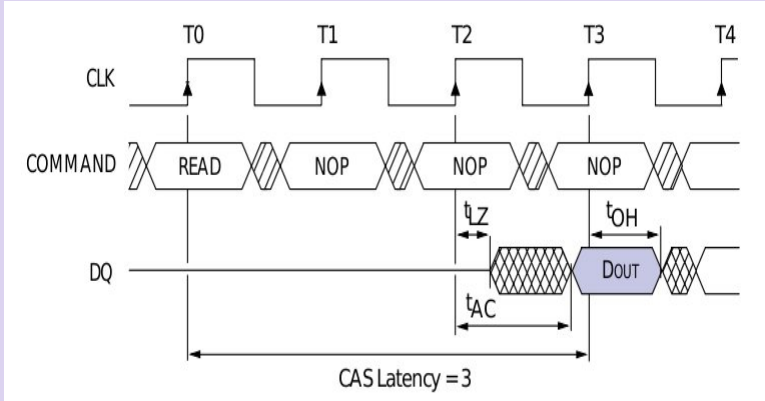
SDRAM(Synchronous DRAM)

- Synchronized with the system bus
- PC66-, PC100-, PC133- (66MHz, 100MHz, 133MHz)
- SIMM/DIMM(in-line memory modules) DDR(dual data rate)-SDRAM

