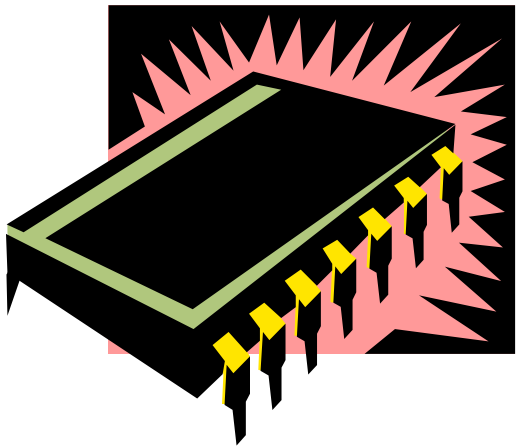
Random access memory

- Sequential circuits all depend upon the presence of memory.
  - A flip-flop can store one bit of information.
  - A register can store a single "word," typically 32-64 bits.
- Random access memory, or RAM, allows us to store even larger amounts of data. Today we'll see:
  - The basic interface to memory.
  - How you can implement static RAM chips hierarchically.
- This is the last piece we need to put together a computer!



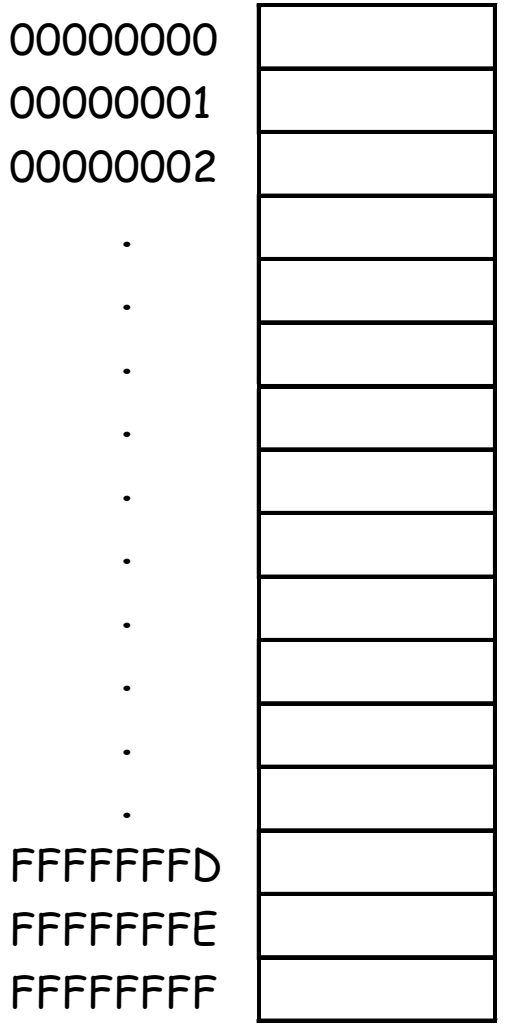
Introduction to RAM

- Random-access memory, or RAM, provides large quantities of temporary storage in a computer system.
- Remember the basic capabilities of a memory:
  - It should be able to store a value.
  - You should be able to read the value that was saved.
  - You should be able to change the stored value.
- A RAM is similar, except that it can store *many* values.
  - An address will specify which memory value we're interested in.
  - Each value can be a multiple-bit word (e.g., 32 bits).
- We'll refine the memory properties as follows:

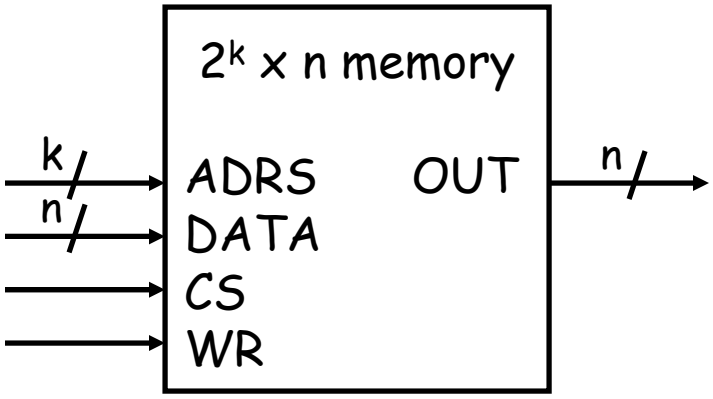
A RAM should be able to:
- Store many words, one per address
- Read the word that was saved at a particular address
- Change the word that's saved at a particular address

Picture of memory

- You can think of computer memory as being one big array of data.
  - The address serves as an array index.
  - Each address refers to one word of data.
- You can read or modify the data at any given memory address, just like you can read or modify the contents of an array at any given index.
- If you've worked with pointers in C or C++, then you've already worked with memory addresses.



Block diagram of RAM

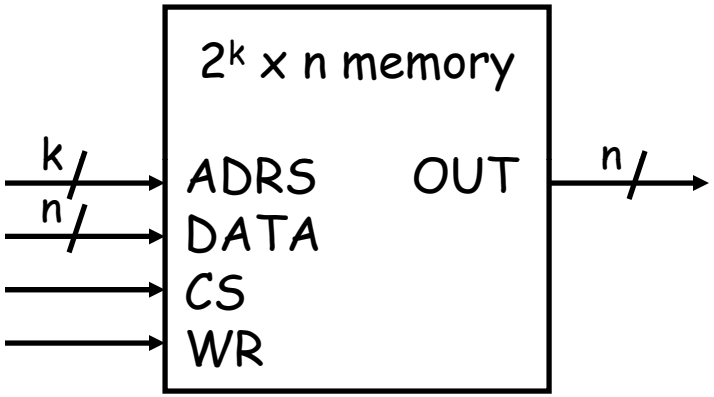


CS	WR	Memory operation
0	x	None
1	0	Read selected word
1	1	Write selected word

- This block diagram introduces the main interface to RAM.
  - A Chip Select, CS, enables or disables the RAM.
  - ADRS specifies the address or location to read from or write to.
  - WR selects between reading from or writing to the memory.
    - To read from memory, WR should be set to 0.
      OUT will be the n-bit value stored at ADRS.
    - To write to memory, we set WR = 1.
      DATA is the n-bit value to save in memory.
- This interface makes it easy to combine RAMs together, as we'll see.

Memory sizes

- We refer to this as a 2^k x n memory.
  - There are k *address lines*, which can specify one of 2^k addresses.
  - Each address contains an n-bit word.



- For example, a 2^{24} x 16 RAM contains 2^{24} = 16M words, each 16 bits long.
  - The RAM would need 24 address lines.
  - The total storage capacity is 2^{24} x 16 = 2^{28} bits.

Size matters!

- Memory sizes are usually specified in numbers of bytes (8 bits).
- The 2^{28}-bit memory on the previous page translates into:

$$2^{28} \text{ bits} / 8 \text{ bits per byte} = 2^{25} \text{ bytes}$$

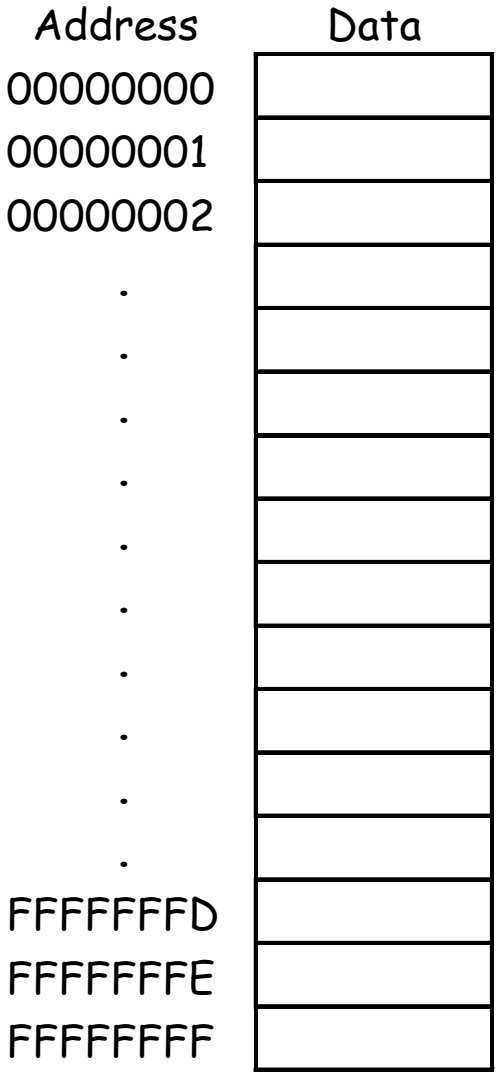
- With the abbreviations below, this is equivalent to 32 megabytes.

	Prefix	Base 2	Base 10
K	Kilo	$2^{10} = 1{,}024$	$10^3 = 1{,}000$
M	Mega	$2^{20} = 1{,}048{,}576$	$10^6 = 1{,}000{,}000$
G	Giga	$2^{30} = 1{,}073{,}741{,}824$	$10^9 = 1{,}000{,}000{,}000$

- To confuse you, RAM size is measured in base 2 units, while hard drive size is measured in base 10 units.
  - In this class, we'll only concern ourselves with the base 2 units.

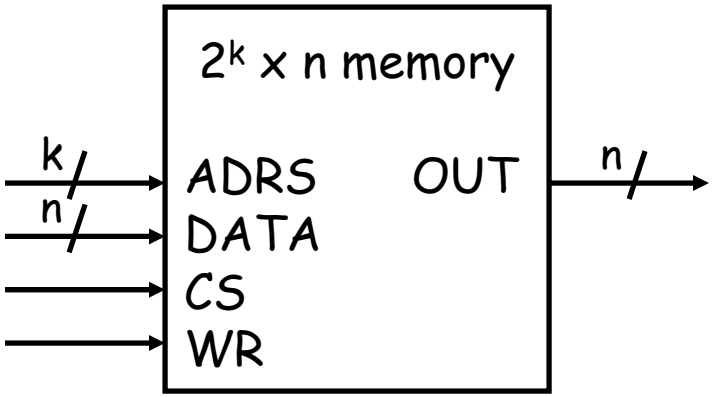
Typical memory sizes

- Some typical memory capacities:
  - PCs usually come with 512MB – 2GB RAM.
  - PDAs have 16-64MB of memory.
  - Digital cameras and MP3 players can have 32MB-8GB or more of onboard storage.
- Many operating systems implement virtual memory, which makes the memory seem larger than it really is.
  - Most systems allow up to 32-bit addresses. This works out to 2^{32}, or about four billion, different possible addresses.
  - With a data size of one byte, the result is apparently a 4GB memory!
  - The operating system uses hard disk space as a substitute for "real" memory.



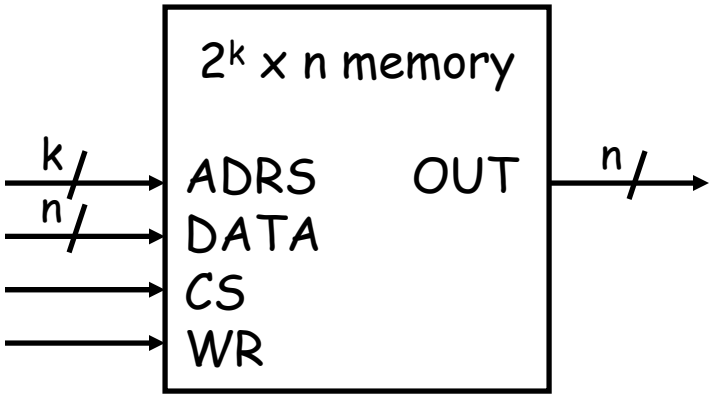
Reading RAM

- To *read* from this RAM, the controlling circuit must:
  - Enable the chip by ensuring CS = 1.
  - Select the read operation, by setting WR = 0.
  - Send the desired address to the ADRS input.
  - The contents of that address appear on OUT after a little while.
- Notice that the DATA input is unused for read operations.



Writing RAM

- To *write* to this RAM, you need to:
  - Enable the chip by setting CS = 1.
  - Select the write operation, by setting WR = 1.
  - Send the desired address to the ADRS input.
  - Send the word to store to the DATA input.
- The output OUT is not needed for memory write operations.

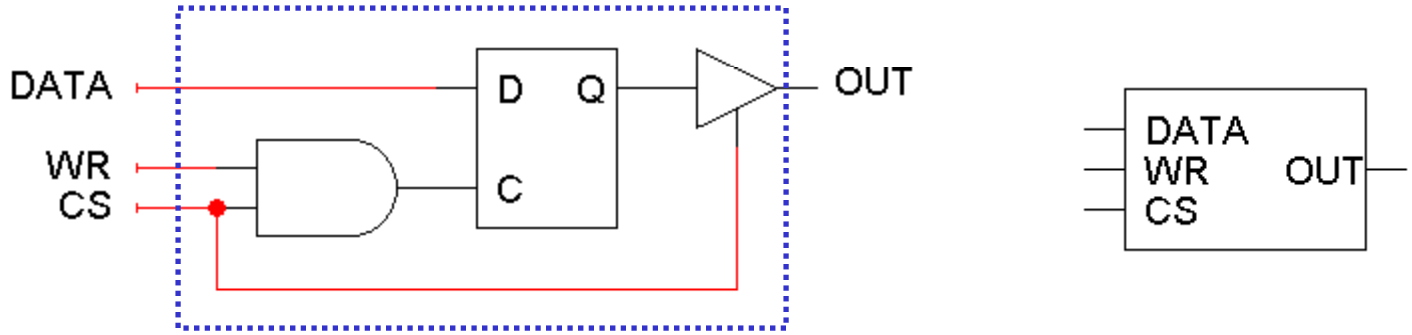


Static memory

- How can you implement the memory chip?
- There are many different kinds of RAM.
  - We'll start off discussing static memory, which is most commonly used in caches and video cards.
  - Later we mention a little about dynamic memory, which forms the bulk of a computer's main memory.
- Static memory is modeled using one *latch* for each bit of storage.
- Why use latches instead of flip flops?
  - A latch can be made with only two NAND or two NOR gates, but a flip-flop requires at least twice that much hardware.
  - In general, smaller is faster, cheaper and requires less power.
  - The tradeoff is that getting the timing exactly right is a pain.

Starting with latches

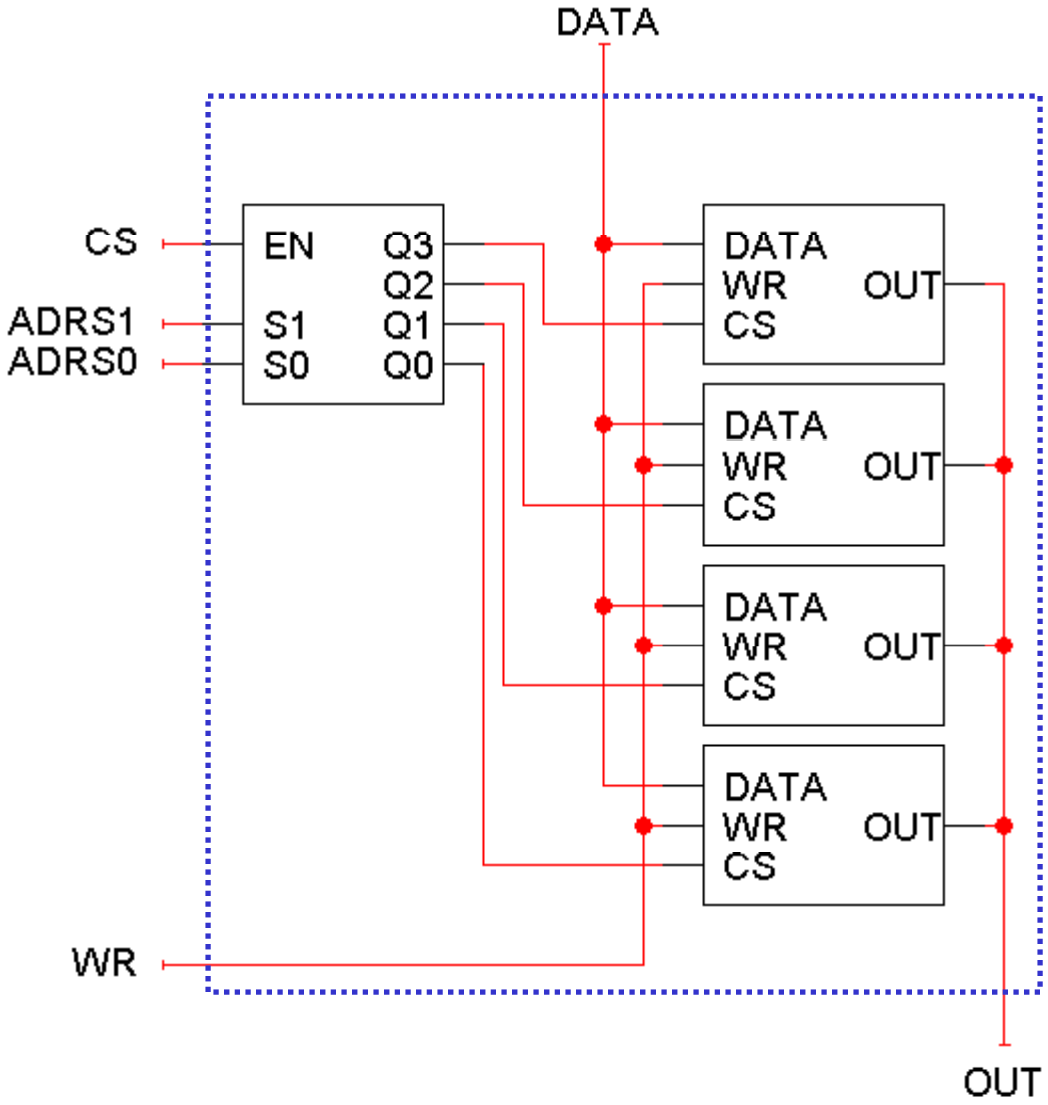
- To start, we can use one latch to store each bit. A one-bit RAM cell is shown here.



- Since this is just a one-bit memory, an ADRS input is not needed.
- Writing to the RAM cell:
  - When CS = 1 and WR = 1, the latch control input will be 1.
  - The DATA input is thus saved in the D latch.
- Reading from the RAM cell and maintaining the current contents:
  - When CS = 0 or when WR = 0, the latch control input is also 0, so the latch just maintains its present state.
  - The current latch contents will appear on OUT.

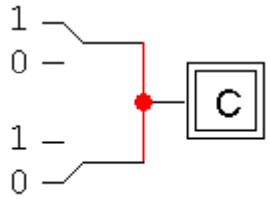
My first RAM

- We can use these cells to make a 4 x 1 RAM.
- Since there are four words, ADRS is two bits.
- Each word is only one bit, so DATA and OUT are one bit each.
- Word selection is done with a decoder attached to the CS inputs of the RAM cells. Only one cell can be read or written at a time.
- Notice that the outputs are connected together with a *single* line!



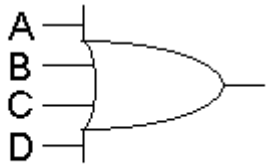
Connecting outputs together

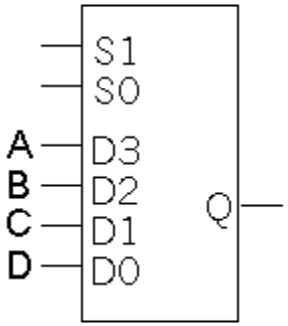
- In normal practice, it's bad to connect outputs together. If the outputs have different values, then a conflict arises.



The "C" in LogicWorks means "conflict."

- The standard way to "combine" outputs is to use OR gates or muxes.

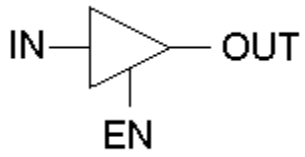




- This can get expensive, with many wires and gates with large fan-ins.

Those funny triangles

- The triangle represents a three-state buffer.
- Unlike regular logic gates, the output can be one of *three* different possibilities, as shown in the table.

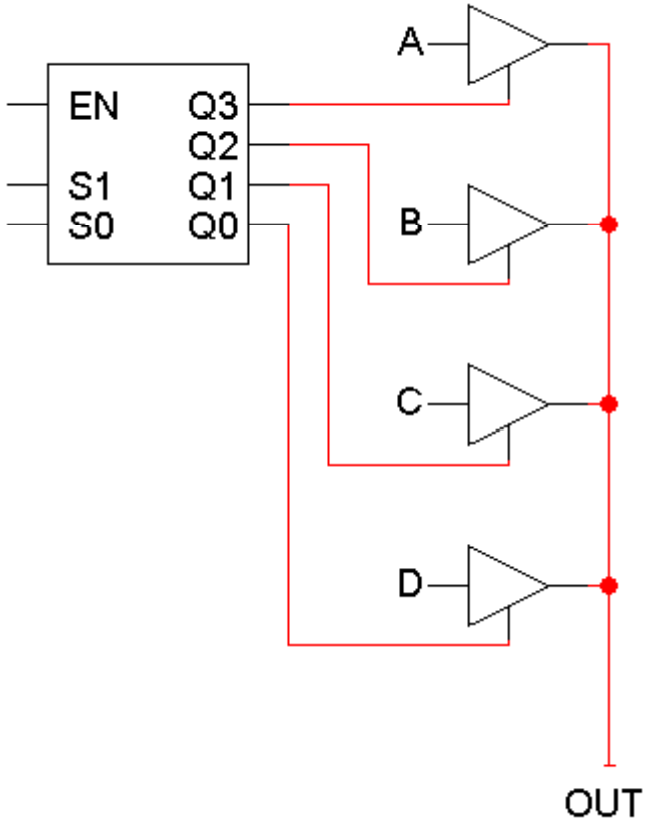


EN	IN	OUT
0	x	Disconnected
1	0	0
1	1	1

- "Disconnected" means no output appears at all, in which case it's safe to connect OUT to another output signal.
- The disconnected value is also sometimes called high impedance or Hi-Z.

