Overview of Memory Technology

Minsoo Ryu

Department of Computer Science and Engineering
Hanyang University

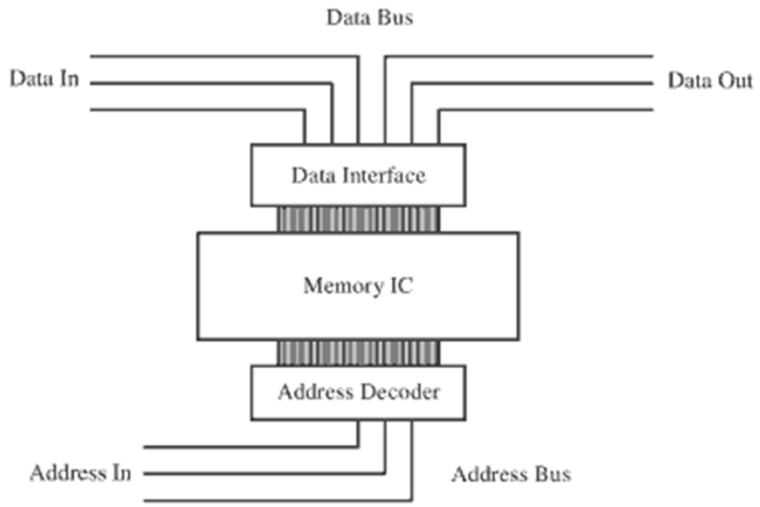
msryu@hanyang.ac.kr

Contents

1. Read Only Memory (ROM)
2. Random Access Memory (RAM)
3. Memory Interfacing
4. Flash Memory

Memory IC

- A memory IC is made up of three units
  - **Memory array**
    - Two dimensional array of memory cells addressed by a unique row and column, in which each cell can store 1 bit
  - **Address decoder**
  - **Data interface**



Read-Only Memory (ROM)

❒ ROM is a type of non-volatile memory

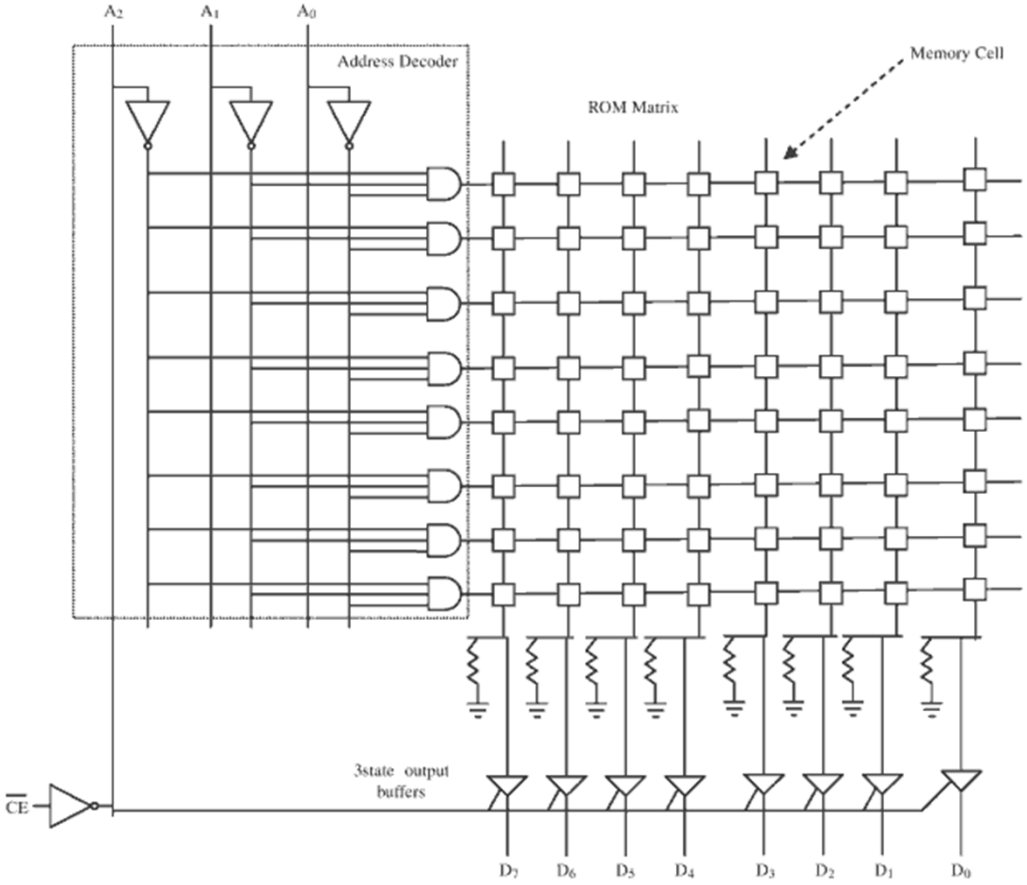


Figure 5-5: 8 x 8 ROM logic circuit [5-1]

어드레스 갯수 x 비트 크기

A x B ROM means that
the number of addresses is A
and the size of data is B

Read-Only Memory (ROM)

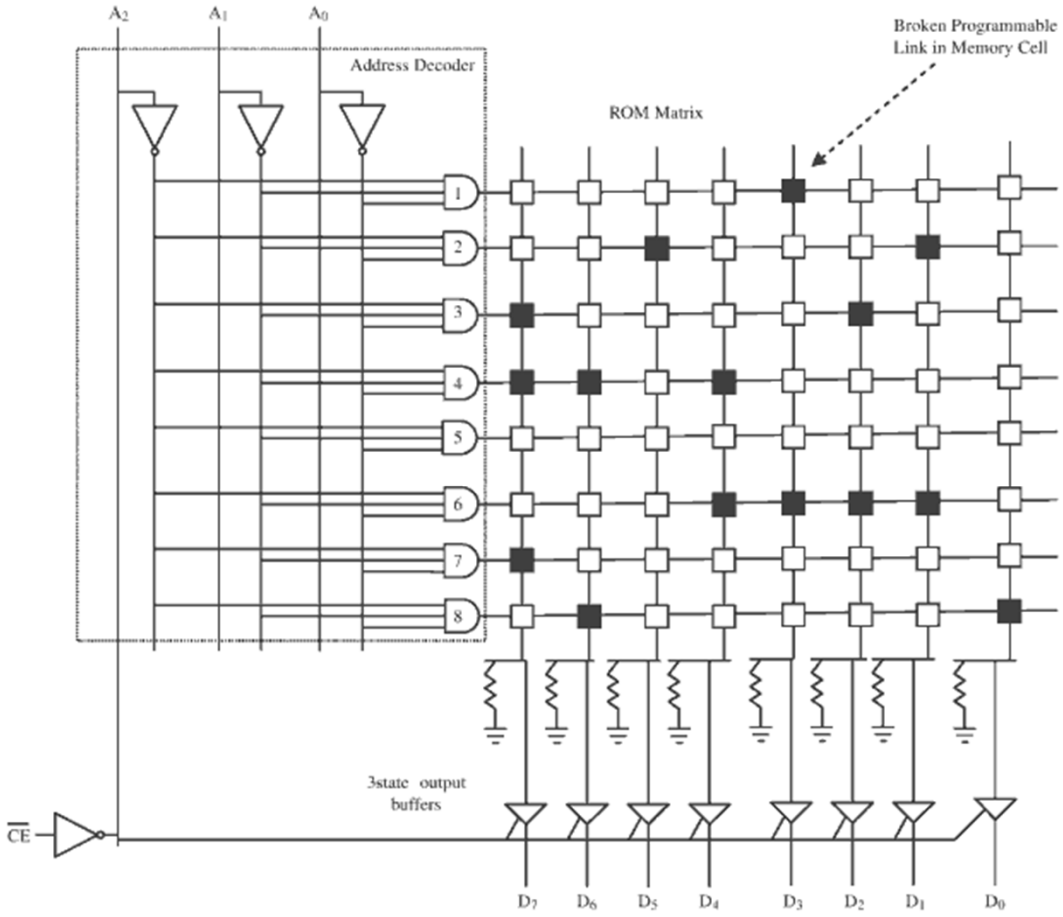


Figure 5-7: 8 x 8 reading ROM circuit [5-1]

When a programmable link is in place, the transistor is biased ON, resulting in a “1” being stored

When a programmable link is broken, a “0” is stored

Types of ROM

- **Mask ROM (MROM)**
  - Data bits are permanently programmed by the manufacturer
  - Usually only produced in high volumes
- **Programmable ROM (PROM)**
  - One-Time Programmable ROM (OTP)
  - Can be programmed outside the manufacturing factory using a ROM burner that burns out fuses of cells using high voltage/current pulses
- **Erasable Programmable ROM (EPROM)**
  - Can be erased using a device that outputs intense short-wavelength ultraviolet light

Types of ROM

- **Electrically erasable programmable ROM (EEPROM)**
  - **Can be electrically erased, then rewritten electrically**
  - **Writing or flashing an EEPROM is much slower (milliseconds per bit) than reading from a ROM or writing to a RAM (nanoseconds in both cases).**
  - **Electrically alterable read-only memory (EAROM)**
    - Can be modified one bit at a time
  - **Flash memory (or simply flash) is a modern type of EEPROM invented in 1984**
    - Flash memory can be erased and rewritten faster than ordinary EEPROM

Contents

1. Read Only Memory (ROM)
2. Random Access Memory (RAM)
3. Memory Interfacing
4. Flash Memory

Random Access Memory (RAM)

- RAM is a type of volatile memory
  - Often referred to as main memory
- Two types of RAM
  - Static RAM (SRAM)
  - Dynamic RAM (DRAM)

Static RAM (SRAM)

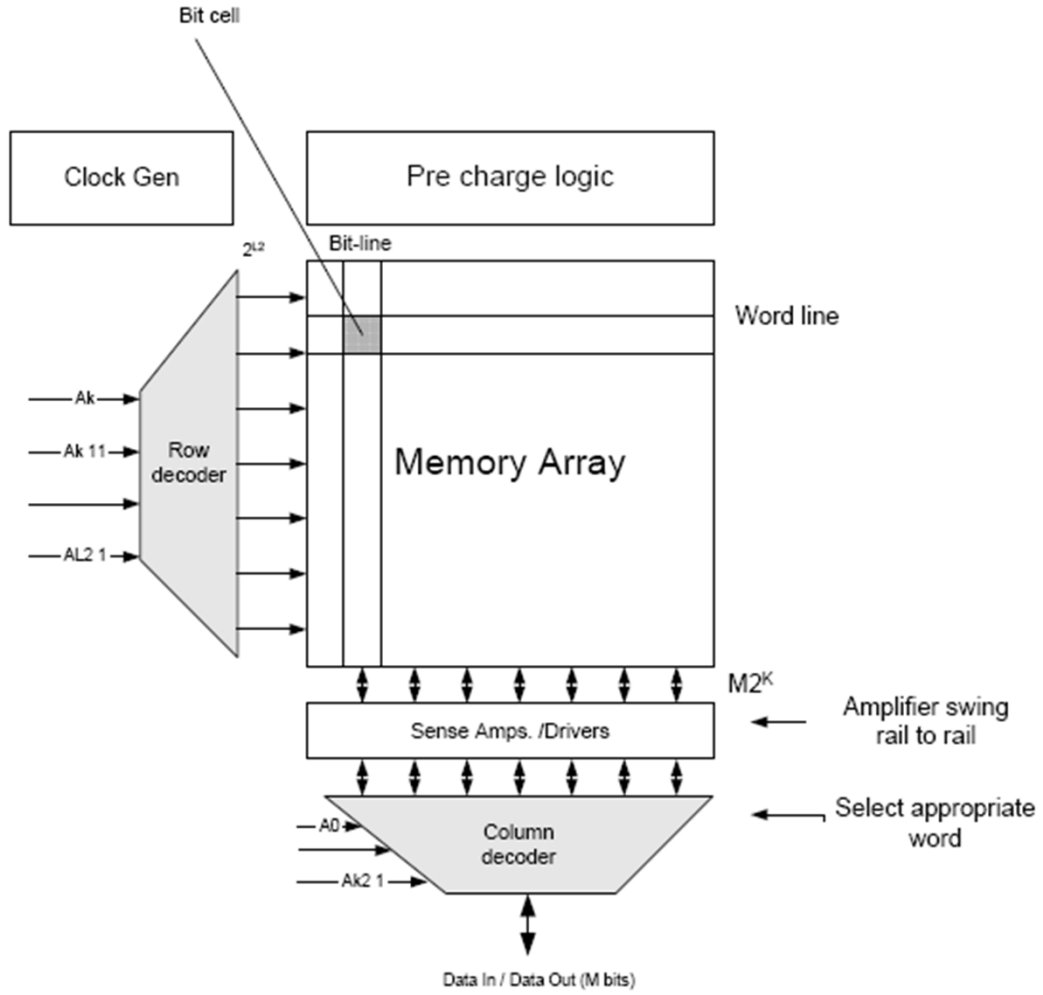
- Store a bit of data in the state of a D latch (rather than edge-triggered D flip-flop)
- Fast, low-power, expensive
- No need to be periodically refreshed

Dynamic RAM (DRAM)

- Store a bit of data as a charge in a capacitor (or transistor)
  전하가 샌다. 따라서 refresh 필요
- Slower and consumes more power than SRAM, cheap

- Need to be periodically refreshed (refresh cycle)
  - Refreshed periodically by reading it and then writing it back
  - Typically 1 refresh cycle per every 64 ms

Typical SRAM Organization



Dynamic RAM (DRAM)

