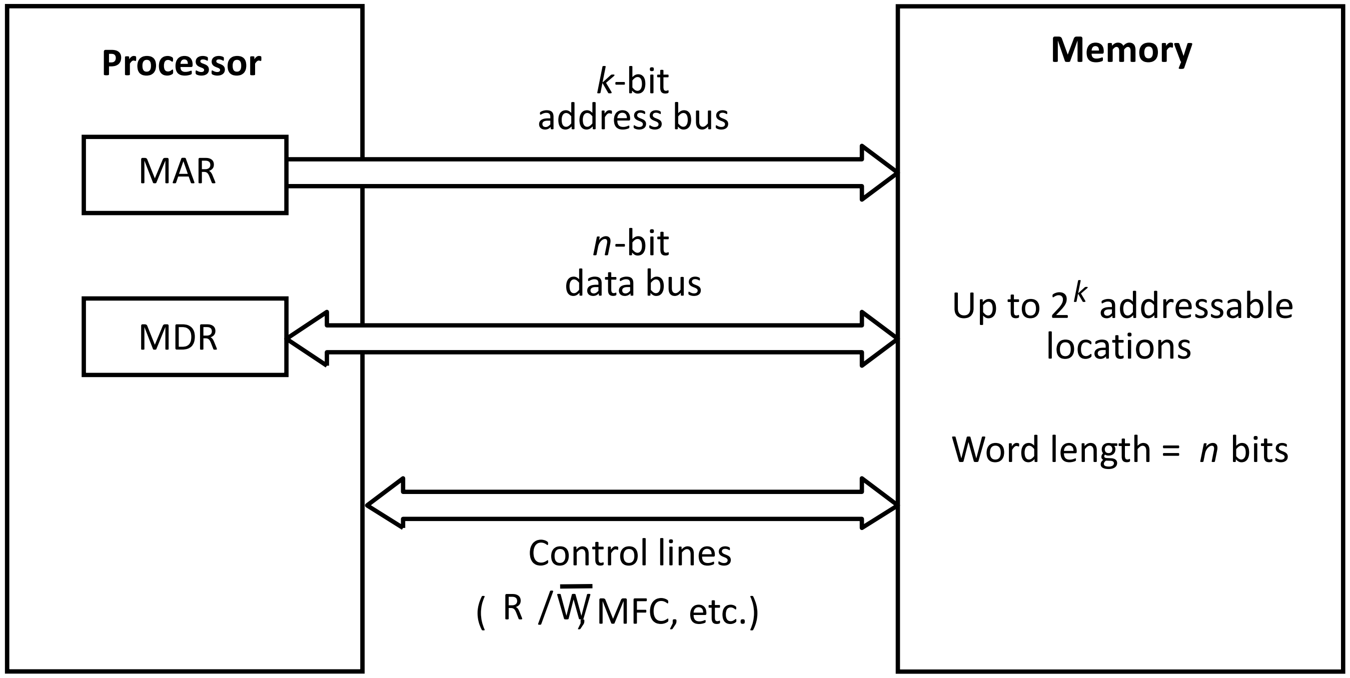
The Memory System

(Chapter 5: Hamacher, Zaky)

Basic concepts [SELF STUDY]

- Maximum size of the memory depends on the addressing scheme:
  - 16-bit computer generates 16-bit addresses and can address up to 2^{16} memory locations.
  - Number of locations represents the size of the address space of a computer.
- Most modern computers are byte-addressable.
- Memory is designed to store and retrieve data in word-length quantities.
  - Word length of a computer is commonly defined as the number of bits stored and retrieved in one memory operation.

Basic concepts (contd..) [SELF STUDY]



Recall that the data transfers between a processor and memory involves two registers MAR and MDR.
If the address bus is k-bits, then the length of MAR is k bits.
If the word length is n-bits, then the length of MDR is n bits.
Control lines include R/$\overline{W}$ and MFC.
For Read operation R/$\overline{W}$ = 1 and for Write operation R/$\overline{W}$ = 0.

Basic concepts (contd..)