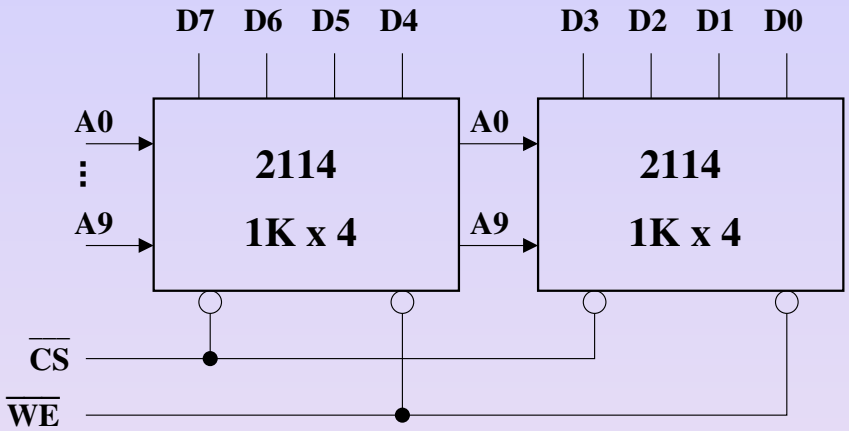
t_{RCD}



CAS Latency

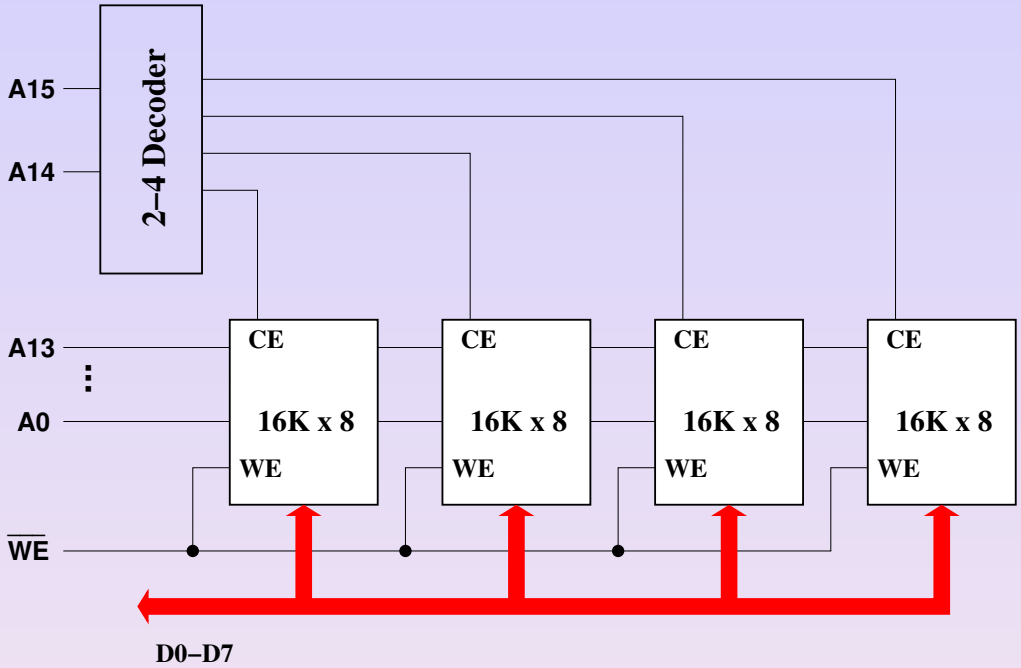


Words



Combining bits to words

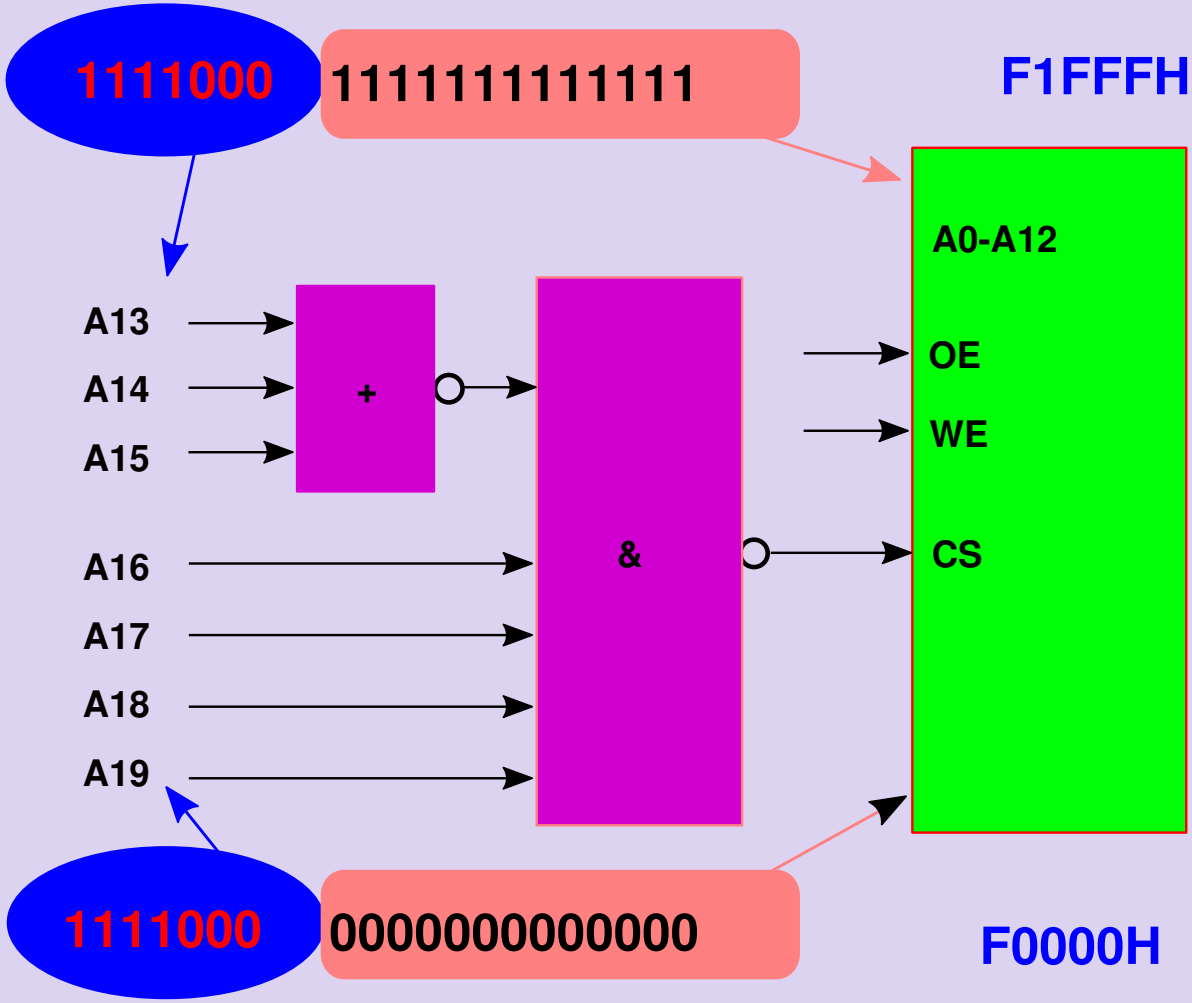
Expansion



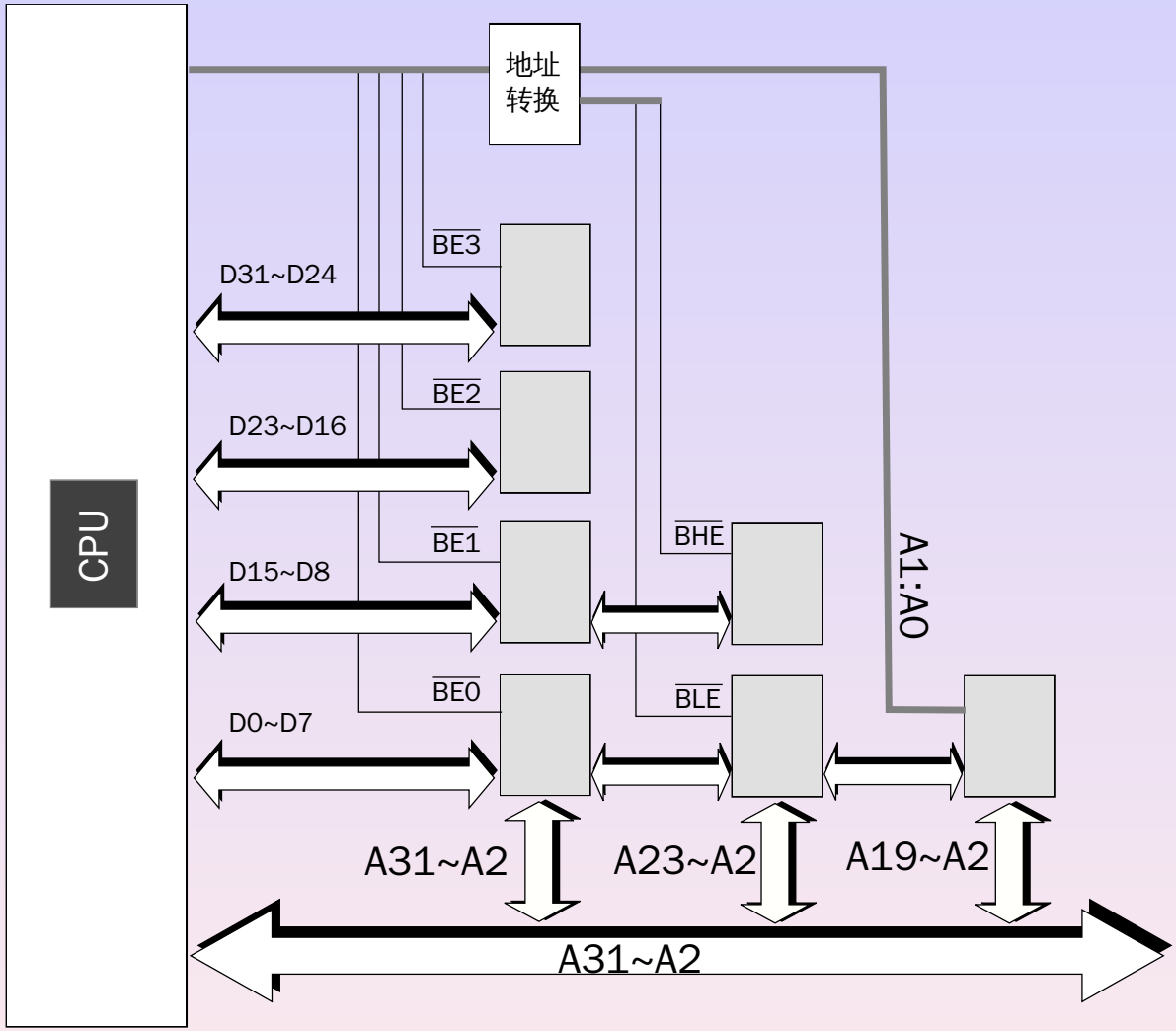
Memory Expansion

Full Decoding

8KB Memory decode



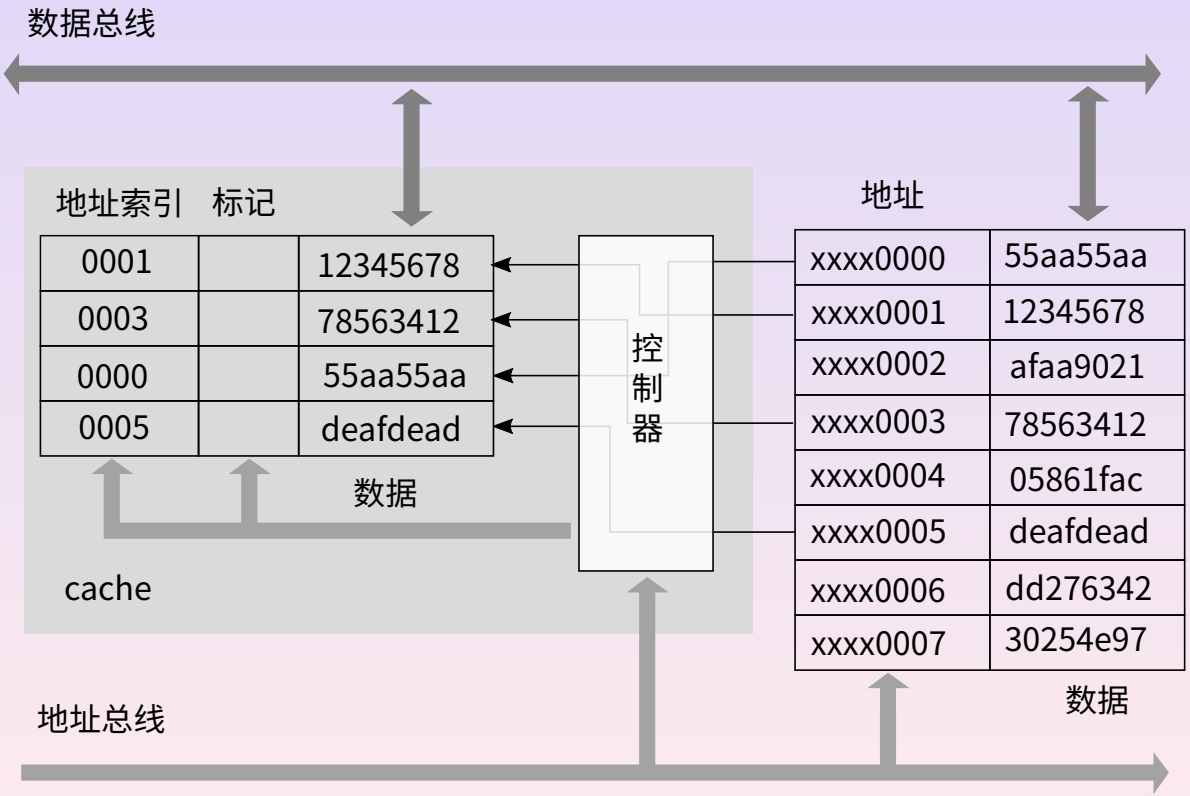
32-bit Bus



Memory system in 32bit computer

Why use Cache

- Memory and I/O need to be synchronized with CPU
- SRAM is faster than DRAM, but also expensive.