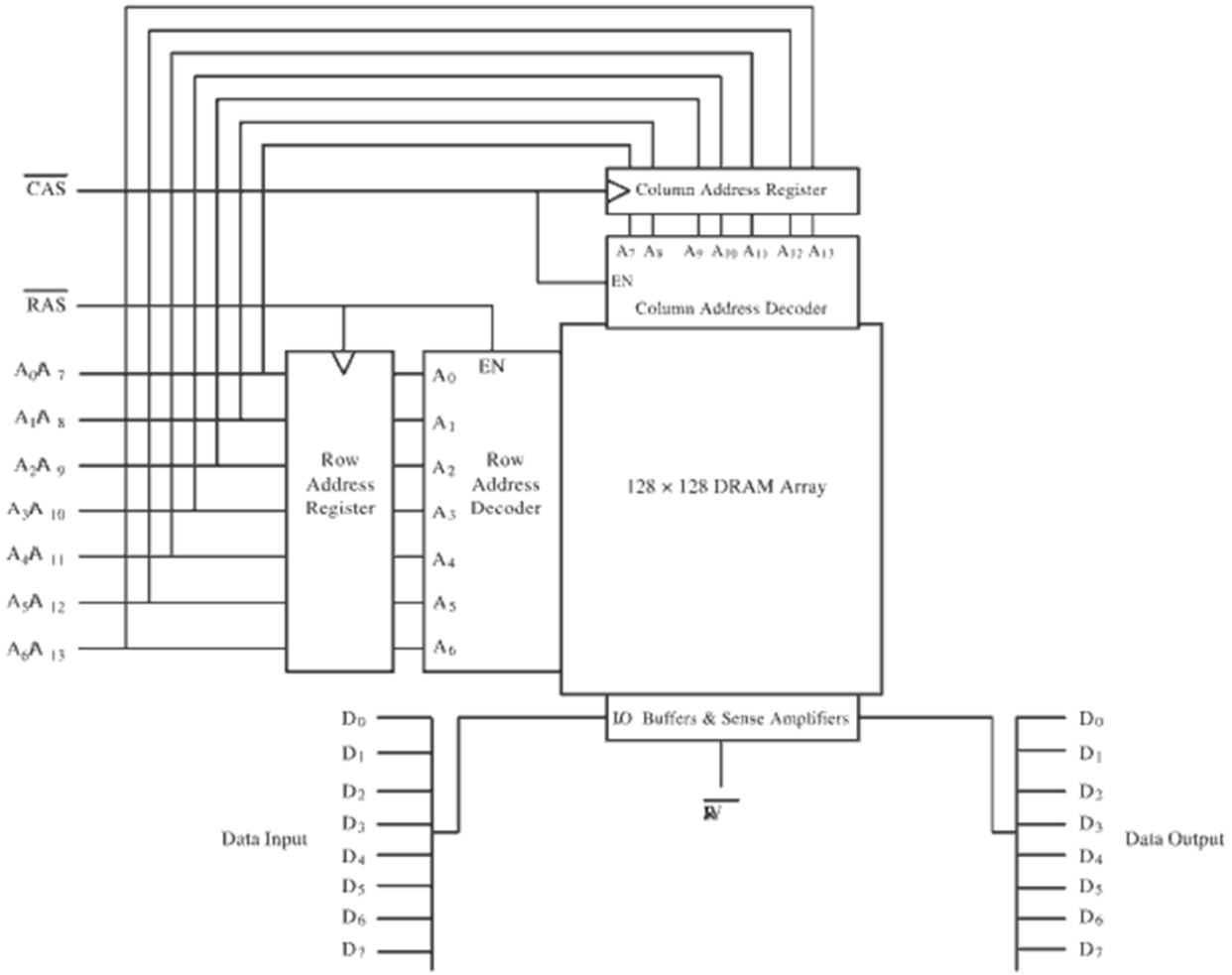


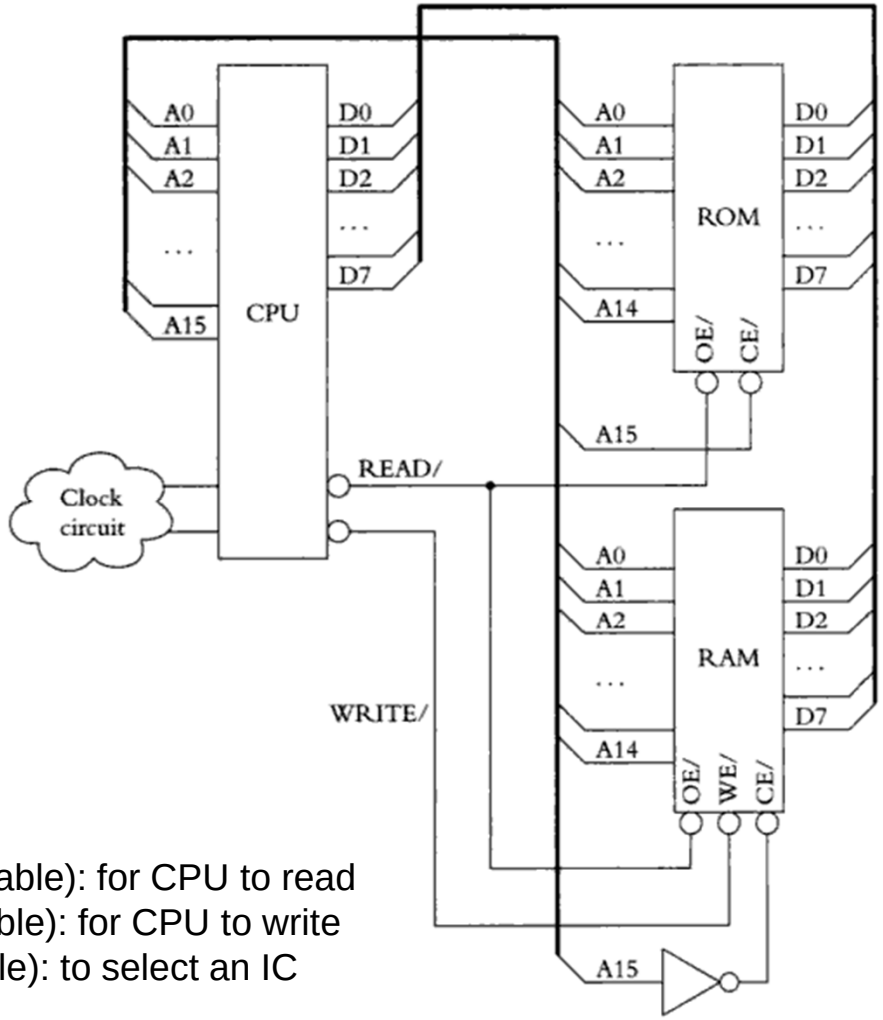
Figure 5-11b: 16K x 8 DRAM logic circuit [5-1]

Contents

1. Read Only Memory (ROM)
2. Random Access Memory (RAM)
3. Memory Interfacing
4. Flash Memory

Simple Memory Interfacing

- CPU + ROM + RAM
  - 64 KB address space
- Hooking up addr. and data buses
  - By using an inverter

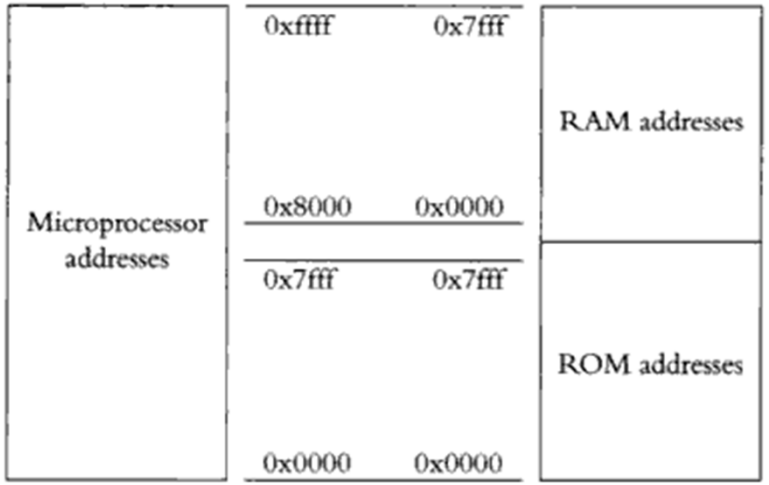


OE (Output Enable): for CPU to read
WE (Write Enable): for CPU to write
CE (Chip Enable): to select an IC

Simple Memory Interfacing

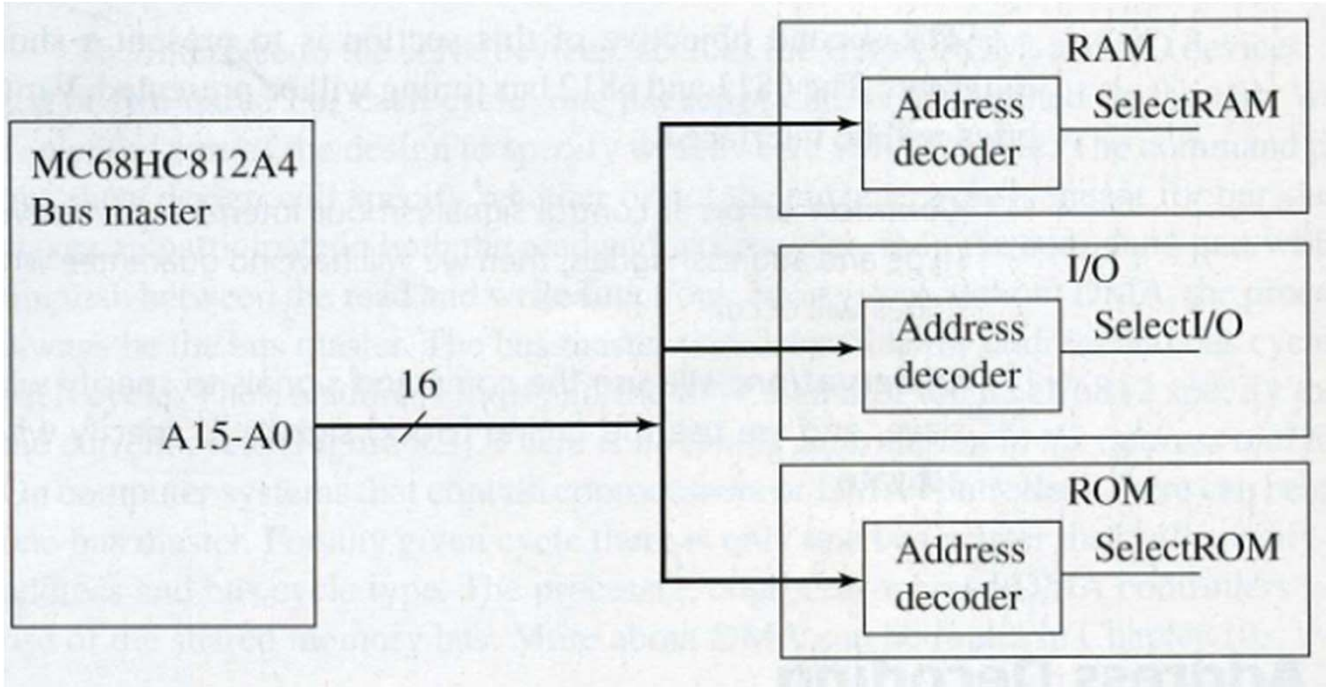
- When A15 signal is “1”
  - RAM is enabled
  - ROM is disabled
- When A15 signal is “0”
  - ROM is enabled
  - RAM is disabled

	Low Address	High Address
ROM	0x0000	0x7fff
	binary: 0000000000000000	binary: 0111111111111111
RAM	0x8000	0xffff
	binary: 1000000000000000	binary: 1111111111111111

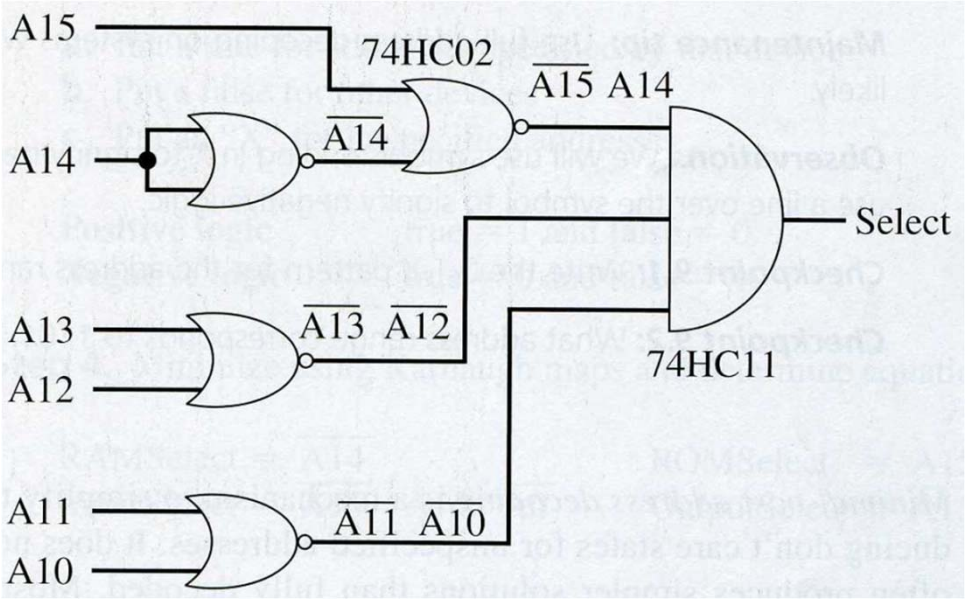


More Complex Memory Interfacing

- **Example**
  - **MC68HC812A4**
  - **Each slave device contains an address decoder**



An Example Address Decoder



Memory Test (During Initialization)

- **Test 1 Walking 1's Test** write & read 00000001 (first address), write&read 00000010 (second address) ...
  - Use a walking ones address pattern to test address bits
- **Test 2 Address Test**
  - This test will write each address location on memory with its own address once is finish will verify each of them for addressing errors
- **Test 3 All 1's and 0's Test**
  - Will use the cache memory to write patterns of all 1's and 0's until memory is full and will detect obvious errors
- **Test 4 8-Bit Pattern Test**
  - Similar to test 1 but instead of moving 1 bit pattern it will move an eight bit pattern

Memory Test (During Initialization)

- Test 5 Random Data Test
  - This test will set a random number as a pattern for walking 1's and 0's, contrary to test 1 or 4
  - By selecting a different width in the pattern, the test will find more complex or hard to detect errors on memory
- Test 6 Block move
  - Memory it segmented in 4 Mb blocks and moved with instruction
  - Once finish it will verify the data patterns written on each block for consistency
- Test 7 32 Bit Shifting Data Test
  - Using an algorithm that changes the data pattern one bit to the left on every pass or loop
  - A total of 32 passes will use all data patterns but the downside is the time required for
- Test 7 Random Data Sequence Test
  - Random numbers are written in sequence into all memory banks
  - Then, using a key for the random numbers duplicates the sequence written in memory which is compared with the original sequence

Walking 1s and 0s Tests

```
/* Walking 1s -- 00000001, 00000010,
   00000100, 00001000, etc. */
  data = 0x01;
  for(data = 1; data <= 128; data <<= 1)
   {
    Write the data
    Read the data
    if(data_in != data_out)  test fails
  }

/* Walking 0s -- 11111110, 11111101,
   11111011, 11110111, etc. */
  data = 0xFE;
  for(data = 254; data >= 127; data =
   ((data << 1) +1)) {
    Write the data
    Read the data
    if(data_in != data_out)  test fails
  }
```

Contents

1. Read Only Memory (ROM)
2. Random Access Memory (RAM)
3. Flash Memory

Flash Memory

- Flash memory is a specific type of EEPROM that is erased and programmed in large blocks
  - In early flash the entire chip had to be erased at once
  - Flash memory costs far less than byte-programmable EEPROM and therefore has become the dominant technology wherever a significant amount of non-volatile storage is needed

- Flash memory (both NOR and NAND types) was invented by Dr. Fujio Masuoka while working for Toshiba in 1984

i-phone 이후 폭발적인 인기 얻음
mp3 에서 비휘발성인 메모리가 필요했다. 초기 작은 hard disk 사용했는데, 1. hard disk의 경우 소비전력이 많다 .
why? 전기 에너지를 물리 에너지로 변경해야 하기 때문에, 2. mp3를 떨어트릴경우 head가 platter를 scratch할 수 있다.
flash memory - 오류가 많다. wear out 가능 (write limit 존재), rom이기 때문에 block 단위로 무조건 write 해야 한다.
i-phone 시리즈는 flash memory의 소재 상의 한계를 소프트웨어적으로 극복

Characteristics

❒ Advantages

- **Flash memory offers fast read access times**
  - Although not as fast as volatile DRAM memory
- **Better kinetic shock resistance than hard disks**
- **Low power consumption**
- **Small size and light weight**

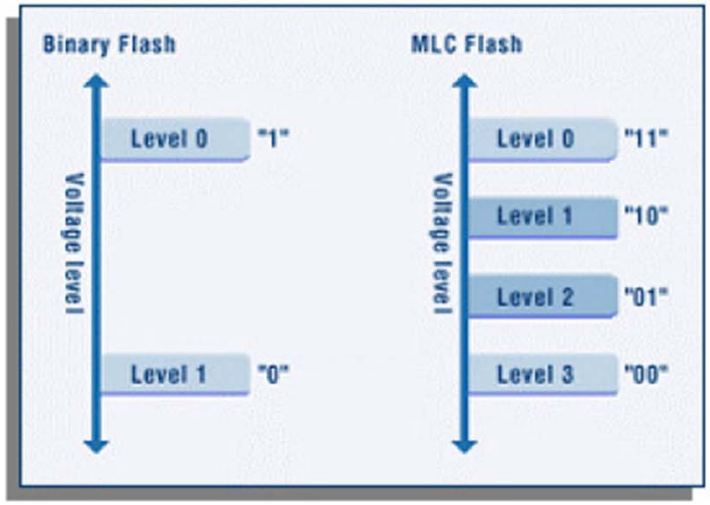
❒ Disadvantages

- **Overwrite operation can be performed only after erase operation**
- **The number of erase operations is limited**
- **Speed asymmetry** Dram - write/read 속도 대칭적이다. (동일하다)
  - Erase (100X) > write (10X) > Read (1X)
- **Size asymmetry**
  - NOR: read two bytes, write and erase a block (16KB or 128KB)
  - NAND: read and write a page (512 or 2048 bytes), erase a block