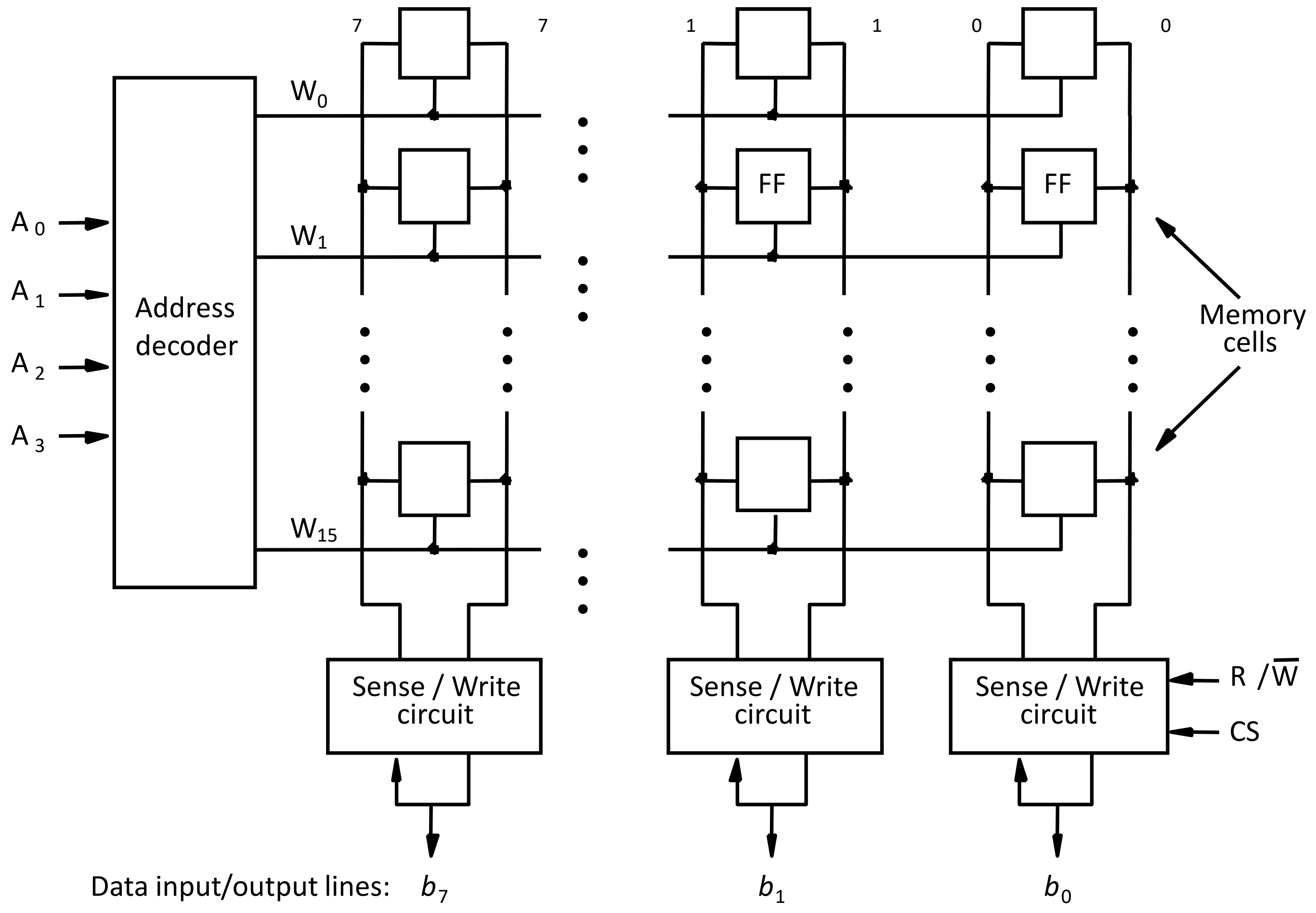
- Measures for the speed of a memory:
  - Elapsed time between the initiation of an operation and the completion of an operation is the memory access time (e.g. the time between the Read & MFC signals).
  - Minimum time between the initiation of two successive memory operations is memory cycle time (e.g. the time between two successive Read operations)
  - Memory Cycle time is slightly longer than memory access time
- In general, the faster a memory system, the costlier it is and the smaller it is.
- An important design issue is to provide a computer system with as large and fast a memory as possible, within a given cost target.
- Several techniques to increase the effective size and speed of the memory:
  - Cache memory (to increase the effective speed).
  - Virtual memory (to increase the effective size).

Semiconductor RAM memories

- Random Access Memory (RAM) memory unit is a unit where any location can be addressed in a fixed amount of time, independent of the location's address.
- Internal organization of memory chips:
  - Each memory cell can hold one bit of information.
  - Memory cells are organized in the form of an array.
  - One row is one memory word.
  - All cells of a row are connected to a common line, known as the "word line".
  - Word line is connected to the address decoder.
  - Sense/write circuits are connected to the data input/output lines of the memory chip.

Semiconductor RAM memories (contd..)

Internal organization of memory chips size=128 bits (16x8)



Semiconductor RAM memories (contd..)

- Static RAMs (SRAMs):
  - Consist of circuits that are capable of retaining their state as long as the power is applied.
  - Volatile memories, because their contents are lost when power is interrupted.
  - Access times of static RAMs are in the range of few nanoseconds.
  - Capacity is low (each cell consists of 6 transistors)
  - However, the cost is usually high.
- Dynamic RAMs (DRAMs):
  - Do not retain their state indefinitely.
  - Contents must be periodically refreshed.
  - Contents may be refreshed while accessing them for reading.
  - Access time is longer than SRAM
  - Capacity is higher than SRAM (each cell consists of 1 transistor)
  - Cost is lower than SRAM

Semiconductor RAM memories (contd..)