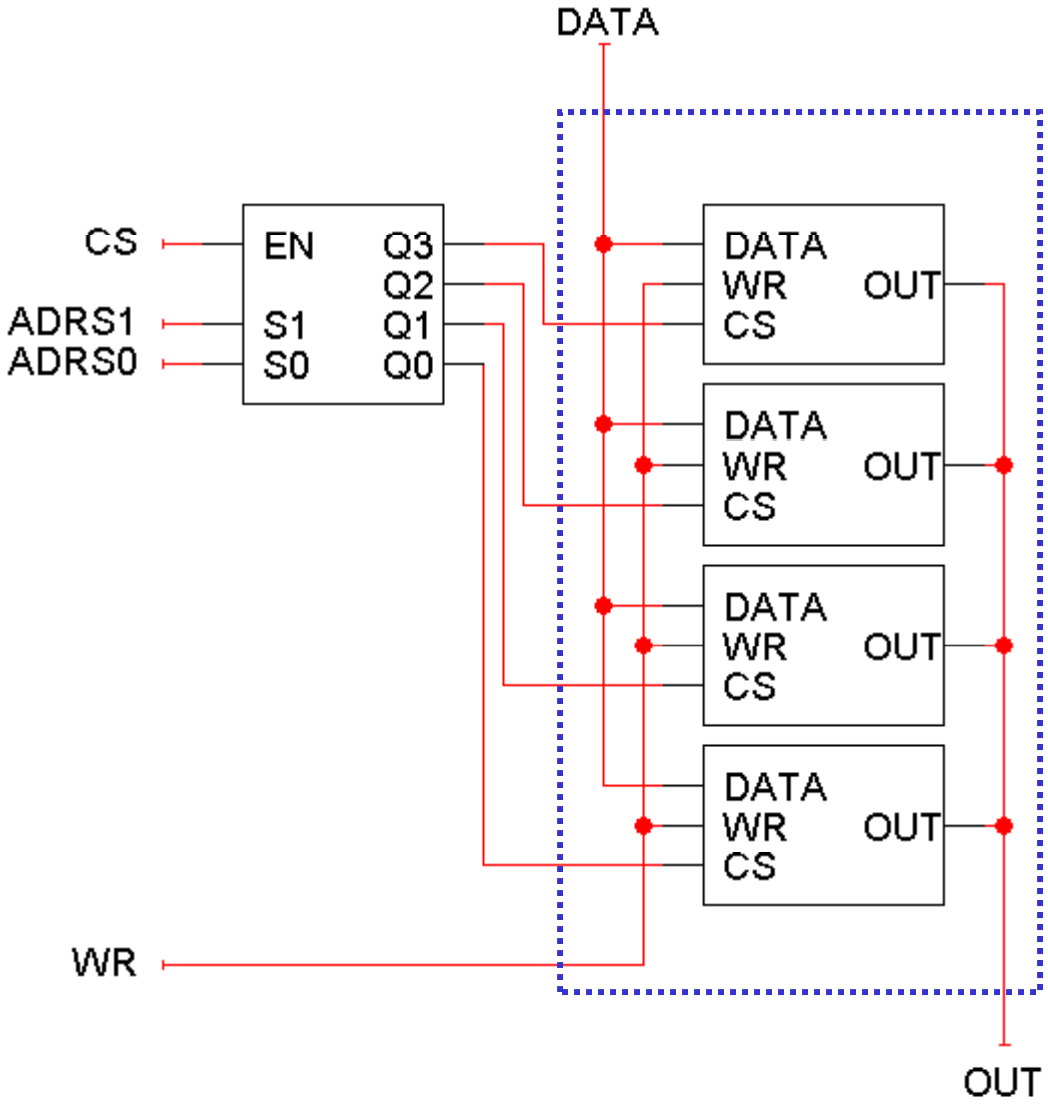
Connecting three-state buffers together

- You can connect several three-state buffer outputs together if you can *guarantee* that only one of them is enabled at any time.
- The easiest way to do this is to use a decoder!
- If the decoder is disabled, then all the three-state buffers will appear to be disconnected, and OUT will also appear disconnected.
- If the decoder is enabled, then exactly one of its outputs will be true, so only one of the tri-state buffers will be connected and produce an output.
- The net result is we can save some wire and gate costs. We also get a little more flexibility in putting circuits together.



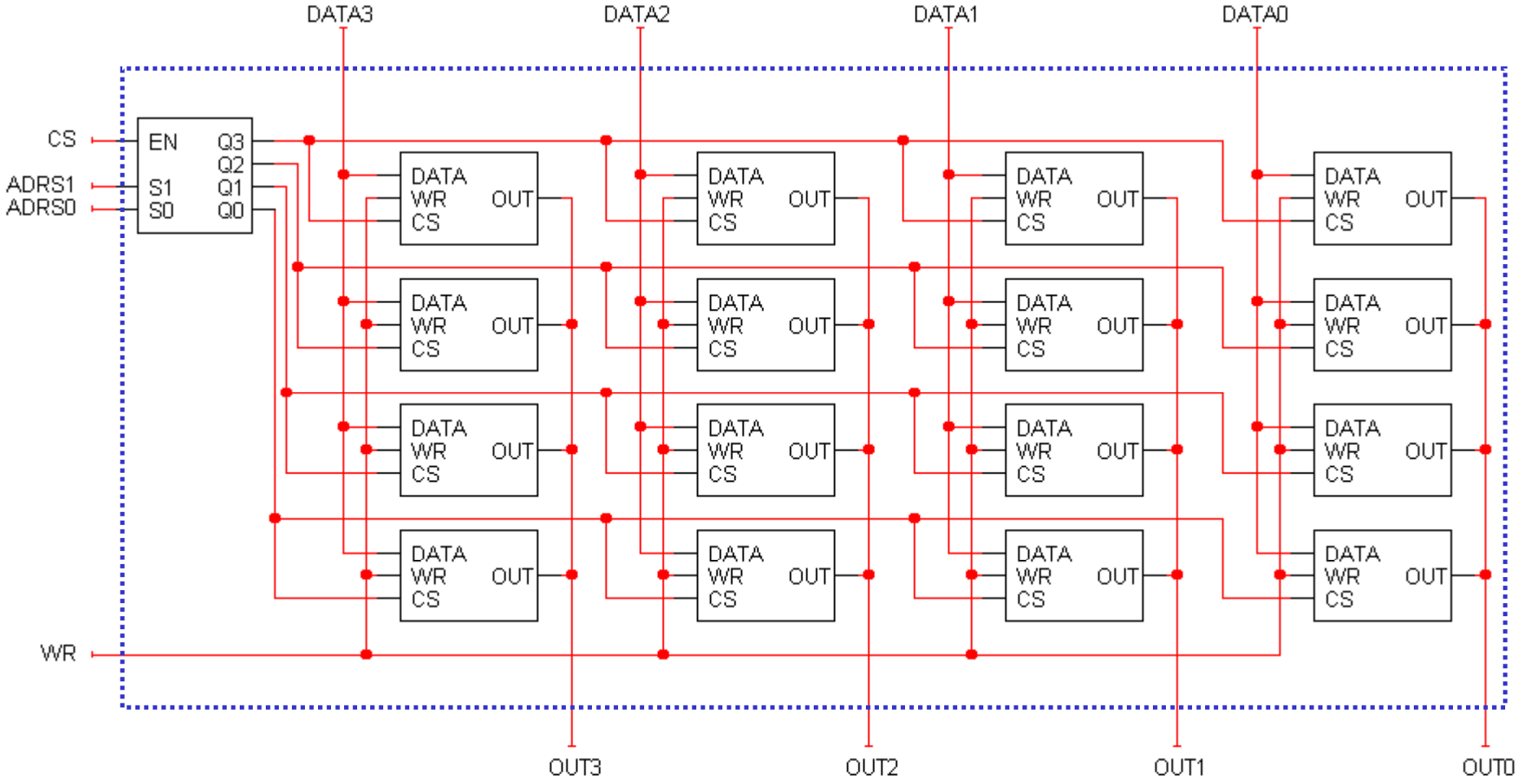
Bigger and better

- Here is the 4 x 1 RAM once again.
- How can we make a "wider" memory with more bits per word, like maybe a 4 x 4 RAM?
- Duplicate the stuff in the blue box!



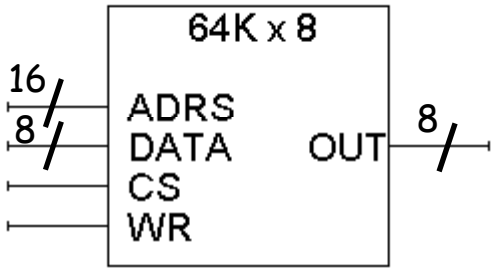
A 4 x 4 RAM

- DATA and OUT are now each *four* bits long, so you can read and write four-bit words.



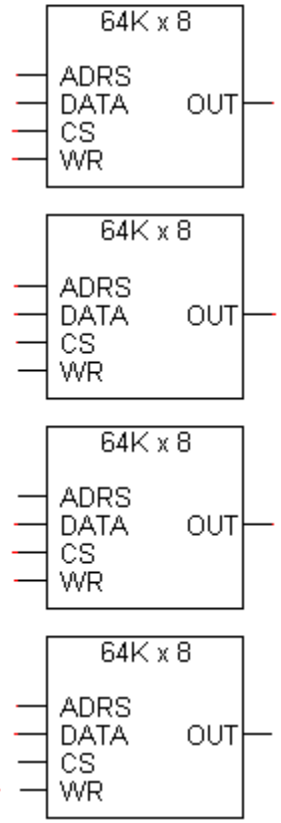
Bigger RAMs from smaller RAMs

- We can use small RAMs as building blocks for making larger memories, by following the same principles as in the previous examples.
- As an example, suppose we have some 64K x 8 RAMs to start with:
  - $64K = 2^6 \times 2^{10} = 2^{16}$, so there are 16 address lines.
  - There are 8 data lines.



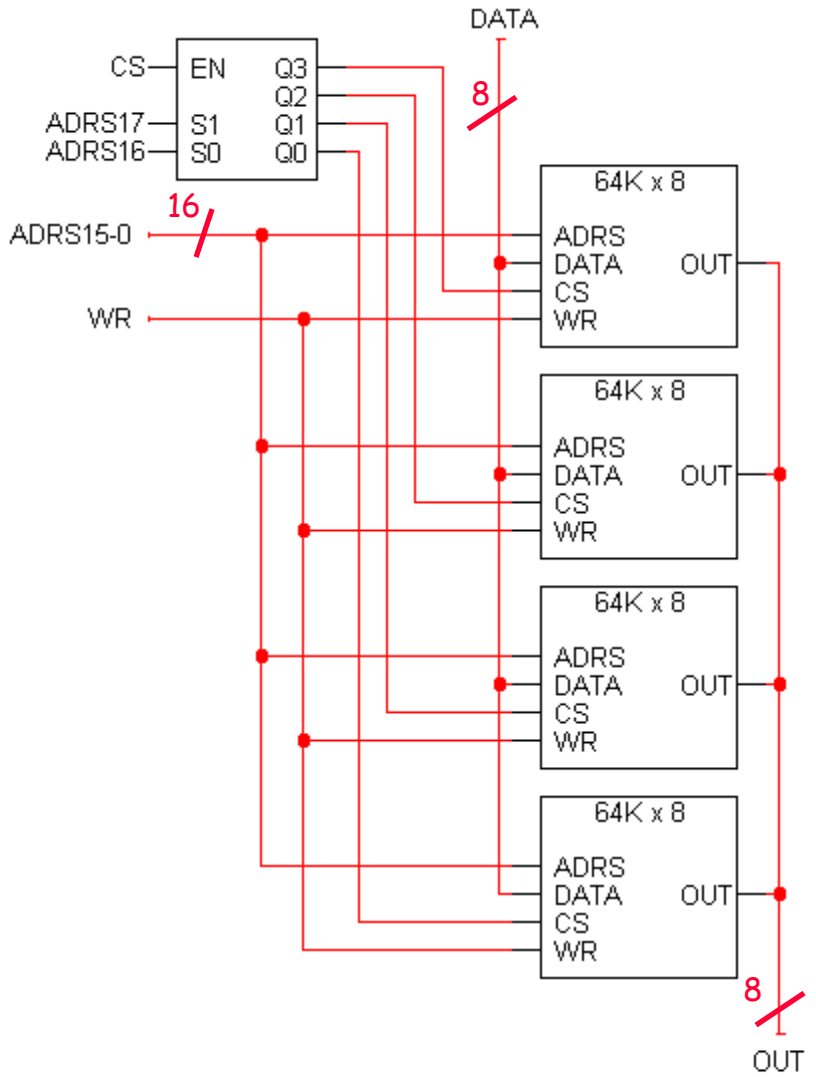
Making a larger memory

- We can put four 64K x 8 chips together to make a 256K x 8 memory.
- For 256K words, we need ?? address lines.



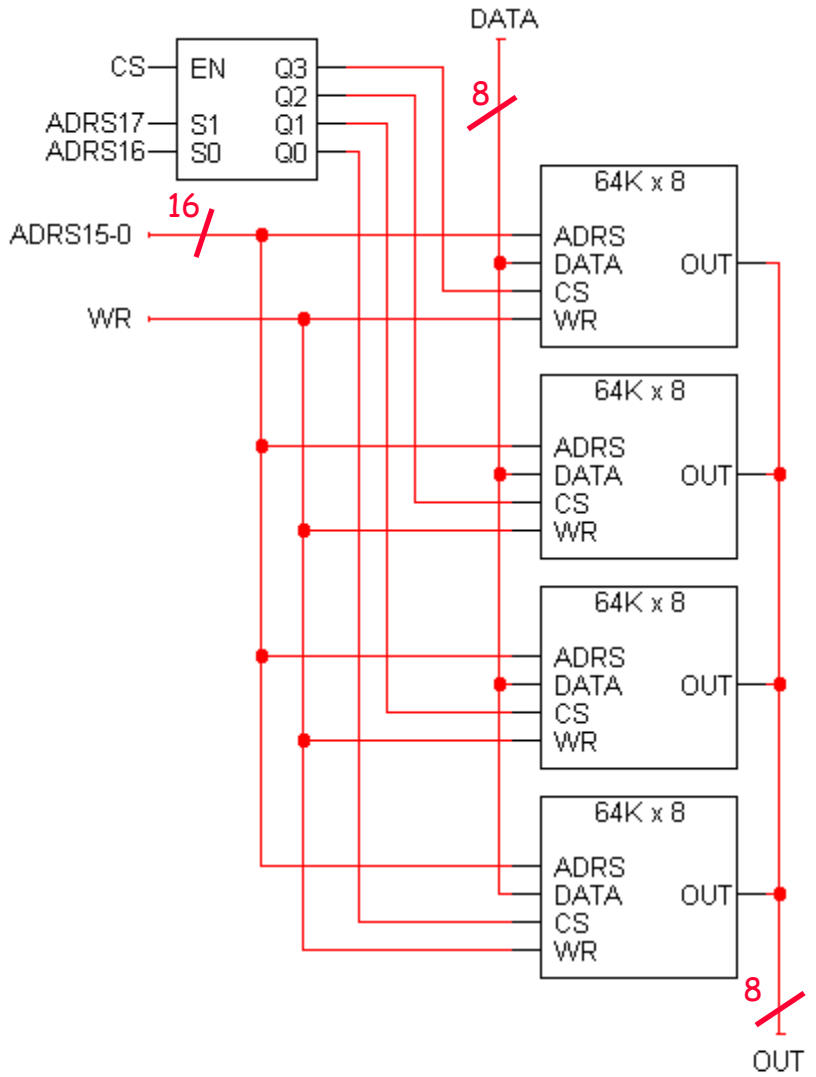
Making a larger memory

- We can put four 64K x 8 chips together to make a 256K x 8 memory.
- For 256K words, we need 18 address lines.
  - The two most significant address lines go to the decoder, which selects one of the four 64K x 8 RAM chips.
  - The other 16 address lines are shared by the 64K x 8 chips.
- The 64K x 8 chips also share WR and DATA inputs.
- This assumes the 64K x 8 chips have three-state outputs.

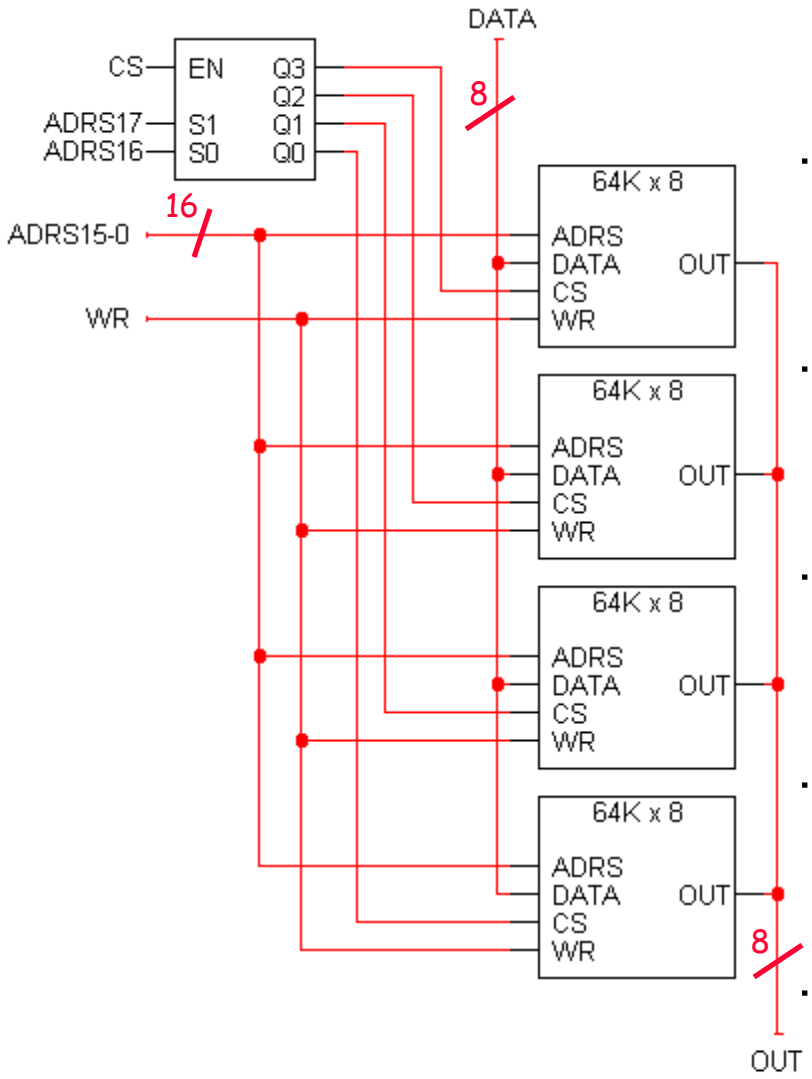


Analyzing the 256K x 8 RAM

- There are 256K words of memory, spread out among the four smaller 64K x 8 RAM chips.
- When the two most significant bits of the address are 00, the bottom RAM chip is selected. It holds data for the first 64K addresses.
- The next chip up is enabled when the address starts with 01. It holds data for the second 64K addresses.
- The third chip up holds data for the next 64K addresses.
- The final chip contains the data of the final 64K addresses.



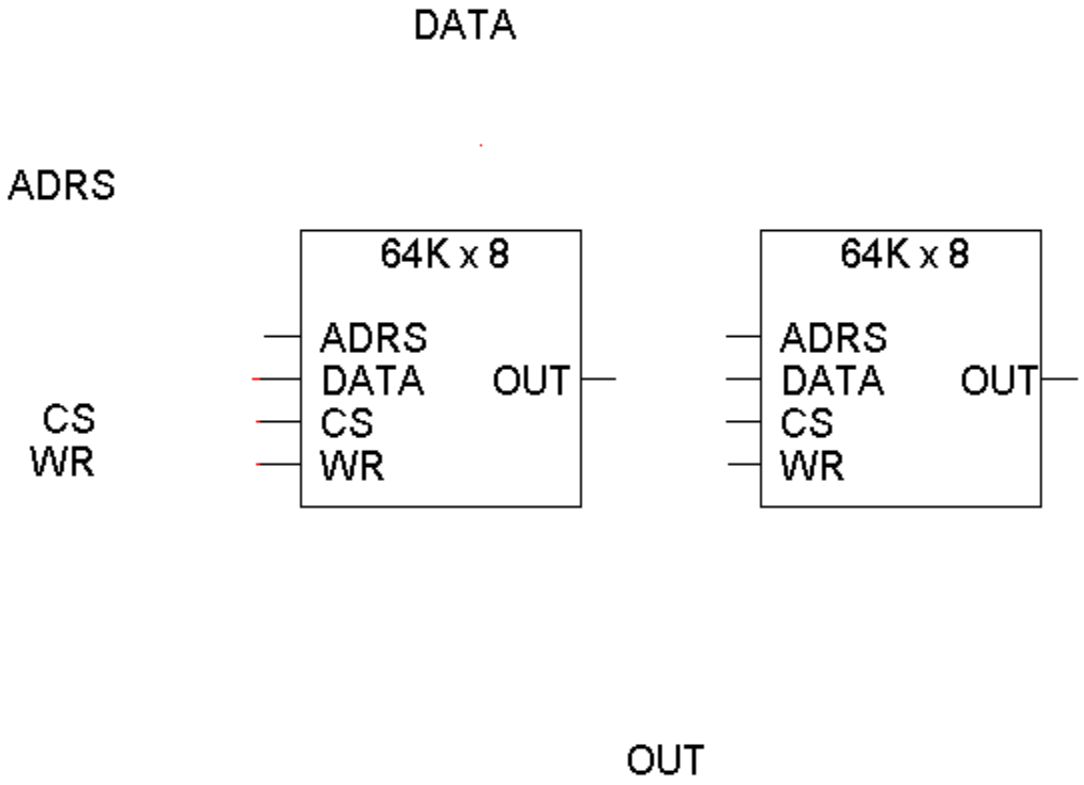
Address ranges




```
11 1111 1111 1111 1111 (0x3ffff)
                  to
11 0000 0000 0000 0000 (0x30000)
10 1111 1111 1111 1111 (0x2ffff)
                  to
10 0000 0000 0000 0000 (0x20000)
01 1111 1111 1111 1111 (0x1ffff)
                  to
01 0000 0000 0000 0000 (0x10000)
00 1111 1111 1111 1111 (0x0ffff)
                  to
00 0000 0000 0000 0000 (0x00000)
```

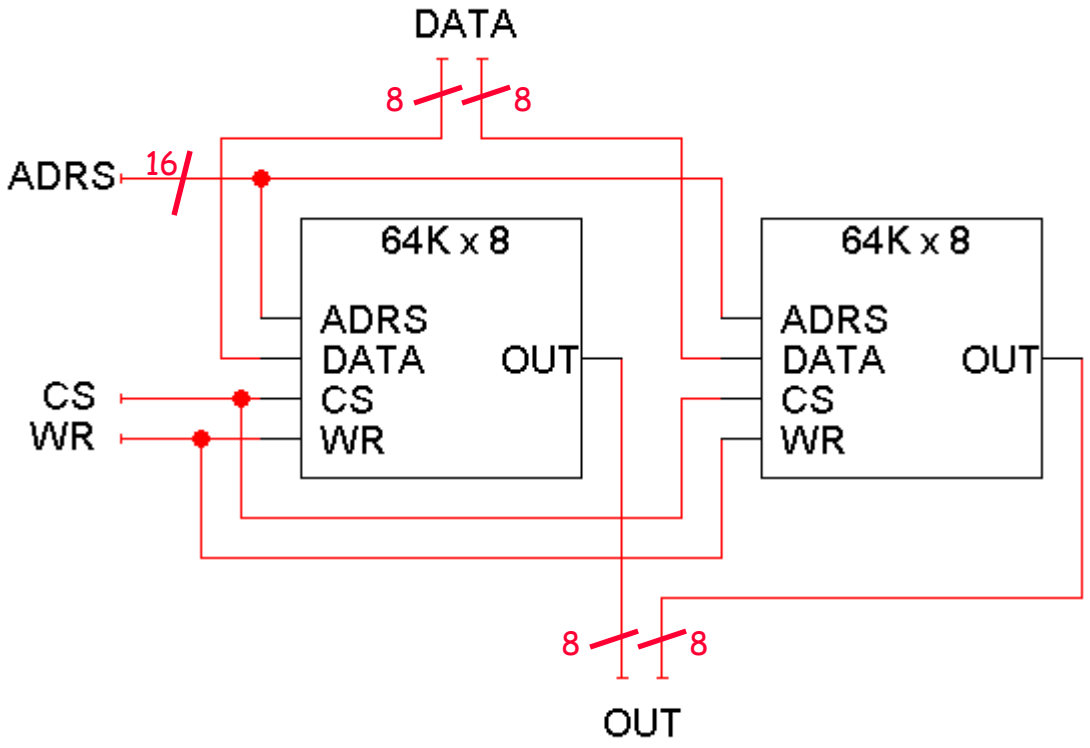
Making a wider memory

- You can also combine smaller chips to make wider memories, with the same number of addresses but more bits per word.
- How do we create a 64K x 16 RAM from two 64K x 8 chips?



Making a wider memory

- You can also combine smaller chips to make wider memories, with the same number of addresses but more bits per word.
- Here is a 64K x 16 RAM, created from two 64K x 8 chips.
  - The left chip contains the most significant 8 bits of the data.
  - The right chip contains the lower 8 bits of the data.



Summary

- A RAM looks like a bunch of registers connected together, allowing users to select a particular address to read or write.
- Much of the hardware in memory chips supports this selection process:
  - Chip select inputs
  - Decoders
  - Tri-state buffers
- By providing a general interface, it's easy to connect RAMs together to make "longer" and "wider" memories.