SLC and MLC

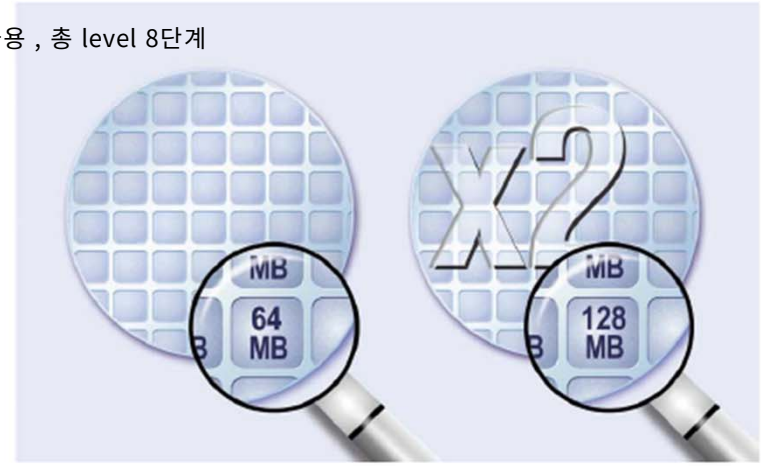
- Flash memory stores information in an array of memory cells made from floating-gate transistors
- In traditional single-level cell (SLC) devices, each cell stores only one bit of information
- Some newer flash memory, known as multi-level cell (MLC) devices, can store more than one bit per cell by choosing between multiple levels of electrical charge to apply to the floating gates of its cells

SLC and MLC



전위차 구분 level 2단계 전위차 구분 level 4단계

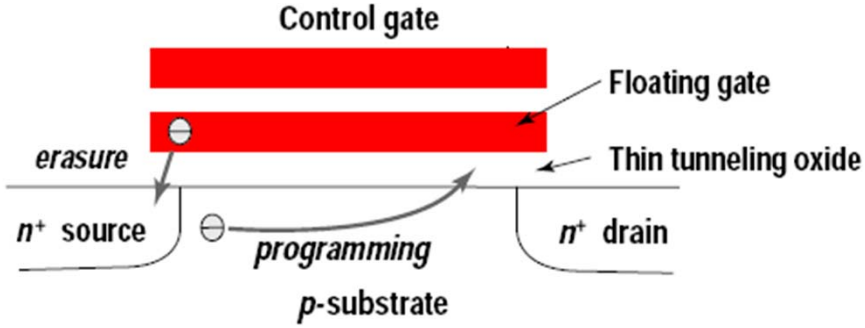
TLC
3비트를 사용 , 총 level 8단계



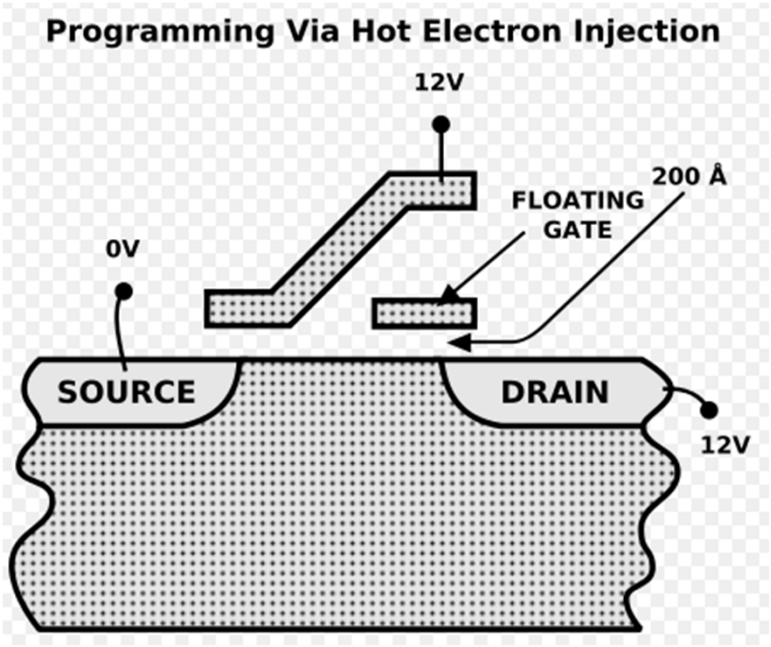
slc - 오류 적다, 직접도 안좋다
mlc - 오류 많다. 직접도 좋다

NOR Flash

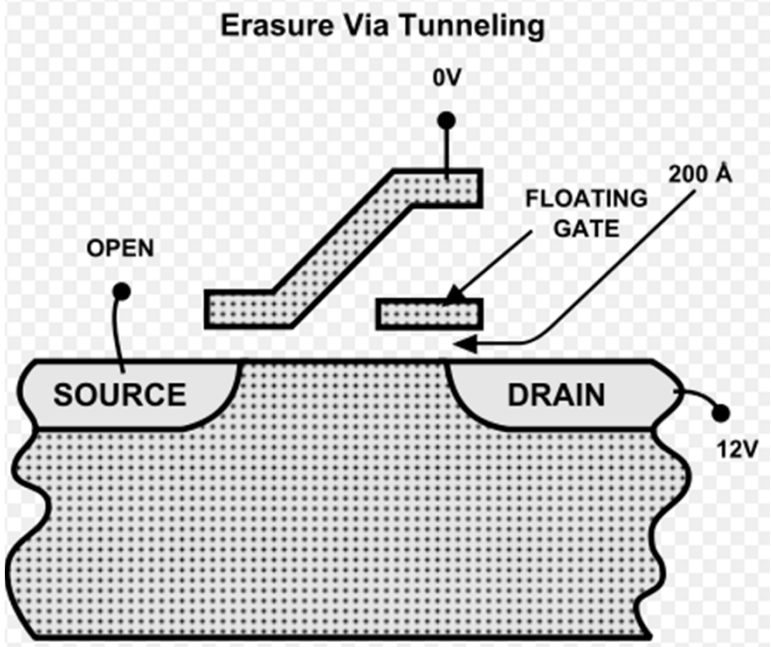
- NOR-based flash has long erase and write times, but provides full address and data buses, allowing random access to any memory location
  - A suitable replacement for older ROM chips, which are used to store program code that rarely needs to be updated
  - Its endurance is 10,000 to 1,000,000 erase cycles
  - NOR-based flash was the basis of early flash-based removable media; CompactFlash was originally based on it, though later cards moved to less expensive NAND flash



NOR Flash Memory Cell

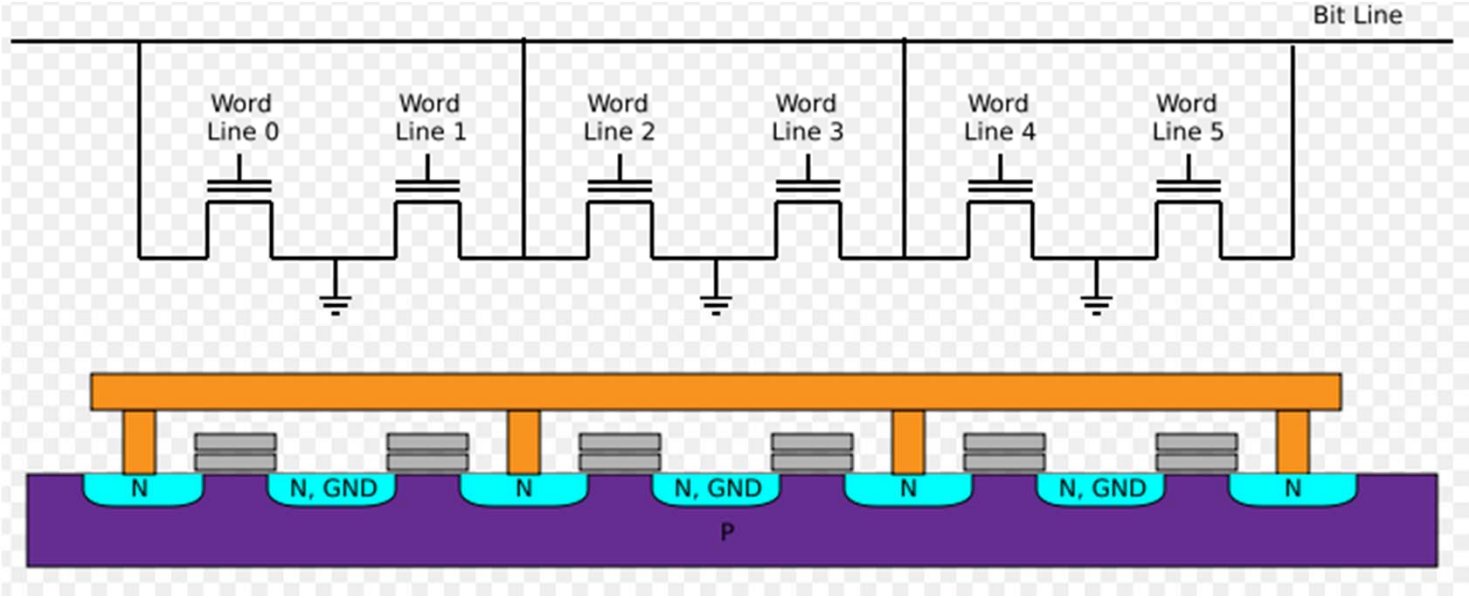


Writing

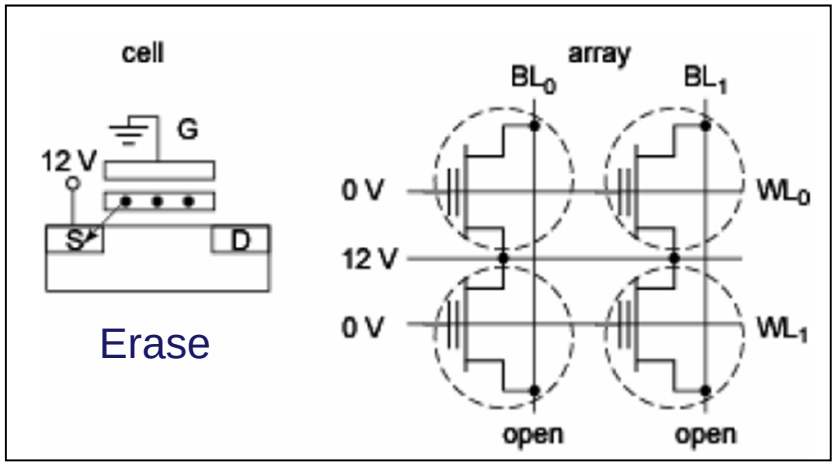


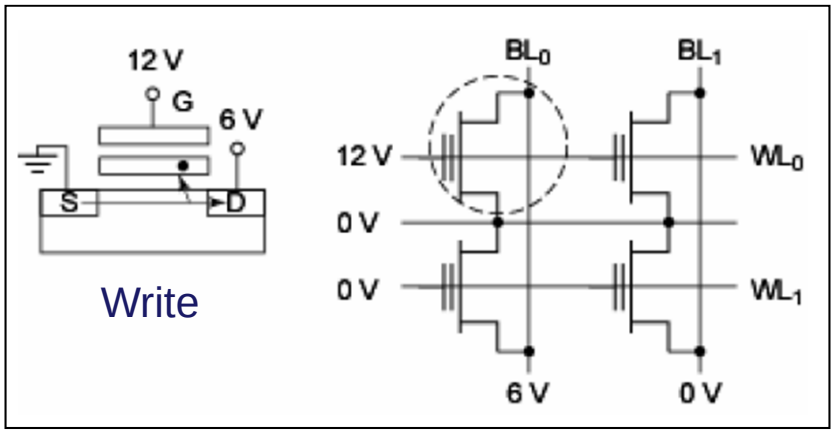
Erasing

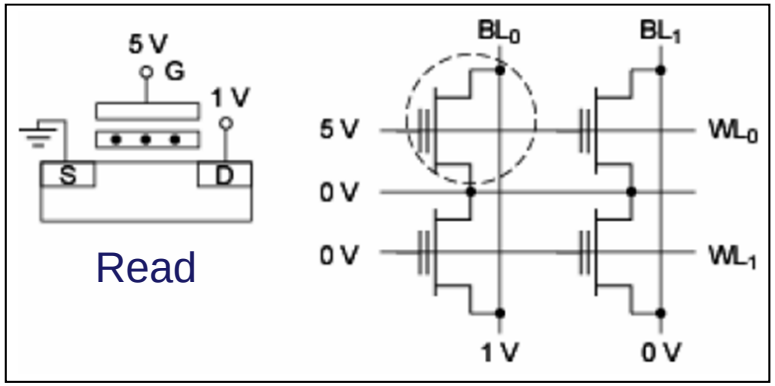
NOR Flash Memory Layout



Erase, Write, and READ in NOR Flash





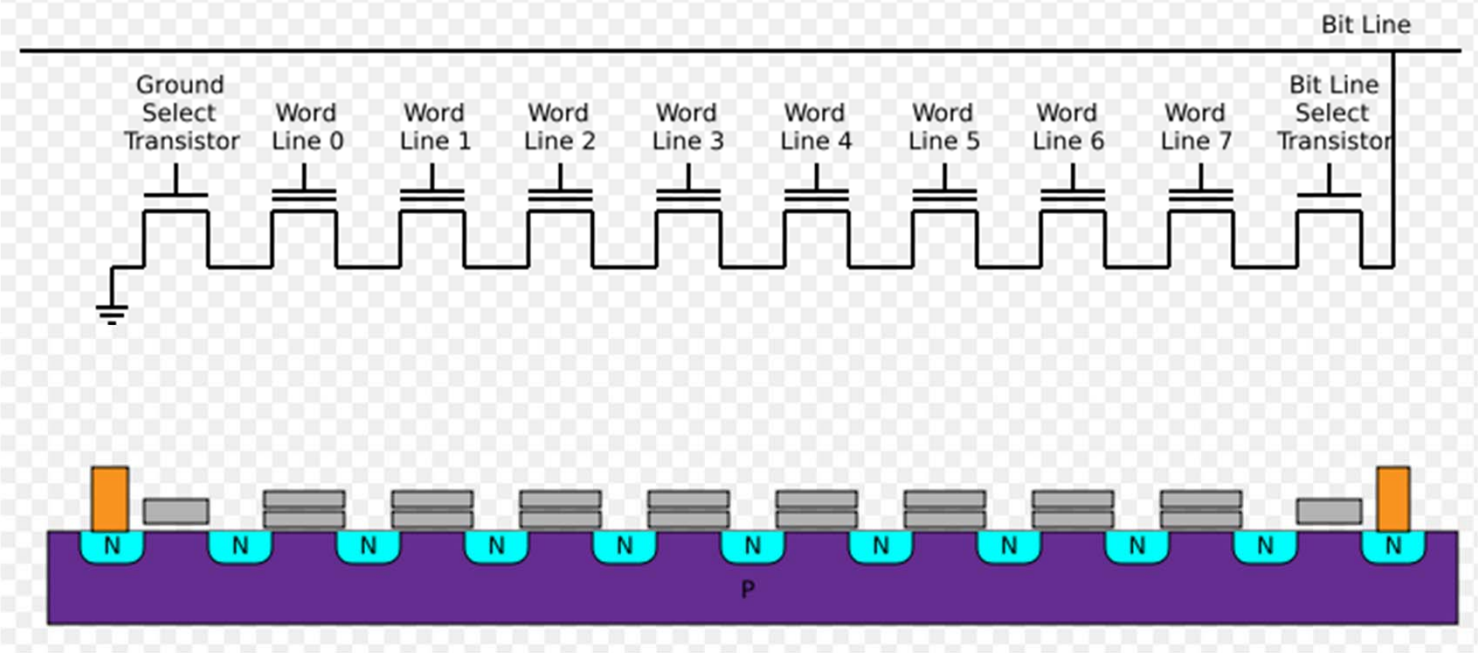


NAND Flash

- NAND flash has faster erase and write times, and requires a smaller chip area per cell, thus allowing greater storage densities and lower costs per bit than NOR flash
  - It also has up to ten times the endurance of NOR flash
  - Toshiba announced NAND flash at ISSCC in 1989