- Data is transferred between the processor and the memory in units of single word or a block.
- Speed and efficiency of data transfer has a large impact on the performance of a computer.
- Two metrics of performance: Latency and Bandwidth.
- **Latency**:
  - **Time taken to transfer a single word of data to or from memory**.
  - Definition is clear if the memory operation involves transfer of a single word of data.
  - In case of a block transfer, latency is the time it takes to transfer first word of data.
  - Time required to transfer first word in a block is substantially larger than the time required to transfer consecutive words in a block.

Semiconductor RAM memories (contd..)

- How much time is needed to transfer a single block of data.
- Blocks can be variable in size, it is useful to define a performance measure in terms of **the number of bits or bytes transferred in one second**.
- This performance measure is referred to as memory **bandwidth**.
- Bandwidth of a memory unit depends on:
  - **Speed** of access to the chip.
  - How many **bits** can be accessed in parallel.
- Bandwidth of data transfer between processor and memory unit also depends on:
  - Transfer capability of the links that connect the processor and memory, or the speed of the bus.

Memory systems

- Various **factors** such as cost, speed, power consumption and size of the chip determine how a **RAM** is chosen for a given application.
- **Static RAMs**:
  - Chosen when **speed** is the primary concern.
  - **Circuit** implementing the basic cell is highly **complex**, so **cost** and **size** are affected.
  - Used mostly in **cache** memories.
- **Dynamic RAMs**:
  - Predominantly used for implementing computer **main** memories.
  - **High densities** available in these chips.
  - **Economically** viable for implementing large memories.

Read-Only Memories (ROMs)

- SRAM and SDRAM chips are volatile:
  - Lose the contents when the power is turned off.
- Many applications need memory devices to retain contents after the power is turned off.
  - For example, computer is turned on, the operating system must be loaded from the disk into the memory.
  - Store instructions which would load the OS from the disk.
  - Need to store these instructions so that they will not be lost after the power is turned off.
  - We need to store the instructions into a non-volatile memory.
- Non-volatile memory is read in the same manner as volatile memory.
  - Separate writing process is needed to place information in this memory.
  - Normal operation involves only reading of data, this type of memory is called Read-Only memory (**ROM**).

Read-Only Memories (contd..) [SELF STUDY]

- Read-Only Memory:
  - Data are written into a ROM when it is manufactured.
- Programmable Read-Only Memory (PROM):
  - Allow the data to be loaded by a user.
  - Process of inserting the data is irreversible.
  - Storing information specific to a user in a ROM is expensive.
  - Providing programming capability to a user may be better.
- Erasable Programmable Read-Only Memory (EPROM):
  - Stored data to be erased and new data to be loaded.
  - Flexibility, useful during the development phase of digital systems.
  - Erasable, reprogrammable ROM.
  - Erasure requires exposing the ROM to UV light.

Read-Only Memories (contd..) [SELF STUDY]

- Electrically Erasable Programmable Read-Only Memory (EEPROM):
  - To erase the contents of EPROMs, they have to be exposed to ultraviolet light.
  - Physically removed from the circuit.
  - EEPROMs the contents can be stored and erased **electrically**.

Flash memory [SELF STUDY]

- Flash memory has similar approach to EEPROM.
- Read the contents of a single cell, but write the contents of an entire block of cells.
- Flash devices have greater density.
  - Higher capacity and low storage cost per bit.
- Power consumption of flash memory is very low, making it attractive for use in equipment that is battery-driven.
- Single flash chips are not sufficiently large, so larger memory modules are implemented using flash cards and flash drives.

Acronyms [SELF STUDY]

RAM	--Random Access Memory	time taken to access any arbitrary location in memory is constant (c.f., disks)
ROM	--Read Only Memory	ROMs are RAMs which can only be written to once; thereafter they can only be read Older uses included storage of bootstrap info
PROM	--Programmable ROM	A ROM which can be bench programmed
EPROM	--Erasable PROM	A PROM which can be erased for rewriting
EEPROM	--Electrically EPROM	A PROM which can be erased electrically.
SRAM	--Static RAM	RAM chip which loses contents upon power off
DRAM	--Dynamic RAM	RAM chip whose contents need to be refreshed.
SDRAM	--Synchronous DRAM	RAM whose memory operations are synchronized with a clock signal.