How Cache Works

Theory of Locality

Most programs trend to access memory within a limited range in a certain time:

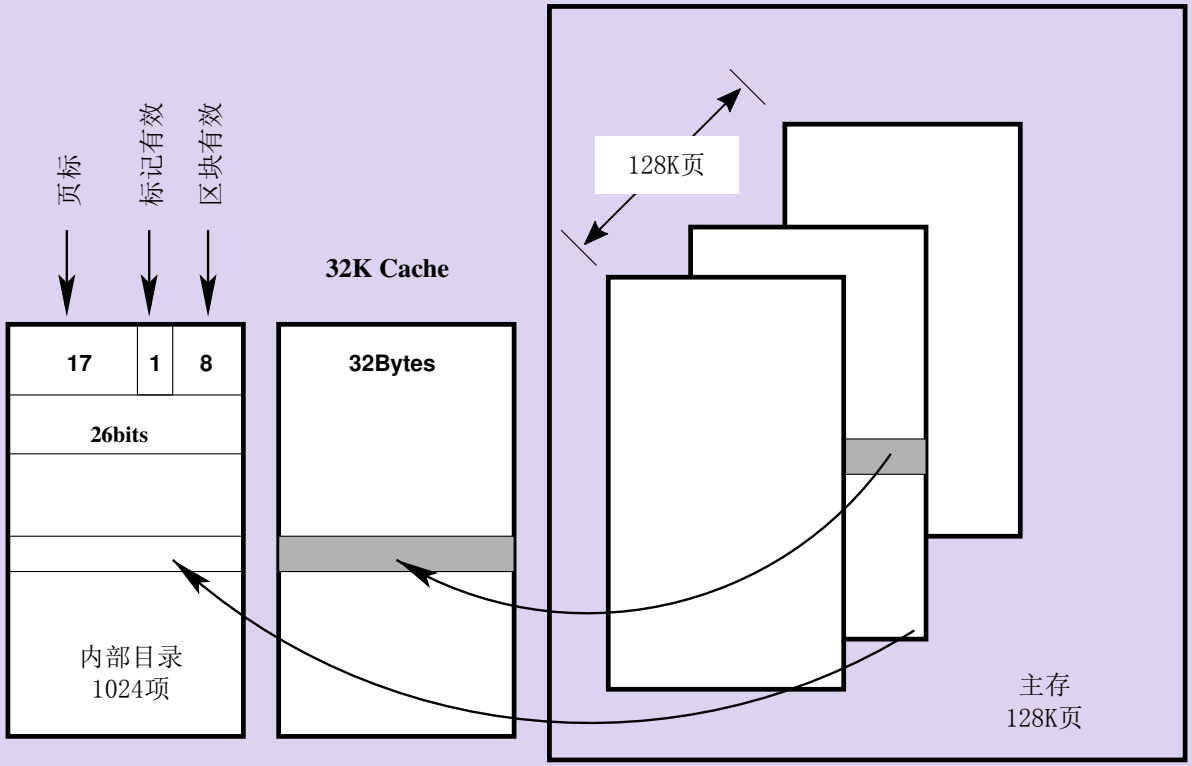
- Spatial Locality
- Temporal Locality

Mapping DRAM to a small amount of SRAM (Cache) to accelerate access.
Available mappings:

- Fully associative cache
- Direct-mapped cache
- Set associative

Cache Map

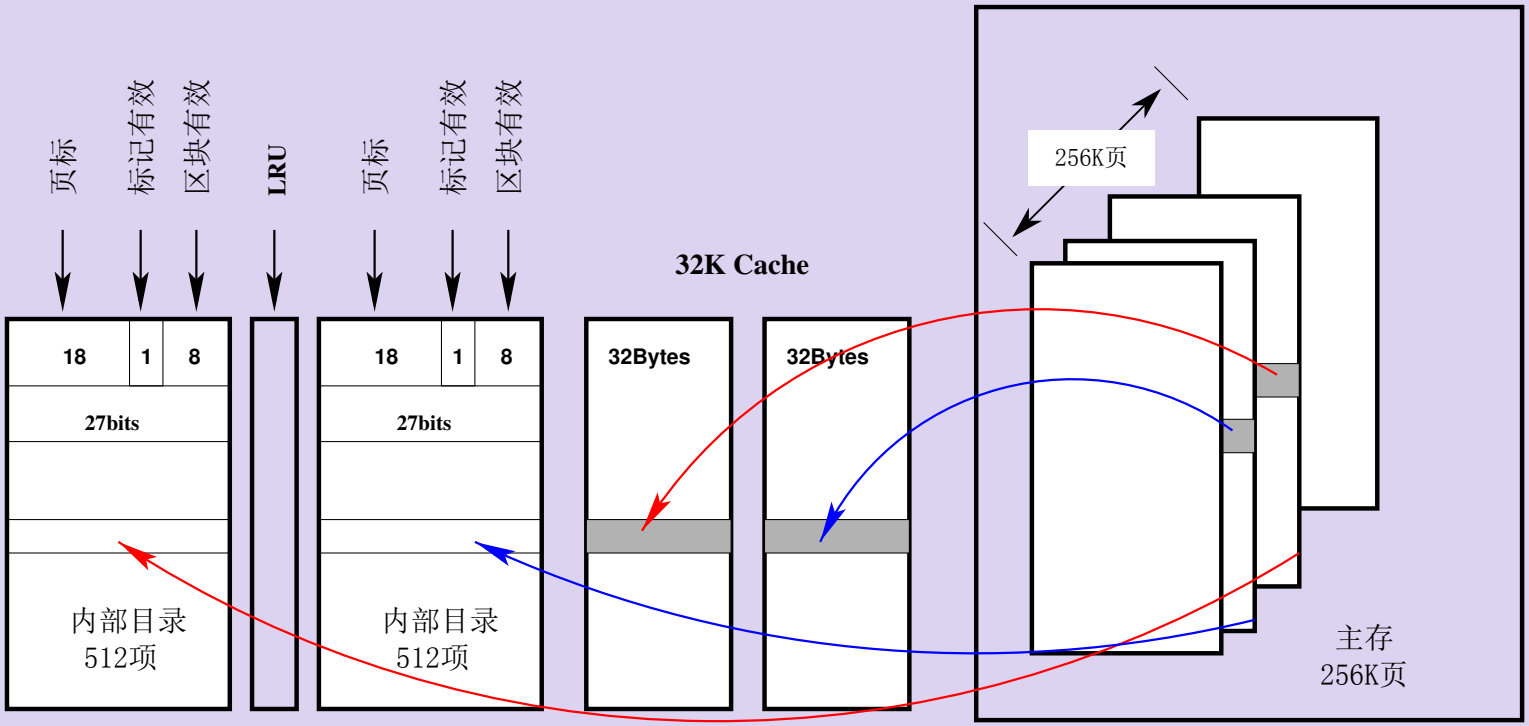
Direct Map



Direct-mapped Cache

Cache Map

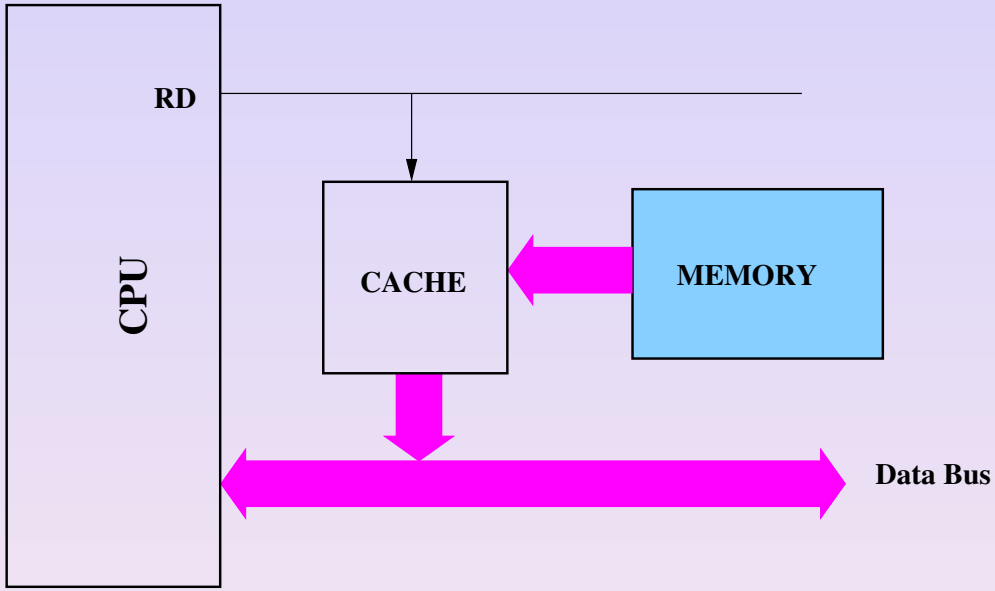
Set associative



2-way set associative

Consistency

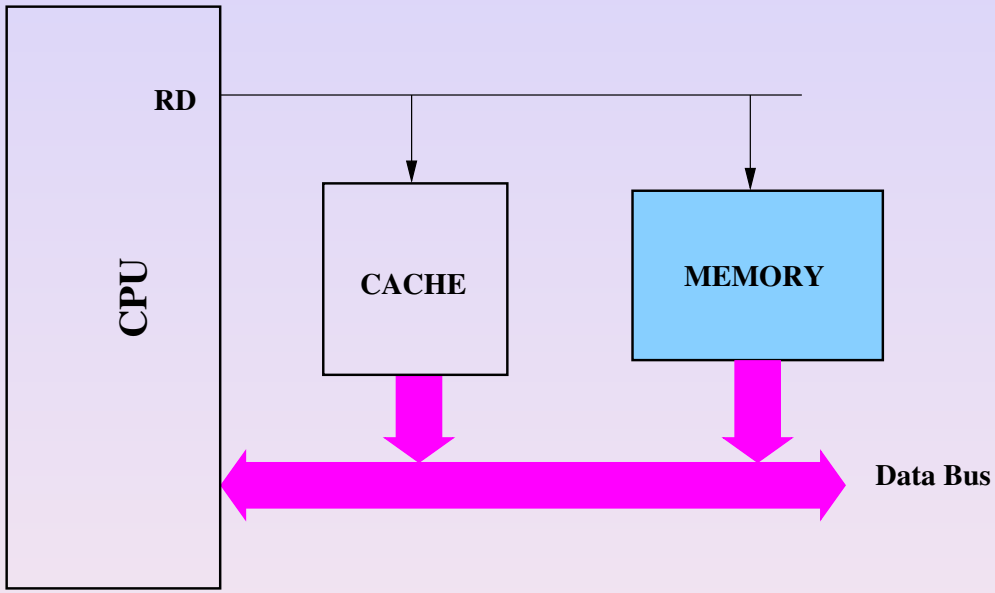
- Reading
  - Look through, always read from cache first