- Next, we'll look at some other types of memories
- We now have all the components we need to build our simple processor.

Organization

- The basic element of a semiconductor memory is the memory cell.
- All semiconductor memory cells share certain properties:
  - They exhibit two stable (or semistable) states, which can be used to represent binary 1 and 0.
  - They are capable of being written into (at least once), to set the state.
  - They are capable of being read to sense the state.

- The cell has three functional terminals capable of carrying an electrical signal. The select terminal is to select a memory cell for a read or write operation.
- The control terminal indicates read or write.
- For writing, the third terminal is to provide an electrical signal that sets the state of the cell to 1 or 0. For reading, that terminal is used for output of the cell's state.

Semiconductor Memory Types

<table>
<tr><th>Memory Type</th><th>Category</th><th>Erasure</th><th>Write Mechanism</th><th>Volatility</th></tr>
<tr><td>Random-access memory (RAM)</td><td>Read-write memory</td><td>Electrically, byte-level</td><td>Electrically</td><td>Volatile</td></tr>
<tr><td>Read-only memory (ROM)</td><td rowspan="2">Read-only memory</td><td rowspan="2">Not possible</td><td>Masks</td><td rowspan="5">Nonvolatile</td></tr>
<tr><td>Programmable ROM (PROM)</td><td rowspan="4">Electrically</td></tr>
<tr><td>Erasable PROM (EPROM)</td><td rowspan="3">Read-mostly memory</td><td>UV light, chip-level</td></tr>
<tr><td>Electrically Erasable PROM (EEPROM)</td><td>Electrically, byte-level</td></tr>
<tr><td>Flash memory</td><td>Electrically, block-level</td></tr>
</table>

Random Access Memory (RAM)

- The most common is referred to as random-access memory (RAM).
- One distinguishing characteristic of RAM is that it is possible both to read data from the memory and to write new data into the memory easily and rapidly.
- The other distinguishing characteristic of RAM is that it is volatile.
- The two traditional forms of RAM used in computers are DRAM and SRAM.

Semiconductor Main Memory

- Random Access Memory (RAM)
  - Misnamed as all semiconductor memory is random access.
  - Read/Write.
  - Volatile.
  - Temporary storage.
  - Static or dynamic.

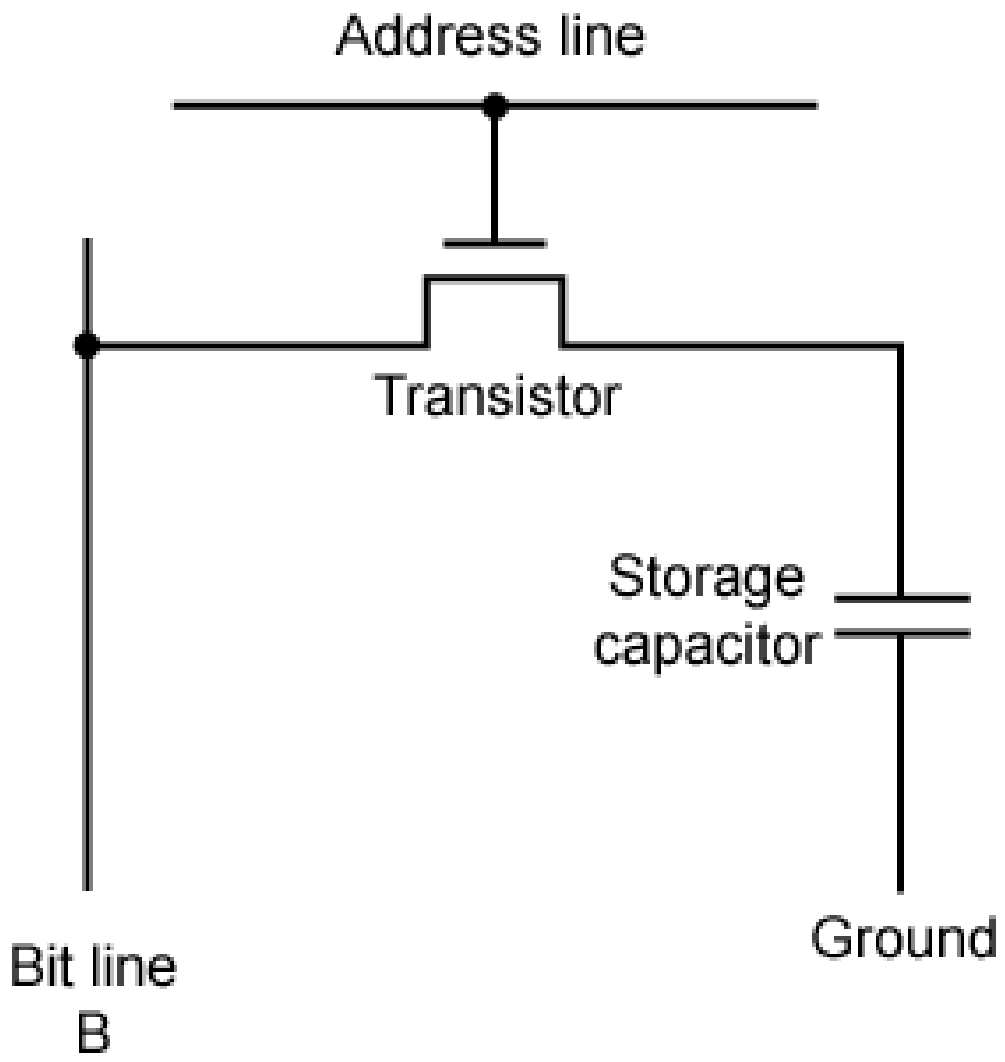
Dynamic RAM (DRAM)

- Bits stored as charge in capacitors.
- Charges leak.
- Need refreshing even when powered.
- Simpler construction.
- Smaller per bit.
- Less expensive.
- Need refresh circuits.
- Slower.
- Used for main memory.
- Essentially an analogue device:
  - Level of charge determines value.

- DRAM is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as a binary 1 or 0.

- The next diagram is a typical DRAM structure for an individual cell that stores 1 bit. The address line is activated when the bit value from this cell is to be read or written. The transistor acts as a switch that is closed (allowing current to flow) if a voltage is applied to the address line and open (no current flows) if no voltage is present on the address line.

Dynamic RAM (DRAM) Cell Structure



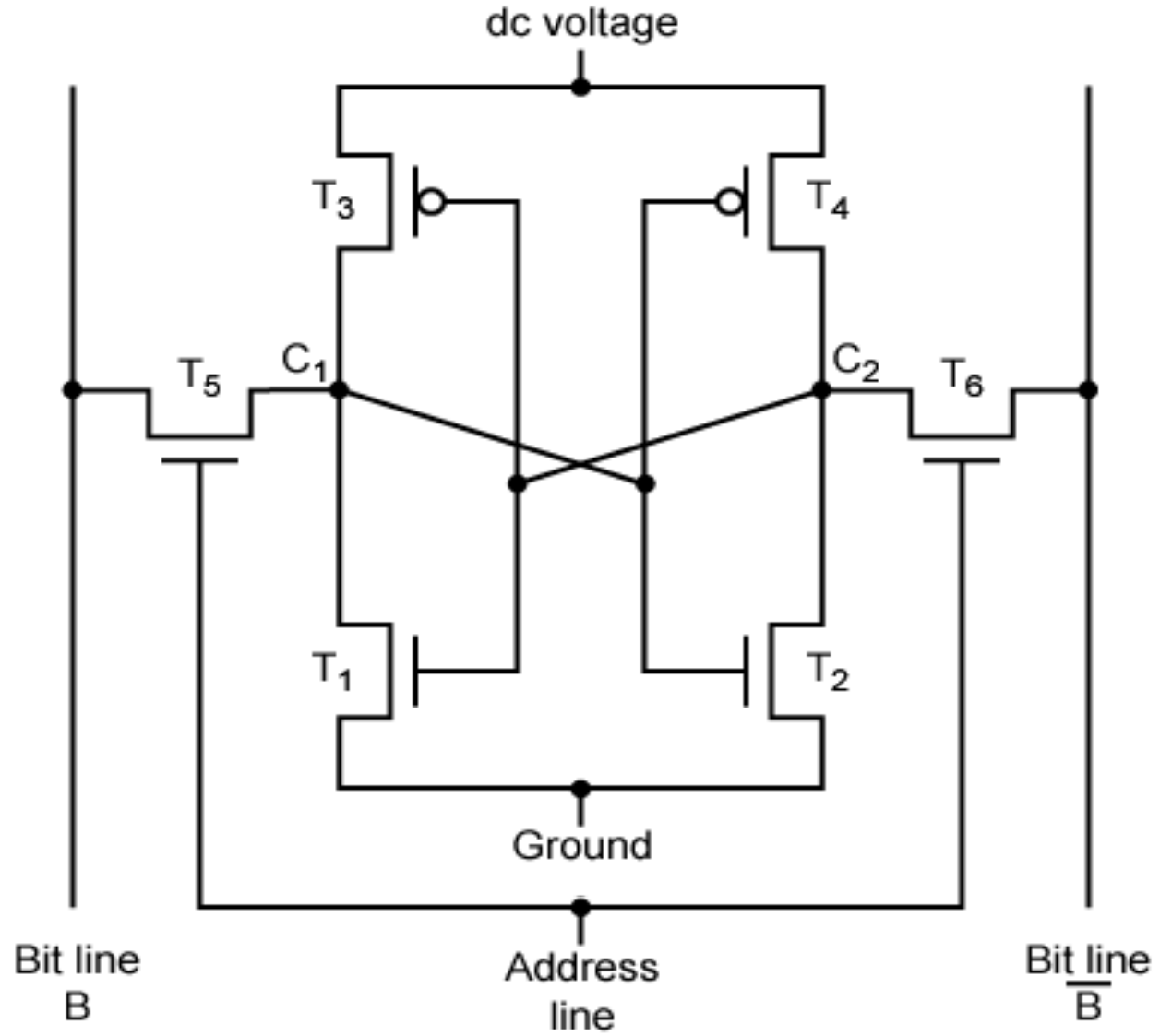
DRAM Operation

- Address line active when bit is to be read or written.
  - Transistor switch closed (current flows).
- For write operation:
  - Voltage signal is applied to the bit line.
    - High voltage for 1, low voltage for 0.
  - Then a signal is applied to the address line.
    - Transfers charge to the capacitor.
- For read operation:
  - Address line is selected.
    - transistor turns on.
  - Charge on capacitor is fed out onto bit line to sense amplifier.
    - Sense amplifier compares capacitor voltage with reference value to determine if the cell has 0 or 1.
  - Capacitor charge must be restored

Static RAM (SRAM)

- Bits stored as on/off switches.
- No charges to leak.
- No refreshing needed when powered.
- More complex construction.
- Larger per bit.
- More expensive.
- Does not need refresh circuits.
- Faster.
- Used for cache memory.
- Digital device:
  - Uses flip-flops.

Static RAM (SRAM) Cell Structure



Static RAM (SRAM) Operation

- Four transistors T_1, T_2, T_3, T_4 connected in an arrangement gives stable logic state.

- State 1:
  - C_1 high, C_2 low
  - T_1 T_4 off, T_2 T_3 on

- State 0:
  - C_2 high, C_1 low
  - T_2 T_3 off, T_1 T_4 on

- Address line controls the two transistors T_5 T_6 by switch it on to allowing read or write operation.

Static RAM (SRAM) Operation

- For write operation:
  - The desired bit value is applied to line B, while its complement is applied to line B’ .
  - This forces the four transistors (T1, T2, T3, T4) into the proper state.

- For a read operation:
  - The bit value is read from line B.

Static RAM (SRAM)

- A static RAM will hold its data as long as power is supplied to it.
- Both states are stable as long as the direct current (dc) voltage is applied.
- Unlike the DRAM, no refresh is needed to retain (hold) data.

SRAM versus DRAM

- Both volatile.
  - Power needed to preserve data.
- Dynamic Memory Cell:
  - Simpler to build, smaller.
  - More dense (smaller cells = more cells per unit area).
  - Less expensive.
  - Needs refresh circuitry.
  - Favoured for larger memory units.
- Static Memory Cell:
  - Faster.
  - Used for cache memory (both on and off chip).

Read Only Memory (ROM)

- It contains a permanent pattern of data that cannot be changed.
- A ROM is nonvolatile.
- While it is possible to read a ROM, it is not possible to write new data into it.
- An important application of ROMs is:
  - Microprogramming.
  - Library subroutines.
  - Systems programs (BIOS).
  - Function tables.

Types of ROM

- Written during manufacture:
  - Very expensive for small runs.
- Programmable (once):
  - PROM.
  - Needs special equipment to program.
- Read “mostly”:
  - Erasable Programmable (EPROM).
    - Erased by UV.
  - Electrically Erasable (EEPROM).
    - Takes much longer to write than read.
  - Flash memory.
    - Erase whole memory electrically.

Programmable ROM (PROM)

- When only a small number of ROMs with a particular memory content is needed, a less expensive alternative is the programmable ROM (PROM).
- It is nonvolatile and may be written into only once. For the PROM,
- The writing process is performed electrically and may be performed by a supplier or customer at a time later than the original chip fabrication.
- Special equipment is required for the writing or “programming” process.
- PROMs provide flexibility and convenience.

Erasable Programmable (EPROM)

- It is read and written electrically, as with PROM.
- It can be altered multiple times and, like the ROM and PROM, holds its data virtually indefinitely.
- For comparable amounts of storage, the EPROM is more expensive than PROM, but it has the advantage of the multiple update capability.

Electrically Erasable Programmable Read-Only Memory (EEPROM)

- It is a read-mostly memory that can be written into at any time without erasing prior contents; only the byte or bytes addressed are updated.
- It combines the advantage of non volatility with the flexibility of being updatable in place, using ordinary bus control, address, and data lines.
- EEPROM is more expensive than EPROM and also is less dense, supporting fewer bits per chip.

Flash Memory

- Like EEPROM, flash memory uses an electrical erasing technology.
- An entire flash memory can be erased in one or a few seconds, which is much faster than EPROM.
- In addition, it is possible to erase just blocks of memory rather than an entire chip.
- Like EPROM, flash memory uses only one transistor per bit, and so achieves the high density.

Interleaved Memory

- Main memory is composed of a collection of DRAM memory chips.
- A number of chips can be grouped together to form a memory bank.
- Banks independently service read or write requests.
- K banks can service k requests simultaneously, increasing memory read or write rates by a factor of K.

EduTechLearners

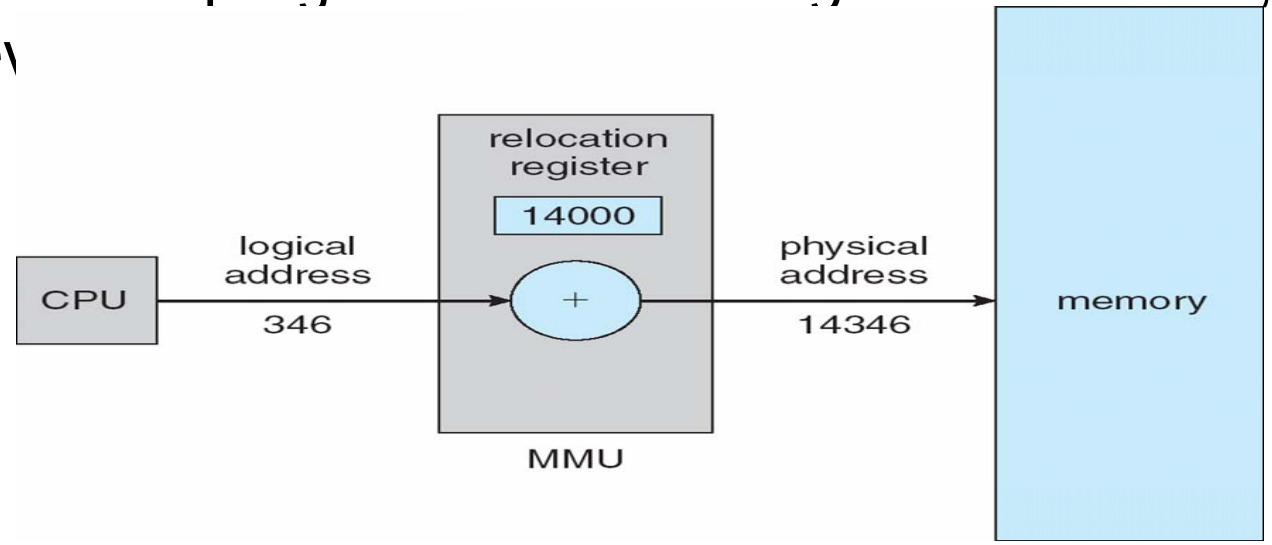
- *Main Memory Management or Allocation*
- *Interleaved Memory*

Memory Management

- Is the task carried out by the OS and hardware to accommodate multiple processes in main memory
- If only a few processes can be kept in main memory, then much of the time all processes will be waiting for I/O and the CPU will be idle
- Hence, memory needs to be allocated efficiently in order to pack as many processes into memory as possible

Memory-Management Unit (MMU)

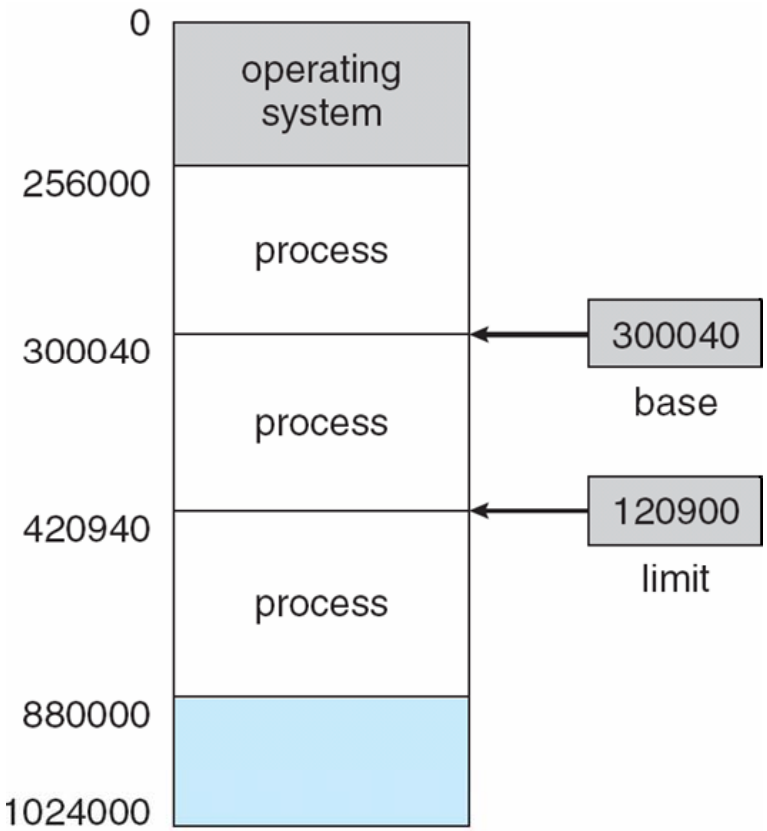
- Hardware device that maps virtual to physical address
- In MMU scheme, the value in the relocation register is added to every address generated by a user process at the time it is sent to memory
- The user program deals with *logical* addresses; it nev



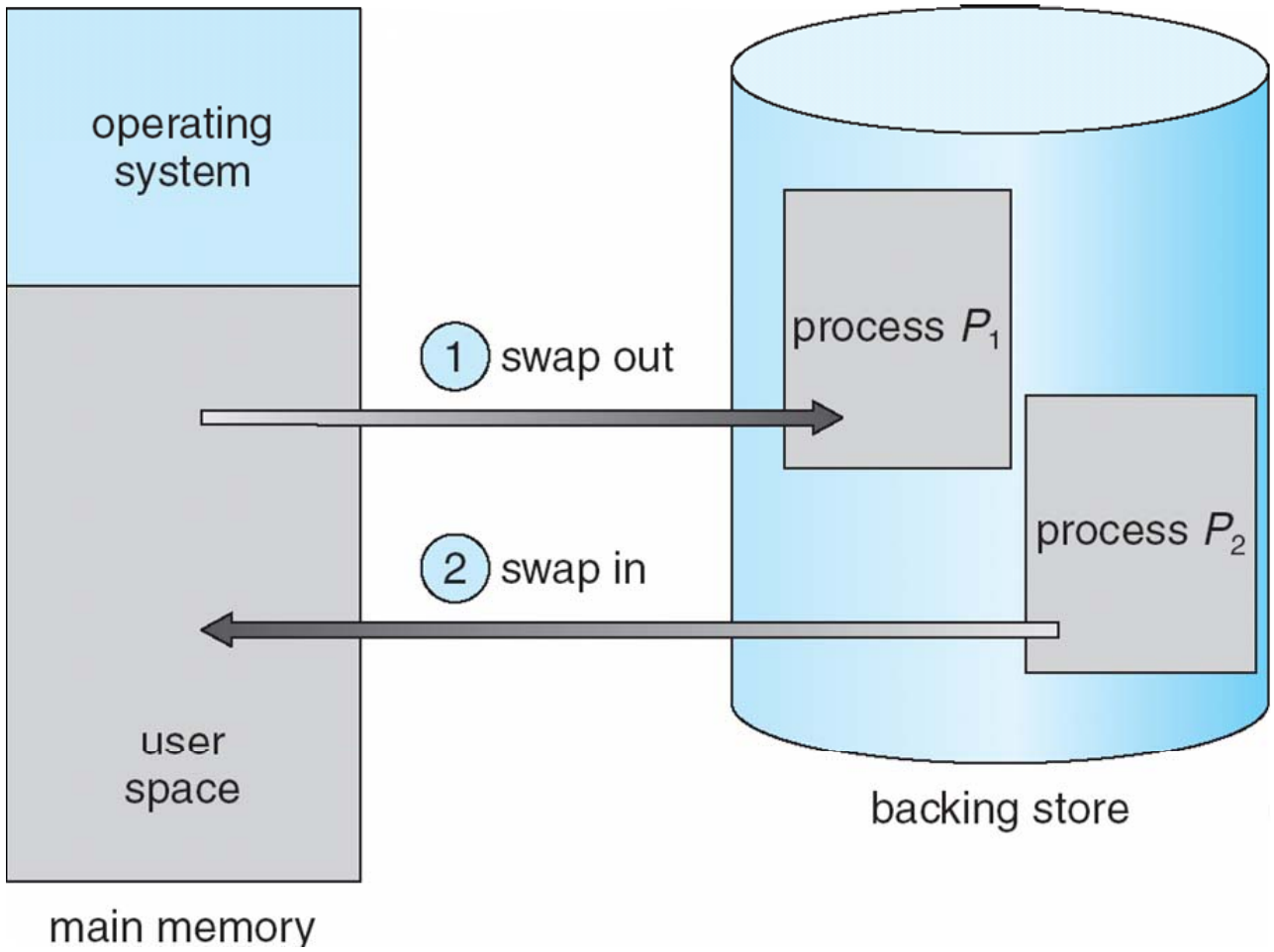
Dynamic relocation using a relocation register

Base and Limit Registers

- A pair of **base** and **limit** registers define the logical address space



Swapping



Memory Management Requirements

- Relocation
  - programmer cannot know where the program will be placed in memory when it is executed
  - a process may be (often) relocated in main memory due to swapping
  - swapping enables the OS to have a larger pool of ready-to-execute processes
  - memory references in code (for both instructions and data) must be translated to actual physical memory address

Memory Management Requirements

- Protection
  - processes should not be able to reference memory locations in another process without permission
  - impossible to check addresses at compile time in programs since the program could be relocated
  - address references must be checked at run time by hardware

