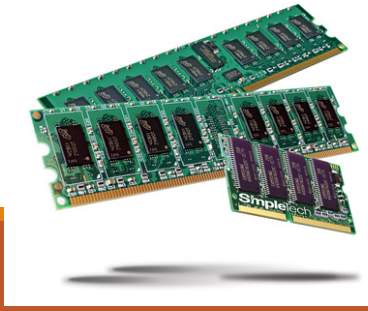
UNIT-5: The Memory System

FUNDAMENTAL CONCEPTS

Table of Contents

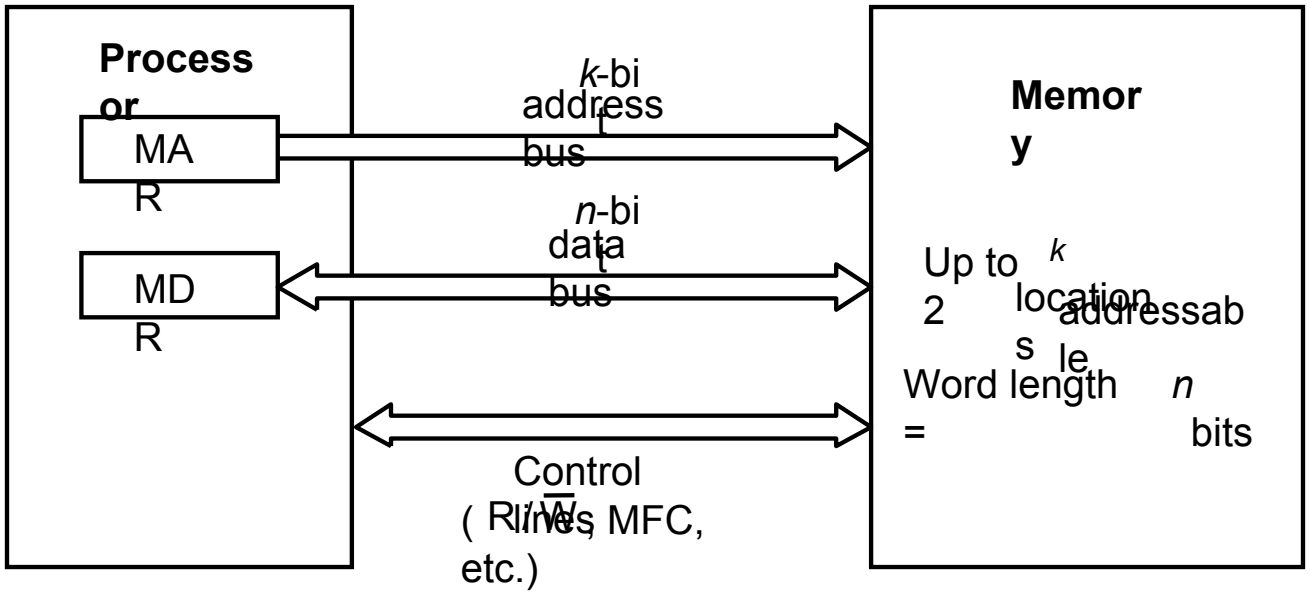


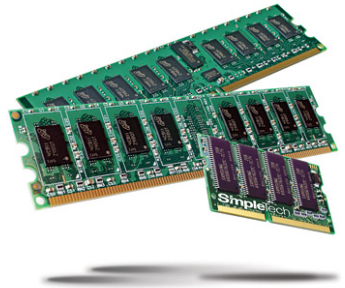
Memory Basic Concepts

Maximum size of the Main Memory

byte-addressable

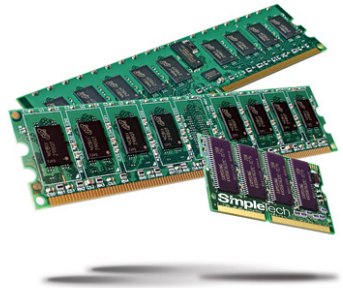
CPU-Main Memory Connection





Memory Basic Concepts(Contd.,)

- Measures for the speed of a memory:
  - Memory access time.
  - Memory cycle time.
- 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).



The Memory System

SEMICONDUCTOR RAM MEMORIES

Internal Organization of Memory Chips

Each memory cell can hold one bit of information.

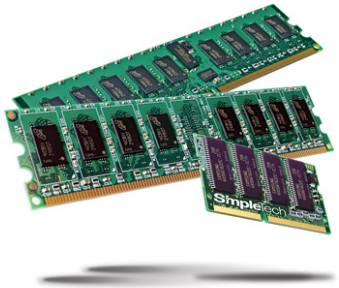
Memory cells are organized in the form of an array.

One row is one memory word.

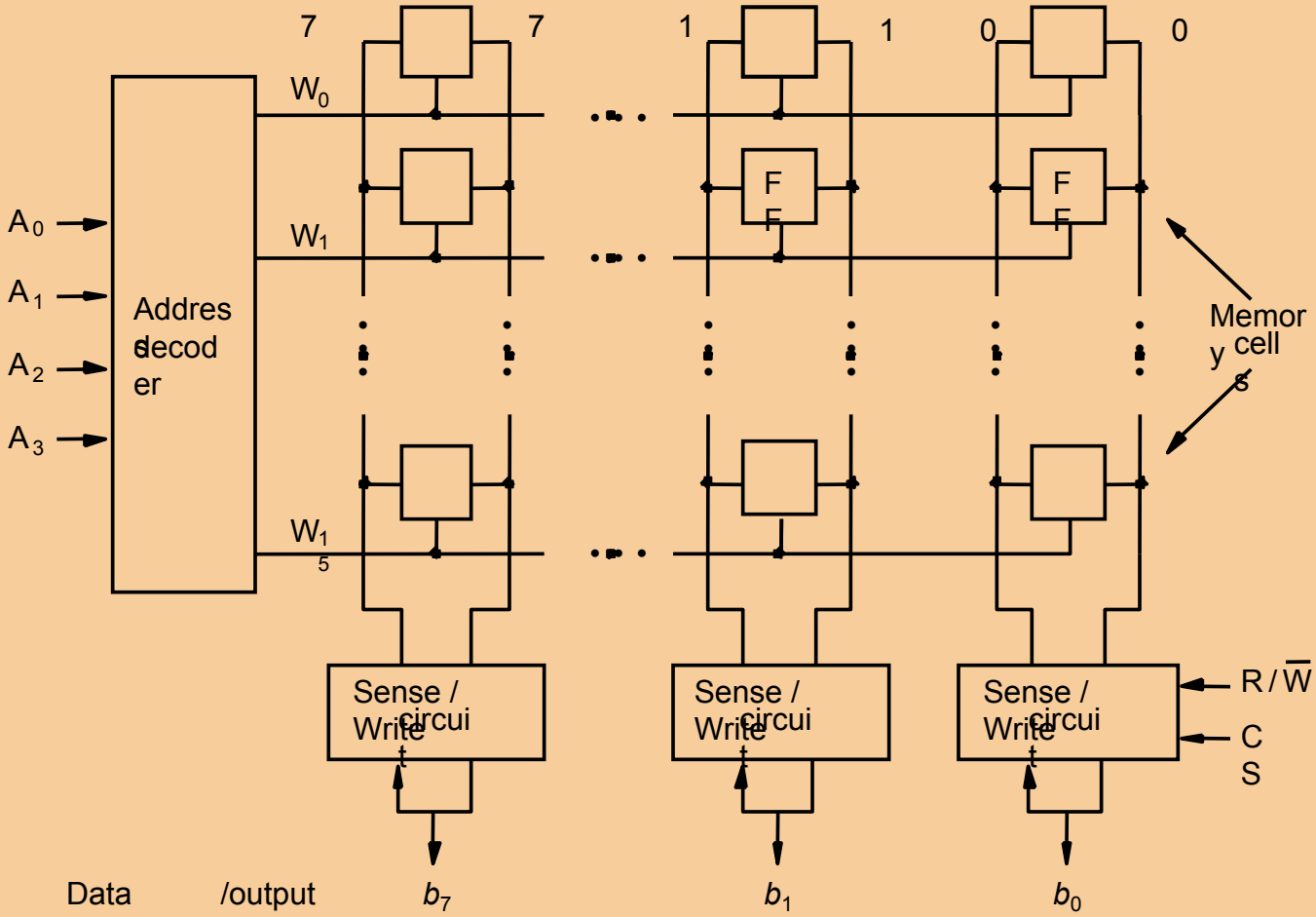
All cells of a row are connected to a common line, known as the "word line".

Word line is connected to the address decoder.

Sense/write circuits are connected to the data input/output lines of the memory chip.



Internal Organization of Memory Chips (Contd.,)



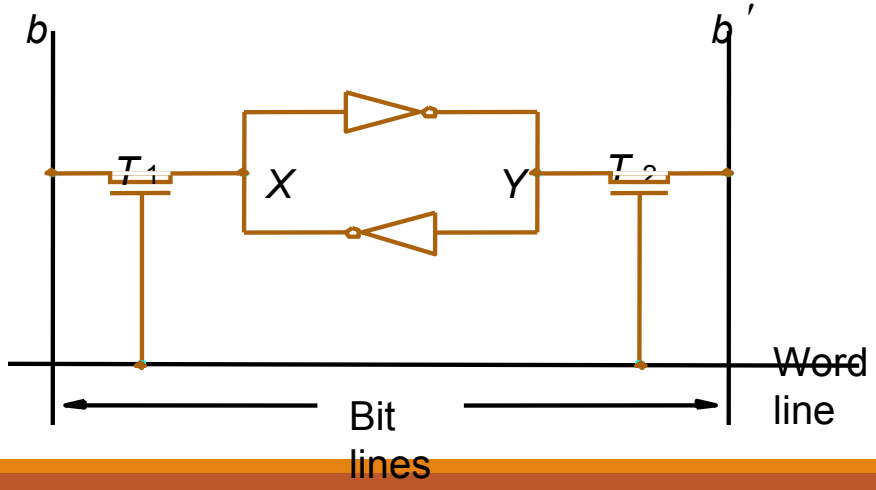
SRAM Cell

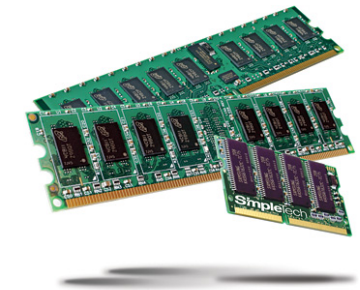
Two transistor inverters are cross connected to implement a basic flip-flop.