  - Line fill

Consistency

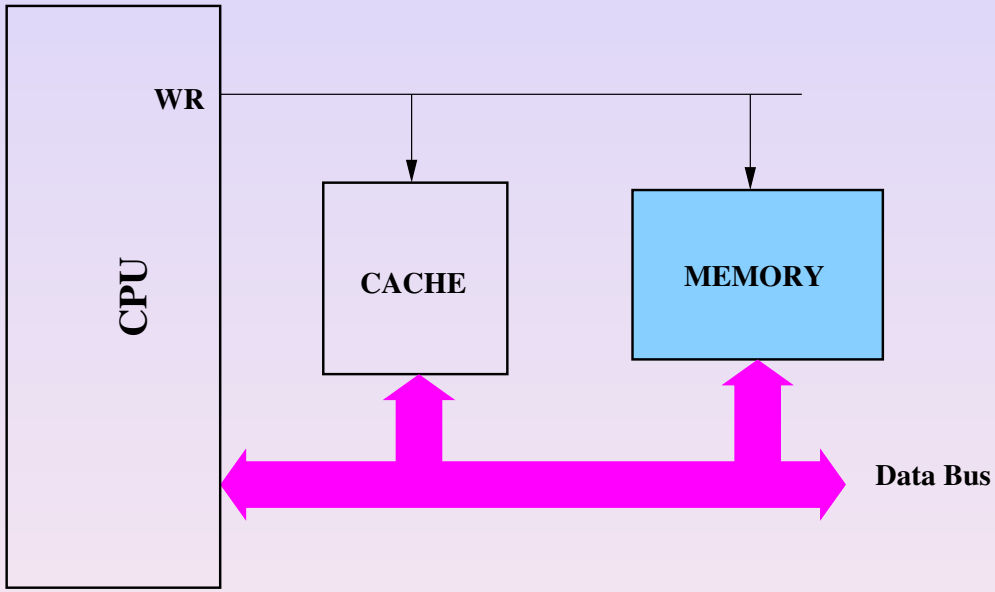
- Reading
  - Look aside, read both from cache and main memory



  - Line fill

Consistency

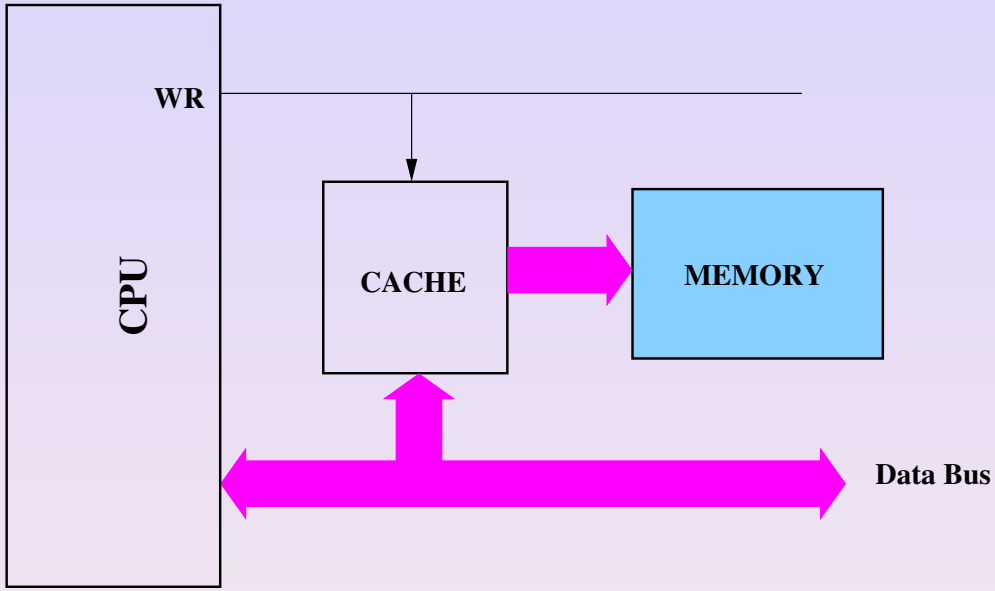
- Writing(updating main memory) when multi-processors or DMA present
  - Write-through at the same time, increase bus access



  - Invalidation

Consistency

- Writing(updating main memory) when multi-processors or DMA present
  - Write-back at certain time(bus watching)



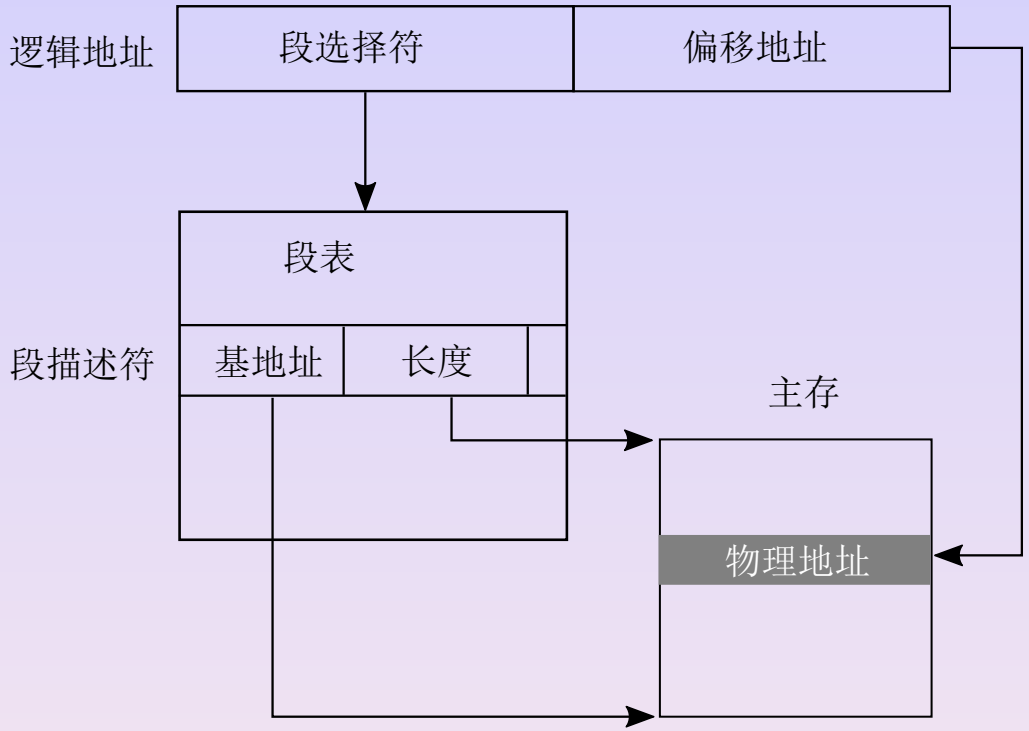
  - Invalidation

What Happens?

```
103ec: e3a02a02 MOV R2 , #8192
103f0: e5c23008 STRB R3 , [R2 , #8]
```