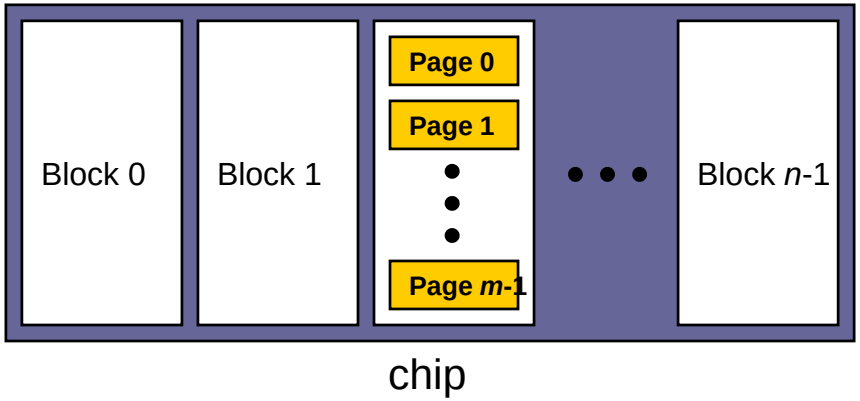
- However, the I/O interface of NAND flash does not provide a random-access external address bus
  - Rather, data must be read on a block-wise basis, with typical block sizes of hundreds to thousands of bits
  - This made NAND flash unsuitable as a drop-in replacement for program ROM since most microprocessors require byte-level random access
  - In this regard NAND flash is similar to other secondary storage devices such as hard disks and optical media

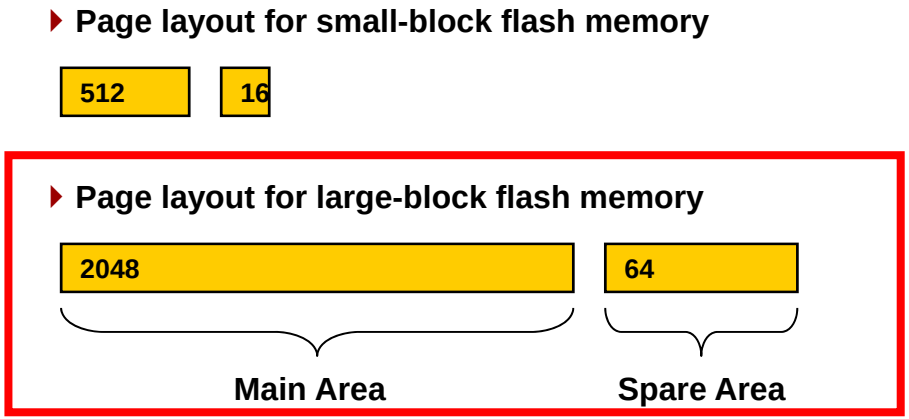
NAND Flash Memory Layout



Organization of NAND Flash memory

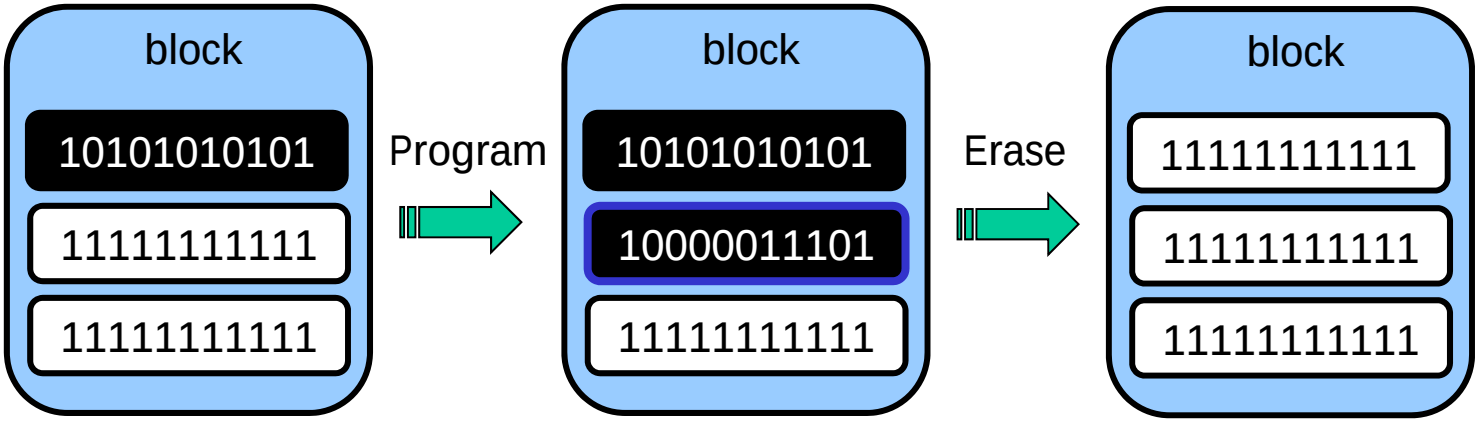
- Small-block flash memory
  - Each page is (512 + 16) bytes
  - 32 pages in each block (16KB block)
- Large-block flash memory
  - Each page is (2048 + 64) bytes
  - 64 pages in each block (128KB Block)





Operations and Data Units

- **Erase**
  - Erase unit is block (16KB or 128KB)
  - Erase operation sets all bits “1”
- **Read and write**
  - Read/write unit is page (512 bytes or 2048 bytes)
  - Write operation sets some bits “0”



- **ECC and some metadata in spare area (16 bytes)**

NOR vs. NAND

- NOR flash is very similar to a random access memory device (RAM)
  - It has enough address pins to map its entire media, allowing for easy access to each and every one of its bytes (memory mapped)
- NAND devices are interfaced serially via a rather complicated I/O interface, which may vary from one device to another or from vendor to vendor
  - The same eight pins convey control, address and data information (I/O mapped)
  - NAND is typically accessed in bursts of bytes; i.e., 512 bytes can be read and written at a time (similar to hard drives)
- Therefore, NOR is ideal for running code while NAND is best used as a data storage device

NOR vs. NAND

- Due to the efficient architecture of NAND flash, its cell size is almost half the size of a NOR cell
  - This, in combination with a simpler production process, enables NAND architecture to offer higher densities, with more capacity on a given die size

- A flash block must be erased before writing to it
  - But the number of times that it can be erased is limited
  - NAND devices offer up to 10 times the life span of NOR devices
  - In fact, since the block size of a NAND device is usually about 8 times smaller than that of a NOR device, each NOR block will be erased relatively more times over a given period of time (especially significant when working with small files) than each NAND block, which further extends the gap in favor of NAND

NOR vs. NAND

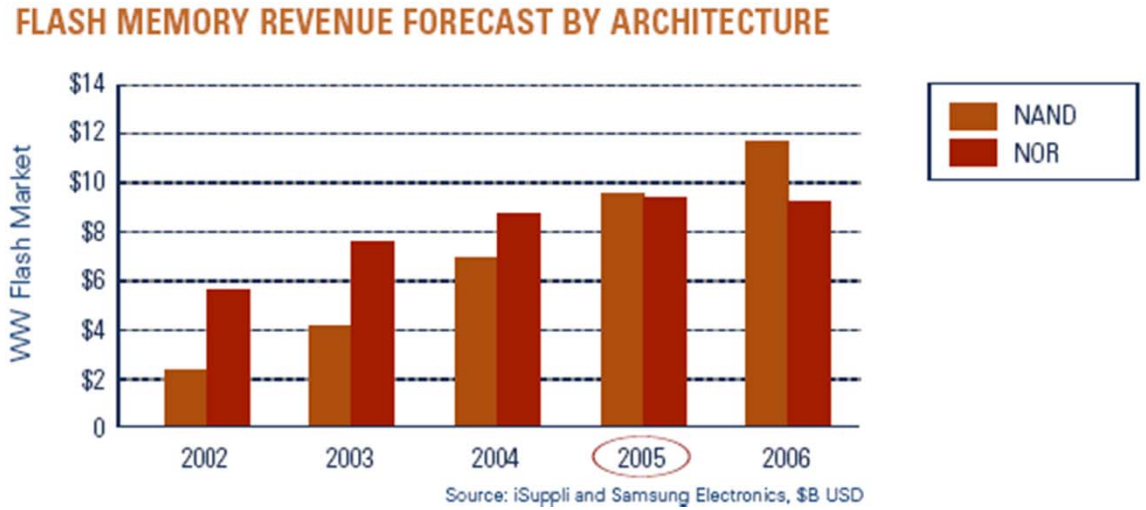
Media	Read	Write	Erase
SDRAM (PC-100)	60ns (2B) 2.56us (512B)	60ns (2B) 2.56us (512B)	N/A
NOR flash (Intel 28F128J3A-150)	150ns (2B) 14.4us (512B)	211us (2B) 3.53ms (512B)	1.2s (128KB)
NAND flash (Samsung K9F5608U0M)	10.2us (2B) 35.9us (512B)	201us (2B) 226us (512B)	2ms (16KB, 128K)
Disk (Segate Barracuda ATA II)	12.5ms (512B) (Average seek)	14.5ms (512B) (Average seek)	N/A

- **General performance**
  - Read: NOR > NAND
  - Write and erase: NAND > NOR
- **Cost per GB**
  - HDD<<NAND<DRAM<NOR

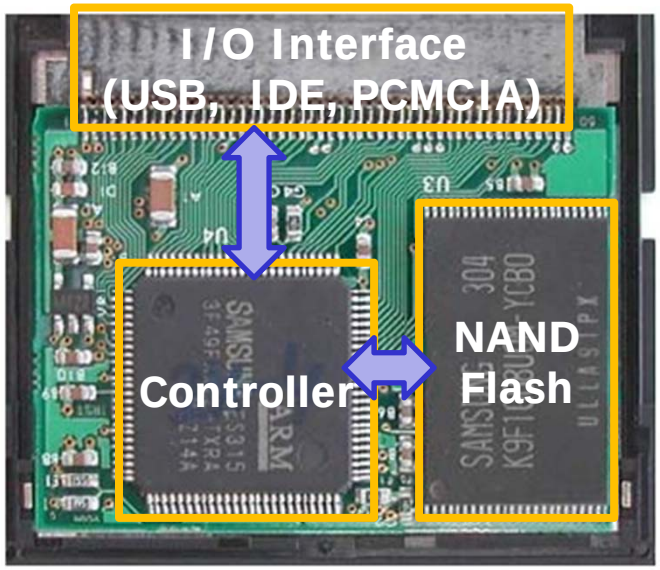
Market Trends

Rank 2005	Rank 2004	Worldwide Supplier	Revenue 2005	Revenue 2004	Market Share	Growth 2005
1	1	Samsung	$5,742	$3,901	52.9%	47.2%
2	2	Toshiba	$2,382	$1,850	21.9%	28.8%
3	4	Hynix	$1,382	$221	12.7%	525%
4	3	Renesas	$735	$600	6.8%	22.5%
5	7	Micron Technology	$238	$8	2.2%	2875%

iSuppli/$M



Typical Flash Storage

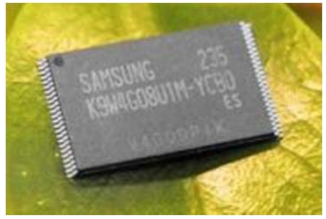


Hybrid Disk

- HDD with NV buffer (NAND flash memory)
  - Flash cost is low ($8~9 for 1GB in 2007)
- Low-power consumption
- Fast start-up

HDD

+

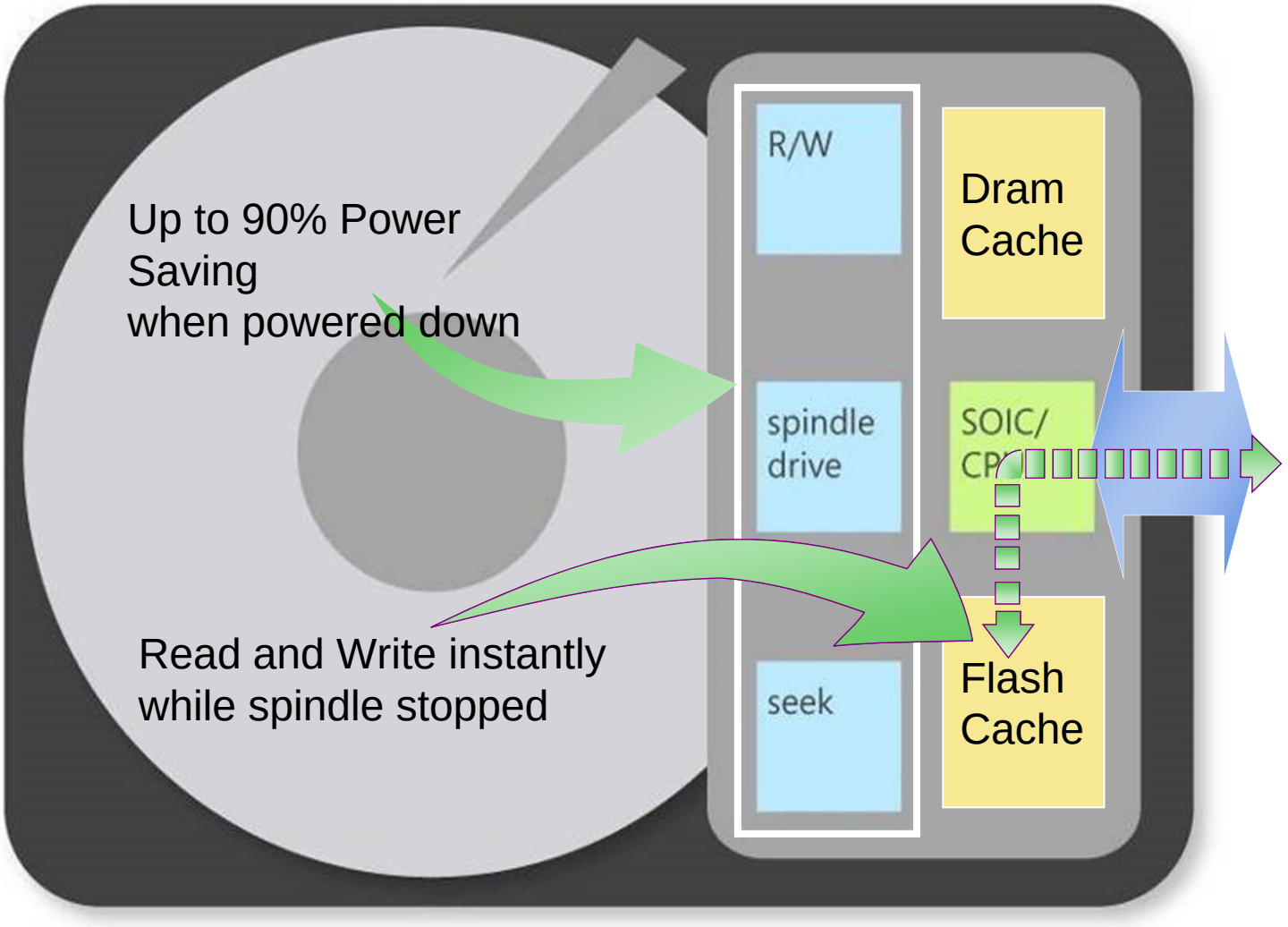


NAND Flash

=

HDD with Low-power consumption and fast start-up time

Hybrid Disk


Up to 90% Power Saving
when powered down
Read and Write instantly
while spindle stopped
R/W
spindle drive
seek
Dram Cache
SOIC/
Flash Cache