The cell is connected to one word line and two bits lines by transistors T1 and T2

When word line is at ground level, the transistors are turned off and the latch retains its state

Read operation: In order to read state of SRAM cell, the word line is activated to close switches T1 and T2. Sense/Write circuits at the bottom monitor the state of b and b'





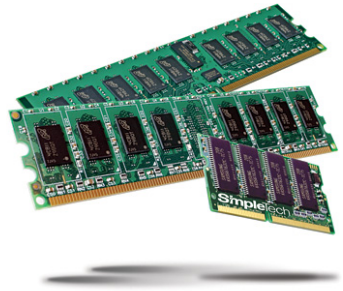
Asynchronous DRAMs

Static RAMs (SRAMs):

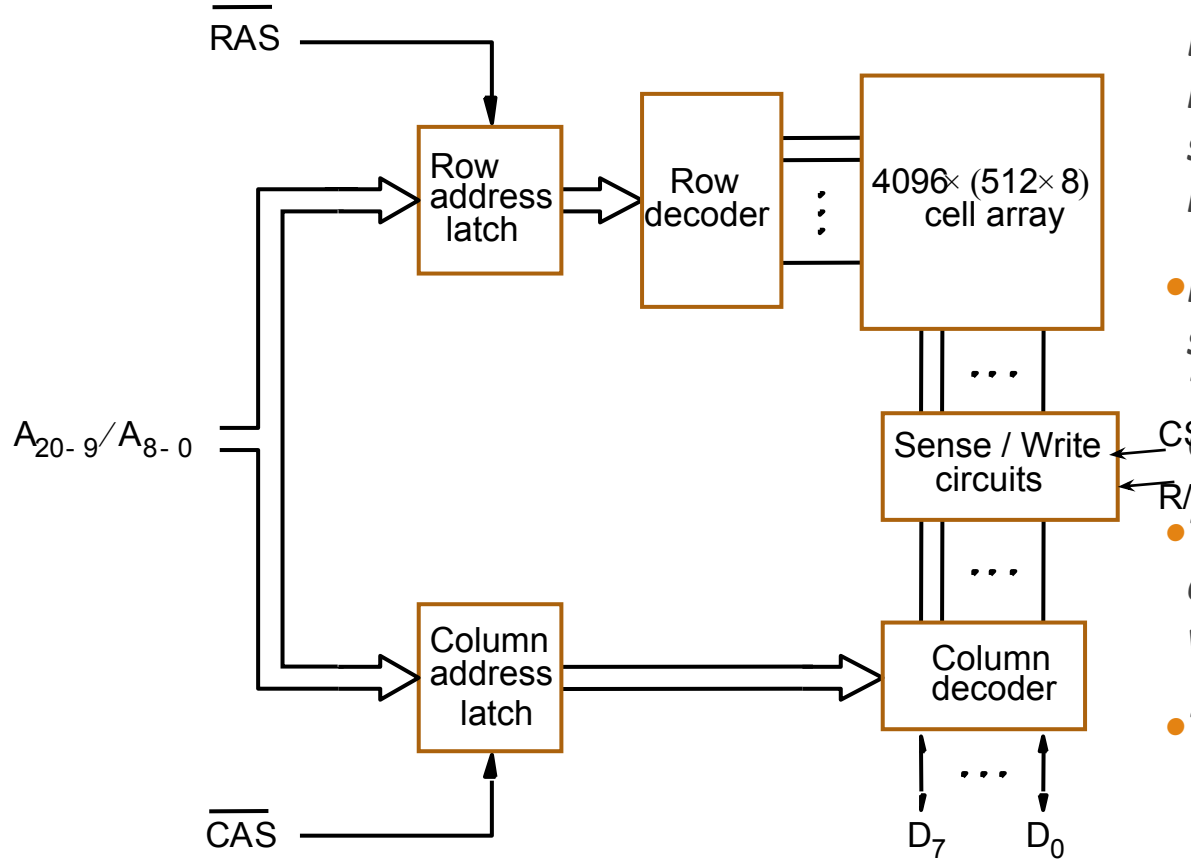
- Consist of circuits that are capable of retaining their state as long as the power is applied.
- Volatile memories, because their contents are lost when power is interrupted.
- Access times of static RAMs are in the range of few nanoseconds.
- However, the cost is usually high.

Dynamic RAMs (DRAMs):

- Do not retain their state indefinitely.
- Contents must be periodically refreshed.
- Contents may be refreshed while accessing them for reading.

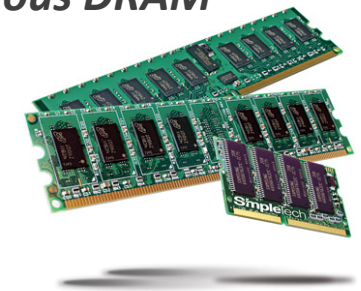


Asynchronous DRAMs



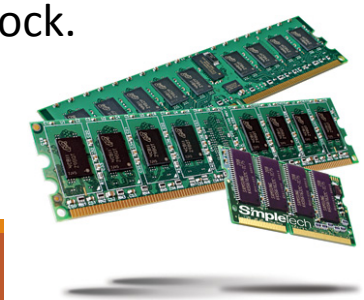
Each row can store 512 bytes. 12 bits to select a row, and 9 bits to select a group in a row. Total of 21 bits.

- ***First apply the row address, RAS signal latches the row address. Then apply the column address, CAS signal latches the address.***
- ***Timing of the memory unit is controlled by a specialized unit which generates RAS and CAS.***
- ***This is asynchronous DRAM***

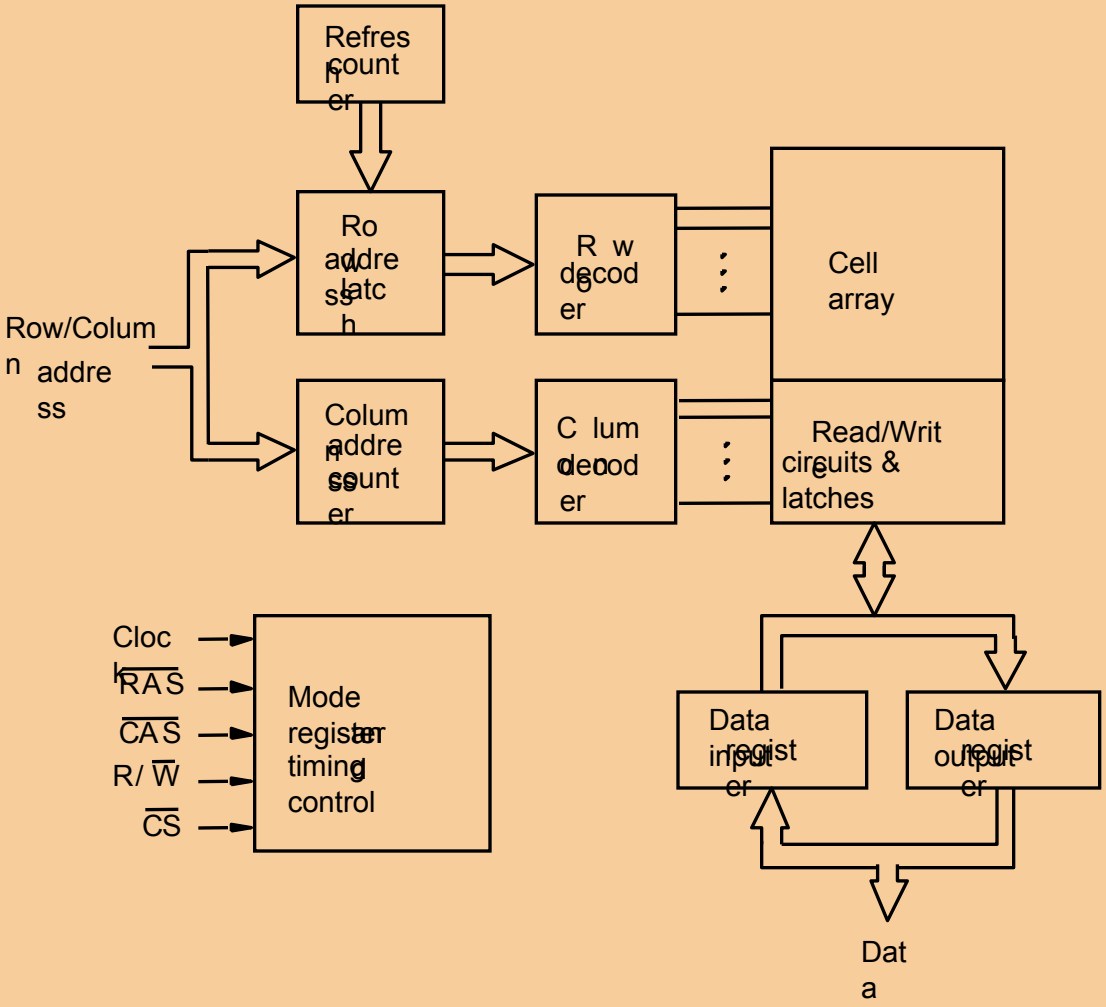


Fast Page Mode

- Suppose if we want to access the consecutive bytes in the selected row.
- This can be done without having to reselect the row.
  - Add a latch at the output of the sense circuits in each row.
  - All the latches are loaded when the row is selected.
  - Different column addresses can be applied to select and place different bytes on the data lines.
- Consecutive sequence of column addresses can be applied under the control signal CAS, without reselecting the row.
  - Allows a block of data to be transferred at a much faster rate than random accesses.
  - A small collection/group of bytes is usually referred to as a block.
- This transfer capability is referred to as the fast page mode feature.