Read instructions and data

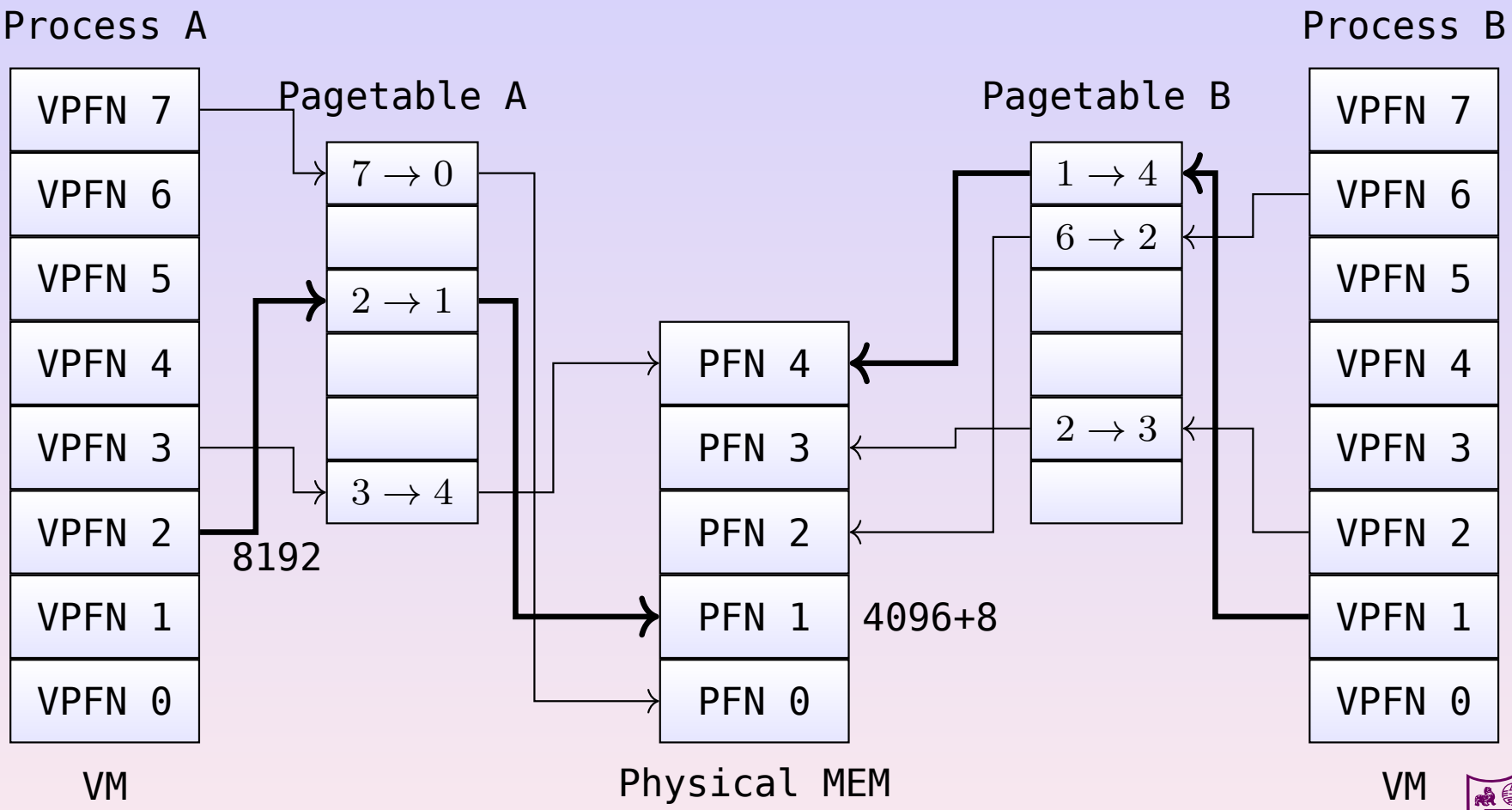
Segment Management



Address Conversion in Segment Unit

A logical (virtual) address consists of two parts: segment selector and offset.

Page Management



Paging Mode

Segment and Page Descriptor

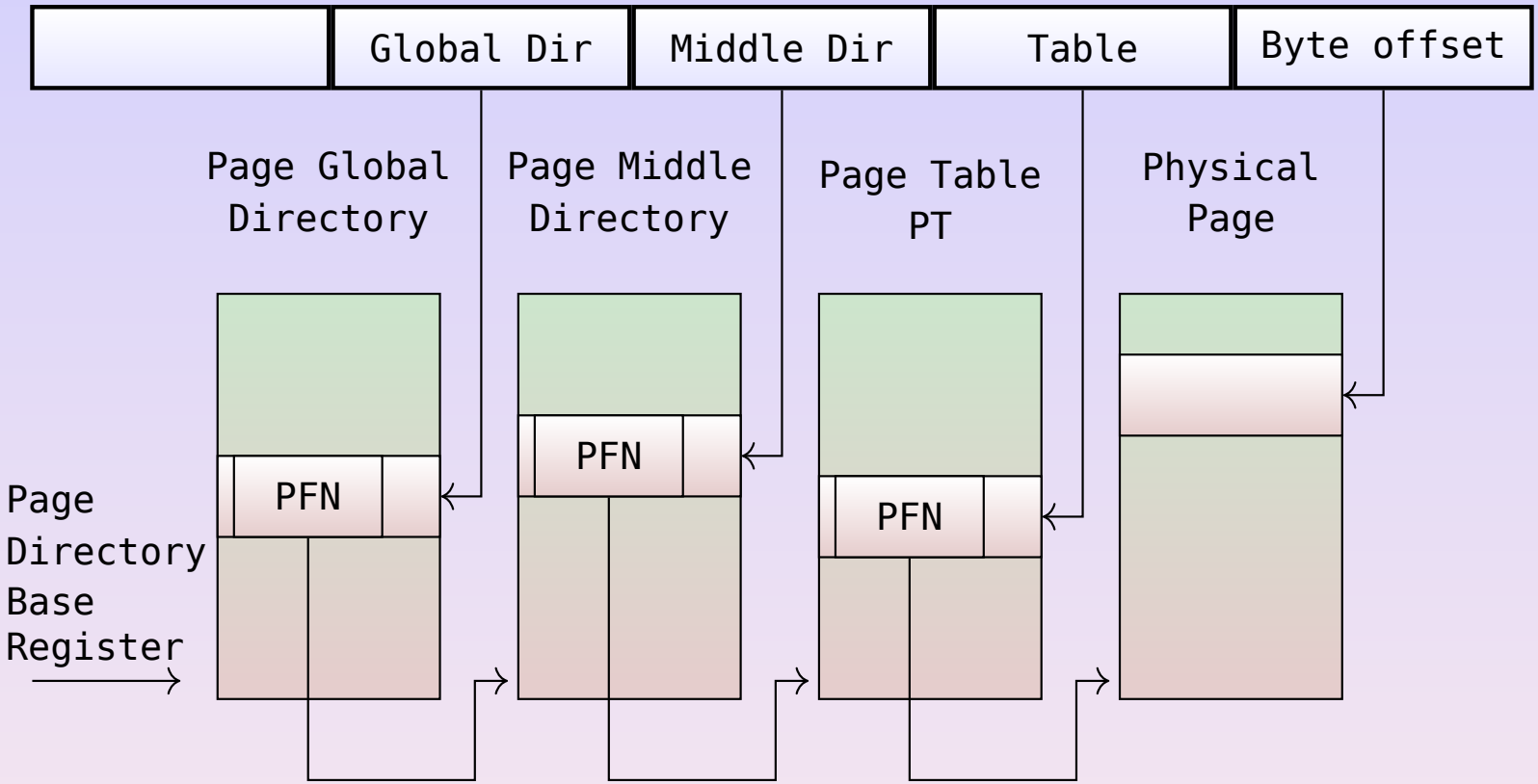
X86(32bit) Segment Descriptor

$BASE_{31-24}$	G	D	0	AVL	$LIMIT_{19-16}$	P	DPL	S	TYPE	A	$BASE_{23-16}$				
$BASE_{15-0}$						$LIMIT_{15-0}$									

Page Item

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Page Physical Base Address																						avl		G	PAT	D	A	PCD	PWT	U/S	R/W	P

Multilevel Page Table



Three level of paging designing can be easily configured to 2 level mode by folding middle level.

TLB

Translation Look-aside Buffer (TLB) is a part of the chip's memory-management unit (MMU)
TLB is used to reduce the time taken to access a user memory location.

Intel's Nehalem (2008, core i7/i5)

cache	4KB	2MB (pagesize)
L1	32KB+32KB	64 entries
L2	256KB (unified TLB)	512 entries/4way
L3	4–12MB	

example: hit – 1 clock cycle, missing – 30 clock cycles, missing rate – 2%.
average cycle: 1*0.98 + 30*0.02≈1.6 cycles

Page Swapping

Accessing an address that is not mapped to physical memory will trigger a page fault exception (interrupt).