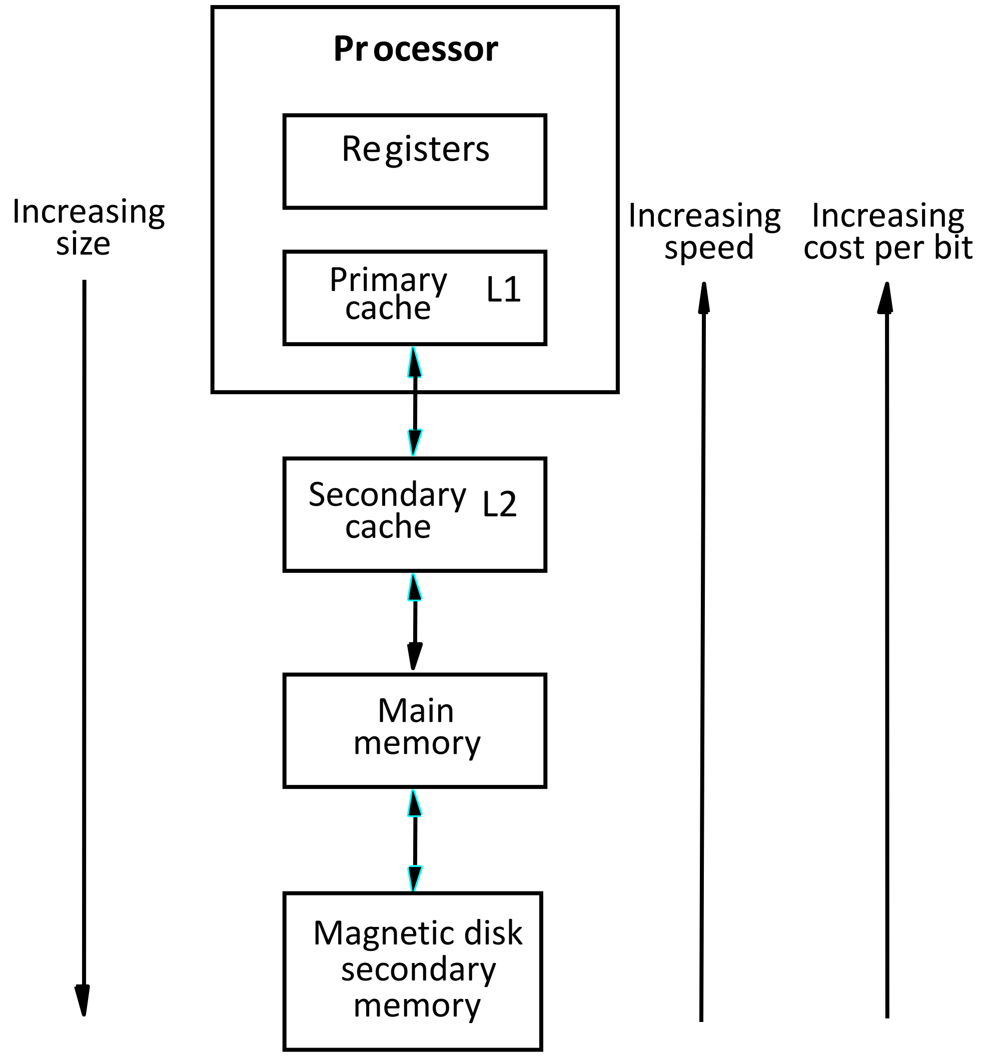
Memory hierarchy

- A big challenge in the design of a computer system is to provide a sufficiently large memory, with a reasonable speed at an affordable cost.
- Static RAM:
  - Very fast, but expensive, because a basic SRAM cell has a complex circuit making it impossible to pack a large number of cells onto a single chip.
- Dynamic RAM:
  - Simpler basic cell circuit, hence are much less expensive, but significantly slower than SRAMs.
- Magnetic disks:
  - Storage provided by DRAMs is higher than SRAMs, but is still less than what is necessary.
  - Secondary storage such as magnetic disks provide a large amount of storage, but is much slower than DRAMs.

Memory hierarchy (contd..)

- All these types of memory units are employed effectively in a computer.
  - SRAM -- Smaller units where speed is critical, namely, cache memories.
  - DRAM -- Large, yet affordable memory, namely, main memory.
  - Magnetic disks -- Huge amount of cost-effective storage.
- Computer memory can be viewed as a hierarchy.

Memory hierarchy (contd..)



- *Fastest access is to the data held in processor registers. Registers are at the top of the memory hierarchy.*
- *Relatively small amount of memory that can be implemented on the processor chip. This is processor cache.*
- *Two levels of cache:*
  Level 1 (L1) cache is on the processor chip.
  Level 2 (L2) cache is in between main memory and processor.
- *Next level is main memory, implemented as SIMMs. Much larger, but much slower than cache memory.*
- *Next level is magnetic disks. Huge amount of inexpensive storage.*
- *Speed of memory access is critical, the idea is to bring instructions and data that will be used in the near future as close to the processor as possible.*

Cache memories

- Processor is much faster than the main memory.
  - As a result, the processor has to spend much of its time waiting while instructions and data are being fetched from the main memory.
  - Major obstacle towards achieving good performance.
- Speed of the main memory cannot be increased beyond a certain point.
- Cache memory is an architectural arrangement which makes the main memory appear faster to the processor than it really is.
- Cache memory is based on the property of computer programs known as “locality of reference”.