Synchronous DRAMs



- *Operation is directly synchronized with processor clock signal.*
- *The outputs of the sense circuits are connected to a latch.*
- *During a Read operation, the contents of the cells in a row are loaded onto the latches.*
- *During a refresh operation, the contents of the cells are refreshed without changing the contents of the latches.*
- *Data held in the latches correspond to the selected columns are transferred to the output.*
- *For a burst mode of operation, successive columns are selected using column address counter and clock. CAS signal need not be generated externally. A new data is placed during raising edge of the clock*

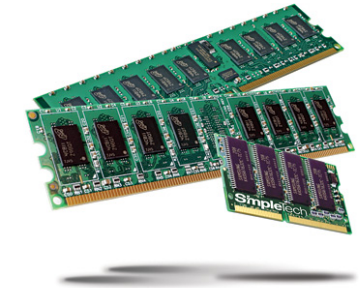
Latency, Bandwidth, and DDRSDRAMs

Memory latency is the time it takes to transfer a word of data to or from memory

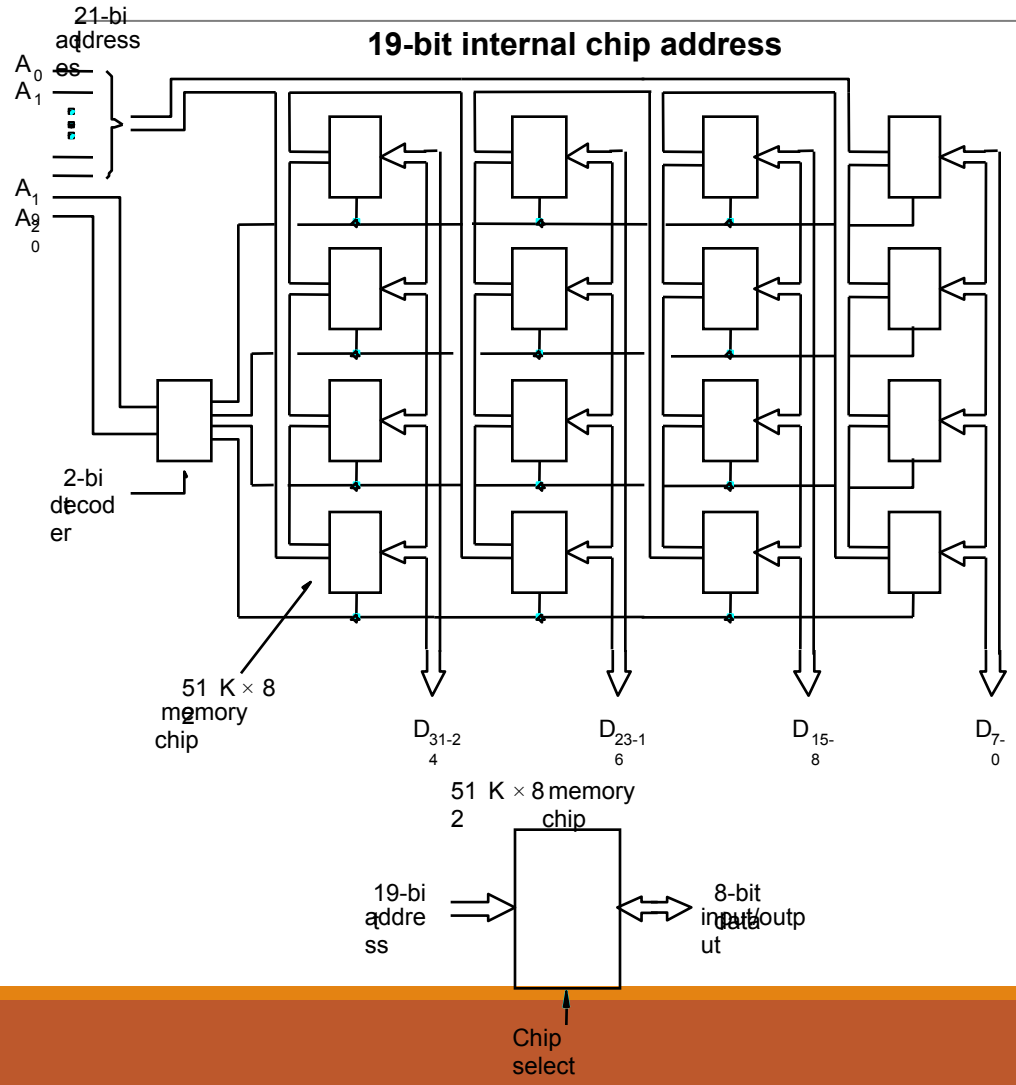
Memory bandwidth is the number of bits or bytes that can be transferred in one second.

DDRSDRAMs-Double-Data-Rate-SDRAM

- Cell array is organized in two banks



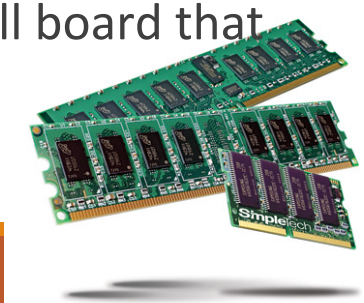
Static Memories



Implement a memory unit of 2M words of 32 bits each.
Use 512x8 static memory chips.
Each column consists of 4 chips.
Each chip implements one byte position.
A chip is selected by setting its chip select control line to 1.
Selected chip places its data on the data output line, outputs of other chips are in high impedance state.
21 bits to address a 32-bit word.
High order 2 bits are needed to select the row, by activating the four Chip Select signals.
19 bits are used to access specific byte locations inside the selected chip.

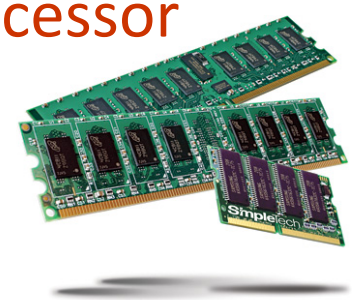
Dynamic Memories

- Large dynamic memory systems can be implemented using DRAM chips in a similar way to static memory systems.
- Placing large memory systems directly on the motherboard will occupy a large amount of space.
- Packaging considerations have led to the development of larger memory units known as SIMMs (Single In-line Memory Modules) and DIMMs (Dual In-line Memory Modules).
- Memory modules are an assembly of memory chips on a small board that plugs vertically onto a single socket on the motherboard.

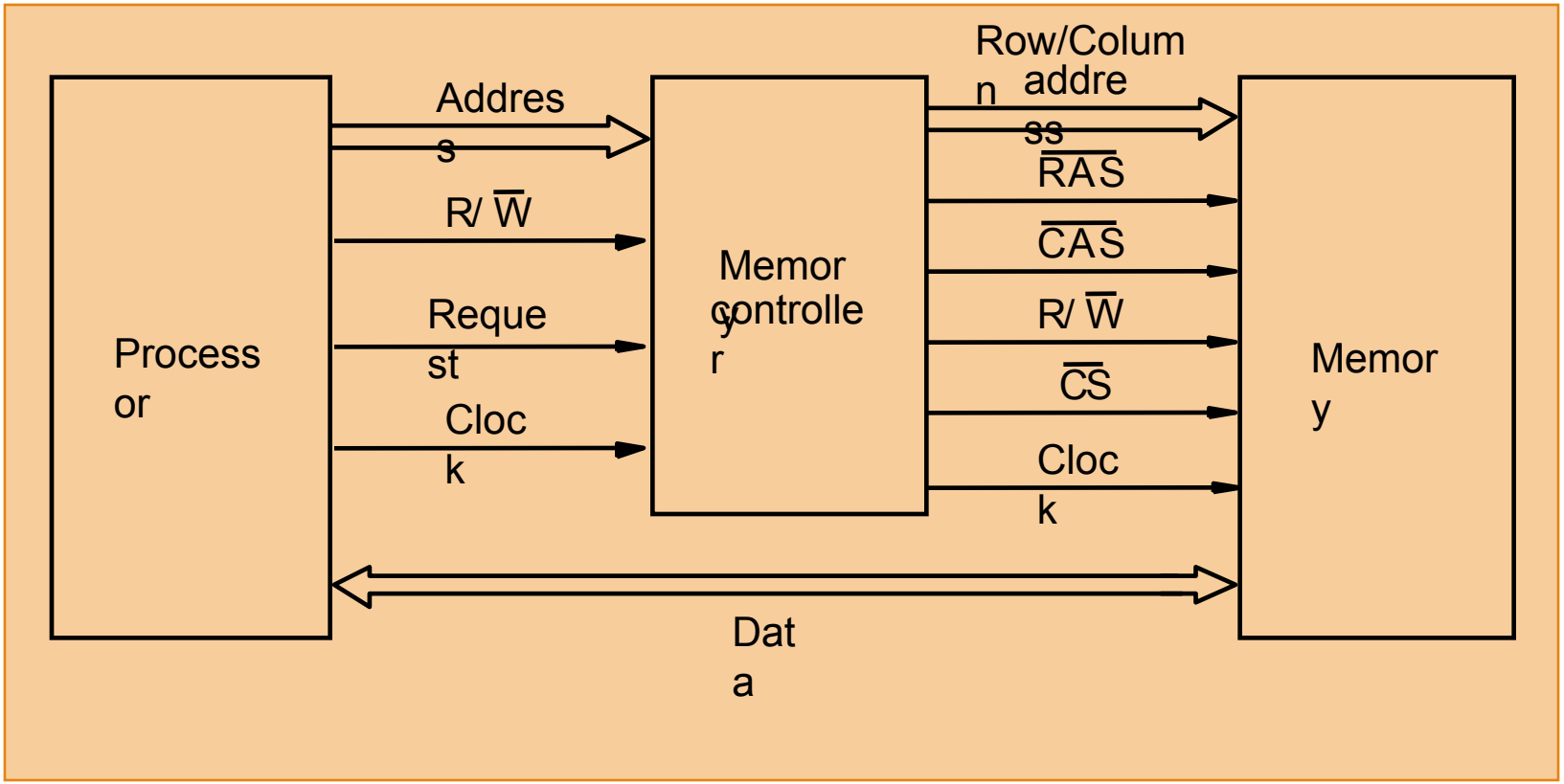


Memory Controller

- Recall that in a dynamic memory chip, to reduce the number of pins, multiplexed addresses are used.
- Address is divided into two parts:
  - High-order address bits select a row in the array.
    - They are provided first, and latched using RAS signal.
  - Low-order address bits select a column in the row.
    - They are provided later, and latched using CAS signal.
- However, a processor issues all address bits at the same time.
- In order to achieve the multiplexing, memory controller circuit is inserted between the processor and memory.