Solid State Drive

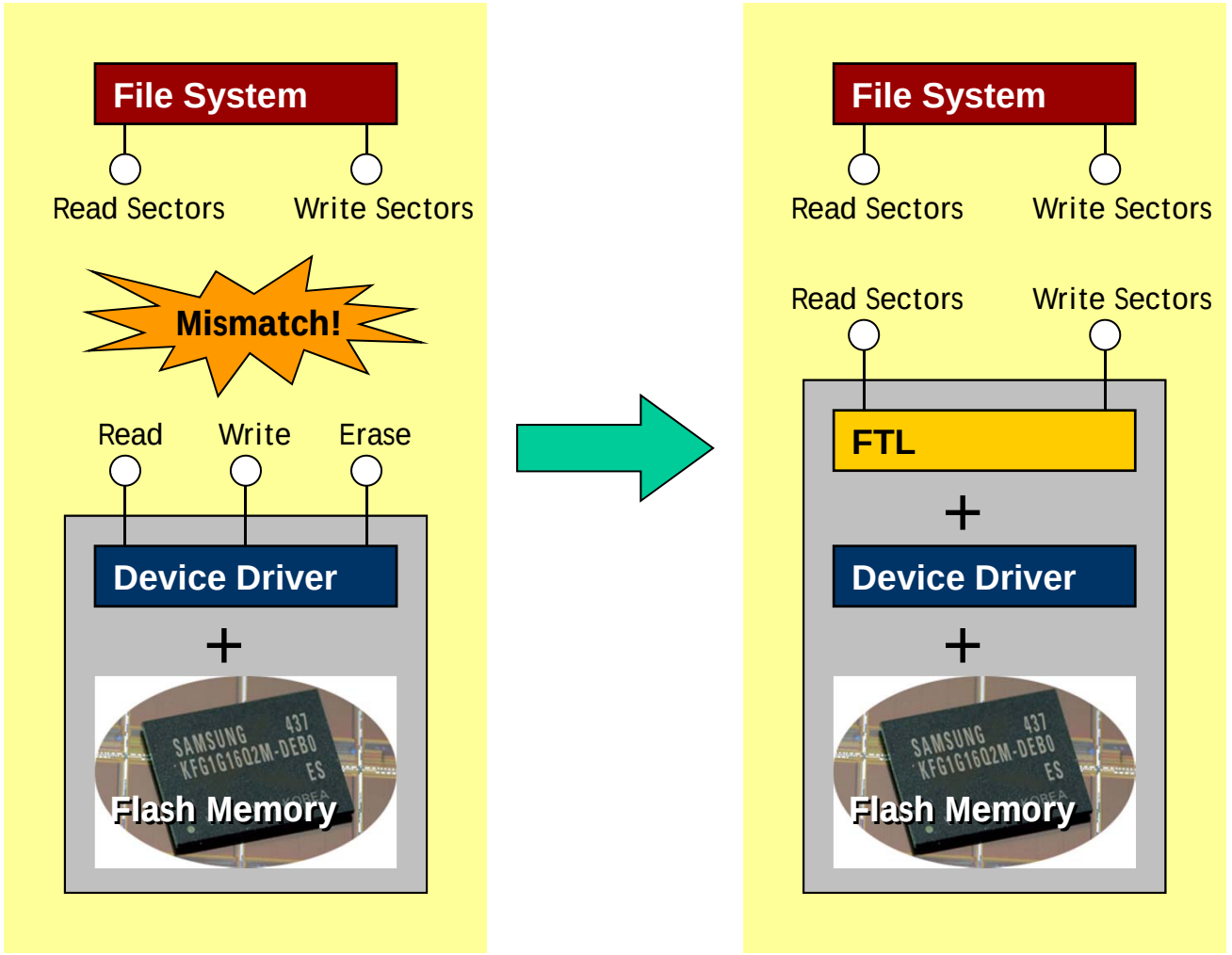
- A solid state drive (SSD) is a data storage device that uses solid-state memory to store persistent data
  - An SSD emulates a hard disk drive, thus easily replacing it in any application
  - An SSD using SRAM or DRAM (instead of flash memory) is often called a RAM-drive

HDD

SSD (bottom)

SSD (top)

Flash Translation Layer (FTL)

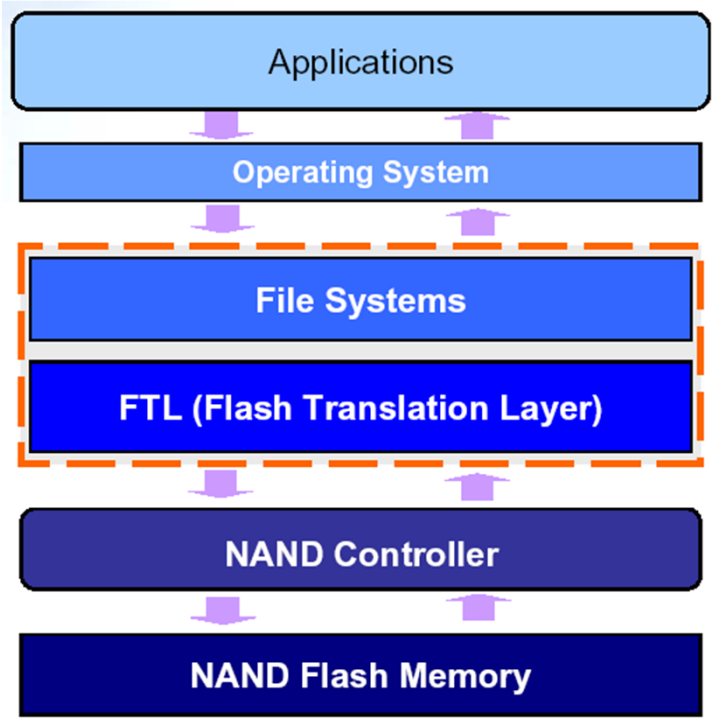


Flash Translation Layer (FTL)

❒ Flash Translation Layer (FTL)

- **A software layer emulating standard block device interface**
  - Read/Write

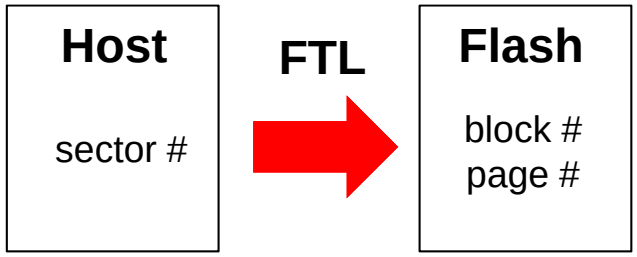
- **Features**
  - Sector mapping
  - Garbage collection
  - Power-off recovery
  - Bad block management
  - Wear-leveling
  - Error correction code (ECC)



SPOR(sudden power-off recovery)
정상적인 전원키를 이용한 시스템 종료가 아닌 갑자기 배터리를 제거햇을 때를 고려하는 것으로 상당한 시스템 부하가 걸리는 작업.
이전과 같은 시스템 상태를 recovery 하는 작업

Simplest Address Translation at FTL

- Host (device driver) generates
  - sector number “A (= 1000)”

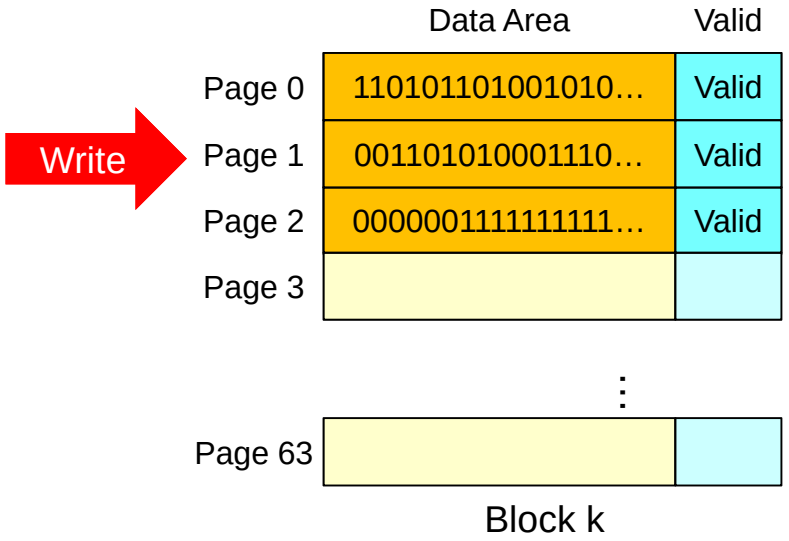


- FTL translates it into
  - block number “B (= 10)”
  - page number “C (= 232)”

- If a sector is 512 bytes, a page is 2048 bytes, and a block has 64 pages
  - block no. = sector no. / 256 (= 1000/256 = 3, quotient)
  - page no. = ~~sector no. % 256~~ (= 1000%256 = 232, remainder)
    (sector/4) % 64

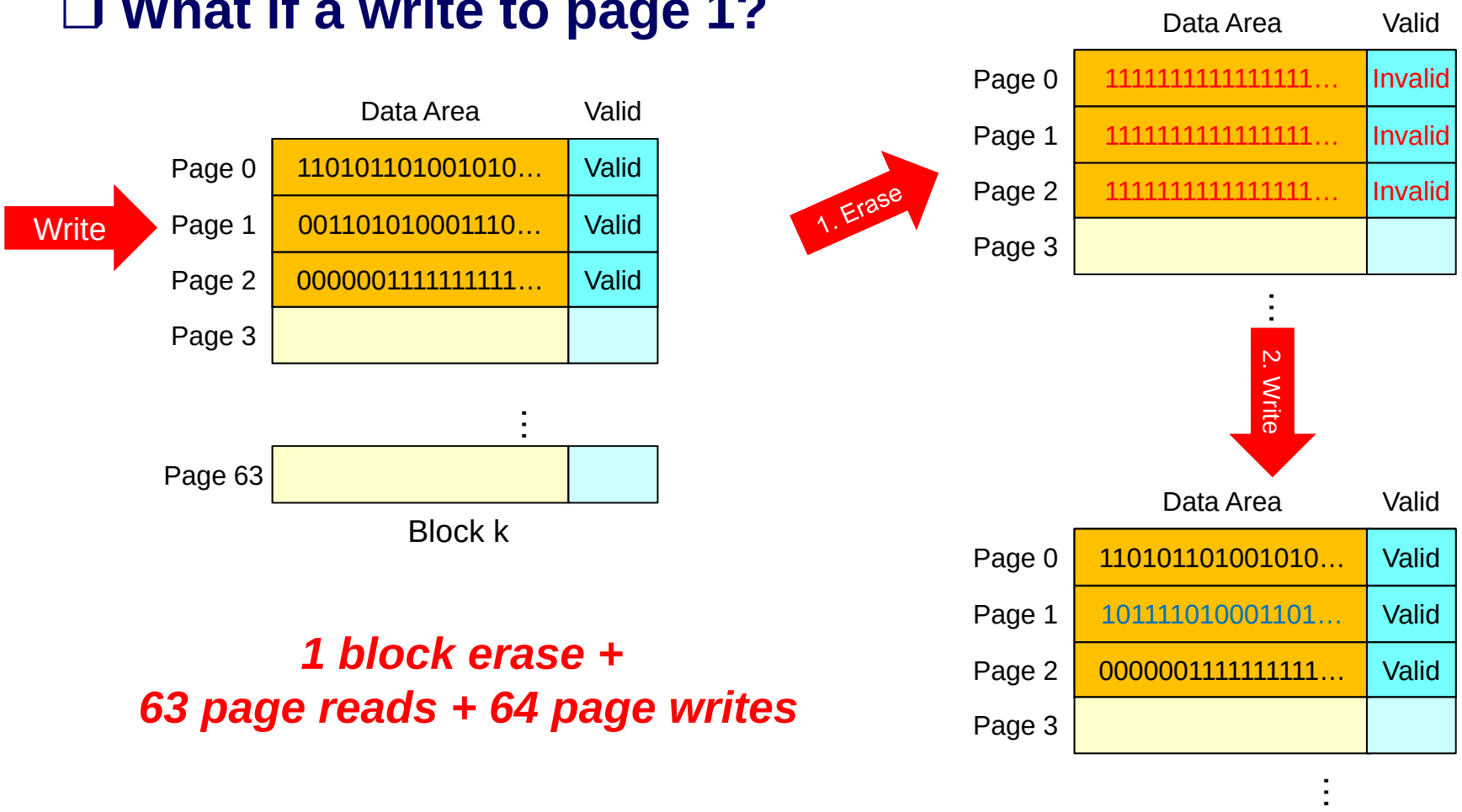
Simplest Address Translation at FTL

❒ What if a write to page 1?



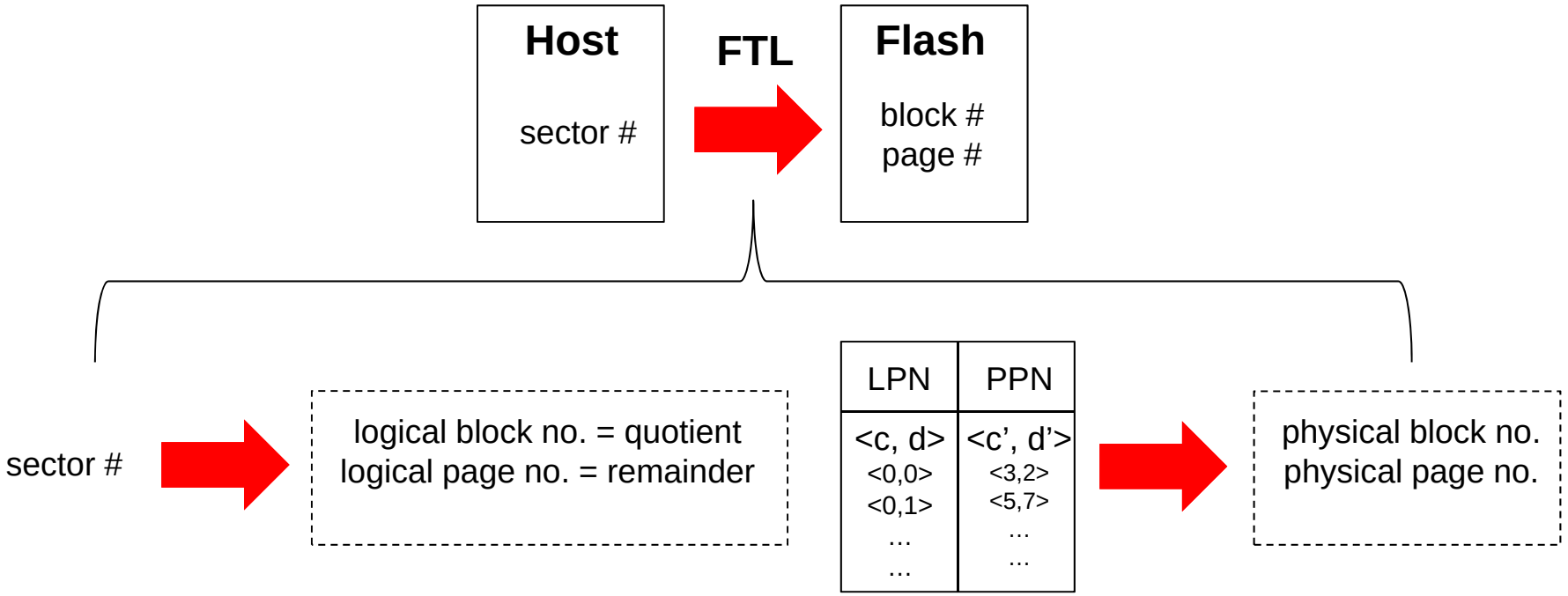
Simplest Address Translation at FTL

❒ What if a write to page 1?