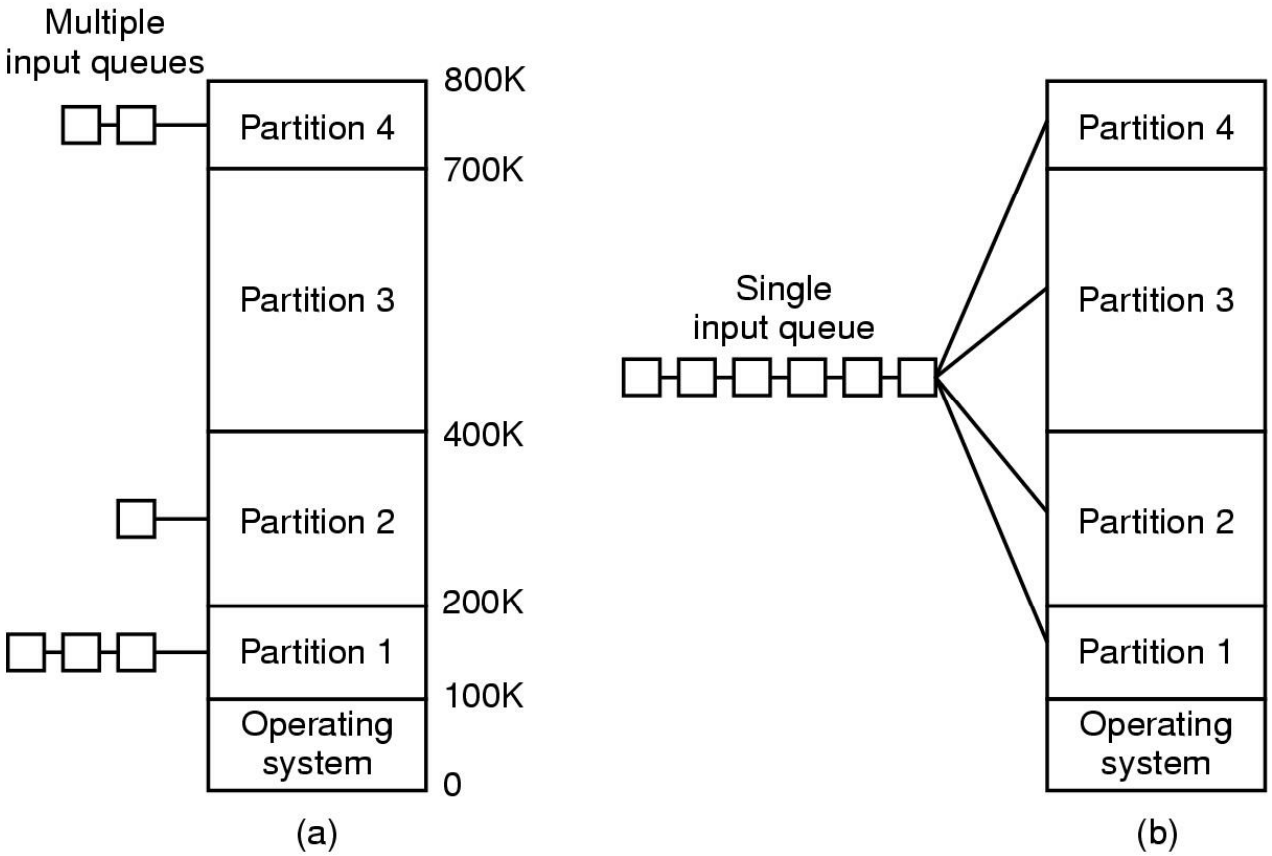
Memory Management Requirements

- Sharing
  - must allow several processes to access a common portion of main memory without compromising protection
    - cooperating processes may need to share access to the same data structure
    - better to allow each process to access the same copy of the program rather than have their own separate copy

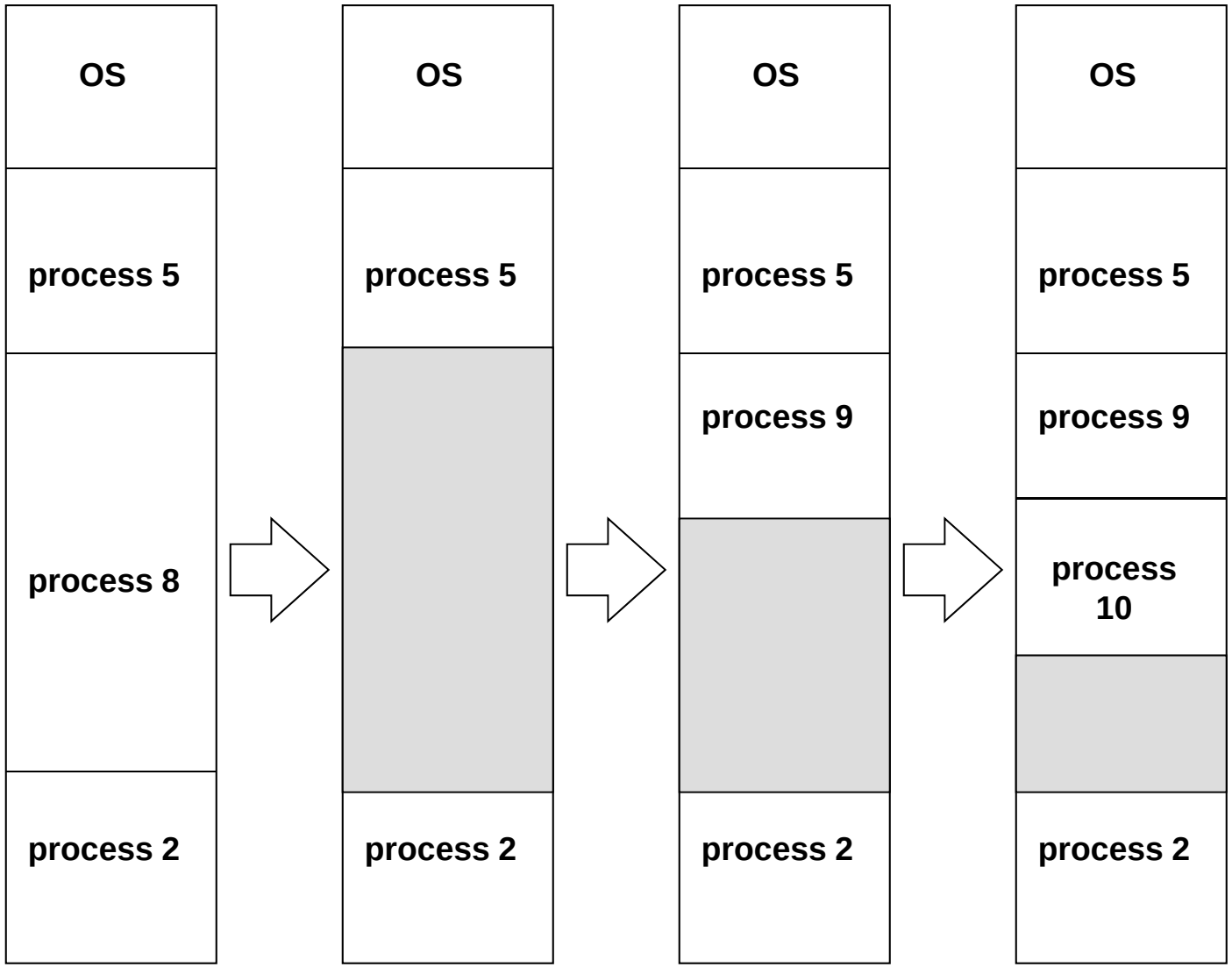
Memory Allocation

- Problem: how to allocate memory for multiple processes (in a multi-programming environment)
- Solutions:
  - Contiguous allocation
    - Fixed partitions
    - Dynamic partitions
  - Paging

Contiguous Allocation (Fixed Partitions)



Contiguous Allocation (Dynamic Partitions)



Information in Memory Map

The information needed for allocation within a two level hierarchy (M1,M2) is as follows:

- OCCUPIED SPACE LIST FOR M1
- AVAILABLE SPACE LIST FOR M1
- DIRECTORY FOR M2

Another Classification of Memory Allocation (ALLOCATION MODES)

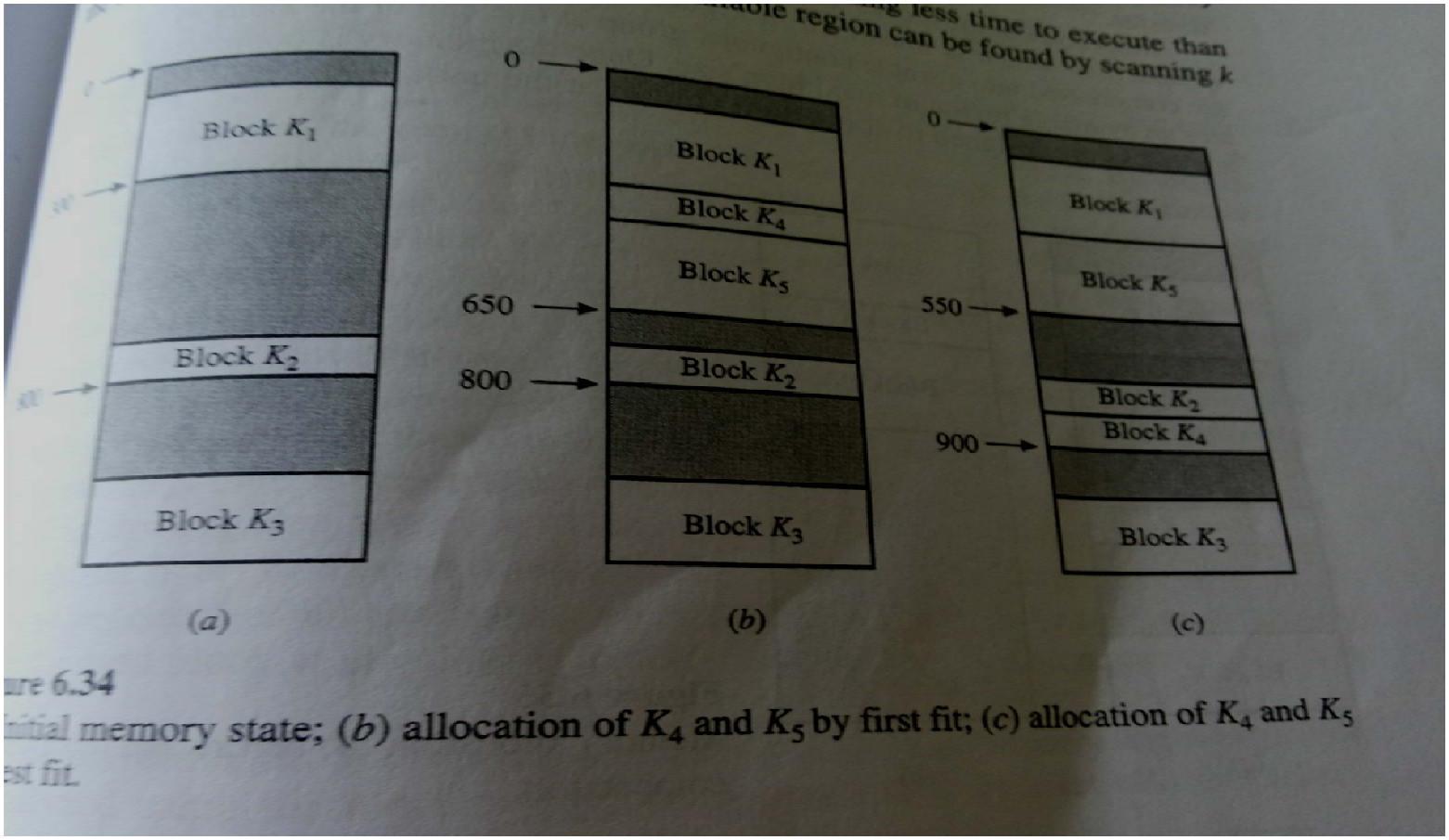
- ***Non-Preemptive allocation:*** *cant make efficient use of memory in all situations. Memory Overflow, that is the rejection of a memory allocation request due to insufficient space. No replacement is allowed here.*
- ***Preemptive allocation:*** *much more efficient use of memory is possible if the occupied space can be relocated to make room for incoming block. Replacement is easily possible here.*

Non-Preemptive Allocation

- Suppose for example at some point in time M1 stores three blocks as in table: (represented by shaded area in figure)
- Two new blocks K4 and K5 of sizes 100 and 250 words are to be assigned to M1.

Region Address	Size(words)
0	50
300	400
800	200

Non-Preemptive Allocation



ure 6.34
itial memory state; (*b*) allocation of K_4 and K_5 by first fit; (*c*) allocation of K_4 and K_5
st fit.

Preemptive Allocation

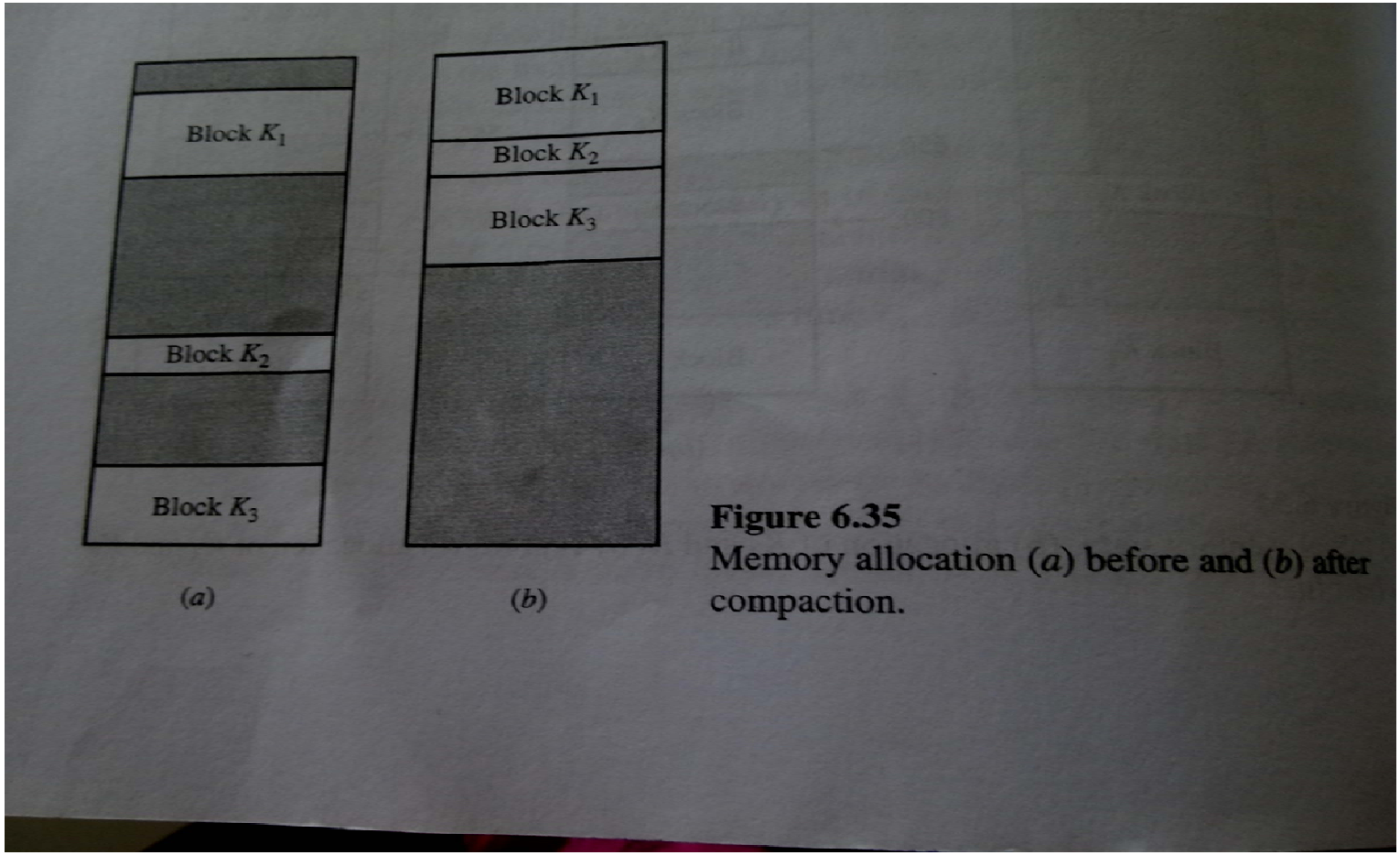


Figure 6.35
Memory allocation (*a*) before and (*b*) after compaction.

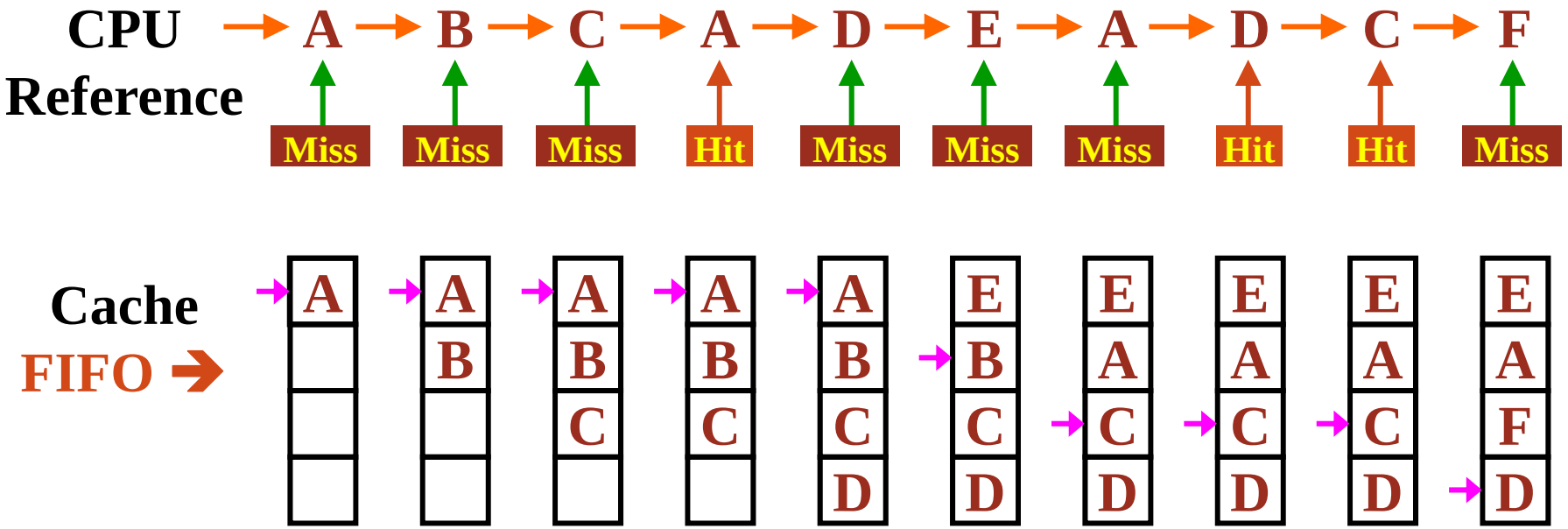
Replacement Algorithm

- When all processes in main memory are blocked, the OS must choose which process to replace
  - A process must be swapped out (to a Blocked-Suspend state) and be replaced by a new process or a process from the Ready-Suspend queue
  - We will discuss later such algorithms for memory management schemes using virtual memory

Replacement Algorithms

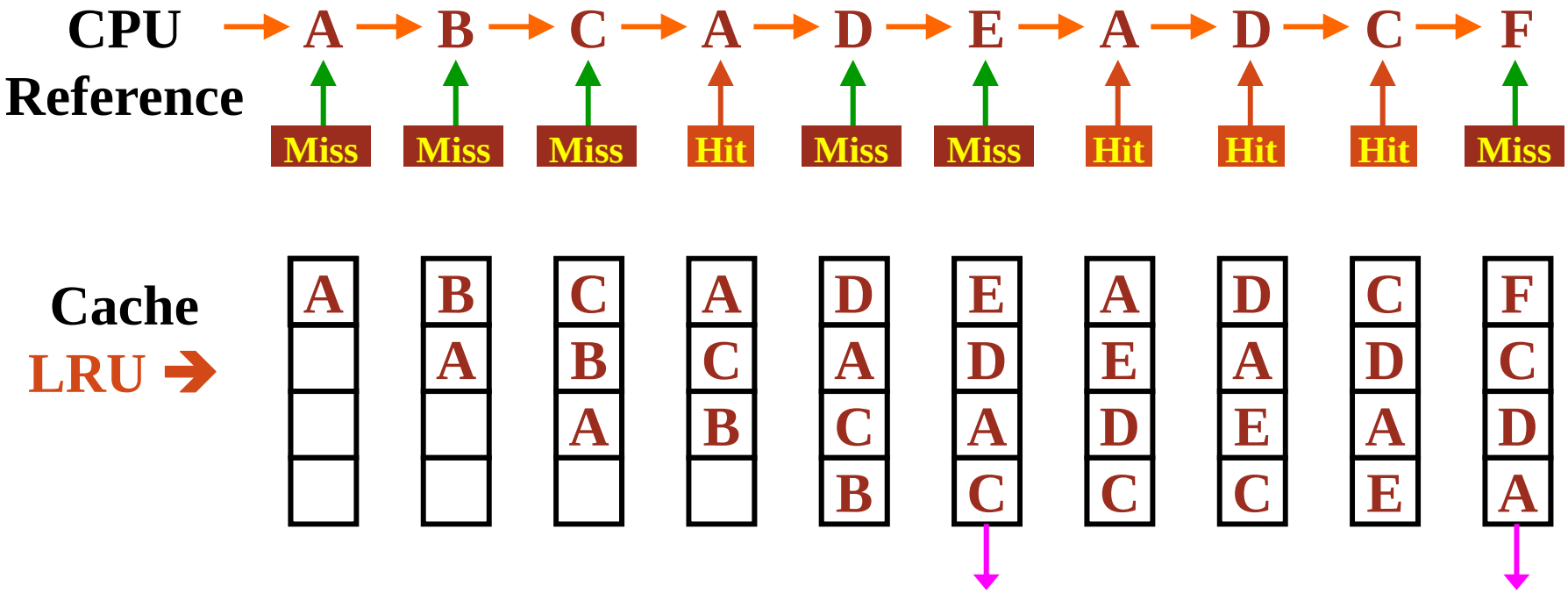
- For Associative & Set-Associative Cache
  Which location should be emptied when the cache is full and a miss occurs?
  - First In First Out (FIFO)
  - Least Recently Used (LRU)
- Distinguish an *Empty* location from a *Full* one
  - Valid Bit

Replacement Algorithms- FIFO



Hit Ratio = 3 / 10 = 0.3

Replacement Algorithms-LRU



Hit Ratio = 4 / 10 = 0.4

Interleaved Memory

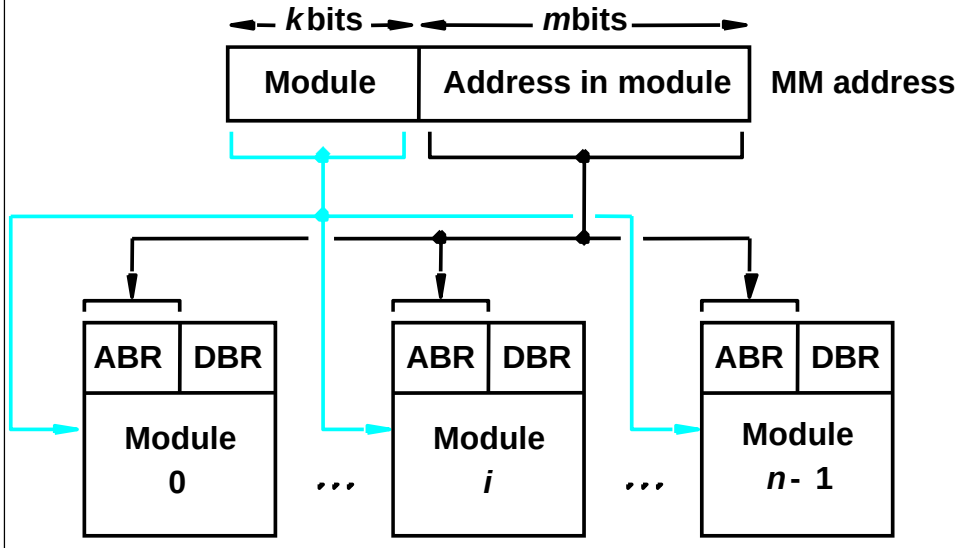
- Two key factors: performance and cost
- Price/performance ratio
- Performance depends on how fast machine instructions can be brought into the processor for execution and how fast they can be executed.
- For memory hierarchy, it is beneficial if transfers to and from the faster units can be done at a rate equal to that of the faster unit.
- This is not possible if both the slow and the fast units are accessed in the same manner.
- However, it can be achieved when parallelism is used in the organizations of the slower unit.