Memory Controller (contd..)

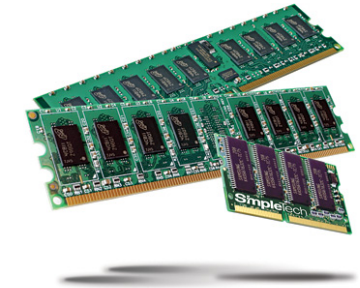


The Memory System

READ-ONLY MEMORIES (ROMS)

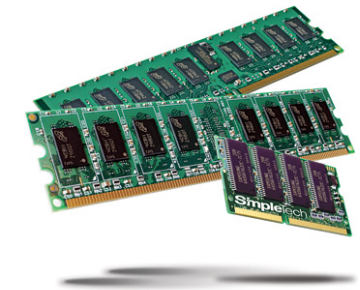
Read-Only Memories (ROMs)

- 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).



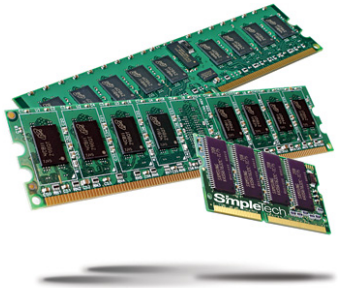
Read-Only Memories (Contd.,)

- 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.,)

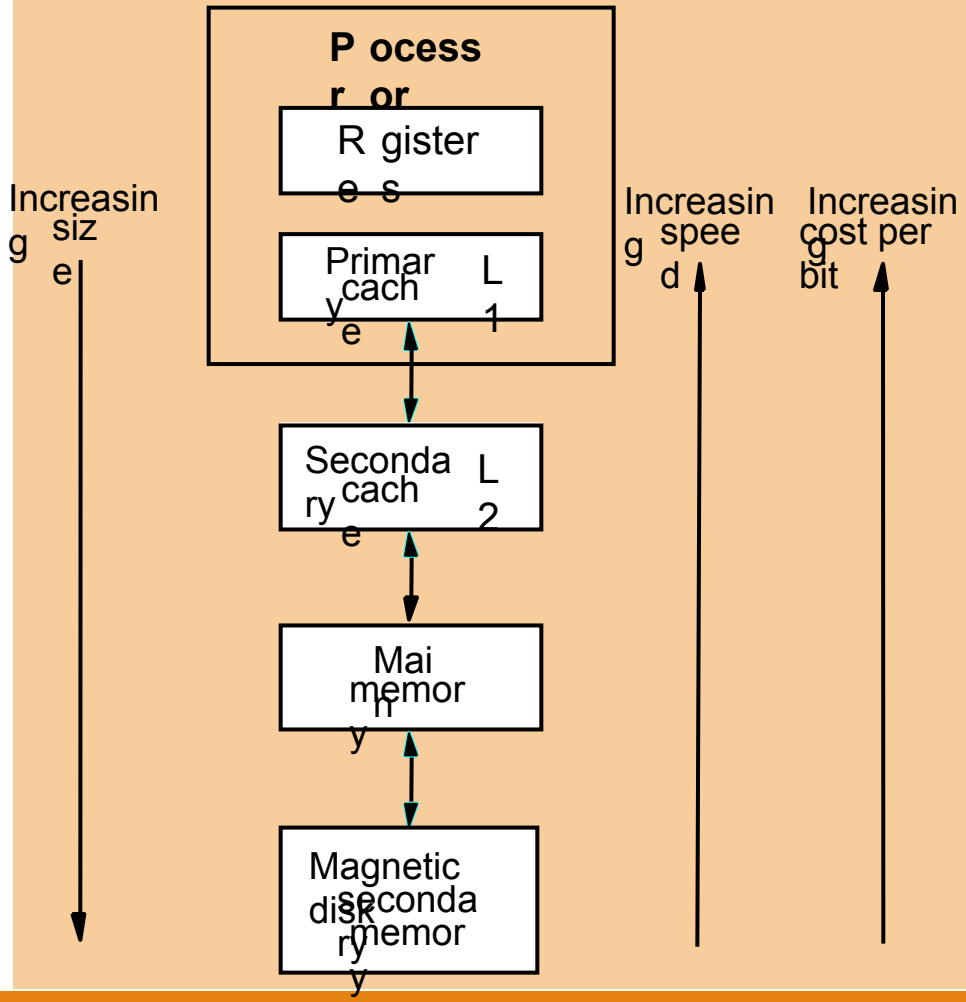
- Electrically Erasable Programmable Read-Only Memory (EEPROM):
  - To erase the contents of EPROMs, they have to be exposed to ultraviolet light.
  - Physically removed from the circuit.
  - EEPROMs the contents can be stored and erased electrically.
- Flash memory:
  - Has similar approach to EEPROM.
  - Read the contents of a single cell, but write the contents of an entire block of cells.



Speed, Size, and Cost

- 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



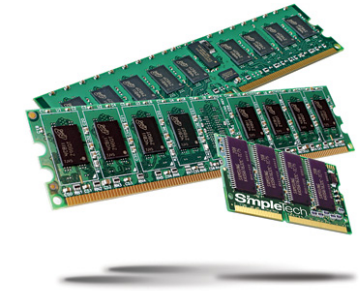
- *Fastest access is to the data held in processor registers. Registers are at the top of the memory hierarchy.*
- *Relatively small amount of memory that can be implemented on the processor chip. This is processor cache.*
- *Two levels of cache. Level 1 (L1) cache is on the processor chip. Level 2 (L2) 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.*

The Memory System

CACHE MEMORIES

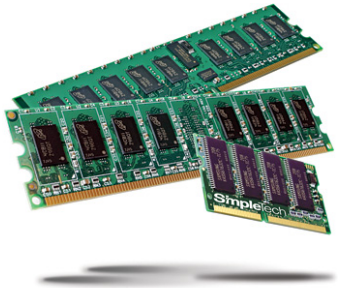
Cache Memory

- Processor is much faster than the main memory.
- 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".

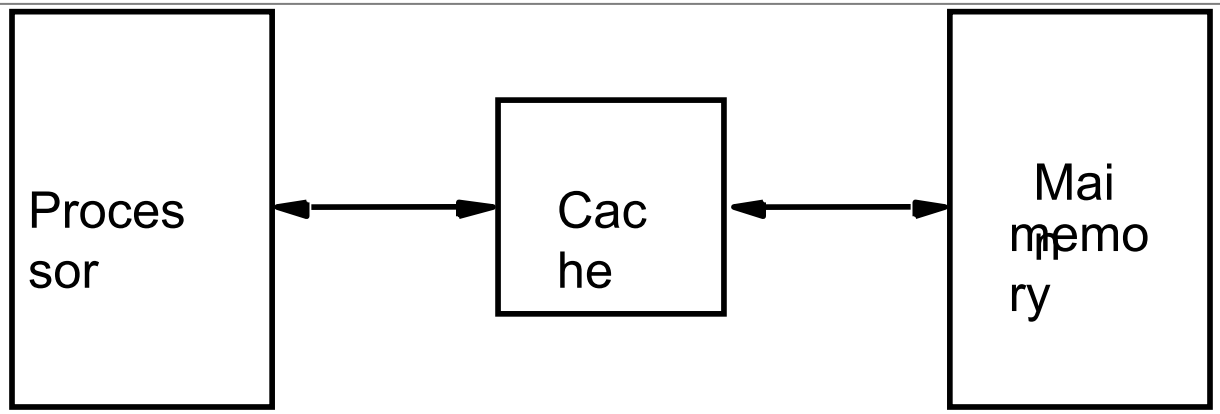


Locality of Reference

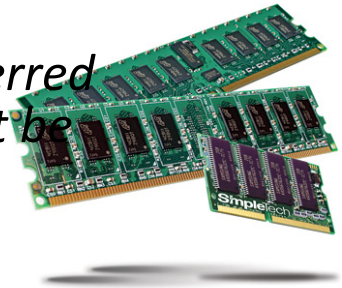
- Analysis of programs indicates that many instructions in localized areas of a program are executed repeatedly during some period of time, while the others are accessed relatively less frequently.
- Temporal locality of reference:
  - Recently executed instruction is likely to be executed again very soon.
- Spatial locality of reference:
  - Instructions with addresses close to a recently instruction are likely to be executed soon.