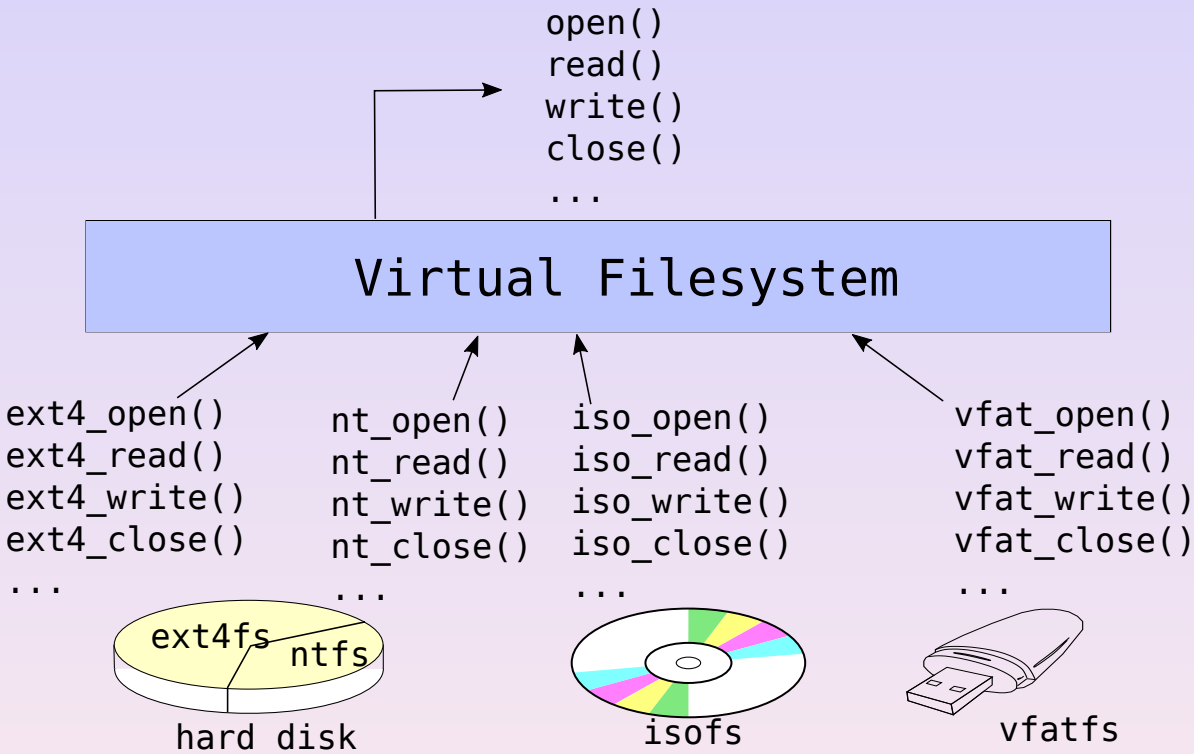
1. MMU finds a physical memory to be swapped (ether drop or backup).
2. The page that is swapped out is marked as “unmapped”.
3. Copy data.
4. Mark PFN in pagetable for this page.

Evolution of X86

- 8086 (1979)
- 80286 (1982), Protected virtual address mode, multi-task
- 80386 (1985), 32bit(80386DX) protected mode, v86 mode supported.
- 80486 (1989), Internal cache(8KB) and math coprocessor(486DX),PLL.
- Pentium(1993), 8KB D-cache and 8KB P-cache,Superscaler pipelining, : SMM
- Pentium-MMX. Multi-Media eXtension : (or maybe Matrix Math eXtension)
- PentiumPro (1995), L1-cache and L2-cache.
- IA-64(Itanium, new ISA) vs. AMD64

VFS

Filesystem describes the way the data is stored and retrieved by filename.



VFS is a common interface for all filesystems.

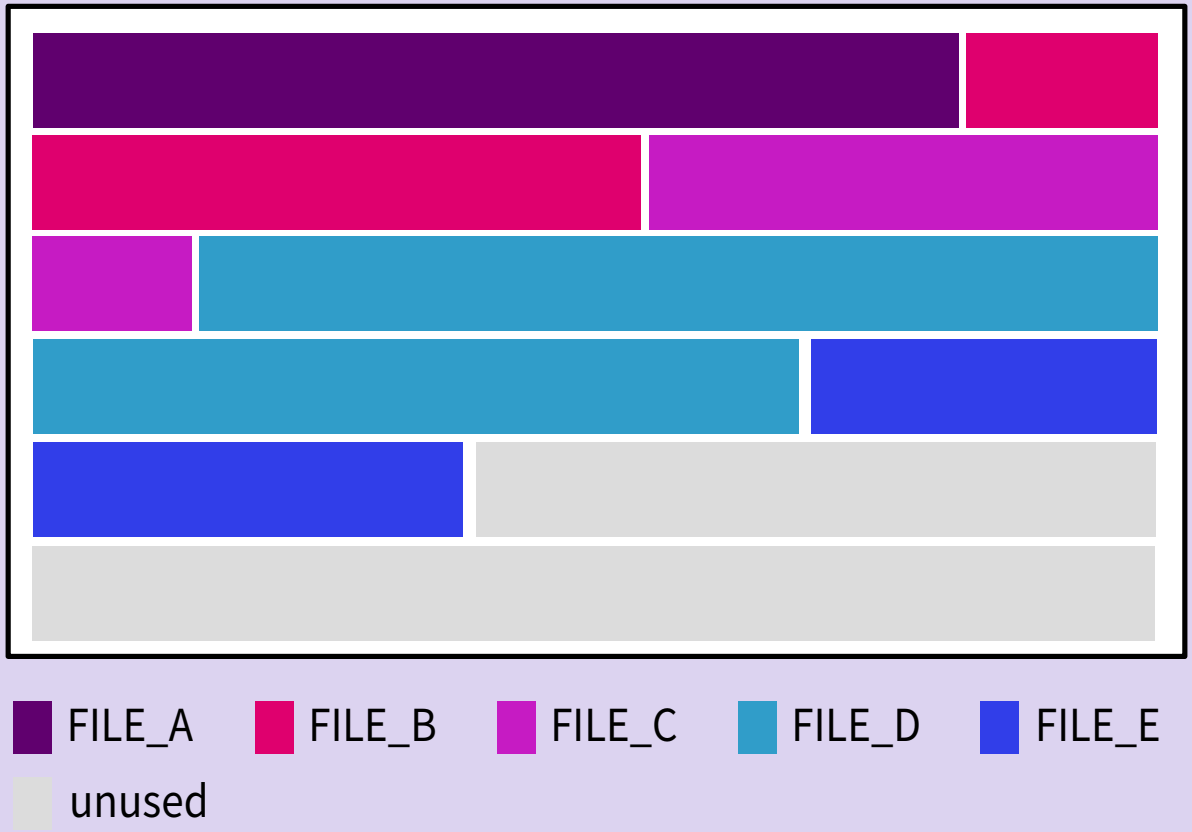
Filesystems in Linux

Types of filesystem (Linux)

- Disk-based: FAT-series, EXT-series, YAFFS2, NTFS, ISO, ...
- Network: NFS, SAMBA, ...
- Pseudo filesystems: PROCFS, SYSFS, ...