Interleaved Memory

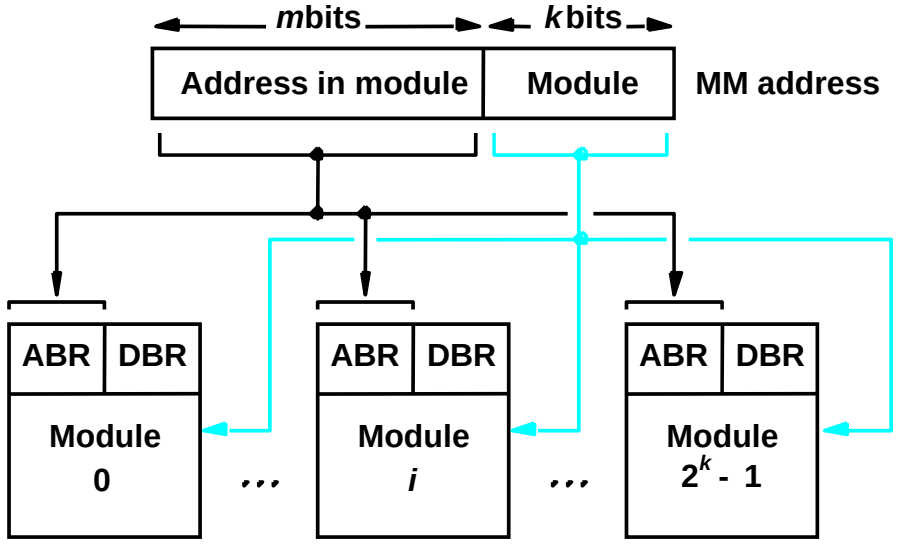
- Main memory is composed of a collection of DRAM memory chips.
- A number of chips can be grouped together to form a memory bank.
- Banks independently service read or write requests.
- K banks can service k requests simultaneously, increasing memory read or write rates by a factor of K.

Interleaving

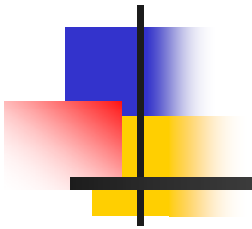
- If the main memory is structured as a collection of physically separated modules, each with its own ABR (Address buffer register) and DBR(Data buffer register), memory access operations may proceed in more than one module at the same time.



(a) Consecutive words in a module

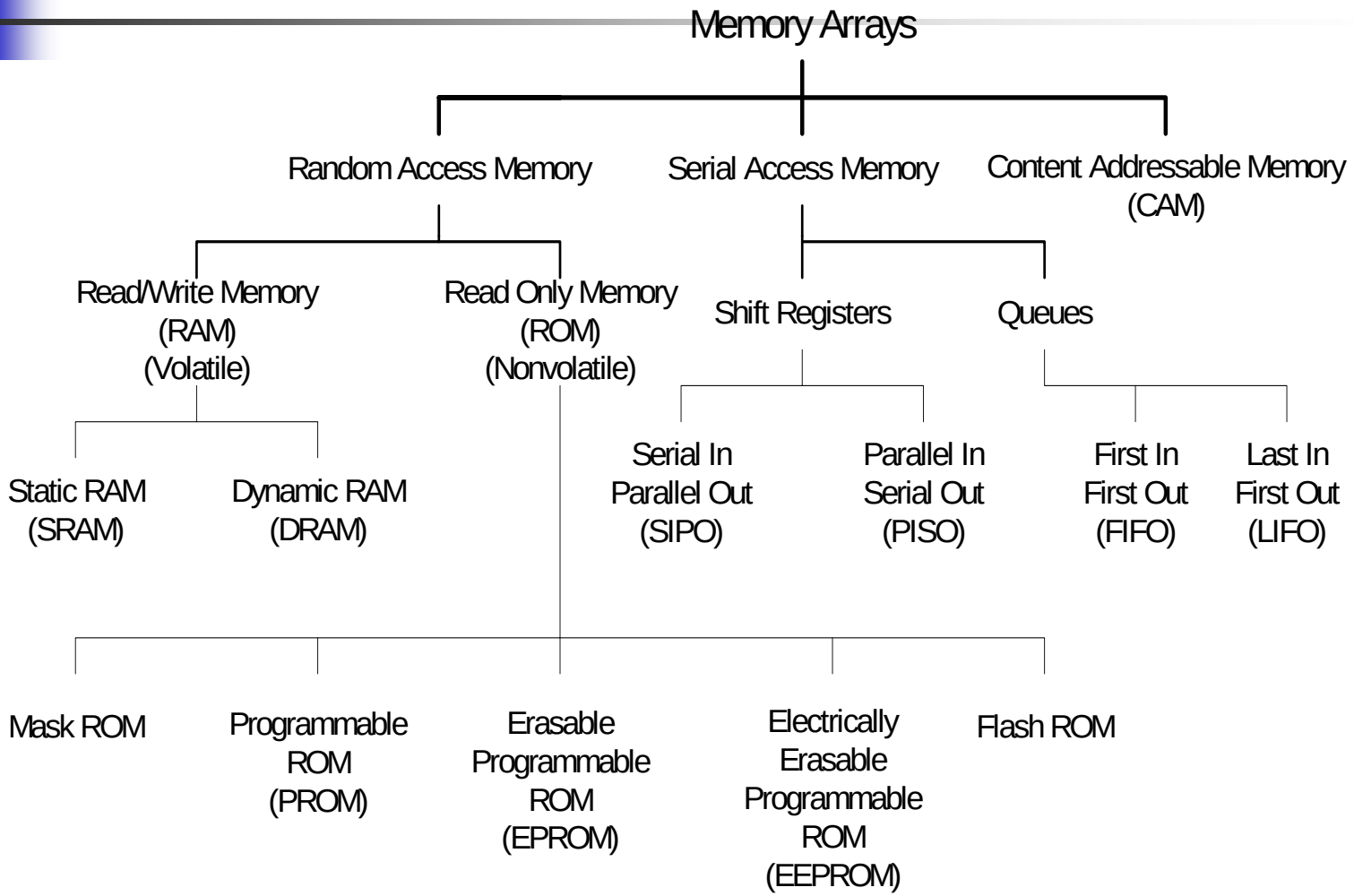


(b) Consecutive words in consecutive modules



Memory Organization

Memory Arrays

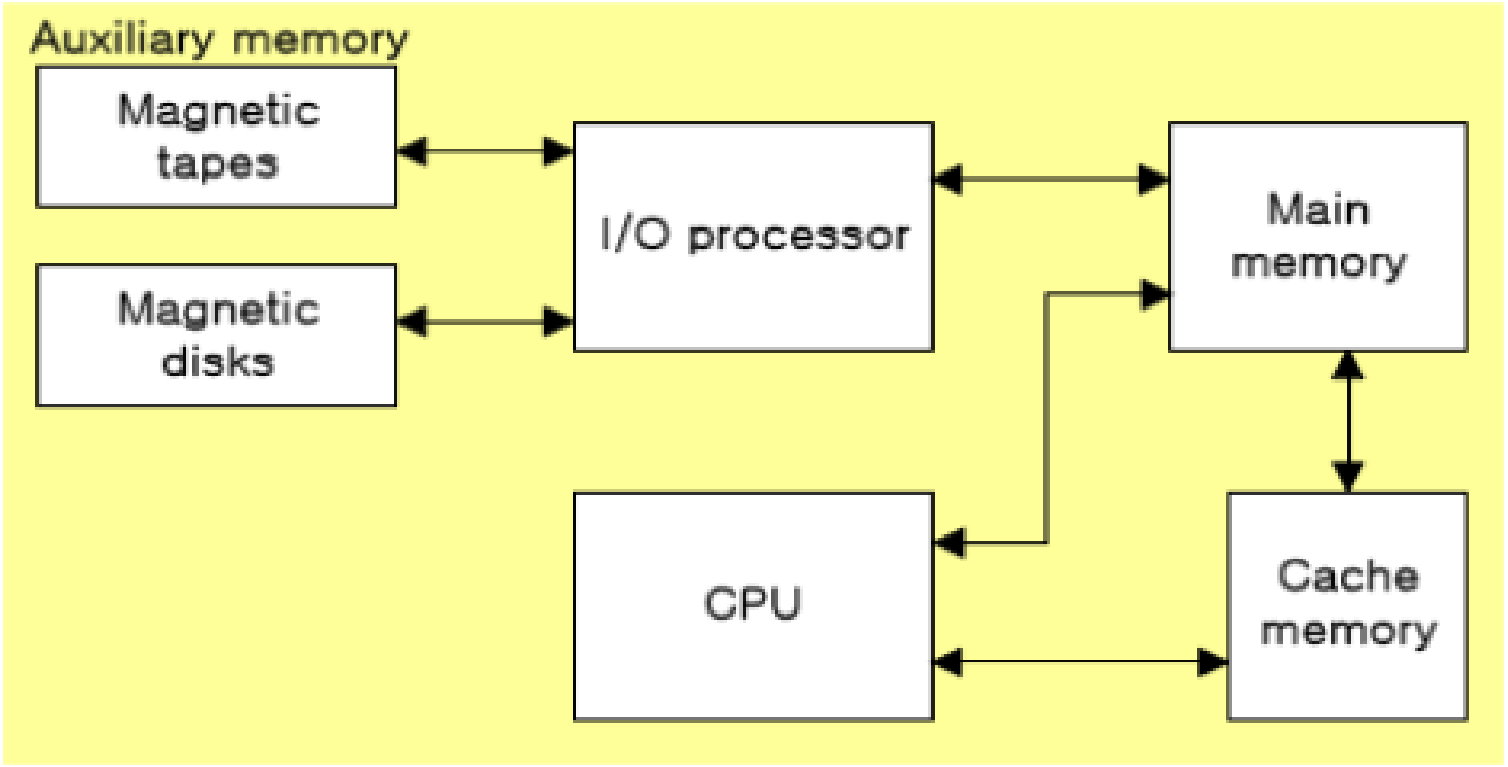


Memory Hierarchy

- The memory unit is an essential component in any digital computer since it is needed for storing programs and data
- Not all accumulated information is needed by the CPU at the same time
- Therefore, it is more economical to use low-cost storage devices to serve as a backup for storing the information that is not currently used by CPU

MEMORY HIERARCHY

Memory Hierarchy is to obtain the highest possible access speed while minimizing the total cost of the memory system



Memory Hierarchy

- Computer Memory Hierarchy is a pyramid structure that is commonly used to illustrate the significant differences among memory types.
- The memory unit that directly communicate with CPU is called the *main memory*
- Devices that provide backup storage are called *auxiliary memory*
- The memory hierarchy system consists of all storage devices employed in a computer system from the slow by high-capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory

Main Memory

Two main types of memory exist: RAM and ROM.

Random access memory (RAM)

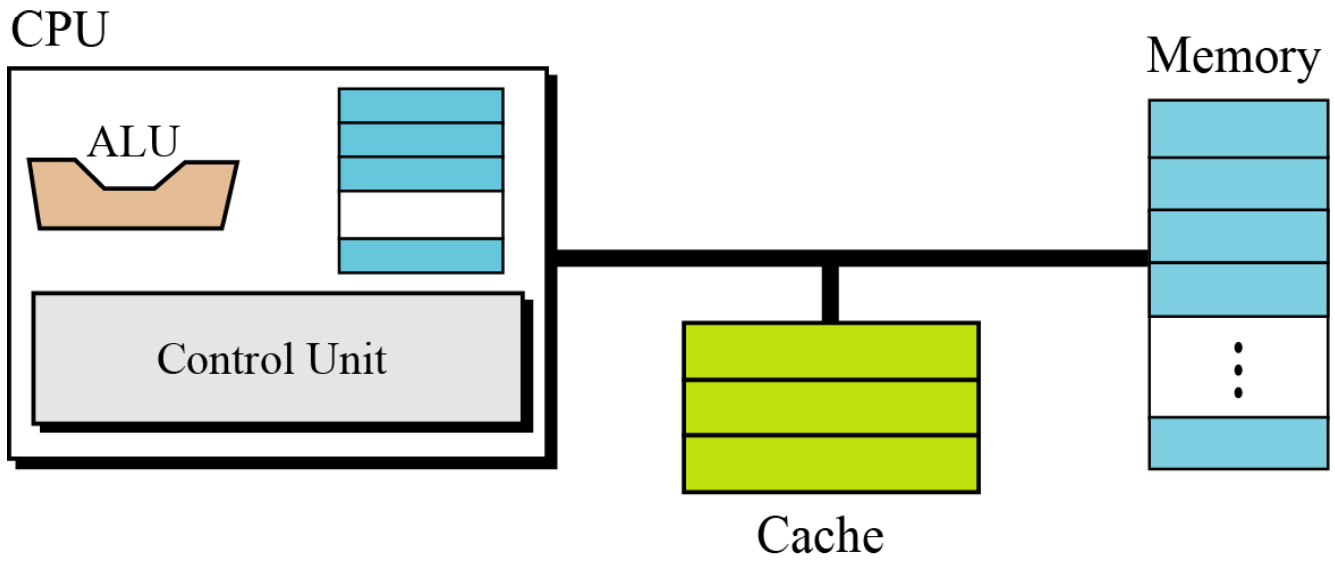
- Static RAM (SRAM)
- Dynamic RAM (DRAM)

Read-only memory (ROM)

- Programmable read-only memory (PROM).
- Erasable programmable read-only memory (EPROM).
- Electrically erasable programmable read-only memory (EEPROM).

Cache memory

Cache memory is faster than main memory, but slower than the CPU and its registers. Cache memory, which is normally small in size, is placed between the CPU and main memory .



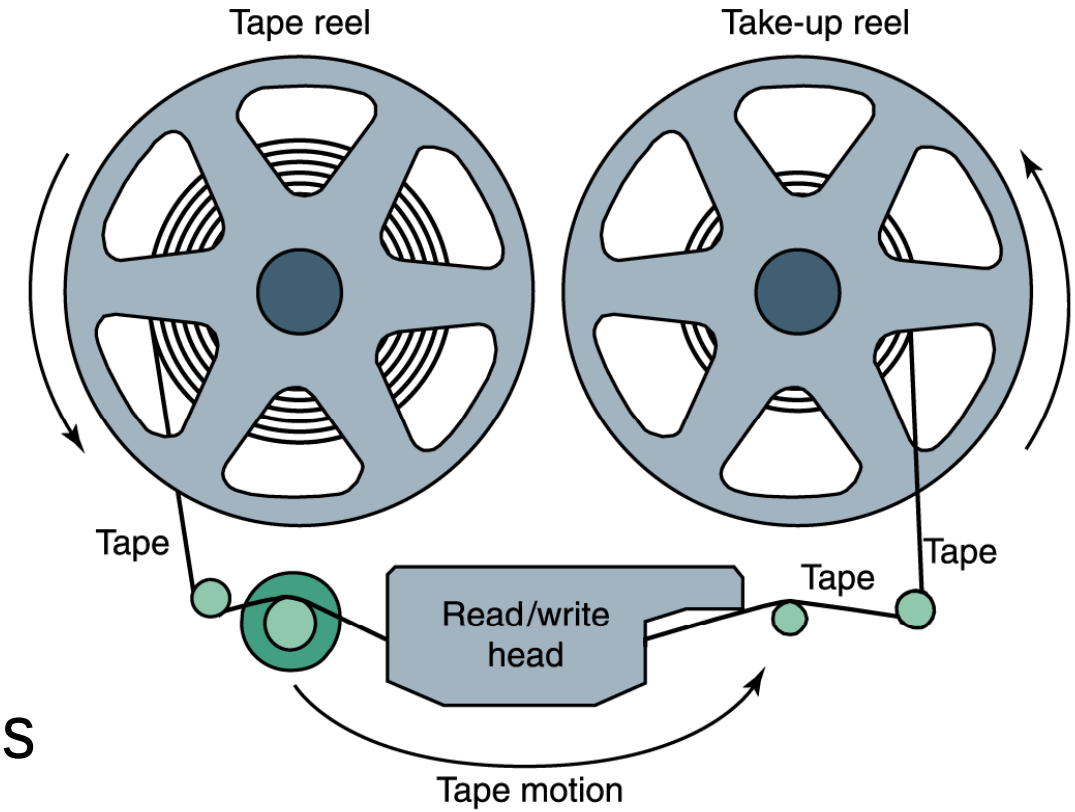
Cache memory

Auxiliary Storage devices

Storage devices, although classified as I/O devices, can store large amounts of information to be retrieved at a later time. They are cheaper than main memory, and their contents are nonvolatile—that is, not erased when the power is turned off. They are sometimes referred to as auxiliary storage devices. We can categorize them as either magnetic or optical.

Magnetic Tape

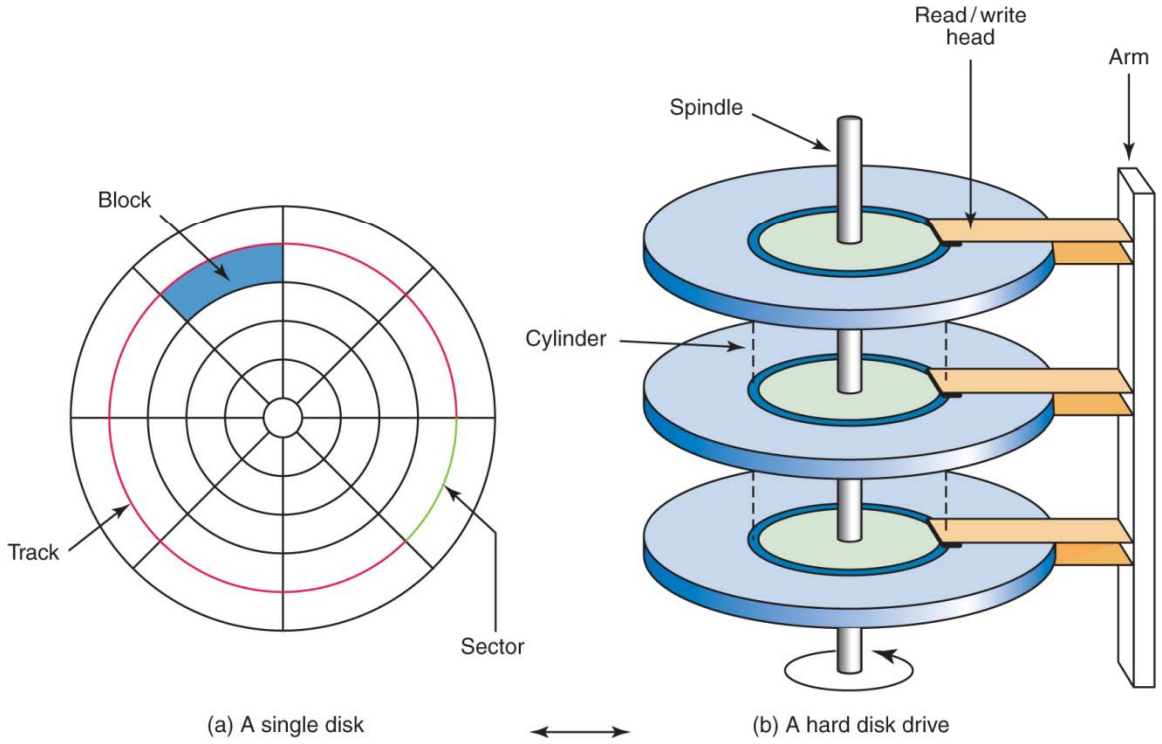
- The first truly mass auxiliary storage device was the magnetic tape drive
- Cassette Tapes are still used for large data backups



A magnetic tape storage mechanism

Magnetic Disks

- A read/write head travels across a spinning magnetic disk, retrieving or recording data



The organization of a magnetic disk

MEMORY DEVICE CHARACTEISTICS

To identify the behavior of various memories certain characteristics are considered. These are as follows-

1. Memory Types : On the basis of their location inside the computer, memory can be placed in four groups :

- **CPU Registers**: these high speed registers in the CPU work as memory for temporary storage of instruction and data. The data can be read from or written into a register within a single clock cycle.
- **Main Memory or Primary Memory**: Main memory size is large and fast accessing external memory stores programs and data. This memory is slower compared to CPU registers because of main memory has large storage capacity is typically 1 and 2^{10} megabyte.

Continued…

- **Secondary Memory**: This memory has larger in capacity but slower than main memory. Secondary memory stores system programs, large data files and like the data are not continually required by the CPU. It also acts as an overflow memory when the capacity of the main memory is exceeded. Information in secondary storage is accessed indirectly via input output processor that transfer information between main and secondary memory.
- **Cache Memory**: Most computers have another level of IC memory called cache memory. It is placed between CPU registers and main memory. A cache memory capacity is less than that of main memory but it is faster than that of main memory because some or all of it can reside on the same IC as the CPU. Cache memories are essential components of high performance computers.

Continued…

2. **Location**: The memory which is inside the processor called the internal memory. The memory which is external to the processor is known as external memory.

3. **Access Method**: Each memory is a collection of various memory location. Accessing the memory means finding and reaching desired location and than reading information from memory location. The information from locations can be accessed as follows:

Continued…

(I)Random access

(ii)Sequential access

(iii)Direct access.

- **Random Access**: It is the access mode where each memory location has a unique address. Using these unique addresses each memory location can be addressed independently in any order in equal amount of time. Generally, main memories are random access memories.

Continued…

- **Sequential Access**: If storage locations can be accessed only in a certain predetermined sequence, the access method is known as serial or sequential access.
- **Direct Access**: In this access information is stored on tracks and each track has a separate read/write head. This features makes it a semi random mode which is generally used in magnetic disks.

Continued…

- **Volatile Memories**: The memories that looses their contents when the power is turned off called volatile memories.
- **Non-volatile Memories**: The memories that do not loose their contents when the power is removed called Non-volatile memories.

TYPES OF RAM

- **Static RAM**: It consist of internal latches that store the binary information. The stored information remains valid as long as power is applied to the unit.
- **Dynamic RAM**: It stores the binary information in the form of electric charges on capacitors. The capacitors are provided inside the chip by MOS transistors. The stored charge on the capacitor tends to discharge with time and the capacitors must be periodically recharged by refreshing the dynamic memory.