Locality of reference

- Analysis of programs indicates that many instructions in localized areas of a program are executed repeatedly during some period of time, while the others are accessed relatively less frequently.
  – These instructions may be the ones in a loop, nested loop or few procedures calling each other repeatedly.
  – This is called "**locality of reference**".
- **Temporal locality of reference**:
  – Recently executed instruction is likely to be executed again very soon.
- **Spatial locality of reference**:
  – Instructions with addresses close to a recently instruction are likely to be executed soon.

Locality of reference (contd..)

- Cache memory is based on the concept of locality of reference.
  - If active segments of a program are placed in a fast cache memory, then the execution time can be reduced.
- Temporal locality of reference:
  - Whenever an instruction or data is needed for the first time, it should be brought into a cache. It will hopefully be used again repeatedly.
- Spatial locality of reference:
  - Instead of fetching just one item from the main memory to the cache at a time, several items that have addresses adjacent to the item being fetched may be useful.
  - The term "**block**" refers to a set of contiguous addresses locations of some size.

Hit and Miss [SELF STUDY]

- Focus on *any two adjacent* levels – called, *upper* (closer to CPU) and *lower* (farther from CPU) – in the memory hierarchy, because each block copy is always between two adjacent levels
- **Terminology:**

– *block*: minimum unit of data to move between levels (Usually 1 word)

– *hit*: data requested is in upper level (example:data found in cache!)

– *miss*: data requested is not in upper level (e.g., data not in cache, So, go to lower level (main memory or hard disk) to find that data)

– *hit rate*: fraction of memory accesses that are hits (i.e., found at upper level, say cache)

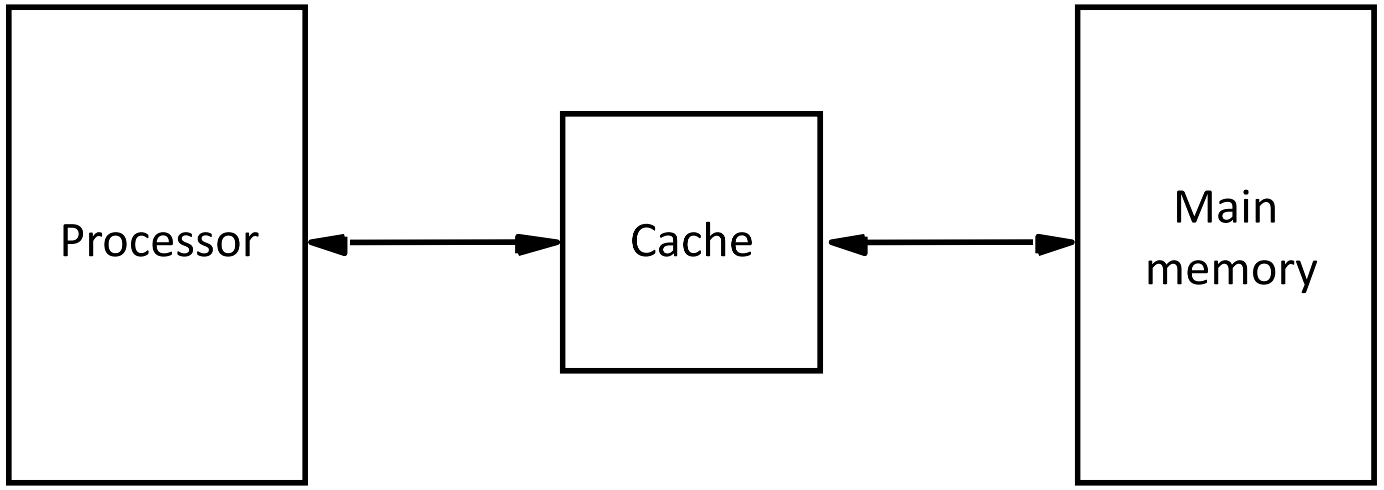
– *miss rate*: fraction of memory accesses that are not hits

- miss rate = 1 – hit rate

– *hit time*: time to determine if the access is really a hit + time to get and deliver the data from the upper level (e.g., cache) to the CPU

– *miss penalty*: time to determine if the access is a miss + time to replace block at upper level (say, cache) with corresponding block at lower level (e.g., RAM) + time to deliver the block to the CPU

Cache memories



•Processor issues a Read request, a block of words is transferred from the main memory to the cache, one word at a time.