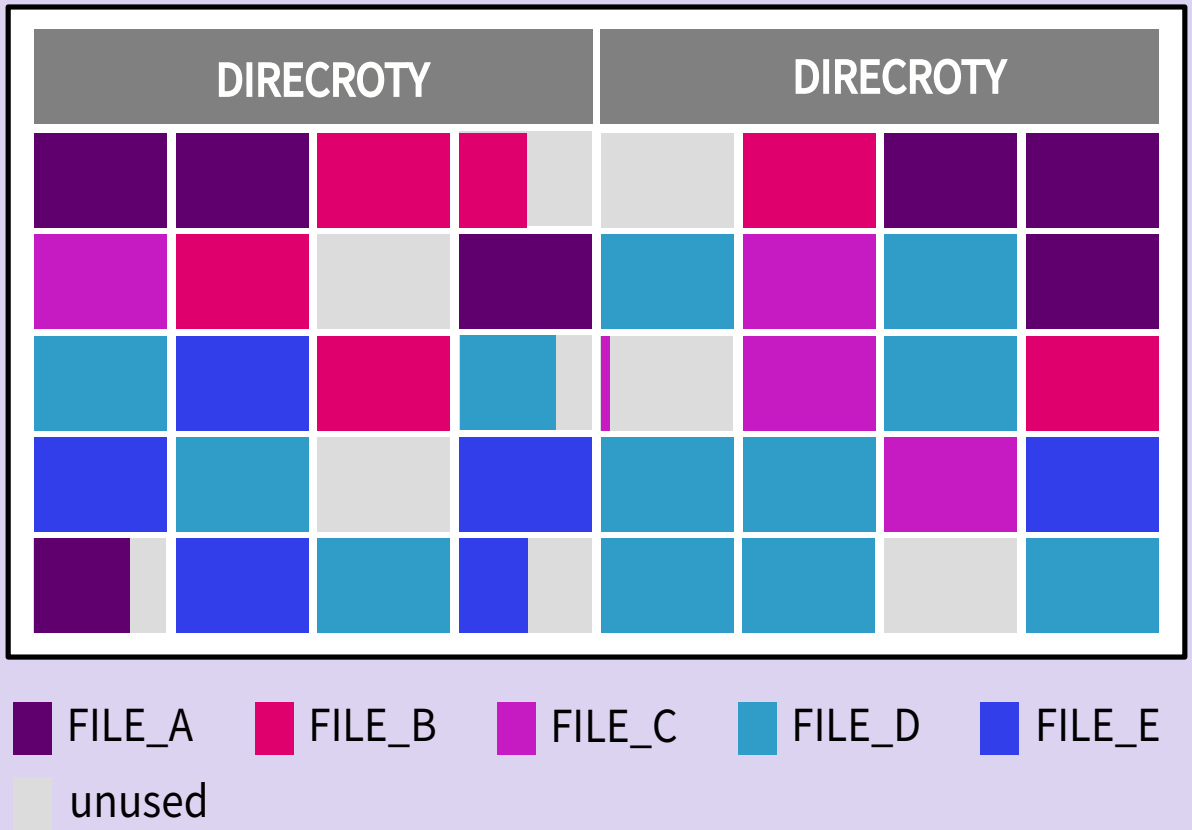
Why Use FS

Non-FS structure



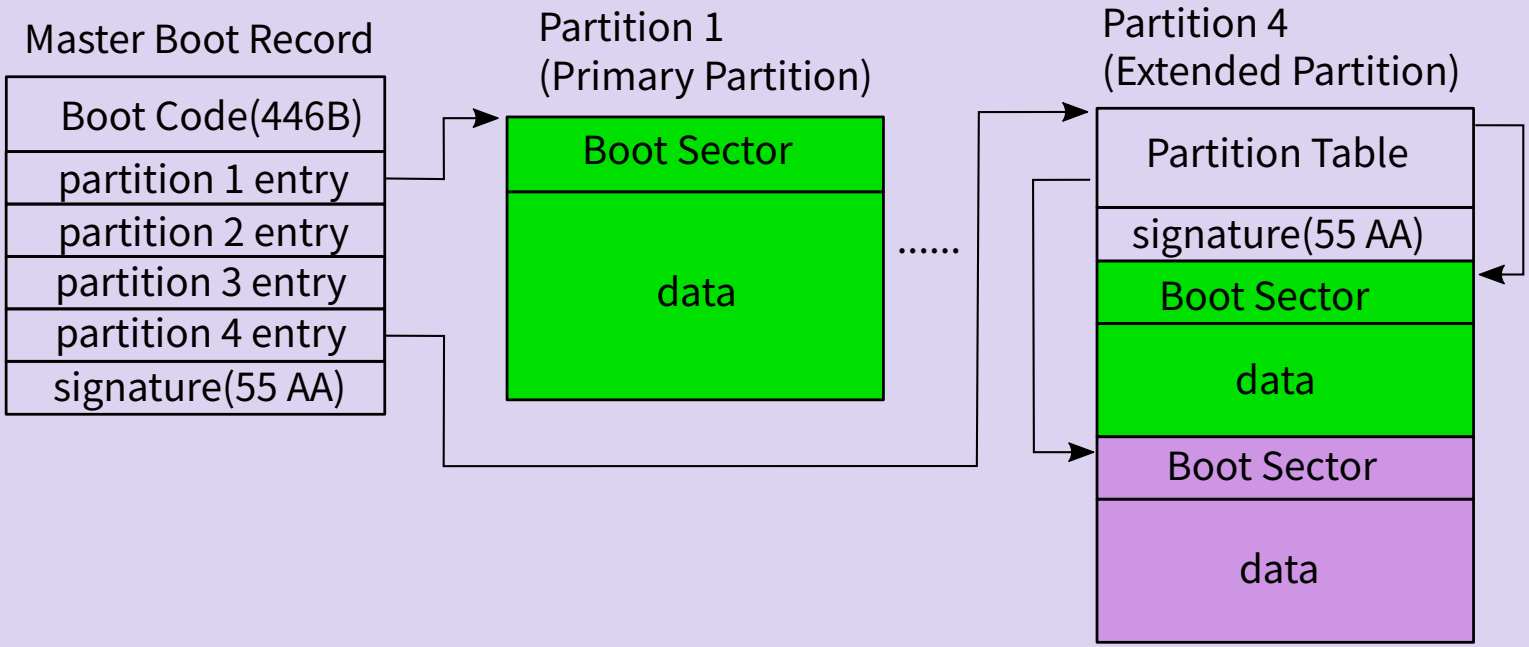
Why Use FS

Filesystem



Disk Partition

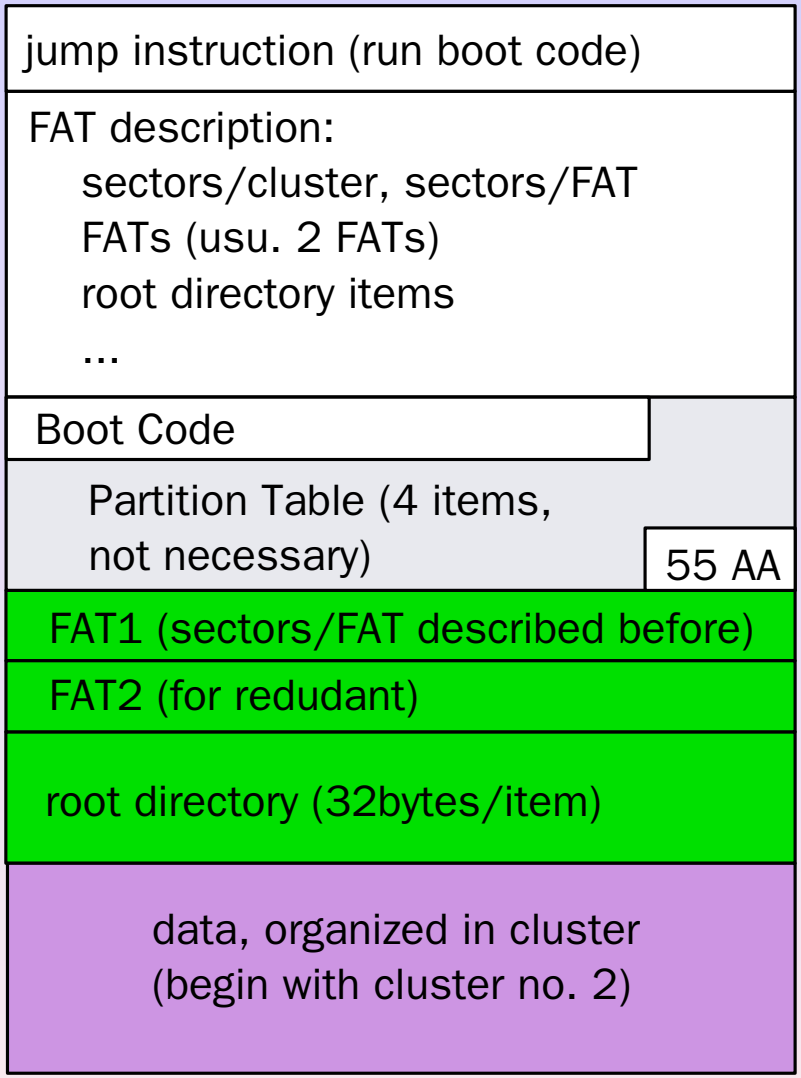
MBR



MBR uses 4 bytes describing a disk, max. 2^{41} bytes of a disk.

Modern disk uses GPT format, which uses LBA (Logical Block Address, 8 bytes), allows $(2^{64}-1)\times 512 = 8$ZiB of a disk size.