ROM & THEIR TYPES

- **Read only memory**: It is non-volatile memory, which retains the data even when power is removed from this memory. Programs and data that can not be altered are stored in ROM. The required paths in a ROM may be programmed in four different ways:

1. **Mask Programming**
2. **Programmable Read only memory(PROM)**
3. **Erasable PROM**
4. **Electrically Erasable PROM**
5. **Flash ROM**

Memory device characteristics (Important parameters)

- Cost
  - *Cost (c) = C (total cost of memory system) / S (storage capacity)*
- Performance
  - *Read access time (t_A)*
  - *Write access time*
  - Calculated from time memory receives a read request to the time at which requested information becomes available at output lines
- Other characteristics
  - *Maximum amount of information transferred to or from memory per unit time (data transfer rate or bandwidth bM bits per sec.)*
- Access time, t_A
  - *Time spent to transfer a data to the output after receiving a read-request.*

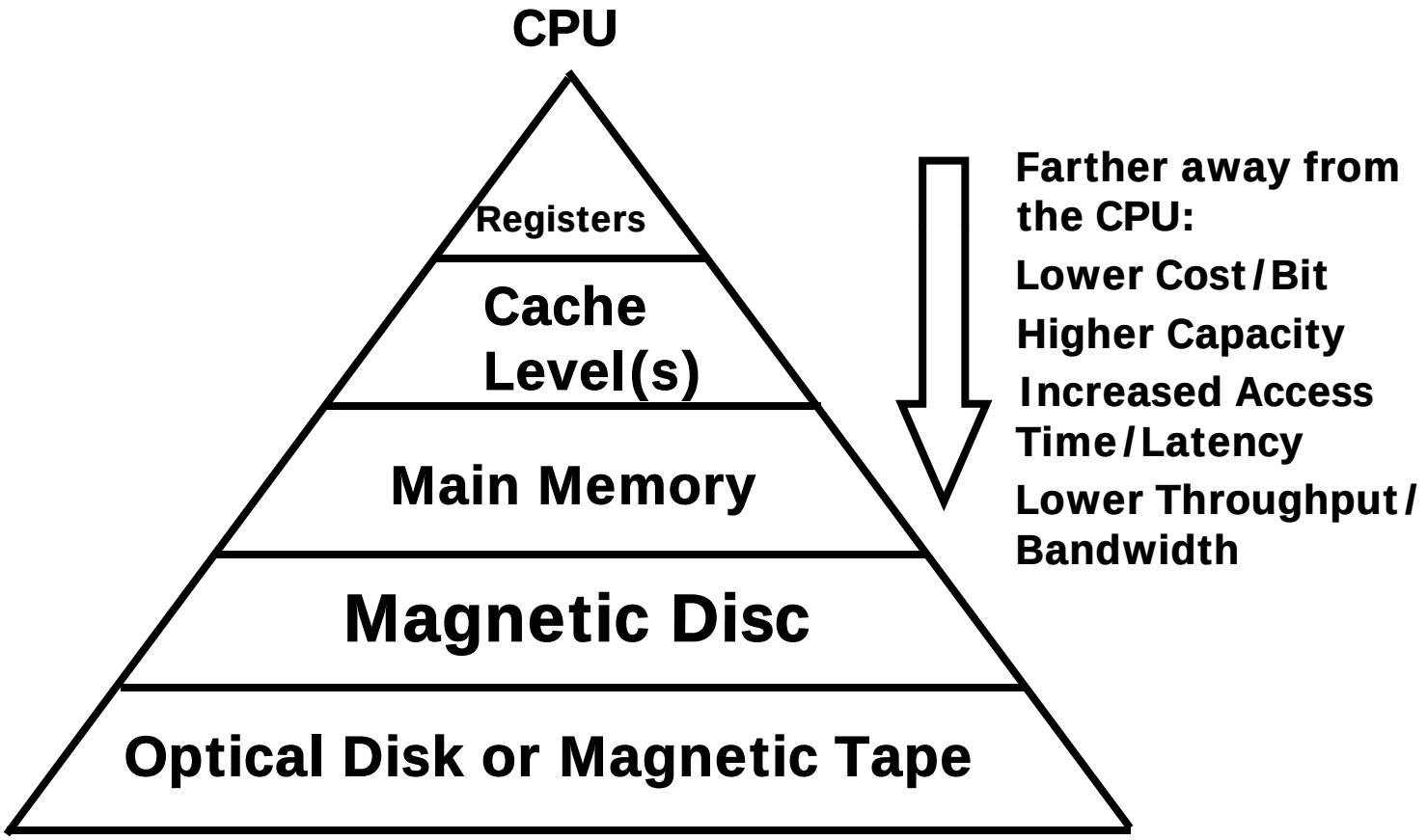
Speed, Size , Cost

- Fastest is SRAM
  - But expensive
  - Costly
- Alternative is DRAM
  - For large volume of data it becomes Costly
- A huge capacity and low cost alternate is
  - Magnetic Disk
  - Magnetic tape
  - Optical Disk

The Goal: Large, Fast, Cheap Memory !!!

- Fact
  - Large memories are slow
  - Fast memories are small
- How do we create a memory that is large, cheap and fast (most of the time) ?
  - Hierarchy
  - Parallelism

Levels of the Memory Hierarchy



Main Memory

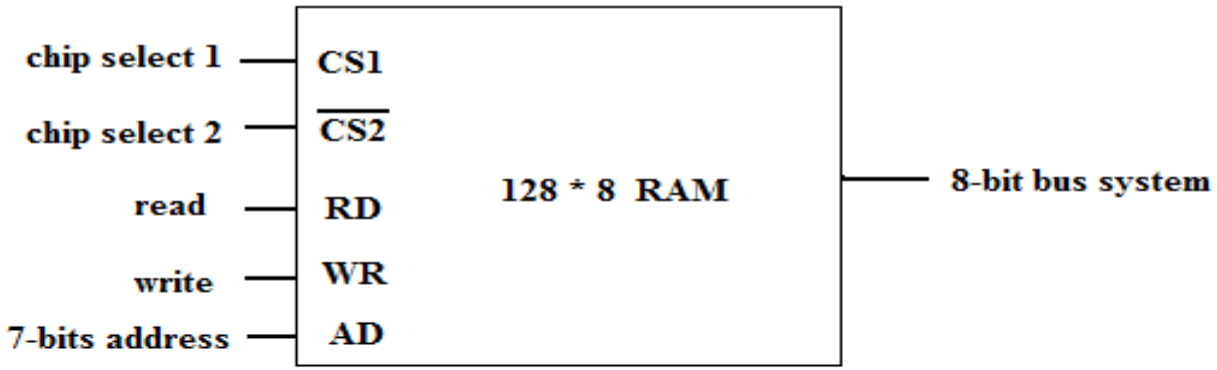
- Most of the main memory in a general purpose computer is made up of RAM integrated circuits chips, but a portion of the memory may be constructed with ROM chips

- RAM– Random Access memory
  - Integrated RAM are available in two possible operating modes, S*tatic and Dynamic*
- ROM– Read Only memory

Main Memory available in chips

- A RAM & ROM chips are better suited for communication with the CPU if it has one or more control inputs that select the chips when needed.
- Available in many sizes.
- If the memory needed for computer is larger than the capacity of one chip, it is necessary to combine a no. of chips to form required size.
- The Block diagram of a RAM chip is shown in next slide, the capacity of the memory is 128 words of 8 bits (one byte) per word.

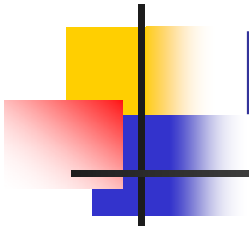
RAM



CS1	$\overline{CS2}$	RD	WD	Memory Function	State of data bus
0	0	*	*	Inhibit	High-impedance
0	1	*	*	Inhibit	High-impedance
1	0	0	0	Inhibit	High-impedance
1	0	0	1	Write	Input data to RAM
1	0	1	*	Read	Output data from RAM
1	1	*	*	Inhibit	High-impedance

RAM

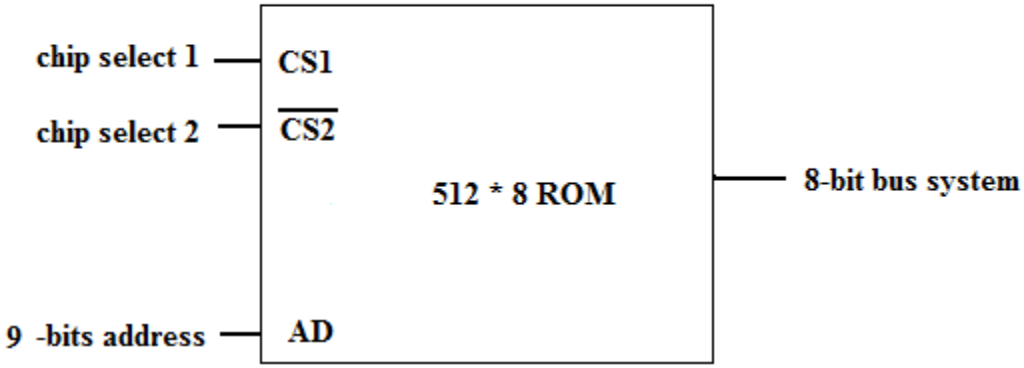
- Read/write memory, that initially doesn't contain any data.
- The computing system that it is used in usually stores data at various locations to retrieve it latter from these locations.
- Its data pins are bidirectional (data can flow into or out of the chip via these pins), as opposite to those of ROM that are output only.
- Bidirectional buses can be constructed with three sets buffers which can be placed in three output states either 0 or 1 and high impedance state.
- It loses its data once the power is removed, so it is a **volatile** memory
- It has a directional select signal R/W'; When R/W'=1, the chip outputs data to the rest of the circuit; when R/W' = 0 it inputs data from the rest of the circuit.

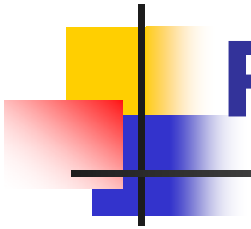


ROM

- ROM is used for storing programs that are **Permanently** resident in the computer and for tables of constants that do not change in value once the production of the computer is completed .
- The ROM portion of main memory is needed for storing an initial program called *bootstrap loader,* witch is to start the computer software operating when power is turned off.

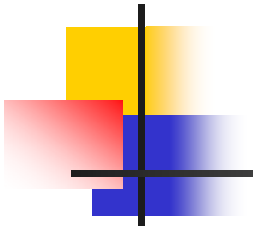
ROM


chip select 1
CS1
chip select 2
CS2
512 * 8 ROM
8-bit bus system
9 -bits address
AD



ROM

- Data is programmed into the chip using an external ROM programmer
- The programmed chip is used as a component into the circuit
- Since ROM can only read, the data bus can only be in output mode.
- For the same size chip, it is possible to have more bits of ROM than of RAMs, because internal binary cells of ROM occupy less space than in RAM.
- When power is removed from a ROM chip, the information is not lost, so it is a nonvolatile type of memory.
- Nine address lines are required here.
- No need of READ/WRITE control pins as ROM can only perform read operation.



ROM Types

- **Masked ROM** – programmed with its data when the chip is fabricated
- **PROM** – programmable ROM, by the user using a standard PROM programmer, by burning some special type of fuses. Once programmed will not be possible to program it again
- **EPROM** – erasable ROM; the chip can be erased and chip reprogrammed; programming process consists in charging some internal capacitors; the UV light (method of erase) makes those capacitors to leak their charge, thus resetting the chip
- **EEPROM** – Electrically Erasable PROM; it is possible to modify individual locations of the memory, leaving others unchanged; one common use of the EEPROM is in BIOS of personal computers.

Memory Address Map

- Memory Address Map is a pictorial representation of assigned address space for each chip in the system
- To demonstrate an example, assume that a computer system needs 512 bytes of RAM and 512 bytes of ROM
- The RAM have 128 byte and need seven address lines, where the ROM have 512 bytes and need 9 address lines

Memory Address Map

Component	Hexadecimal Address	10	9	8	7	6	5	4	3	2	1
RAM1	0000-007F	0	0	0	*	*	*	*	*	*	*
RAM2	0080-00FF	0	0	1	*	*	*	*	*	*	*
RAM3	0100-017F	0	1	0	*	*	*	*	*	*	*
RAM4	0180-01FF	0	1	1	*	*	*	*	*	*	*
ROM	0200-03FF	1	*	*	*	*	*	*	*	*	*

Memory Address Map

- The hexadecimal address assigns a range of hexadecimal equivalent address for each chip
- Line 8 and 9 represent four distinct binary combination to specify which RAM we chose
- When line 10 is 0, CPU selects a RAM. And when it's 1, it selects the ROM

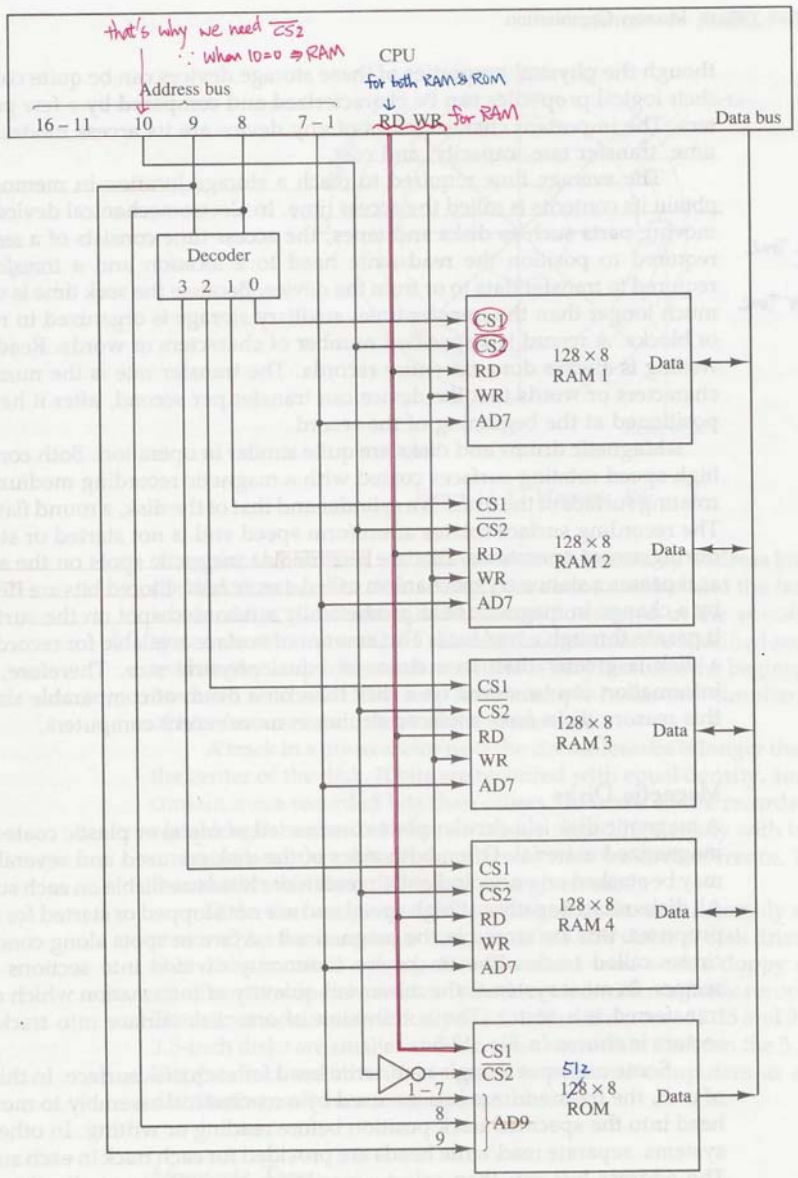


Figure 12-4 Memory connection to the CPU. RAM & ROM connection

Memory Interleaving Concept (p324)

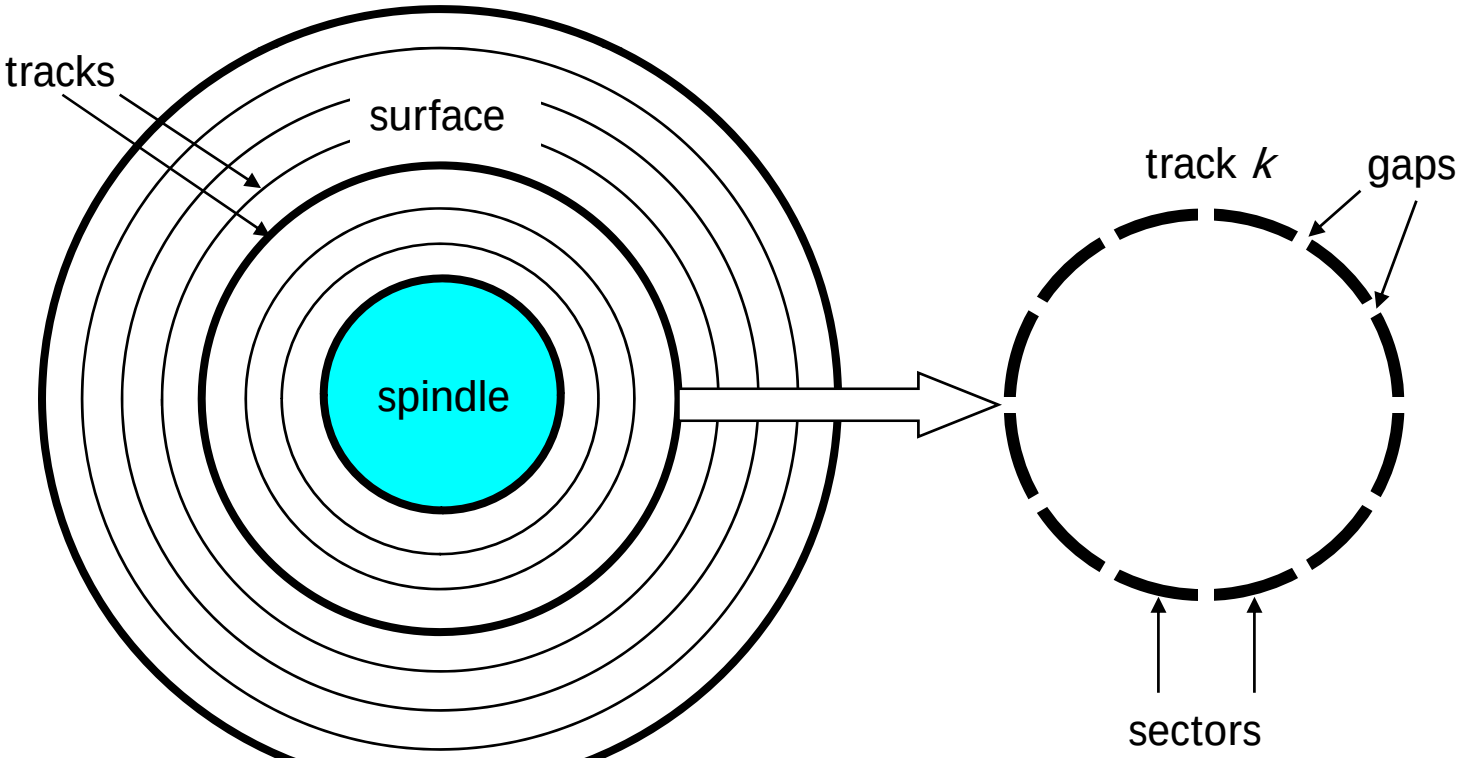
Auxiliary Memory

- The main memory construction is costly. Therefore, it has to be limited in size. The main memory is used to store only those instructions and data which are to be used immediately. However, a computer has to store a large amount of information. The bulk of information is stored in the auxiliary memory. This is also called backing storage or secondary storage. They include hard disk, floppy disks, CD-ROM, USB flash drives, etc.
- When the electricity supply to the computer is off, all data stored in the primary storage is destroyed. On the other hand, this is not true for secondary storage. The data stored in secondary storage can be stored for the desired time.

SOME
T I P S
FOR
AUXILIARY MEMORY

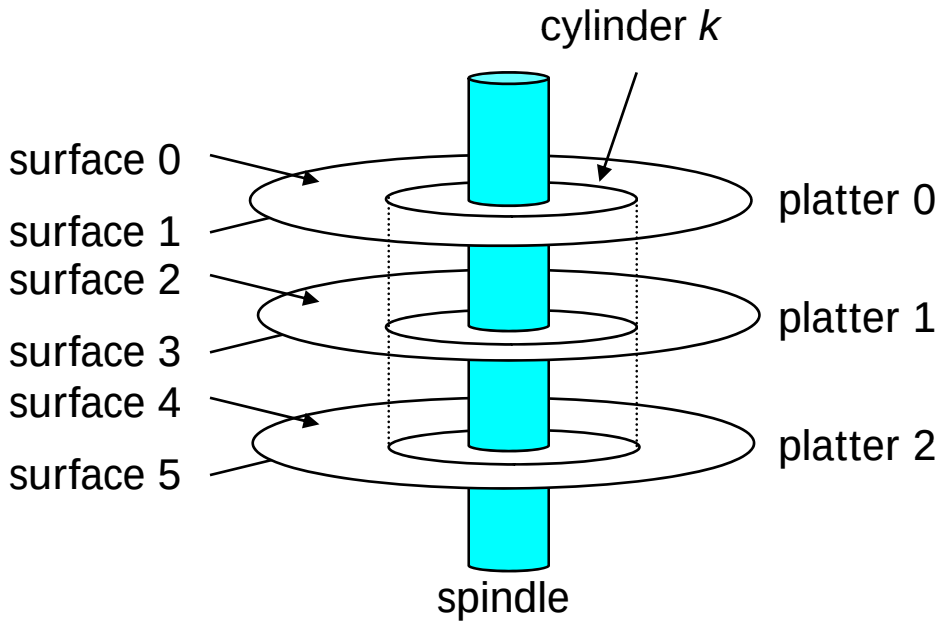
Disk Geometry

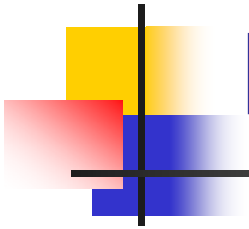
- Disks consist of platters, each with two surfaces.
- Each surface consists of concentric rings called tracks.
- Each track consists of sectors separated by gaps.



Disk Geometry (Multiple-Platter View)

- Aligned tracks form a cylinder.