1 block erase +
63 page reads + 64 page writes

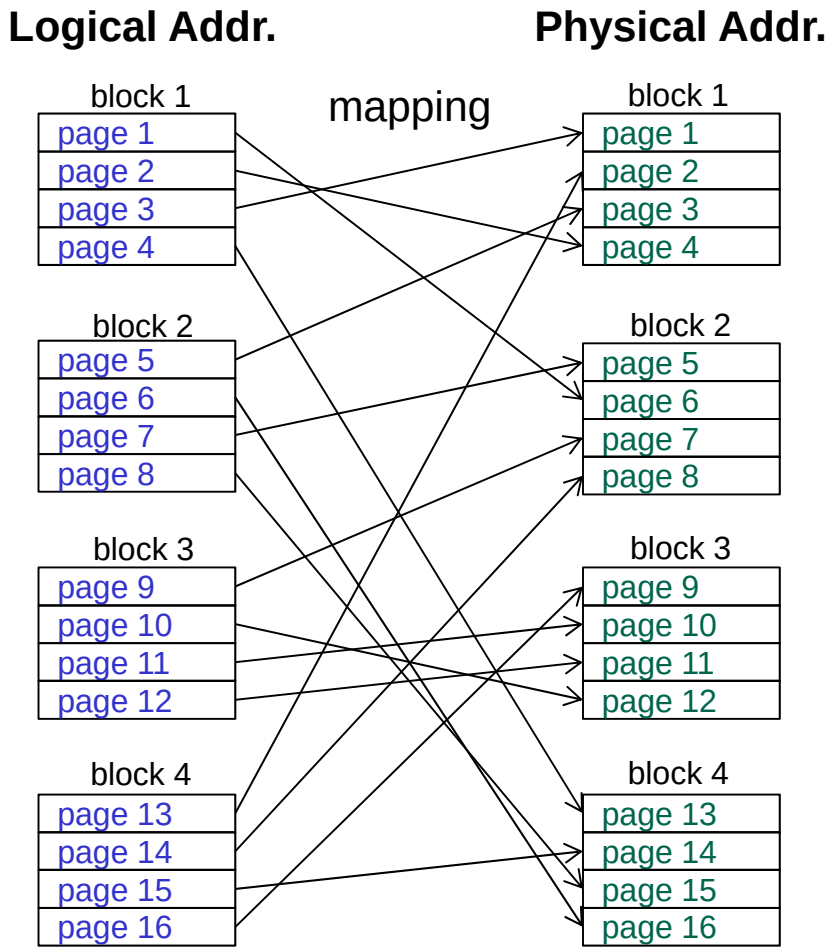
Logical Pages and Physical Pages and Page-Level Address Mapping



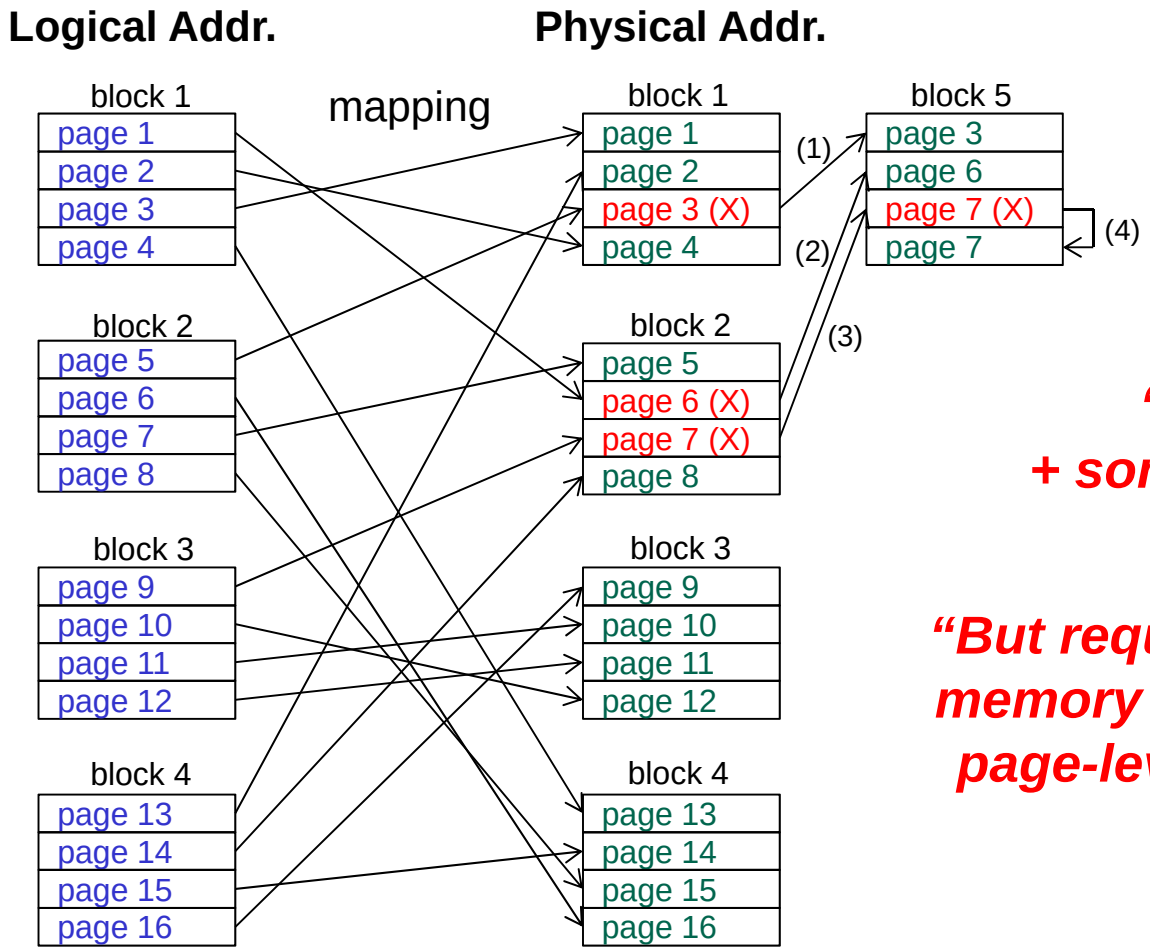
LPN: Logical Page Number = <block no., page no.>

PPN: Physical Page Number = <block no., page no.>

Page-Level Address Mapping



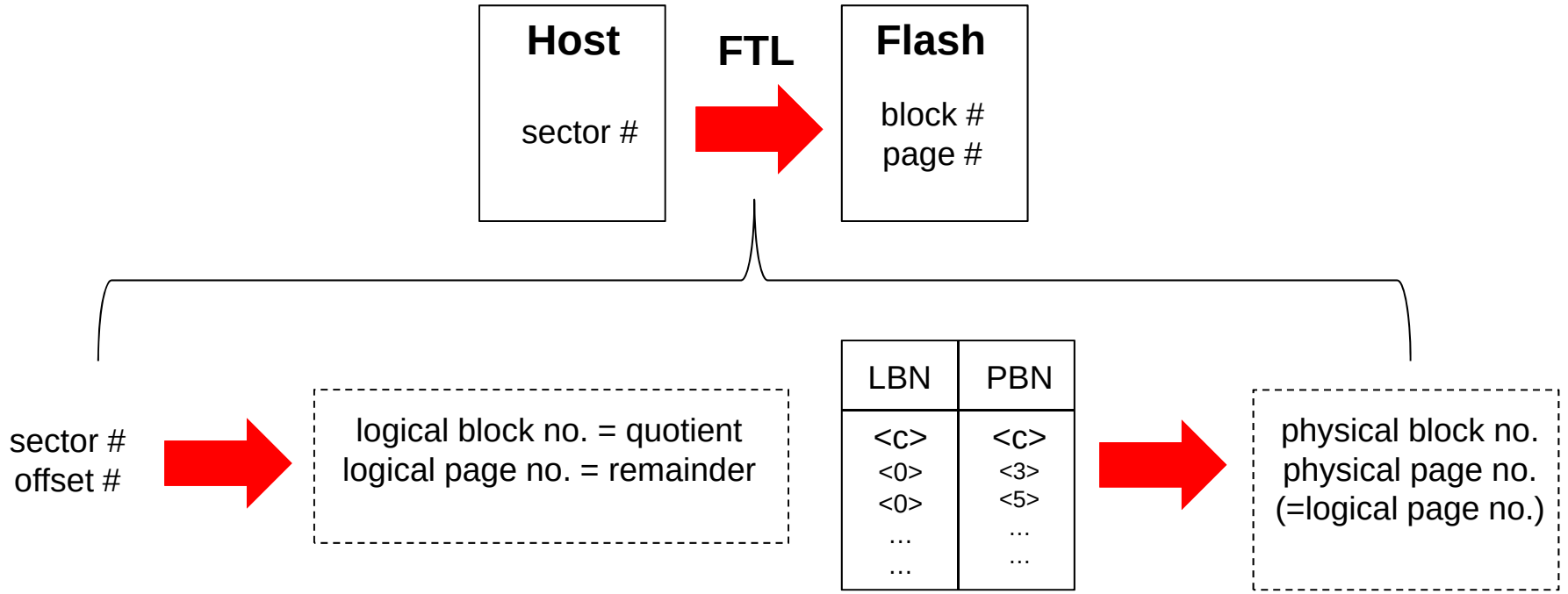
Page-Level Address Mapping with Overwrites <3, 6, 7, 7>



"1 page write + some bookkeeping"

"But requires a large amount of memory space for maintaining page-level address mapping"