FAT File System



Partition Entry

80	Boot indicator
20 21 20	CHS (Start)
07	Filesystem type ID
59 1a bf	CHS (End)
00 80 00 00	Relative sectors
00 0e 2e 00	Total sectors

FAT info

FAT format is described in boot sector:

offset	bytes	meaning
0000	3	JMP instruction
0003	8	Version info
000B	2	Bytes per sector
000D	1	Sectors per cluster
000E	2	Reserved sectors, incl.MBR
0010	1	Number of FATs
0011	2	Root entries,32B/entry
0013	2	Sectors (on small volumes)
0015	1	Media Descriptor
0016	2	Sectors per FAT

Data begin at (Reserved sectors + FATs + Root entries), marked as cluster No.2.

FAT info(cont.)

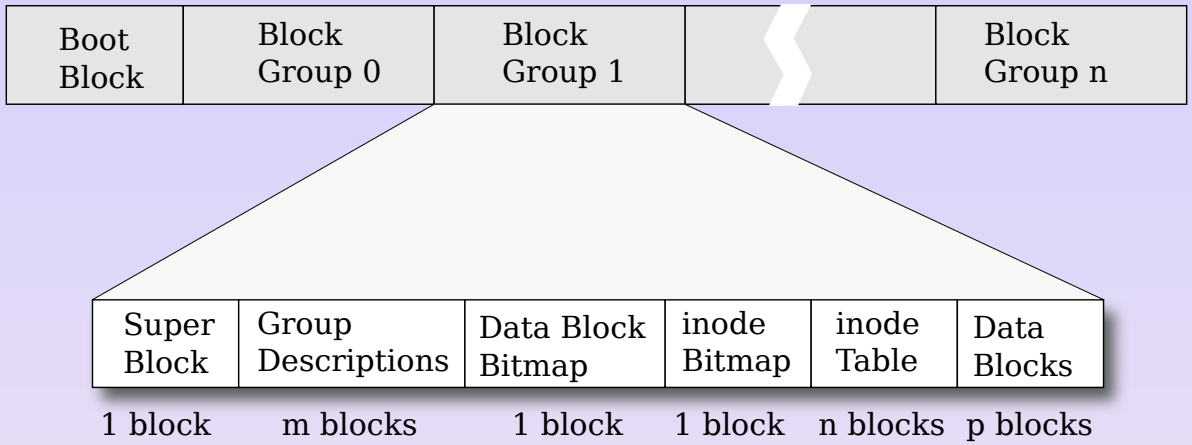
offset	bytes	meaning
0018	2	Sectors per track
001A	2	Heads
001C	4	Hidden sectors
0020	4	Sectors (on large volumes)
0024	1	Physical driver number
0025	1	Reserved
0026	1	Signature (0x28 or 0x29)
0027	4	Serial number (random)
002A	11	Volume label ('NO NAME ')
0036	8	FS type ('FAT12 ')

Directory Entry

offset	bytes	meaning
00	11	Filename(8.3)
0B	1	Attribute
0C	10	Reserved
16	2	Modified time
18	2	Modified date
1A	2	First cluster
1C	4	File length(bytes)

Cluster mark: 000:free, 001:reserved, 002–FEF:next cluster number, FF0–FF6: (reserved values), FF7: bad cluster, FF8-FFF: last cluster
(Cluster No. 0 and 1 are for special purpose).

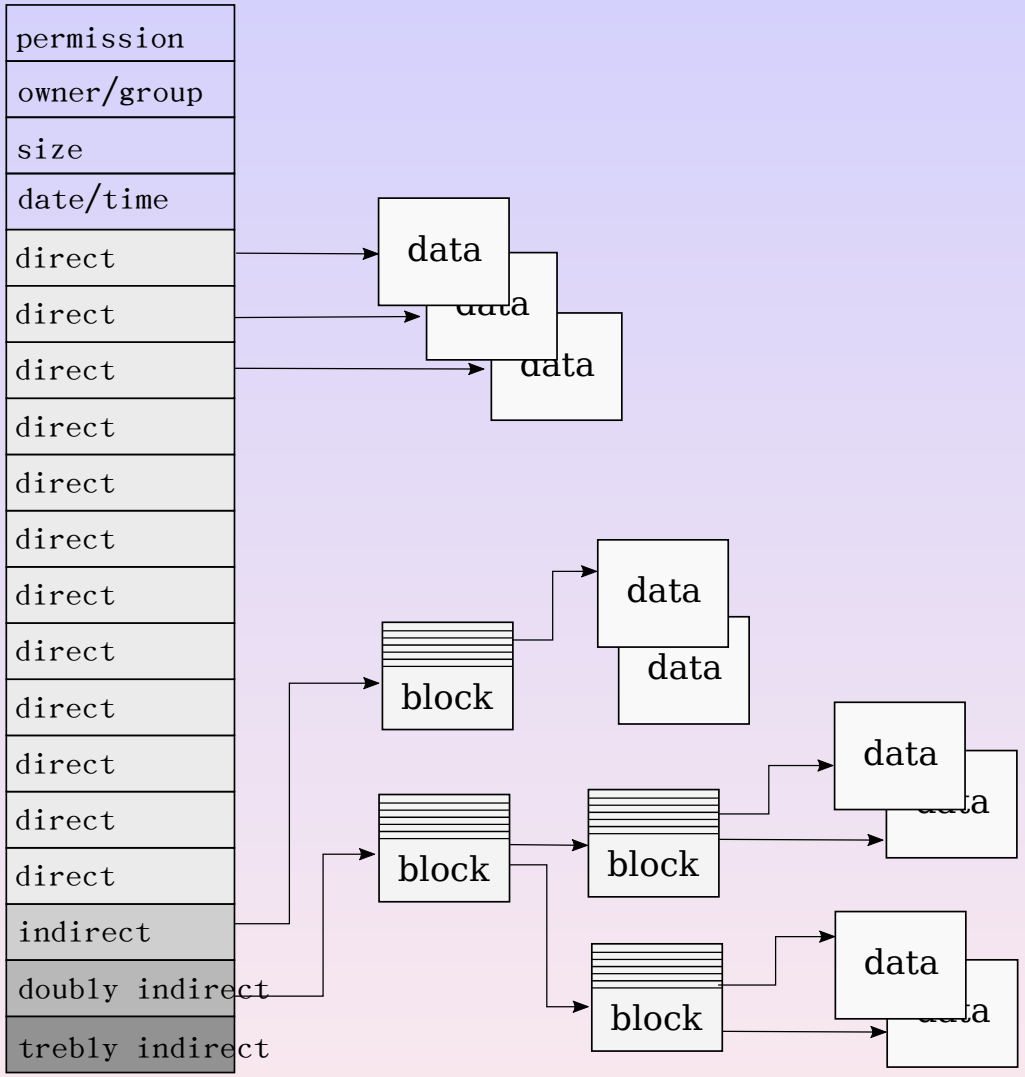
Ext2FS



Ext2 Layout

Inodes and block size are fixed once a disk is formatted. File attributes such as size, permission, owner/group, location, date/time, ... (but not filename) are stored in an inode.

INODE



Direct: 12*1KB

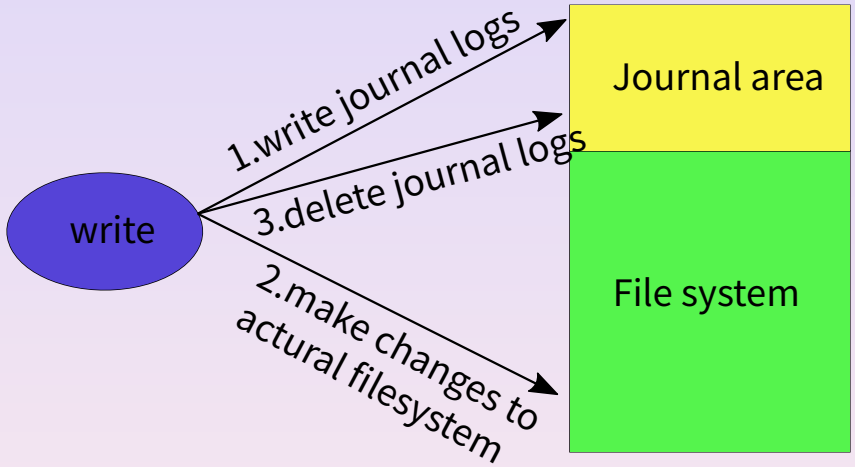
single indirect:
1KB/4*1KB=256KB

doubly indirect:
256^2*1KB=64MB

trebly indirect:
256^3*1KB=16GB

Journal File System

System crash may lead to the loss of data when file written. File system can be corrupted if not closed when the system shuts down.
Using a journal allows data recovery of files and the data within it.



Summary

- Memory (RAM, ROM)
- Memory Organization
- Cache (SRAM vs. DRAM)
- Virtual Memory (Internal memory vs. Disk)
- Filesystem