Disk storage

- Disks are used to store data, applications software and operating systems software. Whereas the primary form of storage in the early days of computing was magnetic tape, this has been replaced by predominantly disk based medium today. The reasons for this trend has been
  - decreasing cost per bit
  - reliability
  - reduced access times
  - higher transfer rates (more data per second)
  - reduced size and power requirements
  - increased capacity
- One trend during the past few years is a move to optical storage medium. Many software companies offer both operating systems software and application software on optical medium (CDROM or DVDROM)

Disk storage technology

- Disk storage systems work on **magnetic principles**.
  - In magnetism, there are two opposing polarities called *poles*, the north and the South pole. Opposite polarity attracts, whilst like polarity repels.
- In computers, data is represented in binary format.
  - Binary data has two states, a 1 or a 0. It just so happens that magnetism also has two states, north and south, so in effect, magnetism is a good way of storing data also
- A rotating disk is coated with very fine ferrous oxide particles, each of which act and behave like little magnets
- All that is required now is a mechanism of converting the digital data of 0's and 1's into magnetic states of north and south poles.
- In a storage disk drive, the mechanism which performs the function of converting the digital 0's and 1's into magnetic states which can magnetize the surface areas of the disk is called the *write head*. A similar head, called the *read head*, is used to detect the magnetic states on the surface of the disk and convert them back into digital states

Disk Capacity

- Capacity: maximum number of bits that can be stored.
  - Vendors express capacity in units of gigabytes (GB), where 1 GB = 10^9.
- Capacity is determined by these technology factors:
  - Recording density (bits/in): number of bits that can be squeezed into a 1 inch segment of a track.
  - Track density (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment.
  - Areal density (bits/in2): product of recording and track density.
- Modern disks partition tracks into disjoint subsets called recording zones
  - Each track in a zone has the same number of sectors, determined by the circumference of innermost track.
  - Each zone has a different number of sectors/track

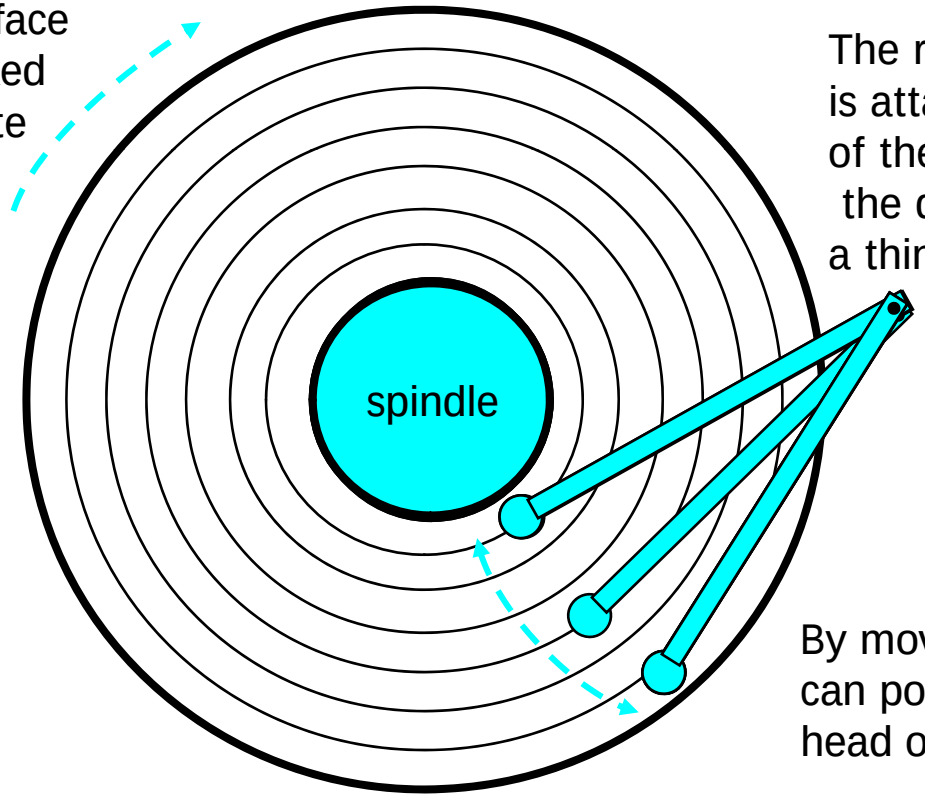
Computing Disk Capacity

- Capacity = (# bytes/sector) x (avg. # sectors/track) x
- (# tracks/surface) x (# surfaces/platter) x
- (# platters/disk)
- Example:
  - 512 bytes/sector
  - 300 sectors/track (on average)
  - 20,000 tracks/surface
  - 2 surfaces/platter
  - 5 platters/disk

- Capacity = 512 x 300 x 20000 x 2 x 5
  = 30,720,000,000
  = 30.72 GB

Disk Operation (Single-Platter View)

The disk surface spins at a fixed rotational rate

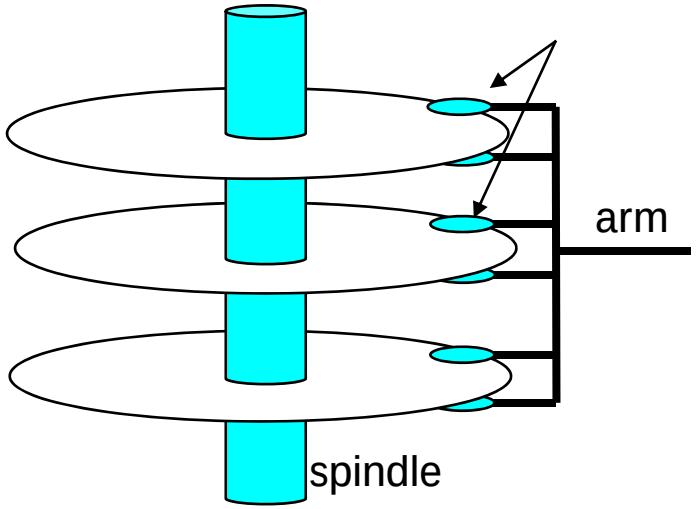


The read/write *head* is attached to the end of the *arm* and flies over the disk surface on a thin cushion of air.

By moving radically, the arm can position the read/write head over any track.

Disk Operation (Multi-Platter View)

- read/write heads move in unison from cylinder to cylinder



Disk Access Time

- Average time to access some target sector approximated by :
  - Access = Tavg seek + Tavg rotation + Tavg transfer
- Seek time (Tavg seek)
  - Time to position heads over cylinder containing target sector.
  - Typical Tavg seek = 9 ms
- Rotational latency (Tavg rotation)
  - Time waiting for first bit of target sector to pass under r/w head.
  - Tavg rotation = 1/2 x 1/RPMs x 60 sec/1 min
- Transfer time (Tavg transfer)
  - Time to read the bits in the target sector.
  - Tavg transfer = 1/RPM x 1/(avg # sectors/track) x 60 secs/1 min.

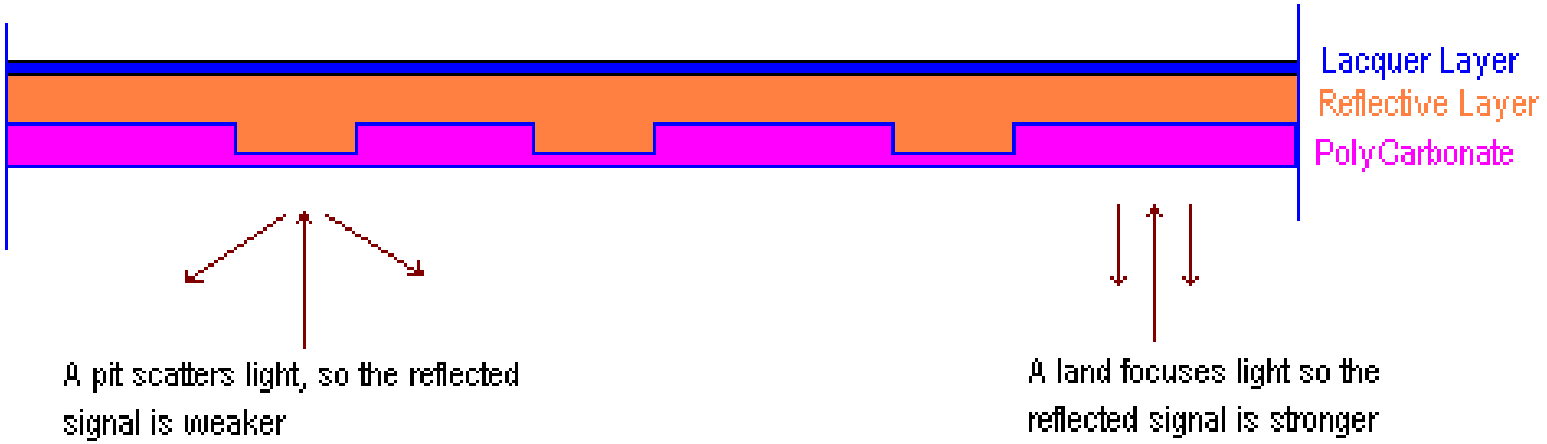
Disk Access Time Example

- Given:
  - Rotational rate = 7,200 RPM
  - Average seek time = 9 ms.
  - Avg # sectors/track = 400.
- Derived:
  - Tavg rotation = 1/2 x (60 secs/7200 RPM) x 1000 ms/sec = 4 ms.
  - Tavg transfer = 60/7200 RPM x 1/400 secs/track x 1000 ms/sec = 0.02 ms
  - Taccess = 9 ms + 4 ms + 0.02 ms
- Important points:
  - Access time dominated by seek time and rotational latency.
  - First bit in a sector is the most expensive, the rest are free.
  - SRAM access time is about 4 ns/doubleword, DRAM about 60 ns
    - Disk is about 40,000 times slower than SRAM,
    - 2,500 times slower then DRAM.

Optical Disks

- The data is accessed from the underside of the CD-ROM. According to the initial specification devised by Philips and Sony, data is stored in a single track which is embedded into a polycarbonate material
- The track starts at the inner of the disk, and ends at the outer radius of the disk. The track length is thus one long tightly wound spiral, the equivalent of over 3 miles long
- The track is comprised of indentations or bumps which are created on a master disc. This master disc is then used to create the actual CDROM's which are shipped to customers. This technique is similar to the technique which was used to create audio records.
- The laser beam is shone onto the surface of the disk. Data is stored as a sequence of surface variations called *lands* (flat surface) and *pits* (bumps or holes). The light is scattered by the pits and reflected by the lands. These two variations encode the binary 0's and 1's. The laser beam is moved to follow the spiral track of the data stored on the disk, detected the pits and lands as it follows the spiral track.
- A light sensitive diode picks up the reflected laser light from the surface of the disk, and converts the light to digital data.

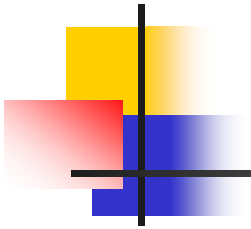
Optical Disk Technology



- The pit and lands vary in length. The speed of rotation of the CD is adjusted so that the speed of the pits and lands passing above the laser is always the same speed (slower when it is in the inner and faster on the outer). This is called *Constant Linear Velocity*.
- The amount of time that occurs between a pit and a land is measured and converted into digital data. Note that the information is stored permanently as pits and lands on the CD-ROM. It cannot be changed once the CD-ROM is mastered, this is why its called CD-ROM.
- Single speed CD-ROM has a transfer speed of 150KB/s

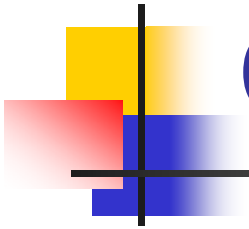
DVD (Digital Versatile Disk)

- This new standard offers higher data storage and faster data transfers than existing CD-ROM. Differences between DVD and CD-ROM
  - standard DVD holds 4.7GB per layer, dual layer single sided DVD holds 8.5GB on a single side
  - error correction is more robust than CD-ROM
  - every DVD is a bonded disc, composed of two 0.6mm substrates joined together
  - smaller pits are used and tracks are closer together than CD-ROM
  - DVD uses MPEG2 compression for high quality full screen pictures
  - a single layer DVD can hold a two hour 13 minute movie, with full digital sound in three languages
  - dual layer single sided DVD can hold a movie greater than 4 hours long
  - DVD-ROM drives have a much faster transfer rate than CD-ROM drives
  - DVD-ROM drives will read and play existing CD-ROM's and CDA disks
- DVD is ideal for companies that wish to deliver enhanced training that includes high quality video. It has both the storage capacity and transfer speeds to support this type of application. In addition, movie companies are producing full length movie pictures on DVD, as MPEG-2 compression provides full screen high quality definition with multiple language track capability.



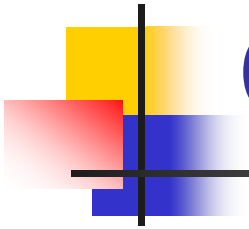
Tape Storage Systems

- Magnetic Reel and Cartridge Tape
- Digital Audio Tape (DAT)
- Digital Data Storage (DDS)
- Digital Linear Tape (DLT)



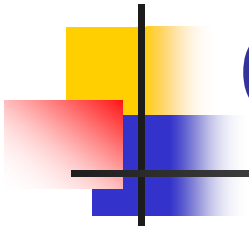
Cache memory

- If the active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced,
- Thus reducing the total execution time of the program
- Such a fast small memory is referred to as cache memory
- The cache is the fastest component in the memory hierarchy and approaches the speed of CPU component



Cache memory

- When CPU needs to access memory, the cache is examined
- If the word is found in the cache, it is read from the fast memory
- If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word



Cache memory

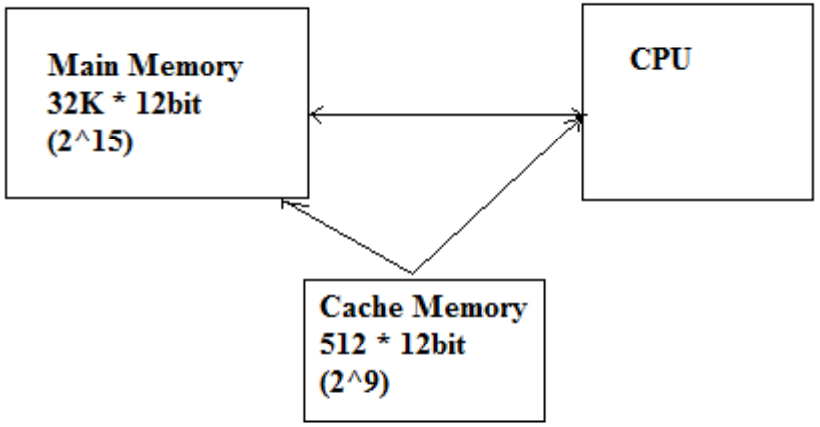
- The performance of cache memory is frequently measured in terms of a quantity called **hit ratio**
- When the CPU refers to memory and finds the word in cache, it is said to produce a **hit**
- Otherwise, it is a **miss**
- **Hit ratio = hit / (hit+miss)**

Cache memory

- The basic characteristic of cache memory is its fast access time,
- Therefore, very little or no time must be wasted when searching the words in the cache
- The transformation of data from main memory to cache memory is referred to as a **mapping** process, there are three types of mapping:
  - Associative mapping
  - Direct mapping
  - Set-associative mapping

Cache memory

- To help understand the mapping procedure, we have the following example:



Memory Address Map

- Memory Address Map is a pictorial representation of assigned address space for each chip in the system
- To demonstrate an example, assume that a computer system needs 512 bytes of RAM and 512 bytes of ROM
- The RAM have 128 byte and need seven address lines, where the ROM have 512 bytes and need 9 address lines

Memory Address Map

Component	Hexadecimal Address	10	9	8	7	6	5	4	3	2	1
RAM1	0000-007F	0	0	0	*	*	*	*	*	*	*
RAM2	0080-00FF	0	0	1	*	*	*	*	*	*	*
RAM3	0100-017F	0	1	0	*	*	*	*	*	*	*
RAM4	0180-01FF	0	1	1	*	*	*	*	*	*	*
ROM	0200-03FF	1	*	*	*	*	*	*	*	*	*

Memory Address Map

- The hexadecimal address assigns a range of hexadecimal equivalent address for each chip
- Line 8 and 9 represent four distinct binary combination to specify which RAM we chose
- When line 10 is 0, CPU selects a RAM. And when it’s 1, it selects the ROM

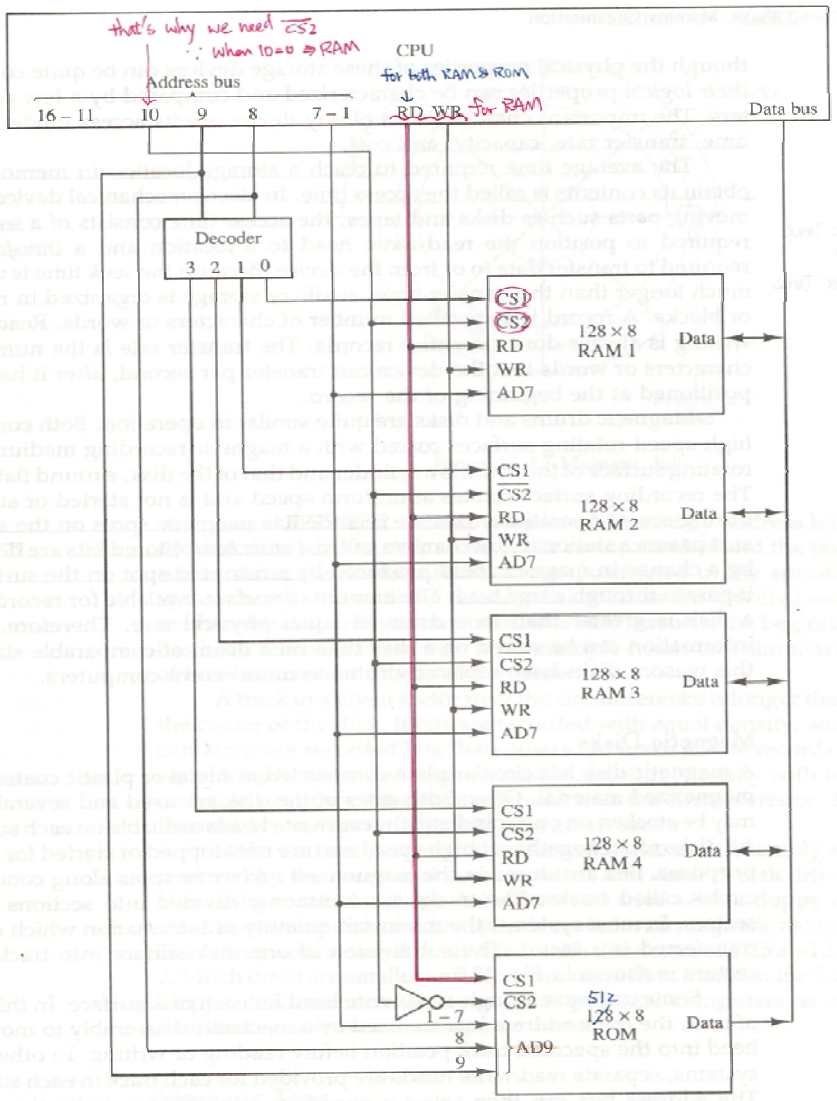


Figure 12-4 Memory connection to the CPU. RAM & ROM connection.

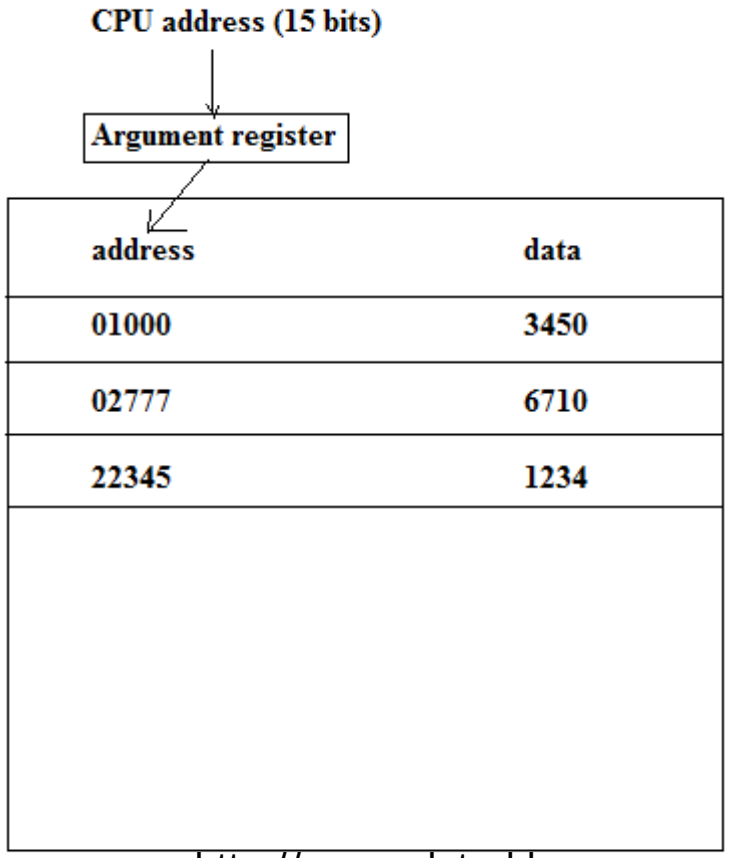
Memory Interleaving Concept (p224)

Associative mapping

- The fastest and most flexible cache organization uses an associative memory
- The associative memory stores both the address and data of the memory word
- This permits any location in cache to store an word from main memory

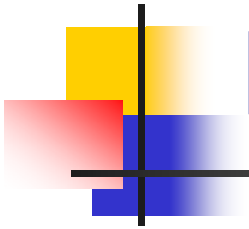
- The address value of 15 bits is shown as a five-digit **octal** number and its corresponding 12-bit word is shown as a four-digit octal number

Associative mapping



Associative mapping

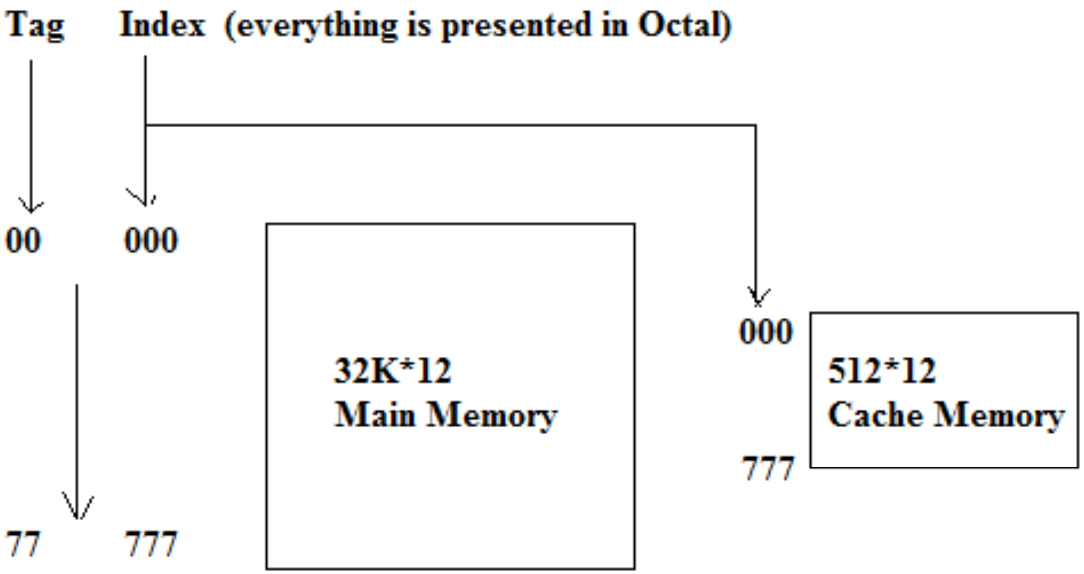
- A CPU address of 15 bits is places in the argument register and the associative memory searched for a matching address
- If the address is found, the corresponding 12-bits data is read and sent to the CPU
- If not, the main memory is accessed for the word
- If the cache is full, an address-data pair must be displaced to make room for a pair that is needed and not presently in the cache



Direct Mapping

- Associative memory is expensive compared to RAM
- In general case, there are 2^k words in cache memory and 2^n words in main memory (in our case, k=9, n=15)
- The n bit memory address is divided into two fields: k-bits for the index and n-k bits for the tag field

Direct Mapping



Direct Mapping

Memory Address	Memory Data
00000	1220
00777	2340
01000	3450
01111	2222
01777	4560
02000	5670
02777	6710

Index Address	Tag	Data
000	00	1220
111	01	2222
777	02	6710

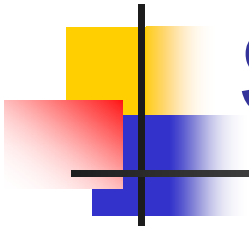
Set-Associative Mapping

- The disadvantage of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time
- Set-Associative Mapping is an improvement over the direct-mapping in that each word of cache can store two or more word of memory under the same index address

Set-Associative Mapping

Memory Address	Memory Data
00000	1220
00777	2340
01000	3450
01111	2222
01777	4560
02000	5670
02777	6710

Index Address	Tag	Data	Tag	Data
000	01	3450	02	5670
111	01	2222		
777	02	6710	00	2340



Set-Associative Mapping

- In the slide, each index address refers to two data words and their associated tags
- Each tag requires six bits and each data word has 12 bits, so the word length is 2*(6+12) = 36 bits