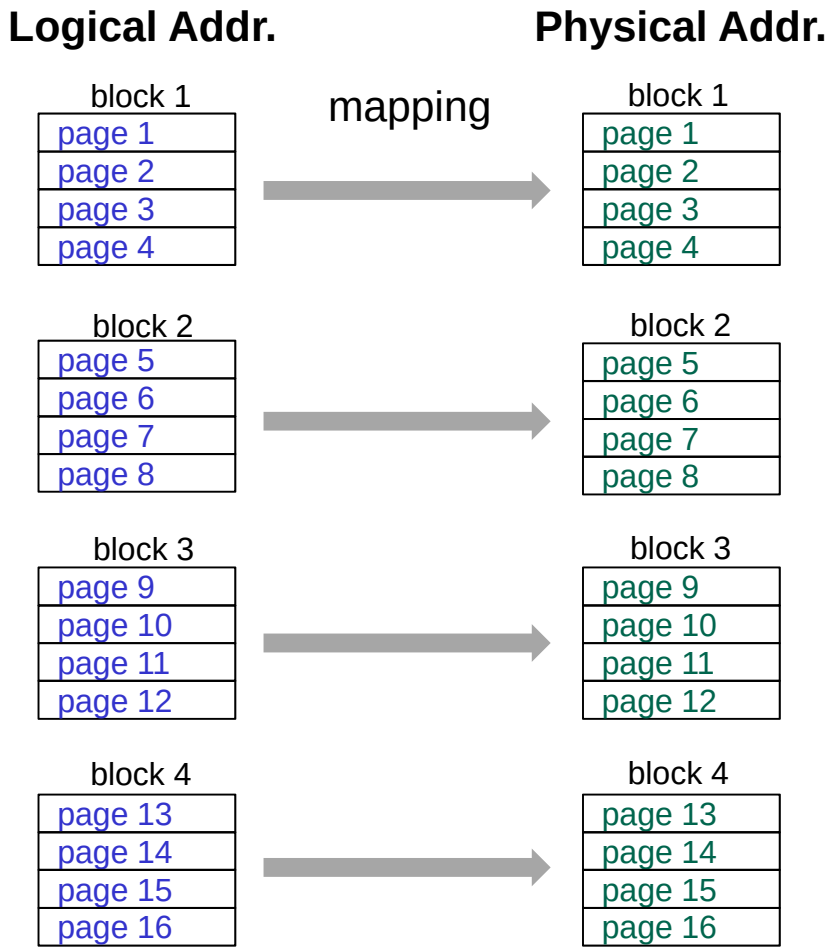
Block-Level Address Mapping



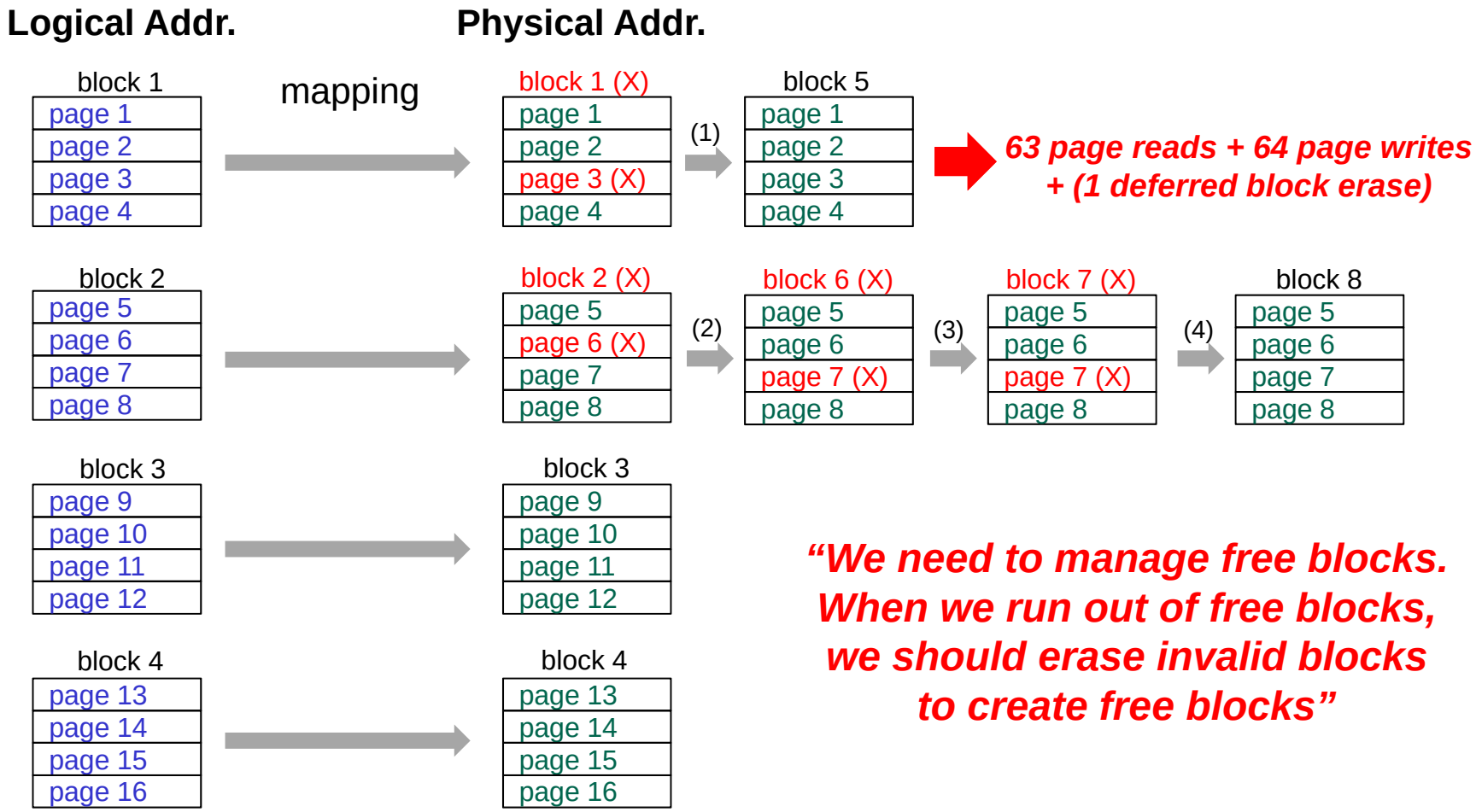
LBN: Logical Block Number = <block no.>

PBN: Physical Block Number = <block no. >

Block-Level Mapping

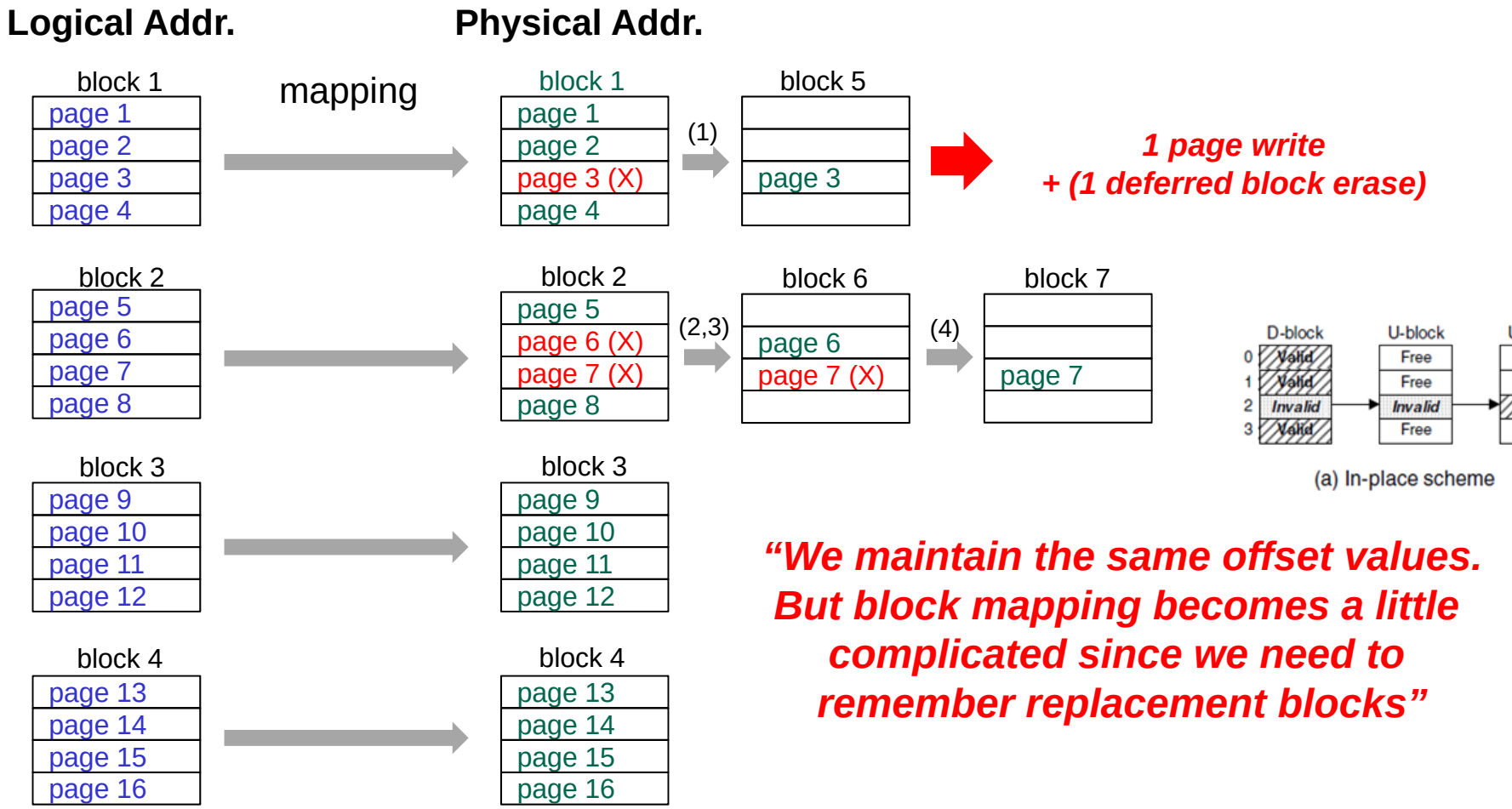


Block-Level Mapping: Vanilla with Overwrites <3, 6, 7, 7>



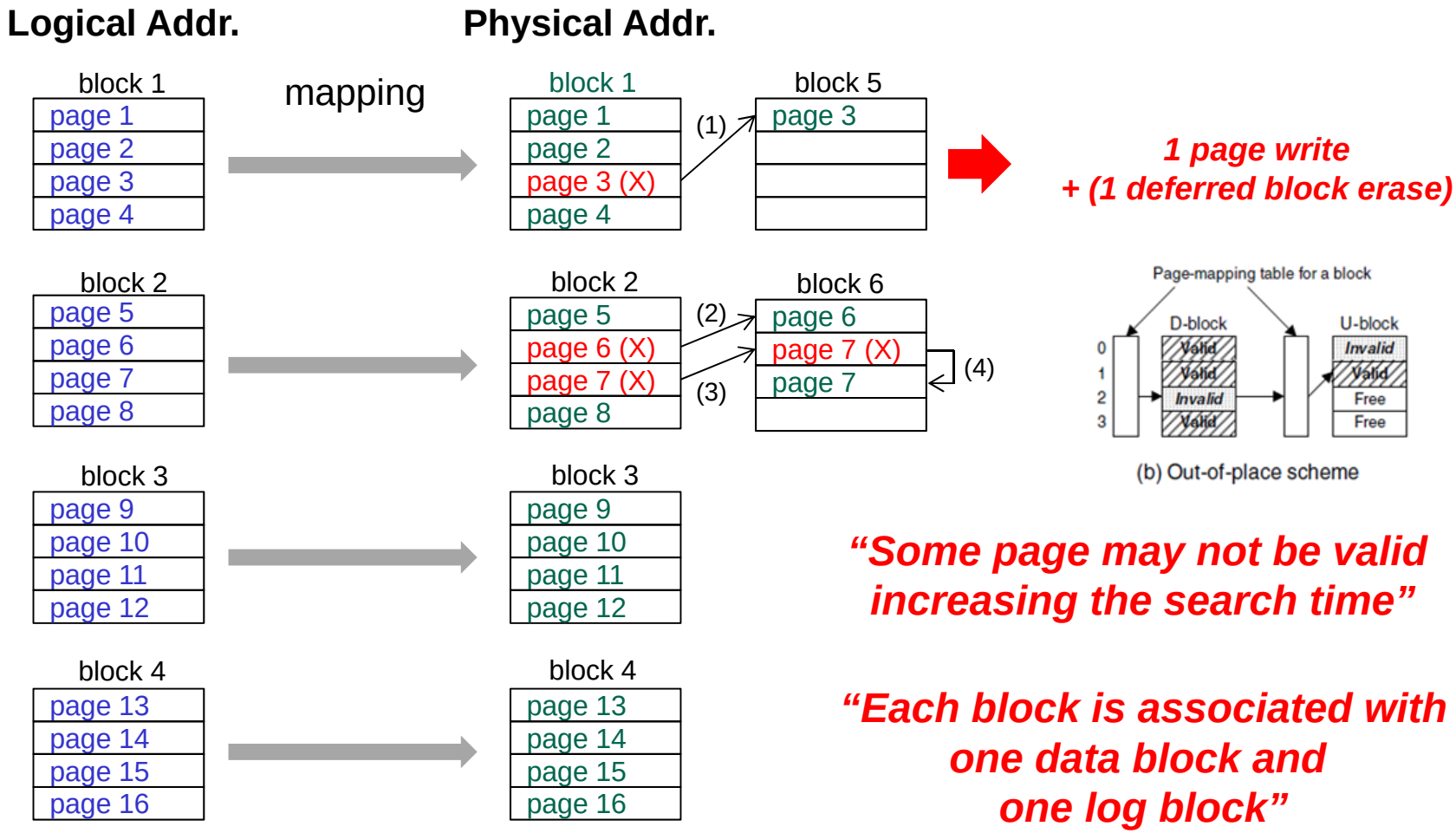
"We need to manage free blocks. When we run out of free blocks, we should erase invalid blocks to create free blocks"

Block Level Mapping: Replacement Block with Overwrites <3, 6, 7, 7>



"We maintain the same offset values. But block mapping becomes a little complicated since we need to remember replacement blocks"

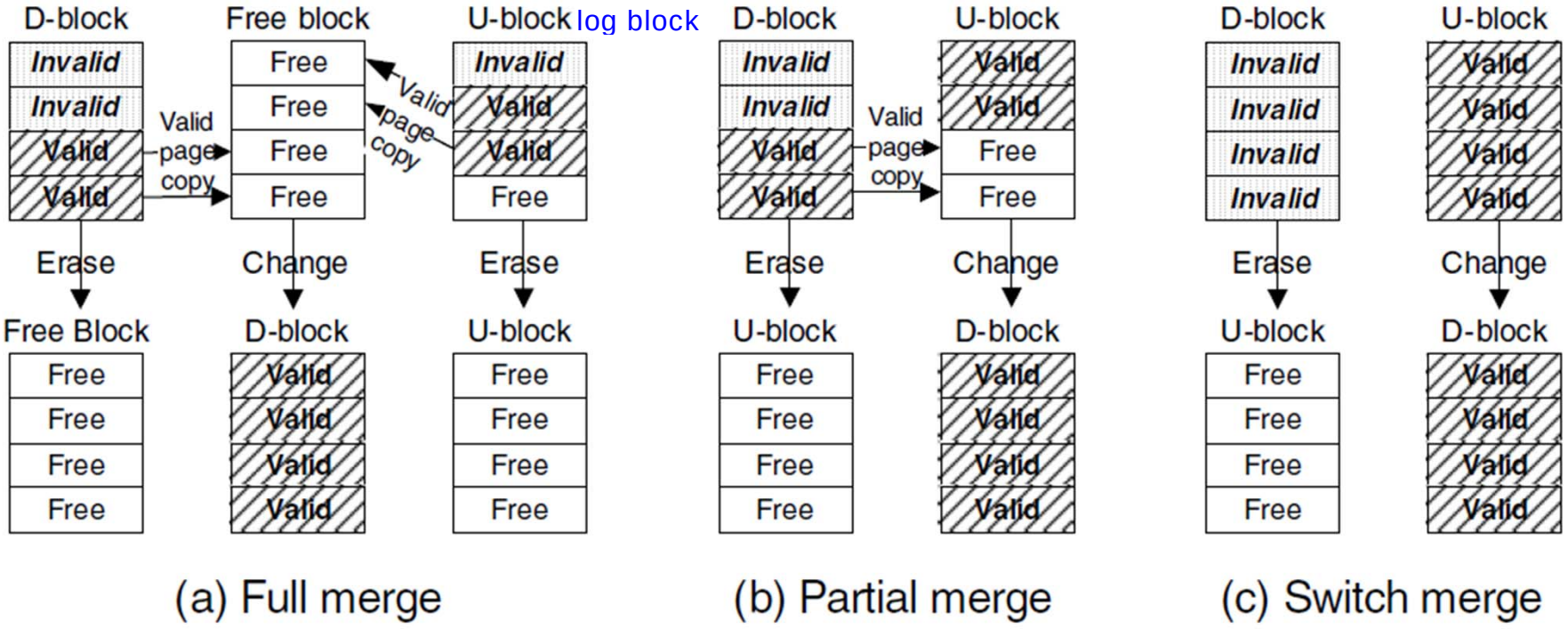
Log Block Scheme (BAST, 2002) with Overwrites <3, 6, 7, 7>



Garbage Collection in Block Level Mapping

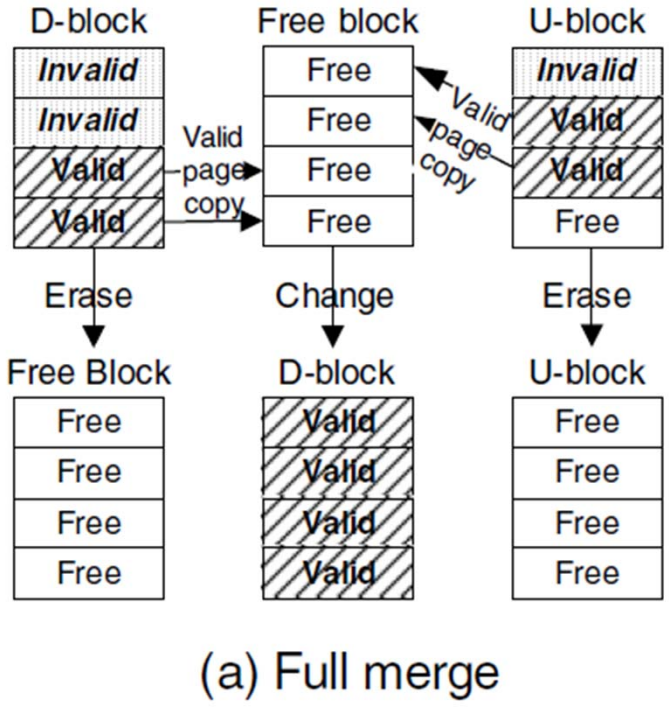
❒ Goal: create free blocks

Log Block Scheme은 쓰기요청이 지속되었을 경우 Log Block이 소진되기 전까지는 별도의 Block-copy가 이루어지지 않는다. 하지만 Log Block이 모두 소진되었을 경우 이를 다시 반환하여야 하는데 이를 위해 Log block과 이에 대응하는 Data block의 데이터가 서로 병합되어야 한다



Full Merge

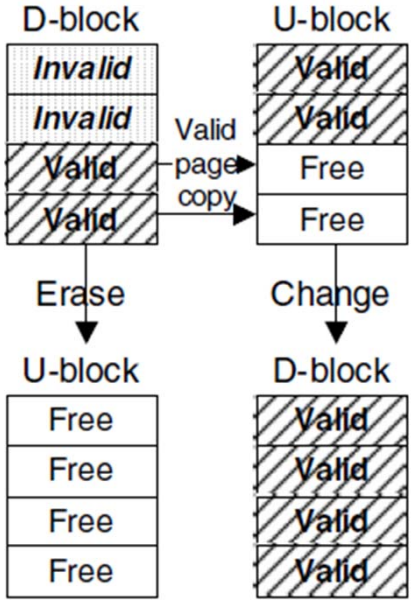
- 1 free block is required
- Cost of full merge
  - "64" page copy operations
    - "n" page reads
    - "n" page writes
  - 2 block erase operations
- 2 free blocks are created



(a) Full merge

Partial Merge

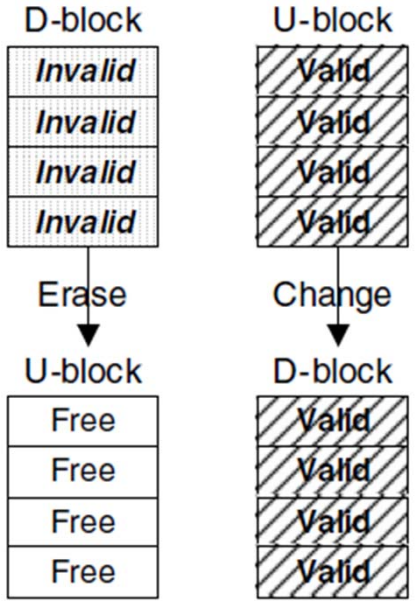
- **No free block is required**
- **Cost of full merge**
  - **"k < 64" page copy operations**
    - "n" page reads
    - "n" page writes
  - **1 block erase operations**
- **1 free block is created**



(b) Partial merge

Switch Merge

- No free block is required
- Cost of full merge
  - No copy operation
  - 1 block erase operation
- 1 free block is created