•Subsequent references to the data in this block of words are found in the cache.

•At any given time, only some blocks in the main memory are held in the cache.

*•Which blocks in the main memory are in the cache is determined by a "**mapping function**".*

*•When the cache is full, and a block of words needs to be transferred from the main memory, some block of words in the cache must be replaced. This is determined by a **"replacement algorithm".***

Cache memories (contd..)

- *Existence of a cache is transparent to the processor. The processor issues Read and Write requests in the same manner.*

- *If the data is in the cache it is called a <u>Read or Write **hit**</u>.*

- ***Read hit**:*
  - *The data is obtained from the cache.*

- ***Write hit**:*
  - *Cache is a replica of the contents of the main memory.*
  - *Contents of the cache and the main memory may be updated **simultaneously**. This is the <u>write-through</u> protocol.*
  - *Update the contents of the cache, and mark it as updated by setting a bit known as the <u>dirty bit</u> or <u>modified</u> bit.*
    The contents of the main memory are updated when this block is replaced. This is <u>write-back</u> or <u>copy-back</u> protocol.

Cache memories (contd..)

- *If the data is not present in the cache, then a* ***Read miss*** *or* ***Write miss*** *occurs.*

- ***Read miss****:*
  - *Block of words containing this requested word is transferred from the memory.*
  - *After the block is transferred, the desired word is forwarded to the processor.*
  - *The desired word may also be forwarded to the processor as soon as it is transferred without waiting for the entire block to be transferred. This is called* ***load-through*** *or* ***early-restart****.*

- ***Write-miss****:*
  - ***Write-through*** *protocol is used, then the contents of the main memory are updated directly.*
  - *If* ***write-back*** *protocol is used, the block containing the addressed word is first brought into the cache. The desired word is overwritten with new information.*

Mapping functions

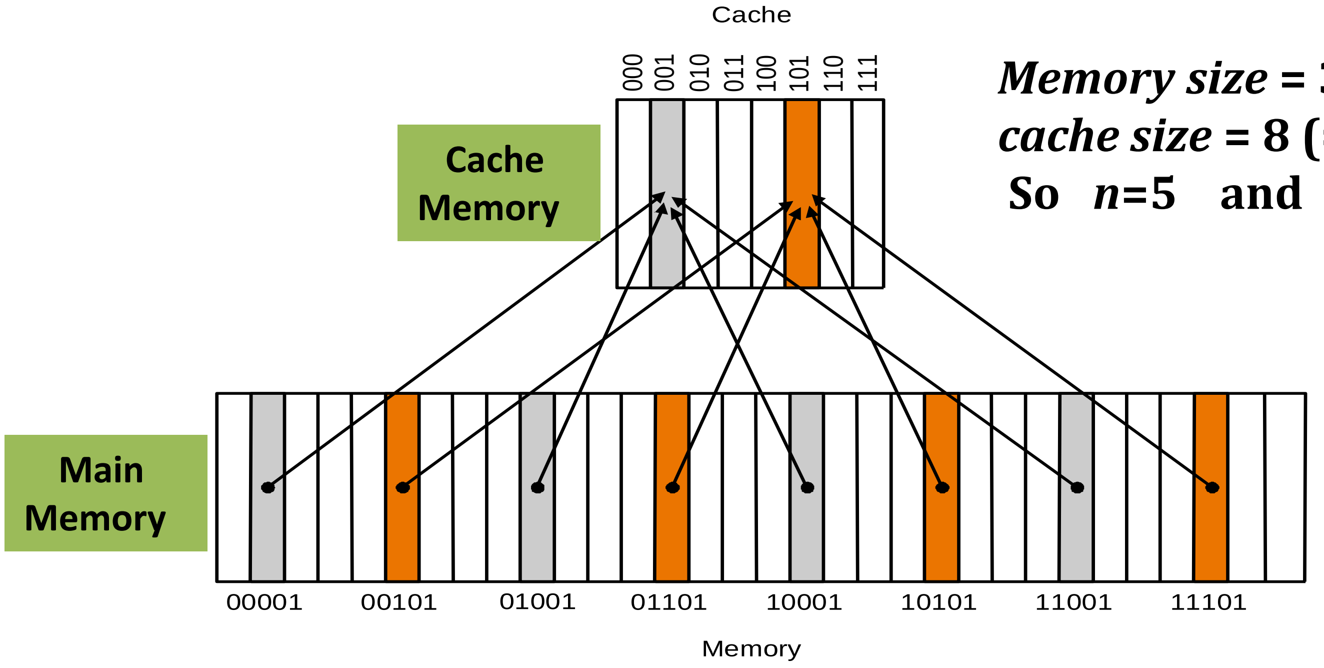
- Mapping functions determine how memory blocks are placed in the cache.
- Three mapping functions:
  - **Direct** mapping
  - **Associative** mapping
  - **Set-associative** mapping.