Cache memories

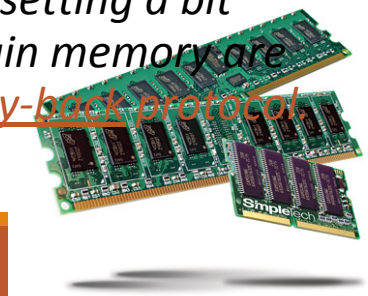


- *Processor issues a Read request, a block of words is transferred from the main memory to the cache, one word at a time.*
- *Subsequent references to the data in this block of words are found in the cache.*
- *At any given time, only some blocks in the main memory are held in the cache. Which blocks 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 Hit

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

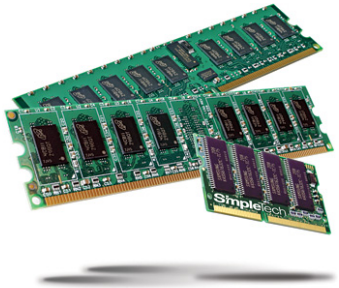


Cache Miss

- *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.*



Cache Coherence Problem

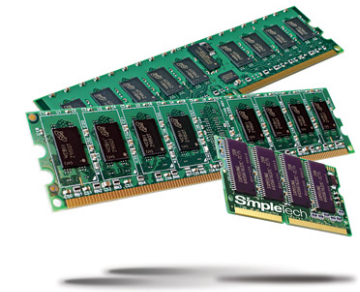
- *A bit called as "valid bit" is provided for each block.*
- *If the block contains valid data, then the bit is set to 1, else it is 0.*
- *Valid bits are set to 0, when the power is just turned on.*
- *When a block is loaded into the cache for the first time, the valid bit is set to 1.*

- *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.*

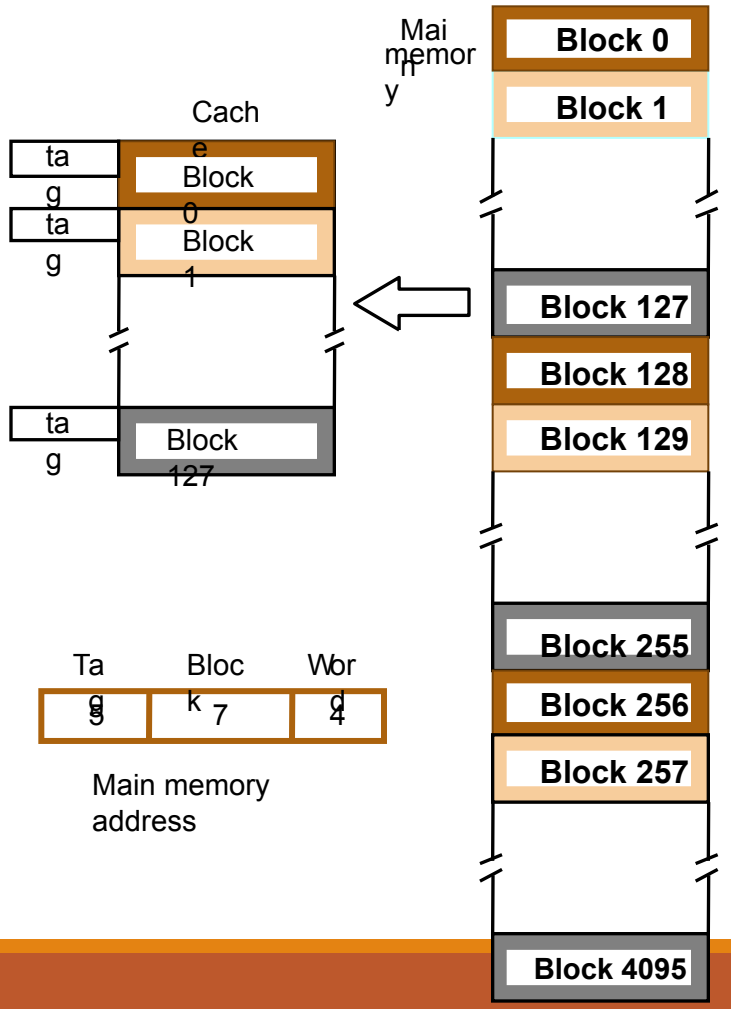
- *What happens if the data in the disk and main memory changes and the write-back protocol is being used?*
- *In this case, the data in the cache may also have changed and is indicated by the dirty bit.*
- *The copies of the data in the cache, and the main memory are different. This is called the cache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

Mapping Functions

- Mapping functions determine how memory blocks are placed in the cache.
- A simple processor example:
  - Cache consisting of 128 blocks of 16 words each.
  - Total size of cache is 2048 (2K) words.
  - Main memory is addressable by a 16-bit address.
  - Main memory has 64K words.
  - Main memory has 4K blocks of 16 words each.
- Three mapping functions:
  - Direct mapping
  - Associative mapping
  - Set-associative mapping.

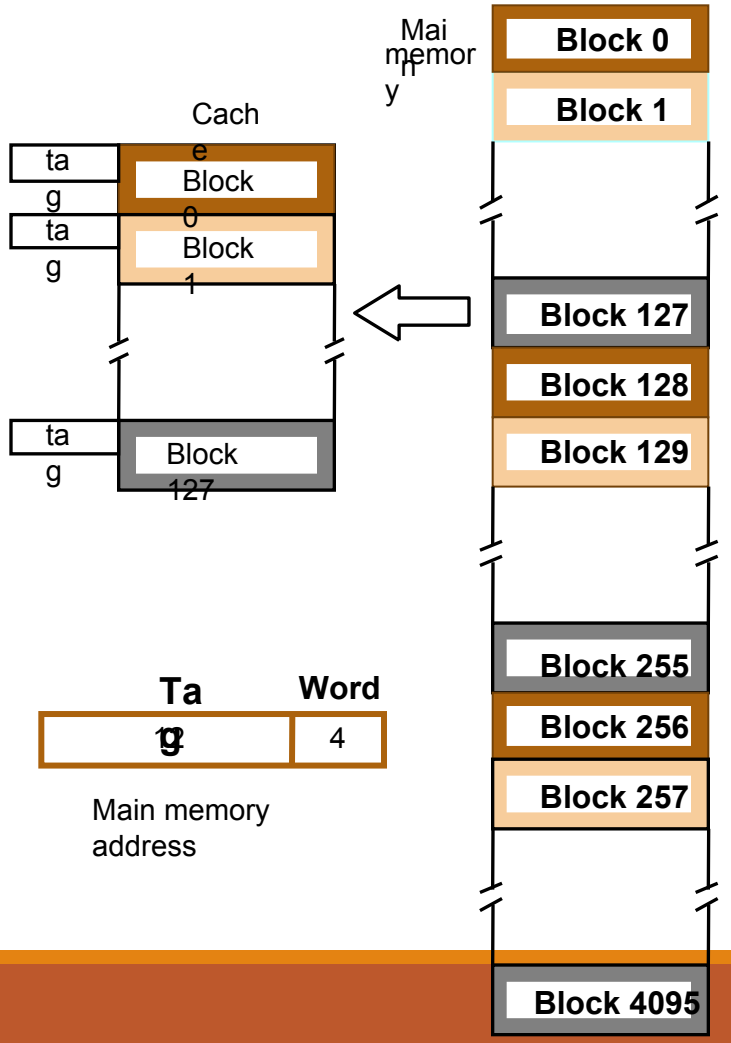


Direct Mapping



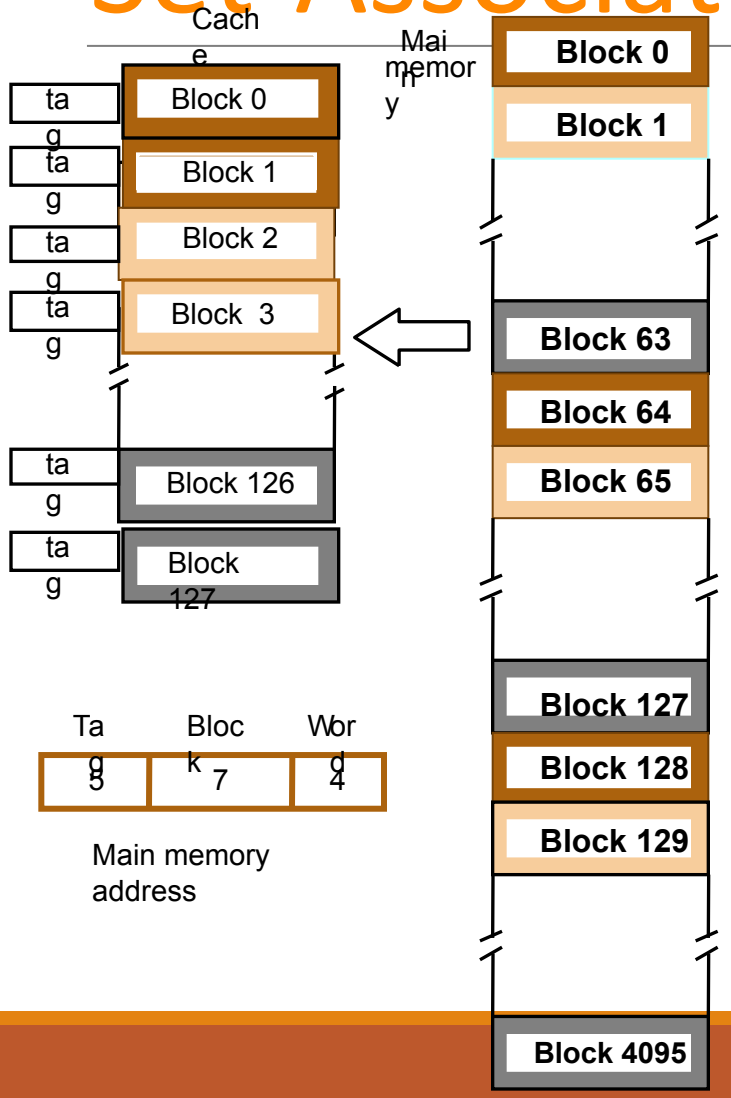
- *Block j of the main memory maps to j modulo 128 of the cache. 0 maps to 0, 129 maps to 1.*
- *More than one memory block is mapped onto the same position in the cache.*
- *May lead to contention for cache blocks even if the cache is not full.*
- *Resolve the contention by allowing new block to replace the old block, leading to a trivial replacement algorithm.*
- *Memory address is divided into three fields:*
  - *Low order 4 bits determine one of the 16 words in a block.*
  - *When a new block is brought into the cache, the next 7 bits determine which cache block this new block is placed in.*
  - *High order 5 bits determine which of the possible 32 blocks is currently present in the cache. These are tag bits.*
- *Simple to implement but not very flexible.*