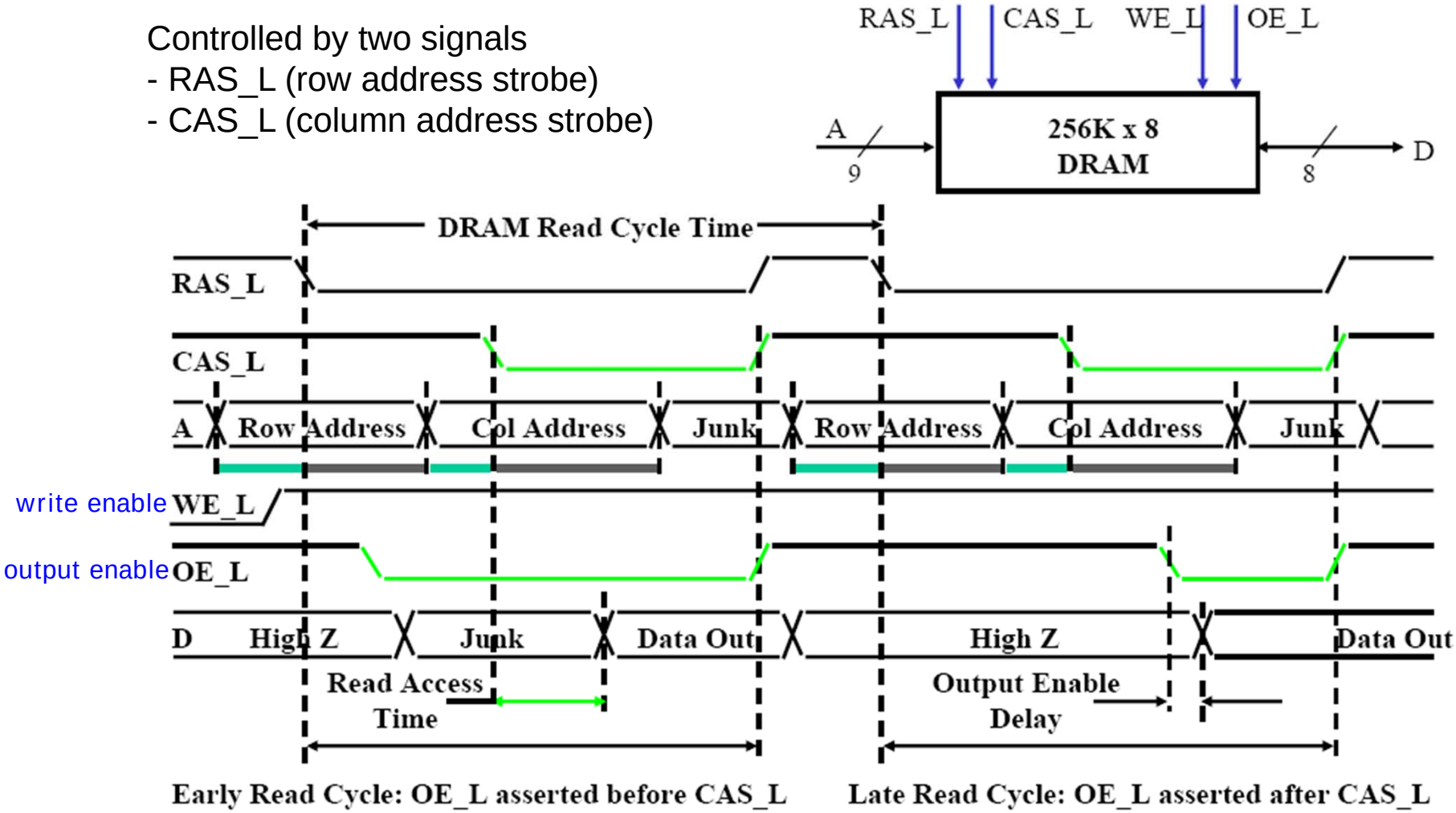


(c) Switch merge

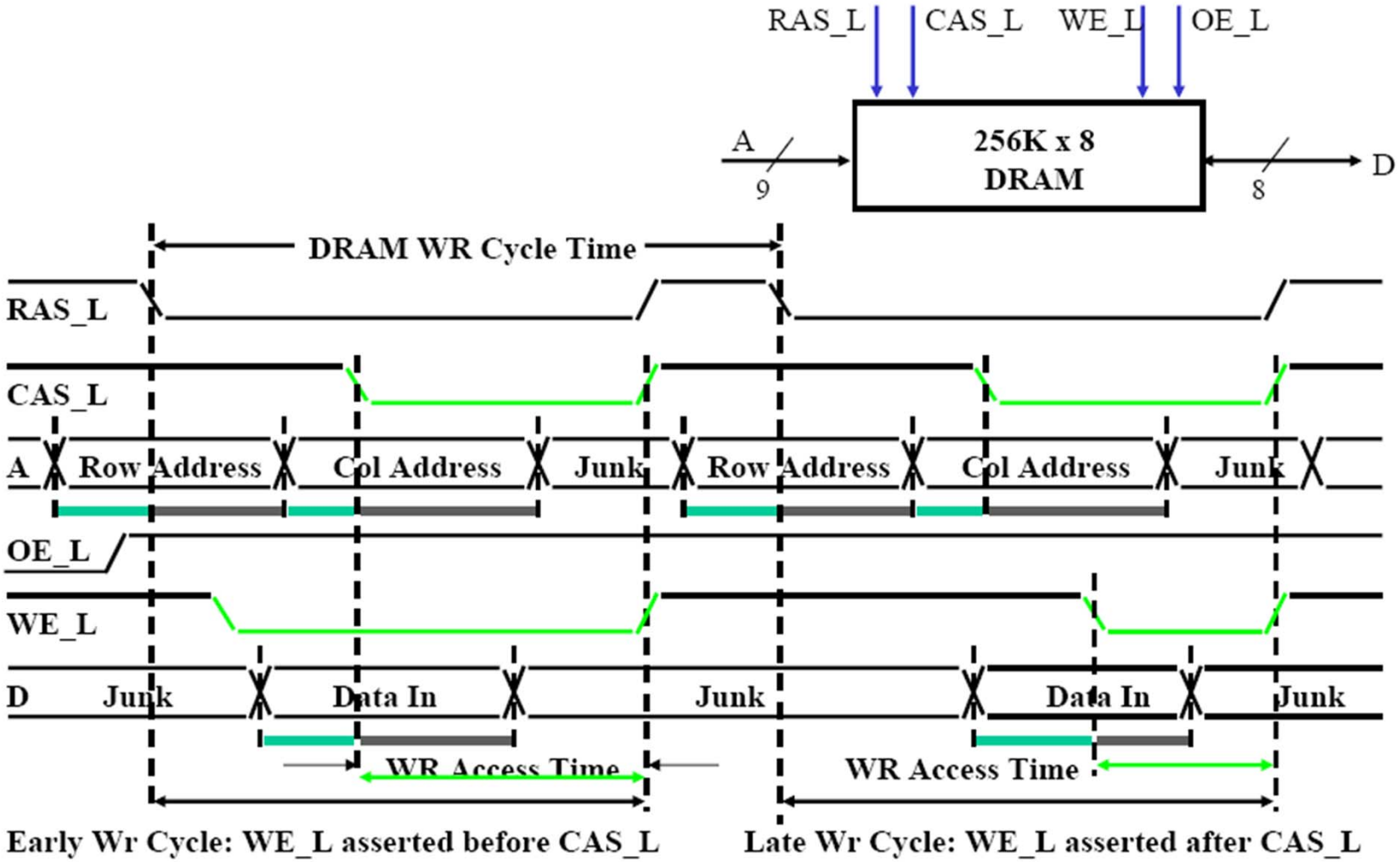
More on RAM

DRAM Read Timing

Controlled by two signals
- RAS_L (row address strobe)
- CAS_L (column address strobe)



DRAM Write Timing



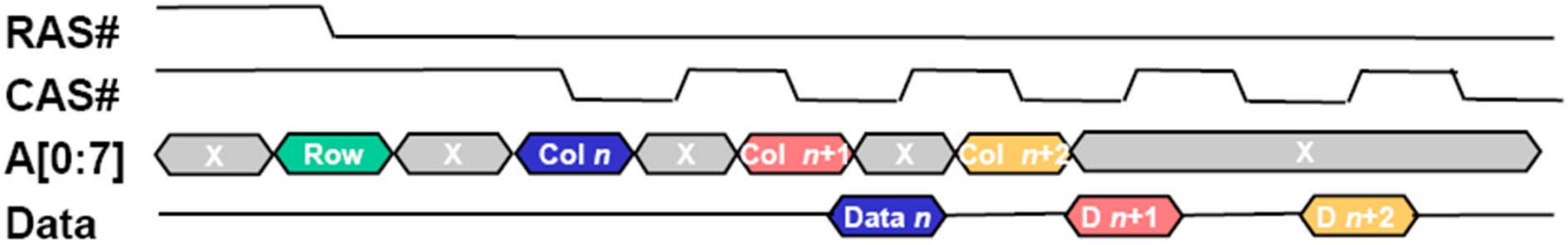
Types of DRAM

- Asynchronous DRAM
  - No need for synchronization
- Synchronous DRAM (SDRAM)
  - Waits for a clock signal before responding to its control inputs
  - Thus it is synchronized with the system bus (the processor)
- Video RAM (VRAM)
  - Dual-ported variant of DRAM
  - Once used to store the frame-buffer in some graphics adaptors
- Others
  - Fast Page Mode DRAM (FPM)
  - Extended Data Out (EDO) DRAM
  - Burst EDO (BEDO) DRAM
  - Synchronous Graphic RAM (SGRAM), Direct Rambus DRAM (DRDRAM), Double Data Rate (DDR) SDRAM, …

Fast Page Mode (FPM) DRAM

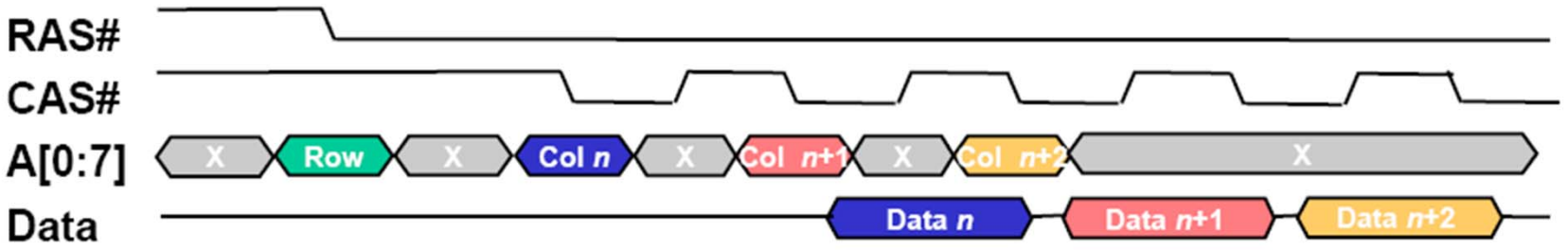
- Each row of memory bit array is viewed as a page
  - Page contains multiple words (word = 4 bytes)
  - Individual words addressed by column address

- Timing diagram
  - Row (page) address sent
  - 3 words read by sending column address for each



Extended Data Out (EDO) DRAM

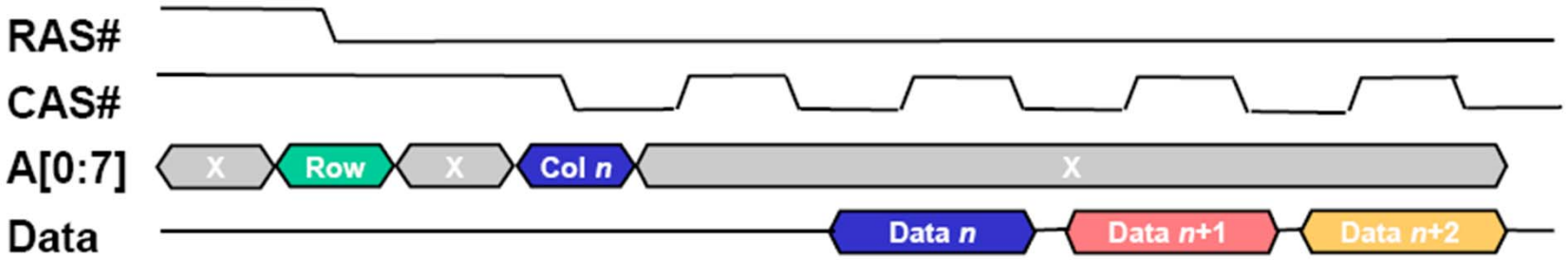
- Improvement of FPM DRAM
  - Uses extra latch before output buffer
  - Allows strobing of CAS before data read operation completed



Speedup through overlap

Burst EDO (BEDO) DRAM

- Generate a consecutive address by itself



Double Data Rate (DDR) SDRAM

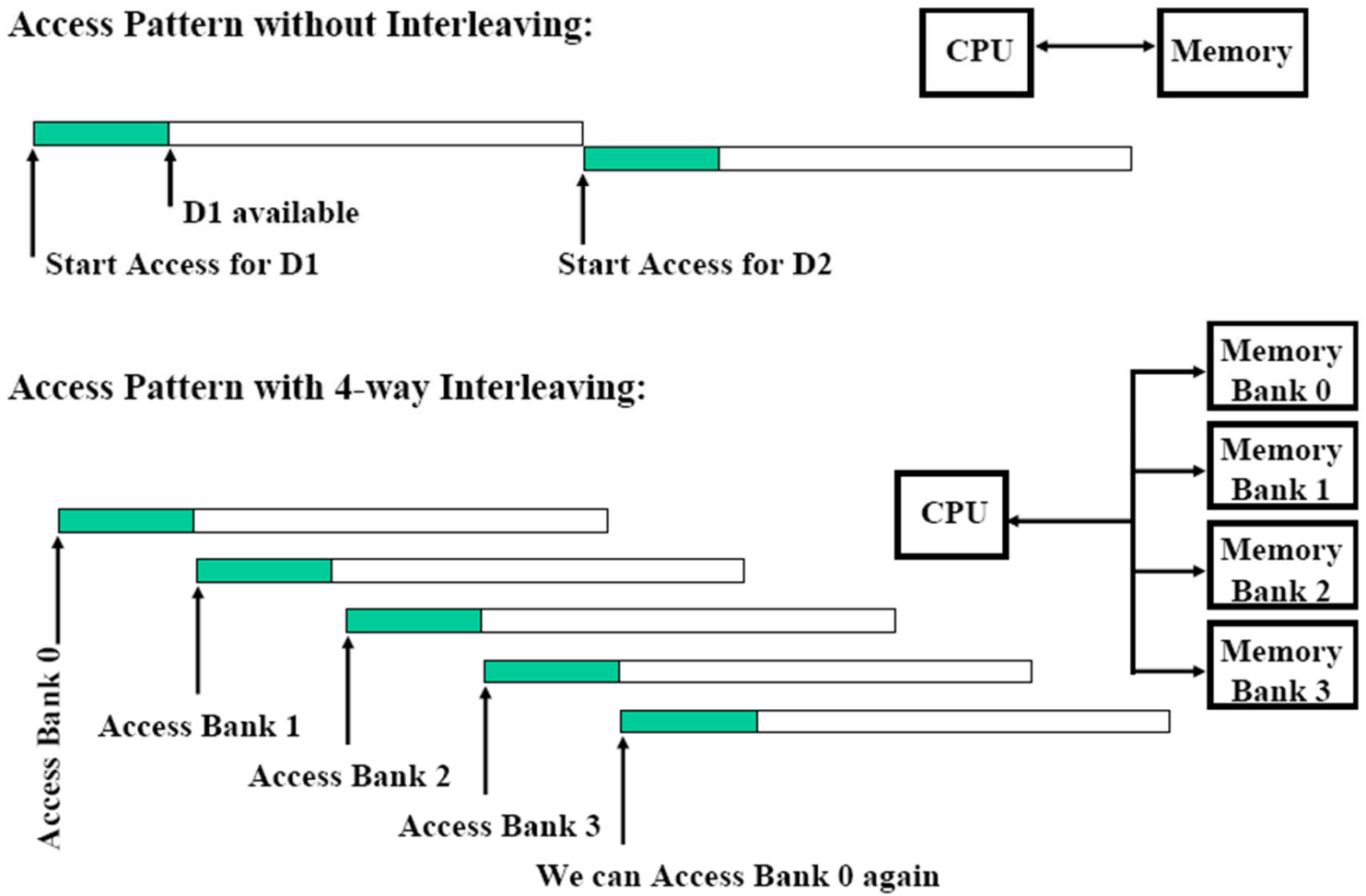
❒ DDR SDRAM specifications

Standard name	Memory clock	Cycle time	I/O Bus clock	Data transfers per second	Module name	Peak transfer rate
DDR-200	100 MHz	10 ns [1]	100 MHz	200 Million	PC-1600	1600 MB/s
DDR-266	133 MHz	7.5 ns	133 MHz	266 Million	PC-2100	2100 MB/s
DDR-300	150 MHz	6.67 ns	150 MHz	300 Million	PC-2400	2400 MB/s
DDR-333	166 MHz	6 ns	166 MHz	333 Million	PC-2700	2700 MB/s
DDR-400	200 MHz	5 ns	200 MHz	400 Million	PC-3200	3200 MB/s

- **PC-XYZ is Synchronous DRAM operating at a clock frequency of XYZ MHz, on a 64-bit bus, at a voltage of 3.3 V**
  - XYZ = 66, 100, 133, 1600, 2100, 2400, 2700, 3200

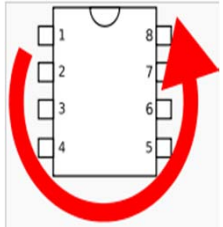
- **It achieves nearly twice the bandwidth of the preceding "single data rate" SDRAM by double pumping (transferring data on the rising and falling edges of the clock signal) without increasing the clock frequency**

Bandwidth Increase with Interleaving

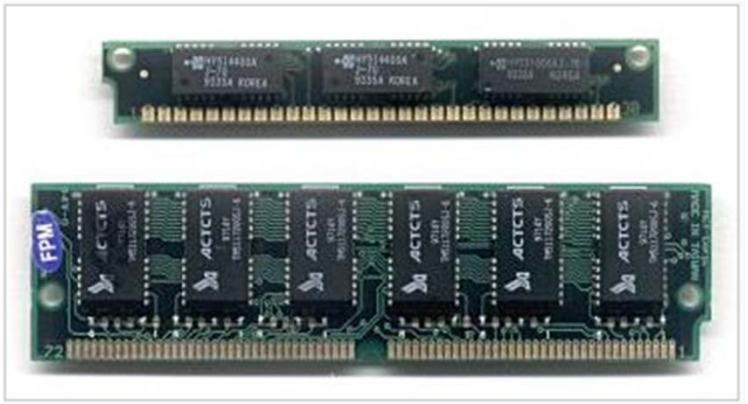


Types of Memory Modules

- **Dual In-line Package (DIP)**
  - Rectangular housing and two parallel rows of electrical connecting pins



- **Single In-line Memory Module (SIMM)**
  - The contacts on a SIMM are redundant on both sides of the module (32-bit data path)
- **Double In-line Memory Module (DIMM)**
  - Separate electrical contacts on each side of the module (64-bit data path)



30- (top) and 72-pin (bottom) SIMMs. Early 30-pin modules commonly had either 256 KB or 1 MB of memory.