Direct Mapped Cache

- Addressing scheme in ***Direct Mapped* Cache:**
- cache block address = memory block address *mod* cache size
- **if cache size = 2^m, cache address = lower m bits of n-bit memory address**

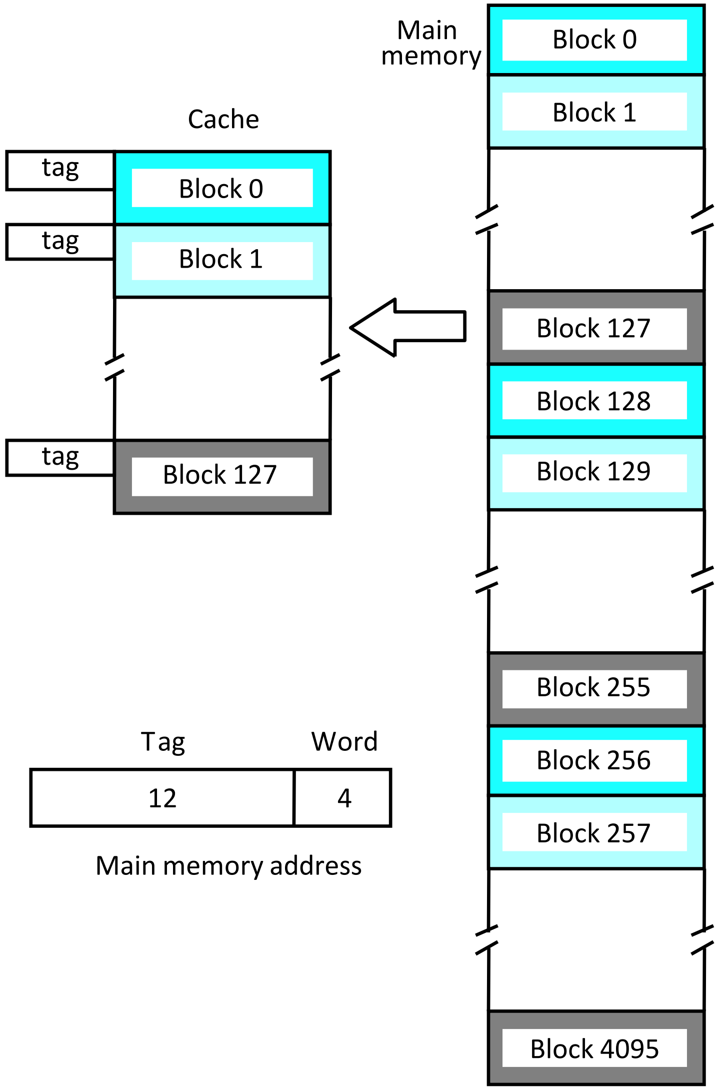
  (equivalently: Cache address = Memory Address MOD cache size)
- **remaining upper n-m bits kept as *TAG bits* at each cache block**
- also need **a *valid bit*** to

  recognize valid entry



Memory size **= 32 (= 2^5) blocks**
cache size **= 8 (= 2^3) blocks**
So *n*=5 and *m*=3

Associative mapping



•Main memory block can be placed into **any** cache position.

•Memory address is divided into **two** fields:

•**(If Main** memory is addressable by a **16-bit** address)