Associative Mapping



- *Main memory block can be placed into any cache position.*
- *Memory address is divided into two fields:*
  - *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 128 patterns to determine whether a given block is in the cache.*

Set-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.
Divide the cache into 64 sets, with two blocks per set.
Memory block 0, 64, 128 etc. map to block 0, and they can occupy either of the two positions.
Memory address is divided into three fields:

- *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.*

Set-associative mapping combination of direct and associative mapping.
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.*

Replacement Algorithms

- For direct mapping where there is only one possible line for a block of memory, no replacement algorithm is needed.
- For associative and set associative mapping, however, an algorithm is needed.
- For maximum speed, this algorithm is implemented in the hardware. Four of the most common algorithms are:

1. Least Recently Used:- This replaces the candidate line in cache memory that has been there the longest with no reference to it.
2. First In First Out:- This replaces the candidate line in the cache that has been there the longest.
3. Least Frequently Used:- This replaces the candidate line in the cache that has had the fewest references.
4. Random Replacement:- This algorithm randomly chooses a line to be replaced from among the candidate lines. This yields only slightly inferior performance than other algorithms.

FIFO (First In First Out)

- Pages in main memory are kept in a list
- First in first out is very easy to implement
- The FIFO algorithm select the page for replacement that has been in memory the longest time

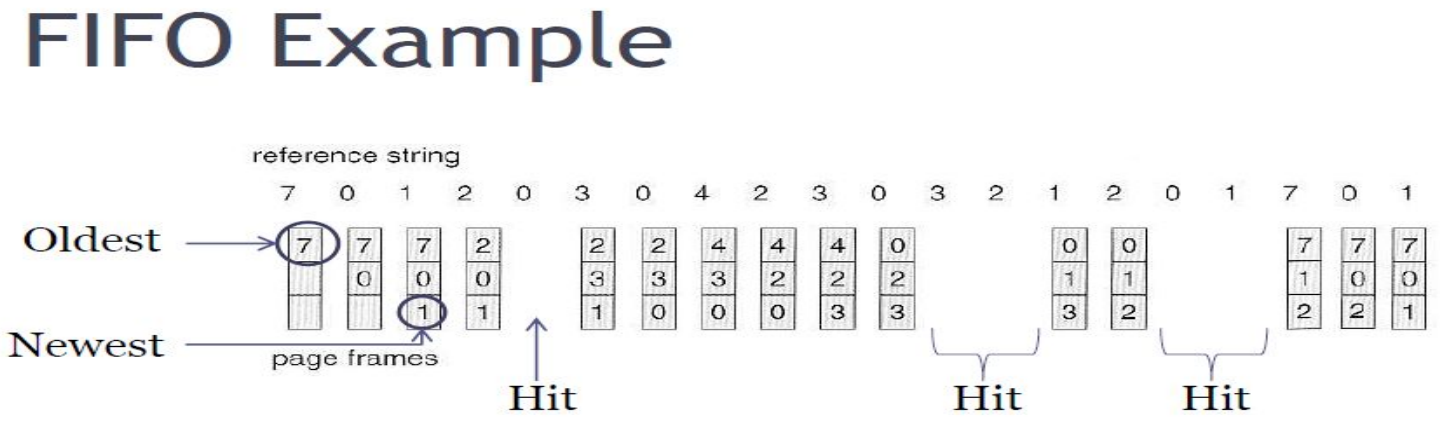


Fig: FIFO example

FIFO (First In First Out)

- Advantages:
  - FIFO is easy to understand.
  - It is very easy to implement.

- Disadvantages:
  - The oldest block in memory may be often used.

LRU (Least Recently Used)

- The least recently used page replacement algorithm keeps track page uses over a short period of time.

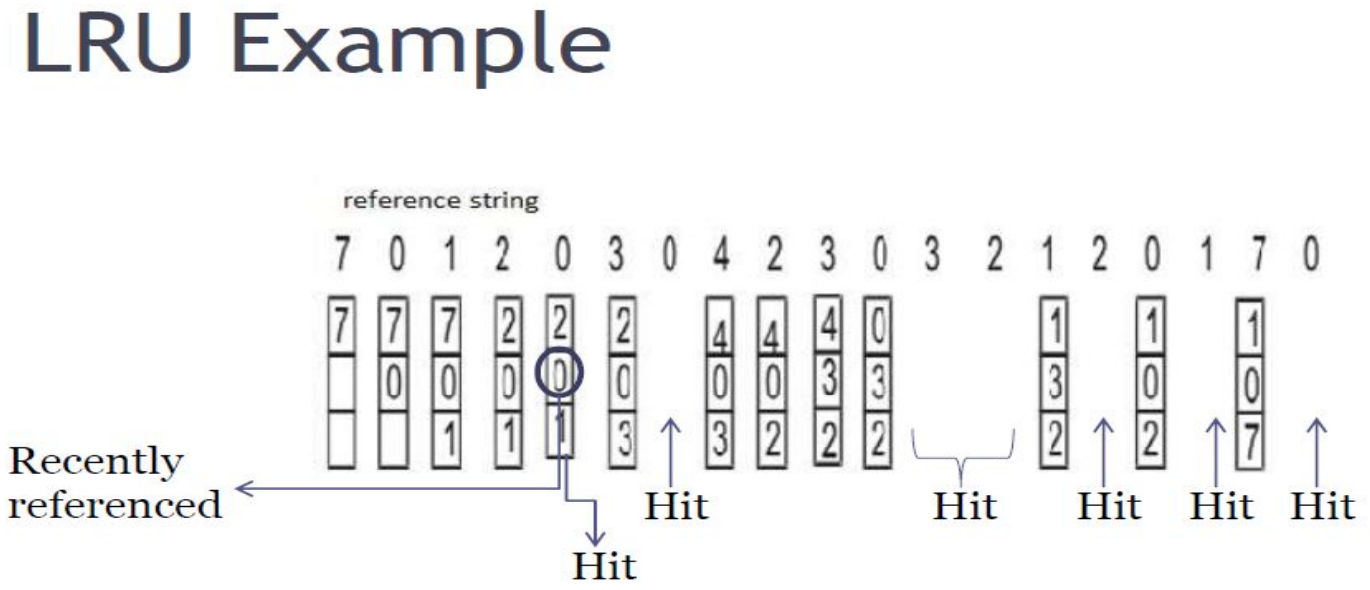
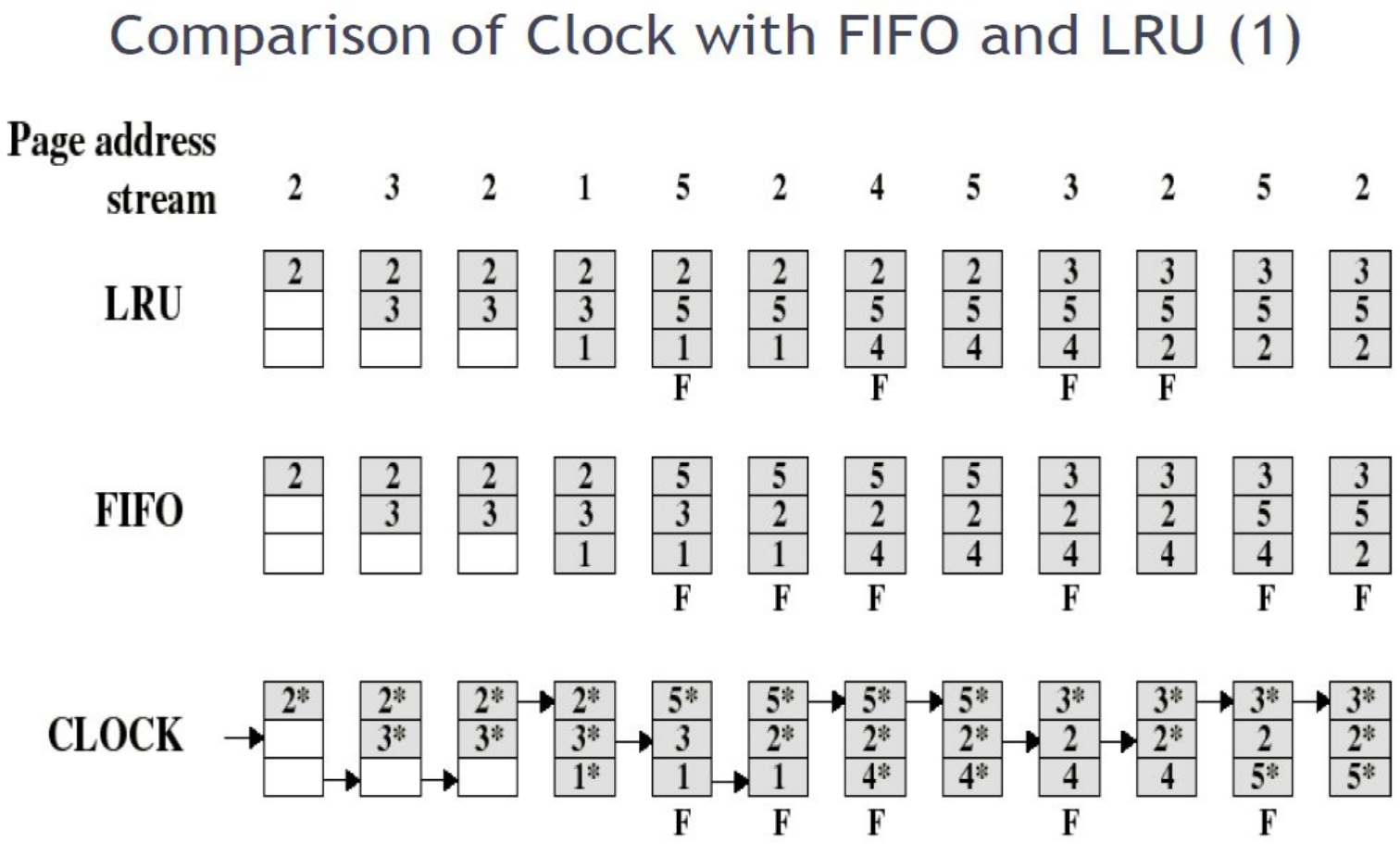