CACHE MEMORY

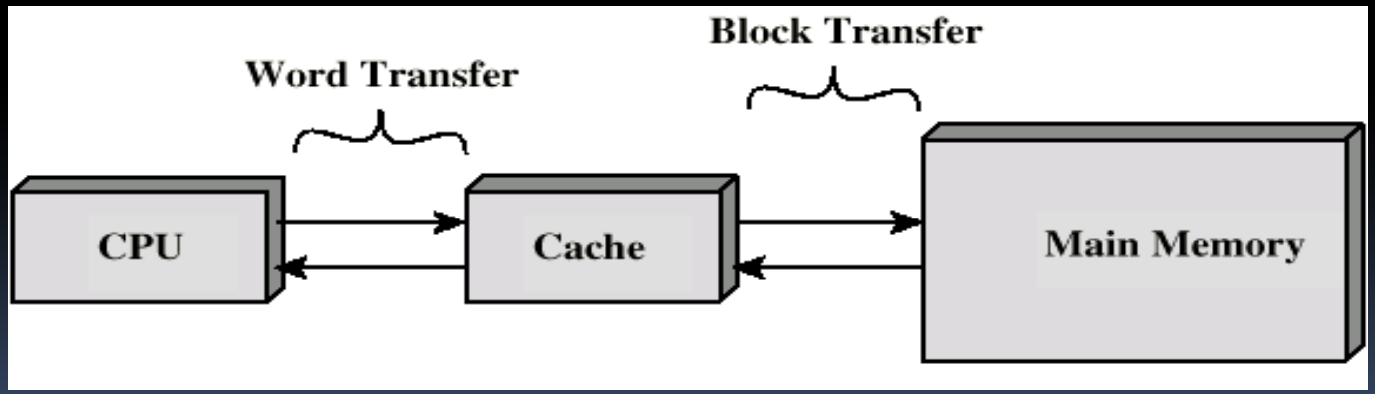
Cache

- If the active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced,
- Thus reducing the total execution time of the program
- Such a fast small memory is referred to as cache memory
- The cache is the fastest component in the memory hierarchy and approaches the speed of CPU component

Cache

CPU

- Cache memory is in between CPU and main memory
- CPU access data from cache



Operation of cache

- When CPU needs to access a memory, cache is examined
- If memory location found than ok! Read (hit)
- If not then memory location is searched in main memory and block of Mem. transferred into cache and then read by CPU (miss)
- Ratio of hit to total CPU access(hit+miss)is called ***HIT RATIO.***

Measuring performance of cache

- Cache access time (c)
  - Time between requested address arrived and requested data placed on the bus
- Main memory access time (m)
  - Time it takes to transfer data from main memory
- Hit ratio (0 < 1) hr

  hr = cache hits / cache hits + cache miss |||| i.e *(total requests made by cpu)*

- Miss ratio (mr)
  - mr = (1 – hr)
- Mean/avarage access time = c + (1-hr)m
  - if hr --> 1 then m.a.t. = c
  - if hr --> 0 then m.a.t. = c + m
- Efficiency of cache (%)
  - Cache access time * 100 / mean access time

Example

- For a CPU cache access time is 160ns and main memory access time is 960ns and hit ratio is .90 . calculate mean access time and cache efficiency
- **Sol.** M.a.t = c + (1-hr)m

  mean = 160 + (1-.90)960 = 256 ns

  efficiency = c / mean = 160/256

Cache design

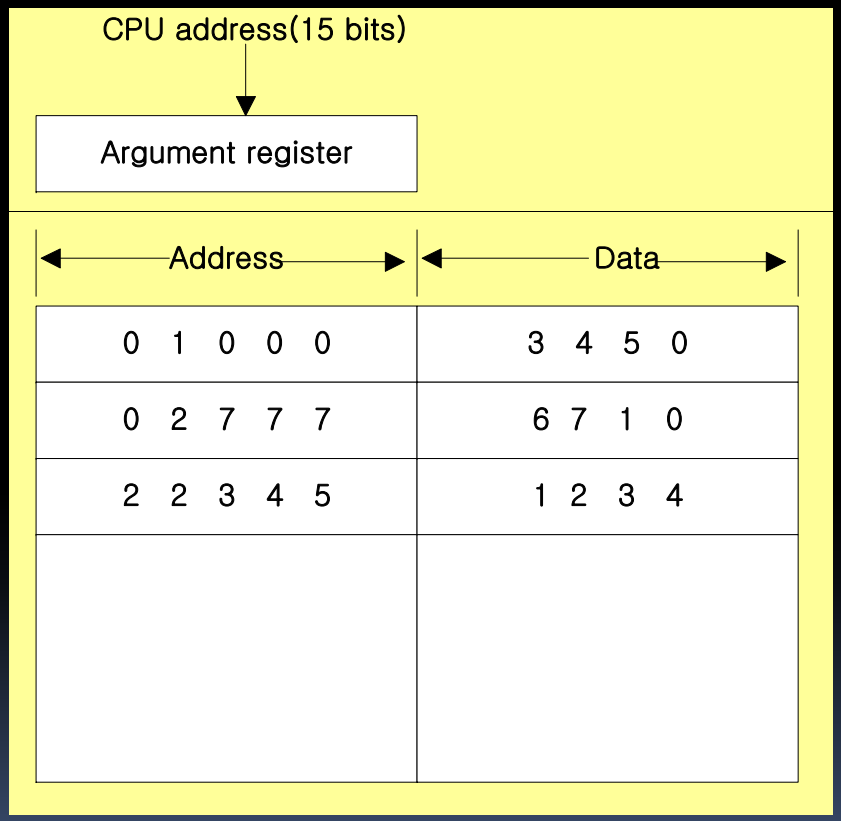
- Size (1Mb … 2Mb…)
  - Cost
    - More cache is expensive
  - More cache is faster (up to a point)
    - Checking cache for data takes time
- Mapping Function
  - The transformation of data from main memory to cache memory is referred to as a mapping process
  - 32K * 12 Bytes main memory
    - 15 bit address
      - (2^{15}=32K)
  - Cache of 512 * 12 Bytes
    - i.e. cache is 512 (2^9) lines of 1 bytes

1) Associative
2) Direct
3) Set-associative

- Replacement policy/Algorithm
  - LRU (Least Recently Used)
  - LFU (Least Frequently Used)
  - FIFO (First In First Out)
- Write policy
  - Write-through
  - Write -back

Cache mapping

- Associative
  - Easiest and fastest mapping method
  - Uses associative memory
    - Associative memory can store address and data both



- CPU place the 15 bit address in argument register and matching address is searched in cache if found then stored data is read
- If no match found then main memory is accessed
- The address – data pair is then transferred in cache
- This replace the previously stored address- data pair
- Which pair should be replaced ? It depends upon replacement policy adhered
- Example - *FIFO*

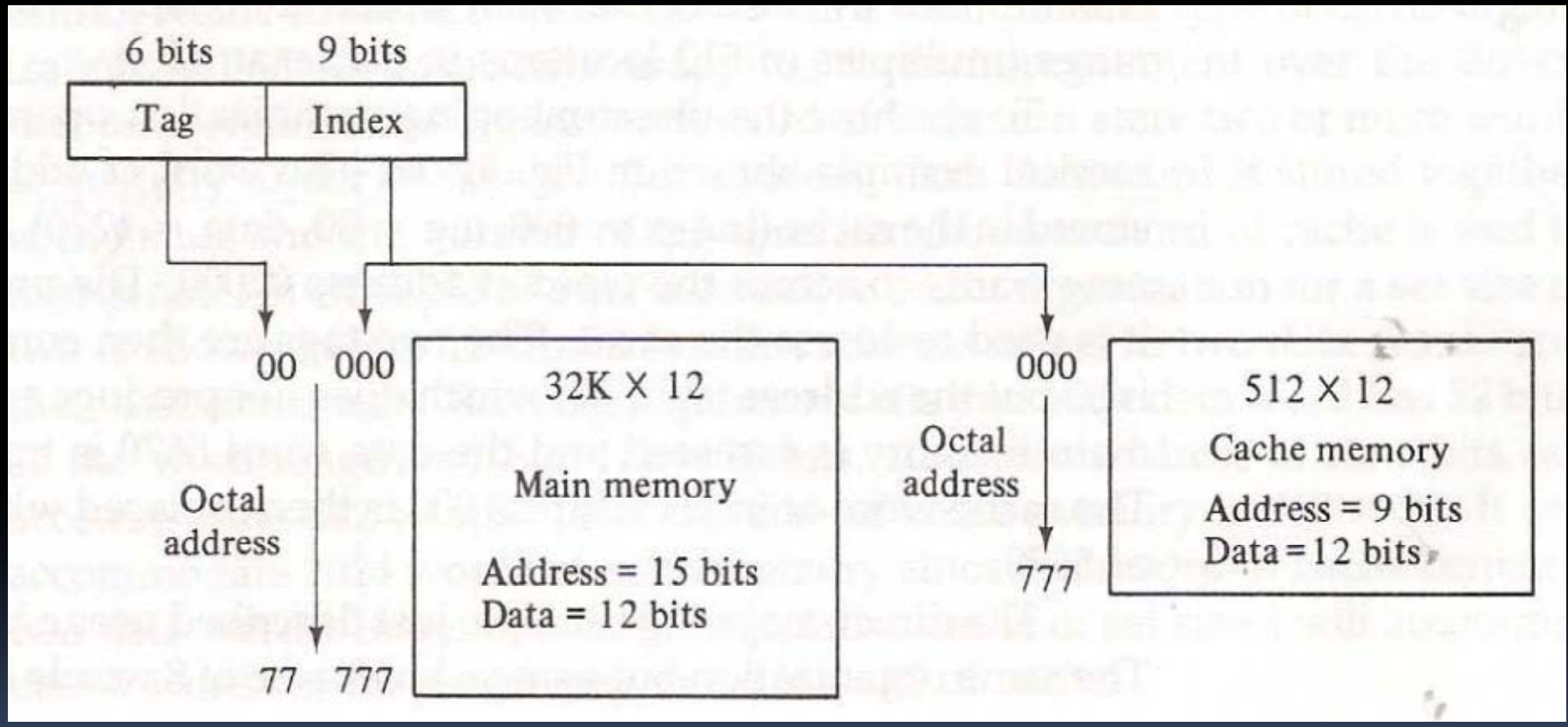
Disadvantage

- Associative memories are expensive.

Direct mapping

- Simple RAM memories with direct mapping can be used
- In this scheme the *15 bit main memory* address is divided in two fields *index field* and *tag field*
- Here, *index bits* = 9 bits = (2^9 = 512) = to access cache
- and remaining 15 – 9 = 6 bits = *tag bits*
- In general, n bit memory address = k bit cache and remaining n - k bits tag

Direct mapping



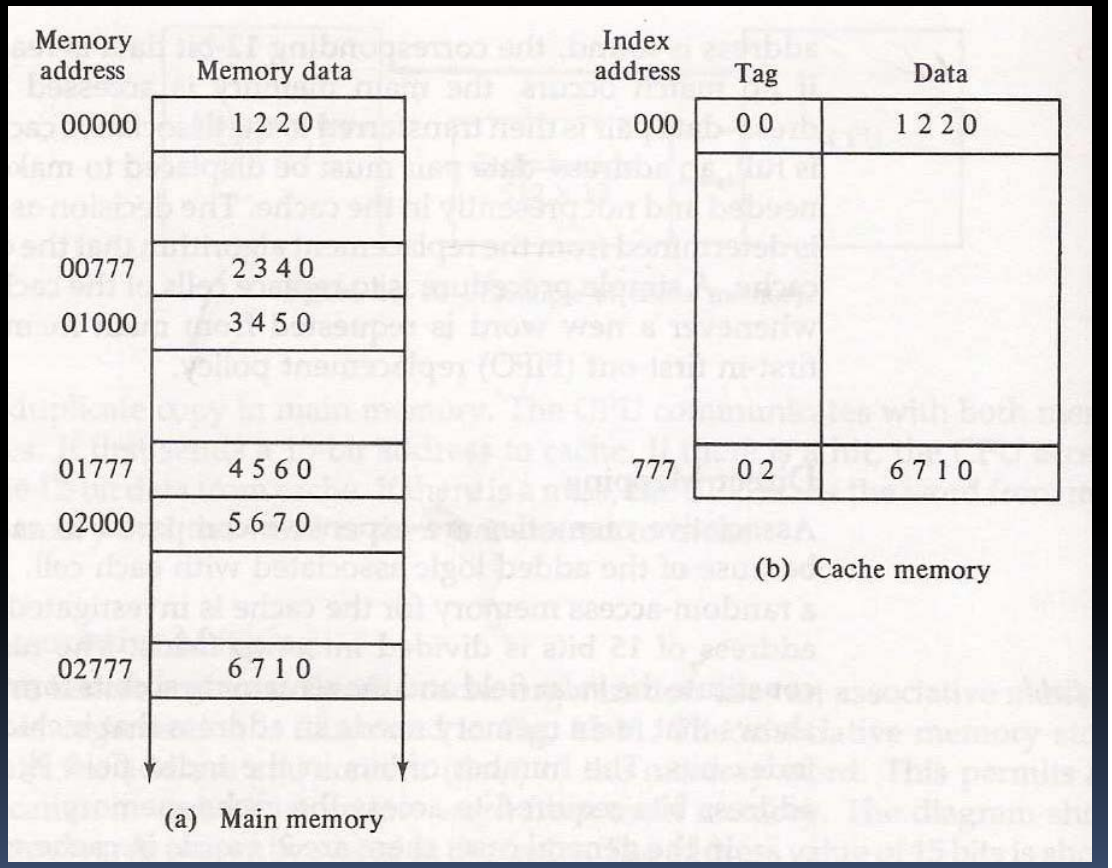
- When a new word is brought into cache the tag bits stored along with data
- When cpu generates the address, then index field is used to access cache
- And then tag field is matched with tag stored in cache
- If both match then it is hit
- If not then miss, it is read from main memory and stored in cache with new tag

Disadvantage

- If two or more words with same index but different tag are accessed again and again
- Hit ratio drops

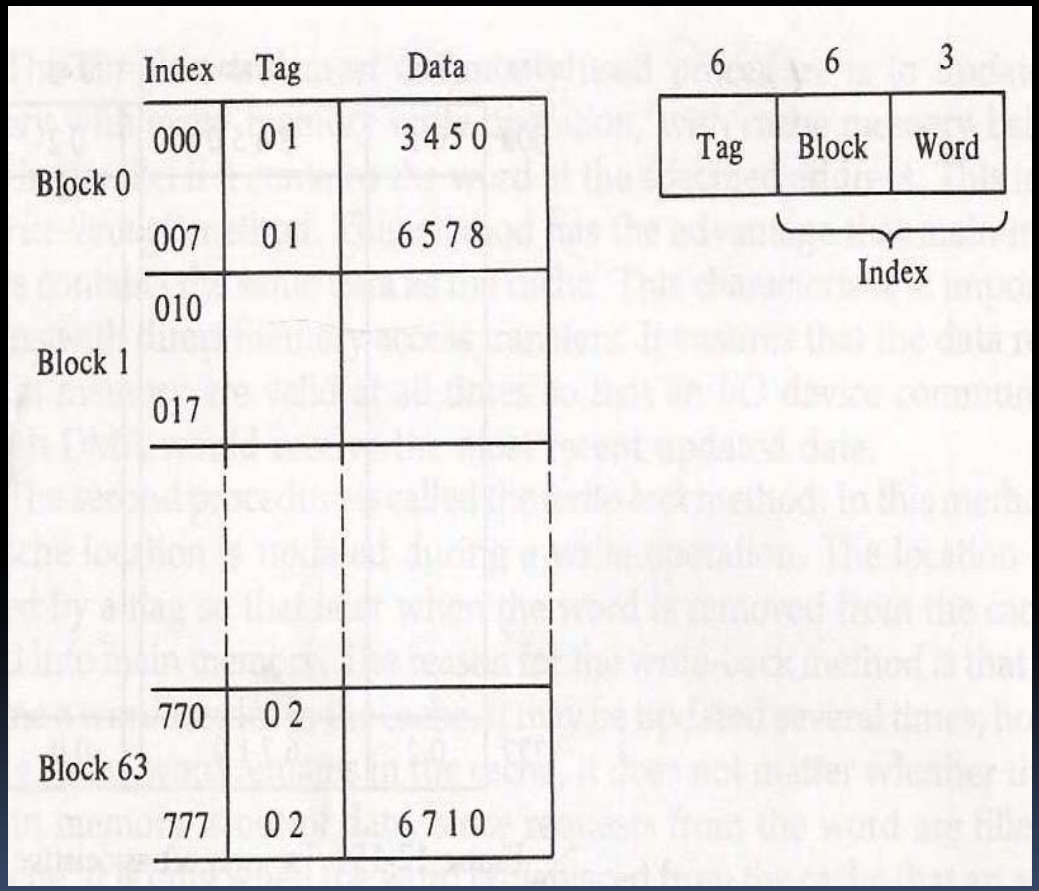
Example

- Here block size is of one word (one memory location)
- Block size may be of more than one word



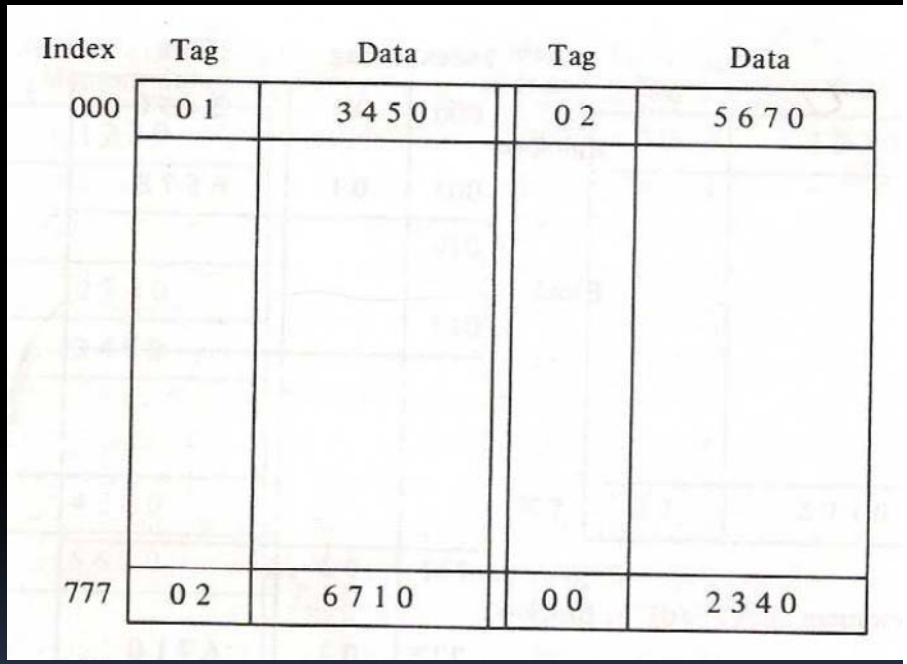
Cache block

- Index field is divided in two 6 bit to identify block
- 3 bits to identify word with in the block



Set associative mapping

- Each word of cache can store more than one word
- Two way set associative memory can store two words with same index but different tags



Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

Replacement Algorithms

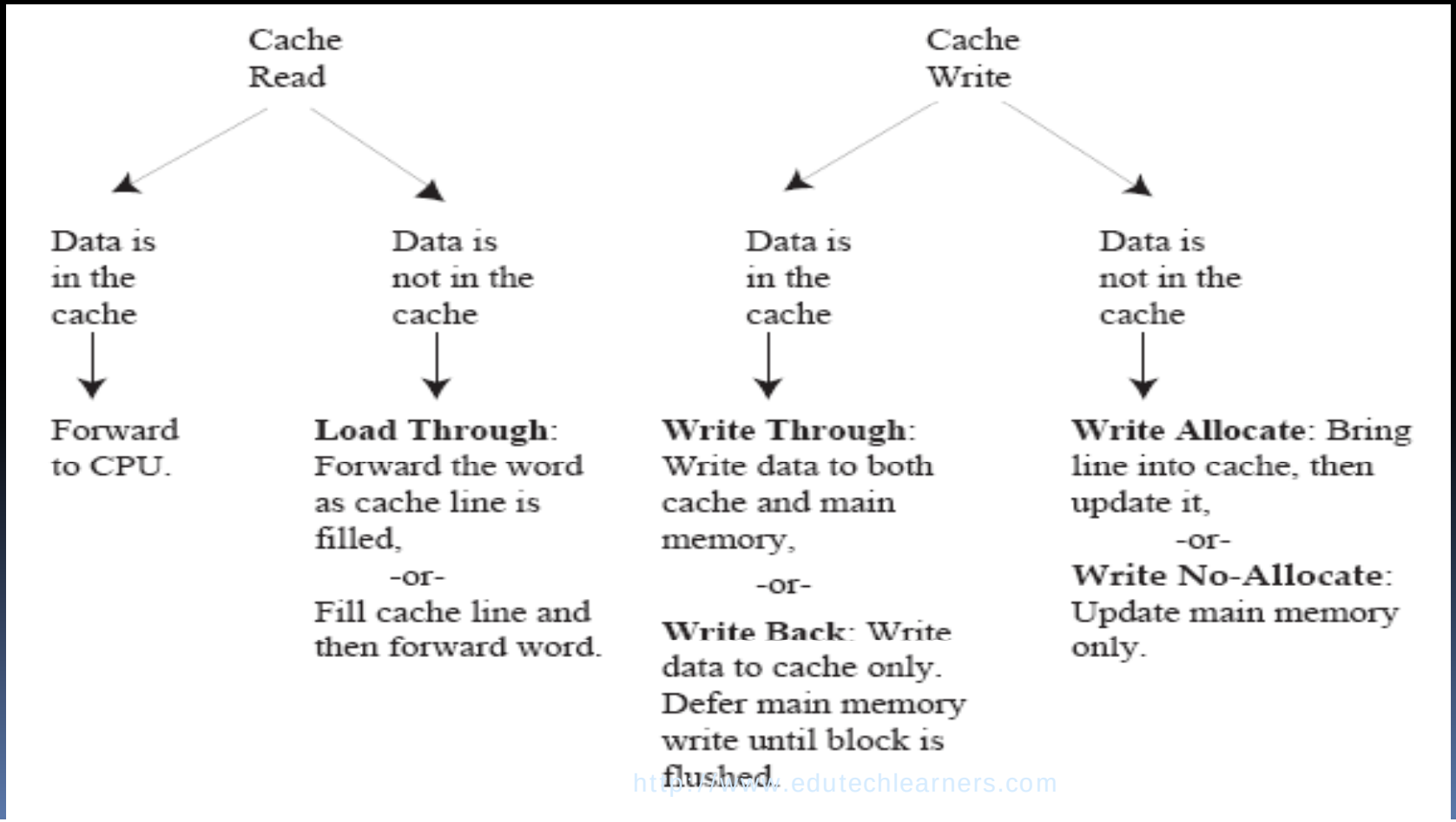
- There must be a method for selecting which line in the cache is going to be replaced when there's no room for a new line
- Hardware implemented algorithm (speed)
- Direct mapping
  - There is no need for a replacement algorithm with direct mapping
  - Each block only maps to one line
  - Replace that line

Associative & Set Associative Replacement Algorithms

- Least Recently used (LRU)
  - Replace the block that hasn't been touched in the longest period of time
- First in first out (FIFO) – replace block that has been in cache longest
- Least frequently used (LFU) – replace block which has had fewest hits
- Random – only slightly lower performance than use-based algorithms LRU, FIFO, and LFU

Cache read and write policies

- If cpu has to write a word in memory location



Cache initialization

- When first system is turned on, program portions are loaded in main memory from aux. memory
- What is stored in cache ?
  - At that time cache is not empty it may contain valid or not valid word
  - So, indicating this we use a special valid bit
    - If 1 then word is valid
    - If 0 then word is not valid
  - Advantage of that, The data will be replaced only when valid bit is 0