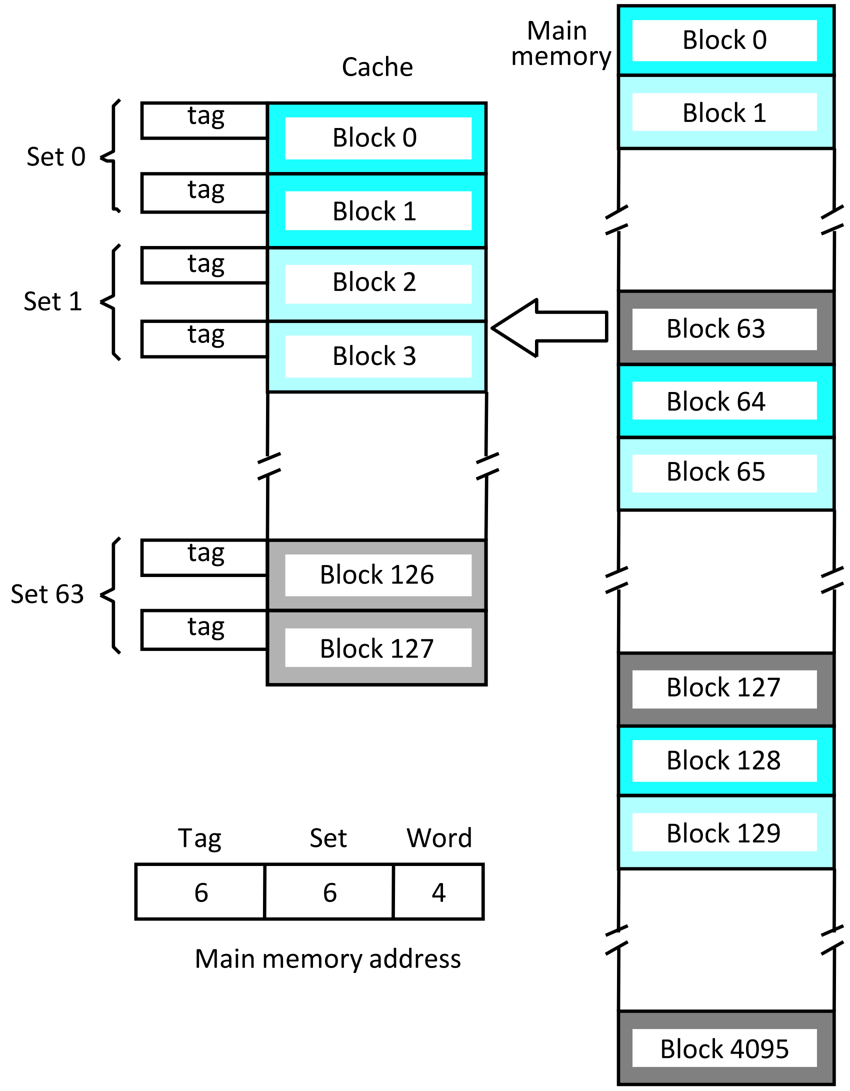
- Low order **4 bits** identify the word within a block.
- High order **12 bits** or **tag** bits identify a memory block when it is resident in the cache.

•Flexible, and uses cache space efficiently.

•Replacement algorithms can be used to replace an existing block in the cache when the cache is full.

•Cost is higher than direct-mapped cache because of the need to search all patterns to determine whether a given block is in the cache.

Set-Associative mapping



*Set-associative mapping combination of **direct** and **associative** mapping.*
*Blocks of cache are grouped into **sets**.*
*Mapping function allows a block of the main memory to reside in any block of a specific set.Assume that, No of **Blocks** of **memory** =4096.Divide the cache into **64 sets**, with two blocks per set. (if cache size 128)*
*Memory block **0, 64, 128** etc. map to **set 0**, and they can occupy either of the two positions.*
*16 bit Memory address is divided into **three** fields:*

- *Low order **4 bits** identify the word within a block.*
- ***6 bit** field determines the **set** number.*
- *High order **6 bit** fields are compared to the **tag** fields of the two blocks in a set.*

*Tag size = No of **Blocks** of **memory** / No of **sets** of **cache** = 4096/64 = **64**= 2^**6***
Number of blocks per set is a design parameter.

- *One extreme is to have all the blocks in one set, requiring no set bits (fully associative mapping).*
- *Other extreme is to have one block per set, is the same as direct mapping.*

Cache memories (contd..)

- *A bit called as "**valid bit**" is provided for each **block**.*
- *If the block contains valid data, then the bit is set to 1, else it is 0.*
- *Valid bits are set to 0, when the power is just turned on.*
- *When a block is loaded into the cache for the first time, the valid bit is set to 1.*

- *Data transfers between main memory and disk occur directly bypassing the cache.*
- *When the data on a disk changes, the main memory block is also updated.*
- *However, if the data is also resident in the cache, then the valid bit is set to 0.*

- *What happens if the data in the disk and main memory changes and the write-back protocol is being used?*
- *In this case, the data in the cache may also have changed and is indicated by the dirty bit.*
- *The copies of the data in the cache, and the main memory are different. This is called the cache coherence problem.*
- *One option is to force a write-back before the main memory is updated from the disk..*

Replacement algorithms

• **Direct-mapped** cache, the position that each memory block occupies in the cache is **fixed**.
As a result, the replacement strategy is **trivial**.
• **Associative** and **set-associative** mapping provide some flexibility in deciding which memory block occupies which cache block.
When a new block is to be transferred to the cache, and all the positions it may occupy are **full**, which block in the cache should be replaced?

• Locality of reference suggests that it may be okay to replace the block that has gone the longest time without being referenced.
This block is called Least Recently Used (**LRU**) block, and the replacement strategy is called **LRU replacement algorithm.**
LRU algorithm has been used extensively.
It provides poor performance in some cases.
Performance of the LRU algorithm may be improved by introducing a small amount of randomness into which block might be replaced.

Other replacement algorithms include removing the "**oldest**" block. However, they disregard the locality of reference principle and do not perform as well.