Fig: LRU example

LRU (Least Recently Used)

- Advantages:
  - LRU page replacement algorithm is quiet efficient.
- Disadvantages:
  - Implementation difficult. This algorithm requires keeping track of what was used when, which is expensive if one wants to make sure
  - the algorithm always discards the least recently used item.

Comparison of Clock with FIFO and LRU



LFU (Least Frequently Used)

- The Least-Frequently-Used (LFU) Replacement technique replaces the least-frequently block in use when an eviction must take place.
- Software counter associated with each block, initially zero is required in this algorithm.
- The operating system checks all the blocks in the cache at each clock interrupt.
- The R bit, which is '0' or '1', is added to the counter for each block. Consequently, the counters are an effort to keep track of the frequency of referencing each block.
- When a block must be replaced, the block that has the lowest counter is selected for the replacement.

LFU (Least Frequently Used)

Reference string:

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
F	F	F	F		F		F	F	F	F			F		F		F		

Number of page faults = 12.
Number of page hits= 8.

LFU (Least Frequently Used)

- Advantages:
  - Frequently used block will stay longer than (fifo)
- Disadvantages:
  - Older blocks are less likely to be removed , even if they are on longer frequently used because this algorithm never forgets anything.
  - Newer blocks are more likely to be replaced even if they are frequently used.
  - Captures only frequency factor.

Random Replacement

- When we need to evict a page, choose one randomly
- Advantage:
  - Extremely simple
- Disadvantages:
  - Can easily make "bad" choices by swapping out pages right before they are needed.

The Memory System

PERFORMANCE CONSIDERATIONS

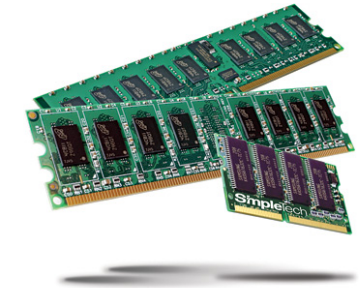
Performance Considerations

A key design objective of a computer system is to achieve the best possible performance at the lowest possible cost.

- Price/performance ratio is a common measure of success.

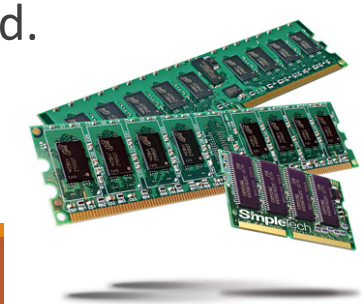
Performance of a processor depends on:

- How fast machine instructions can be brought into the processor for execution.
- How fast the instructions can be executed.

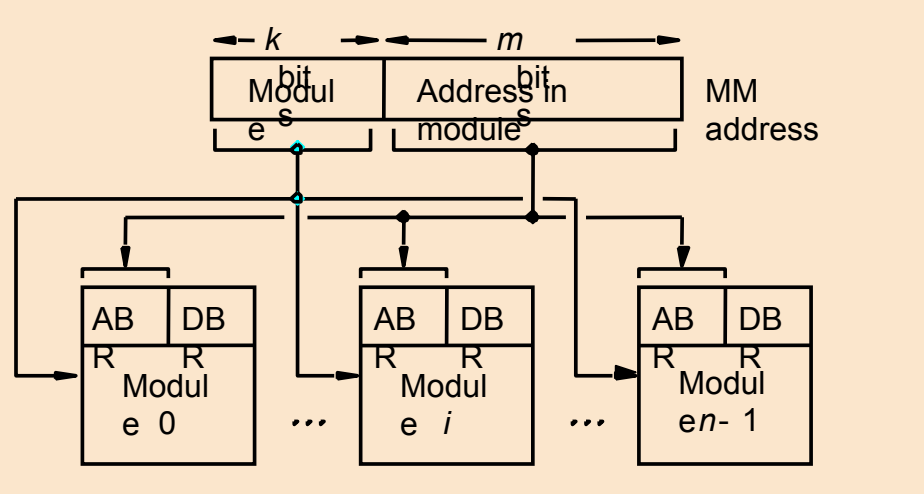


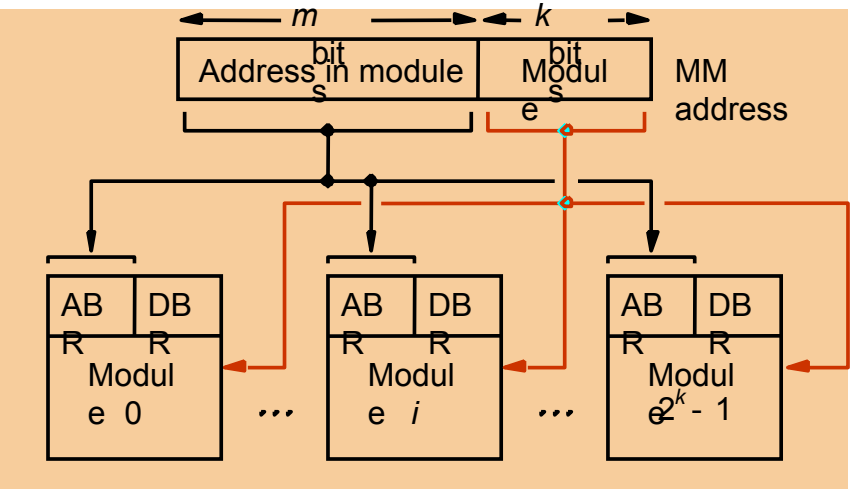
Interleaving

- Divides the memory system into a number of memory modules. Each module has its own address buffer register (ABR) and data buffer register (DBR).
- Arranges addressing so that successive words in the address space are placed in different modules.
- When requests for memory access involve consecutive addresses, the access will be to different modules.
- Since parallel access to these modules is possible, the average rate of fetching words from the Main Memory can be increased.



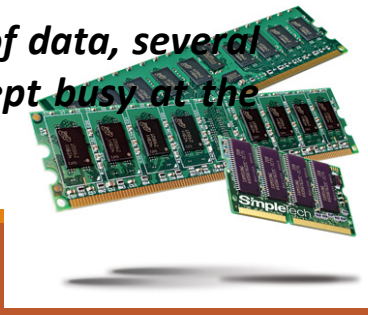
Methods of Address Layouts





- ***Consecutive words are placed in a module.***
- ***High-order k bits of a memory address determine the module.***
- ***Low-order m bits of a memory address determine the word within a module.***
- ***When a block of words is transferred from main memory to cache, only one module is busy at a time.***

- ***Consecutive words are located in consecutive modules.***
- ***Consecutive addresses can be located in consecutive modules.***
- ***While transferring a block of data, several memory modules can be kept busy at the same time.***



Hit Rate and Miss Penalty

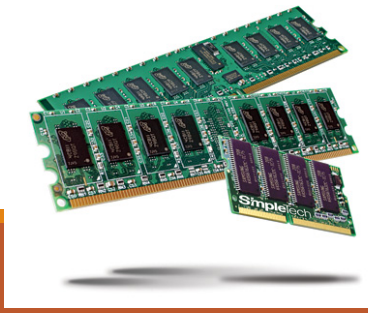
Hit rate

Miss penalty

Hit rate can be improved by increasing block size, while keeping cache size constant

Block sizes that are neither very small nor very large give best results.

Miss penalty can be reduced if load-through approach is used when loading new blocks into cache.

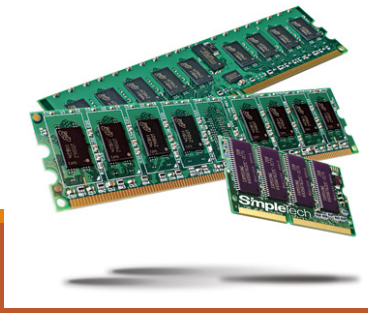


Caches on the Processor Chip

In high performance processors 2 levels of caches are normally used.

Avg access time in a system with 2 levels of caches is

$$T_{ave} = h1c1+(1-h1)h2c2+(1-h1)(1-h2)M$$



Other Performance Enhancements

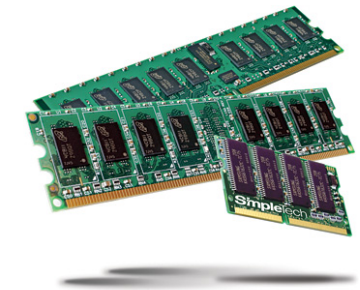
Write buffer

- *Write-through:*

- *Each write operation involves writing to the main memory.*
- *If the processor has to wait for the write operation to be complete, it slows down the processor.*
- *Processor does not depend on the results of the write operation.*
- *Write buffer can be included for temporary storage of write requests.*
- *Processor places each write request into the buffer and continues execution.*
- *If a subsequent Read request references data which is still in the write buffer, then this data is referenced in the write buffer.*

- *Write-back:*

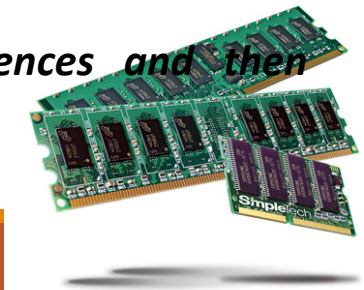
- *Block is written back to the main memory when it is replaced.*
- *If the processor waits for this write to complete, before reading the new block, it is slowed down.*
- *Fast write buffer can hold the block to be written, and the new block can be read first.*



Other Performance Enhancements (Contd.,)

Prefetching

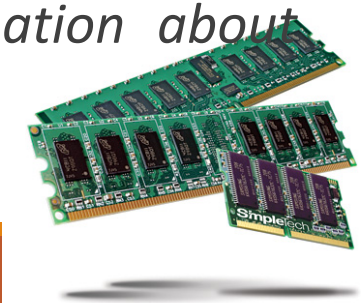
- *New data are brought into the processor when they are first needed.*
- *Processor has to wait before the data transfer is complete.*
- *Prefetch the data into the cache before they are actually needed, or a before a Read miss occurs.*
- *Prefetching can be accomplished through software by including a special instruction in the machine language of the processor.*
  - ***Inclusion of prefetch instructions increases the length of the programs**.*
- *Prefetching can also be accomplished using hardware:*
  - ***Circuitry that attempts to discover patterns in memory references and then prefetches according to this pattern.***



Other Performance Enhancements (Contd.,)

Lockup-Free Cache

- *Prefetching scheme does not work if it stops other accesses to the cache until the prefetch is completed.*
- *A cache of this type is said to be "locked" while it services a miss.*
- *Cache structure which supports multiple outstanding misses is called a lockup free cache.*
- *Since only one miss can be serviced at a time, a lockup free cache must include circuits that keep track of all the outstanding misses.*
- *Special registers may hold the necessary information about these misses.*

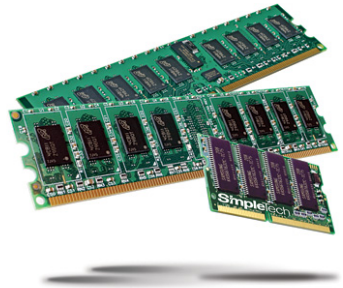


The Memory System

VIRTUAL MEMORY

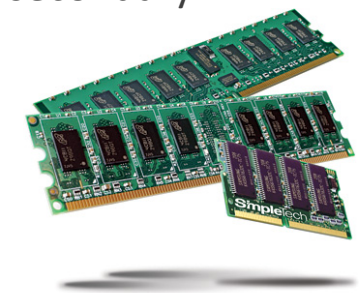
Virtual Memory

- Recall that an important challenge in the design of a computer system is to provide a large, fast memory system at an affordable cost.
- Architectural solutions to increase the effective speed and size of the memory system.
- Cache memories were developed to increase the effective speed of the memory system.
- Virtual memory is an architectural solution to increase the effective size of the memory system.



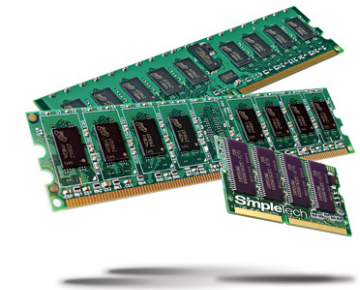
Virtual Memory (contd..)

- Recall that the addressable memory space depends on the number of address bits in a computer.
  - For example, if a computer issues 32-bit addresses, the addressable memory space is 4G bytes.
- Physical main memory in a computer is generally not as large as the entire possible addressable space.
  - Physical memory typically ranges from a few hundred megabytes to 1G bytes.
- Large programs that cannot fit completely into the main memory have their parts stored on secondary storage devices such as magnetic disks.
  - Pieces of programs must be transferred to the main memory from secondary storage before they can be executed.



Virtual Memory (contd..)

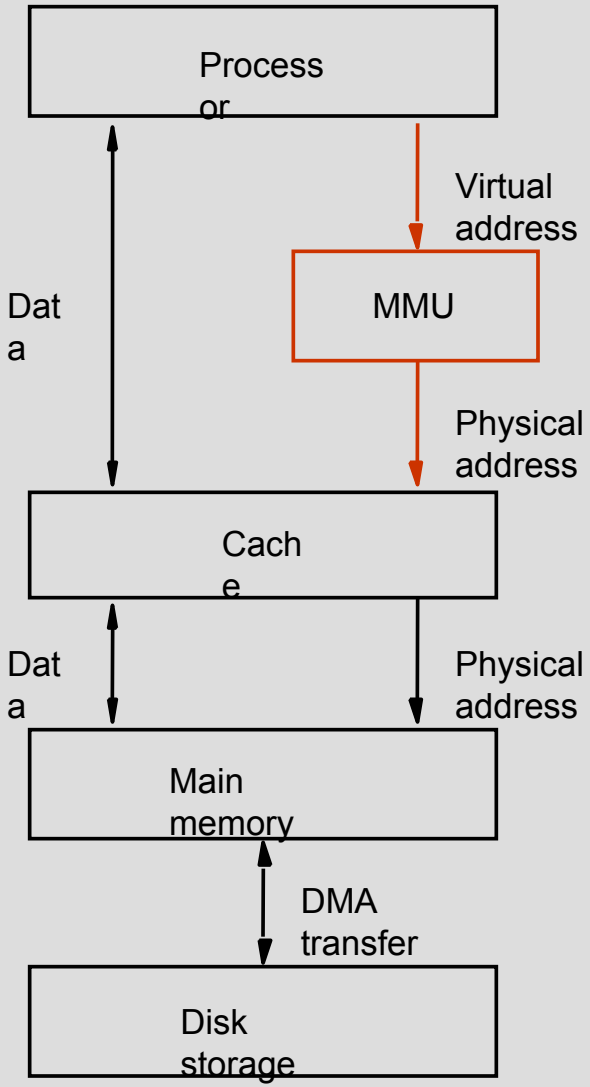
- When a new piece of a program is to be transferred to the main memory, and the main memory is full, then some other piece in the main memory must be replaced.
  - Recall this is very similar to what we studied in case of cache memories.
- Operating system automatically transfers data between the main memory and secondary storage.
  - Application programmer need not be concerned with this transfer.
  - Also, application programmer does not need to be aware of the limitations imposed by the available physical memory.



Virtual Memory (contd..)

- Techniques that automatically move program and data between main memory and secondary storage when they are required for execution are called virtual-memory techniques.
- Programs and processors reference an instruction or data independent of the size of the main memory.
- Processor issues binary addresses for instructions and data called logical or virtual addresses.
- Virtual addresses are translated into physical addresses by a combination of hardware and software subsystems.
  - If virtual address refers to a part of the program that is currently in the main memory, it is accessed immediately.
  - If the address refers to a part of the program that is not currently in the main memory, it is first transferred to the main memory before it can be u

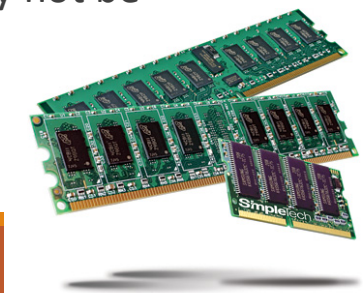
Virtual Memory Organizatio



- *Memory management unit (MMU) translates virtual addresses into physical addresses.*
- *If the desired data or instructions are in the main memory they are fetched as described previously.*
- *If the desired data or instructions are not in the main memory, they must be transferred from secondary storage to the main memory.*
- *MMU causes the operating system to bring the data from the secondary storage into the main memory.*

Address Translation

- Assume that program and data are composed of fixed-length units called pages.
- A page consists of a block of words that occupy contiguous locations in the main memory.
- Page is a basic unit of information that is transferred between secondary storage and main memory.
- Size of a page commonly ranges from 2K to 16K bytes.
  - Pages should not be too small, because the access time of a secondary storage device is much larger than the main memory.
  - Pages should not be too large, else a large portion of the page may not be used, and it will occupy valuable space in the main memory.



Address Translation (contd..)

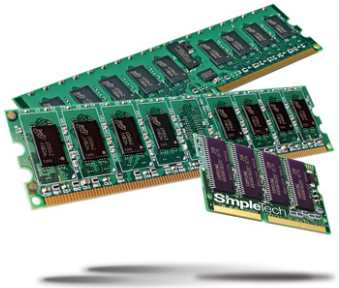
Concepts of virtual memory are similar to the concepts of cache memory.

Cache memory:

- Introduced to bridge the speed gap between the processor and the main memory.
- Implemented in hardware.

Virtual memory:

- Introduced to bridge the speed gap between the main memory and secondary storage.
- Implemented in part by software.



Address Translation (contd..)

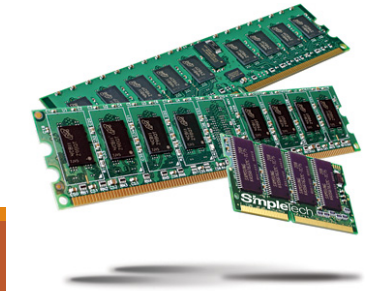
- Each virtual or logical address generated by a processor is interpreted as a virtual page number (high-order bits) plus an offset (low-order bits) that specifies the location of a particular byte within that page.
- Information about the main memory location of each page is kept in the page table.
  - Main memory address where the page is stored.
  - Current status of the page.
- Area of the main memory that can hold a page is called as page frame.
- Starting address of the page table is kept in a page table base register.

Address Translation (contd..)

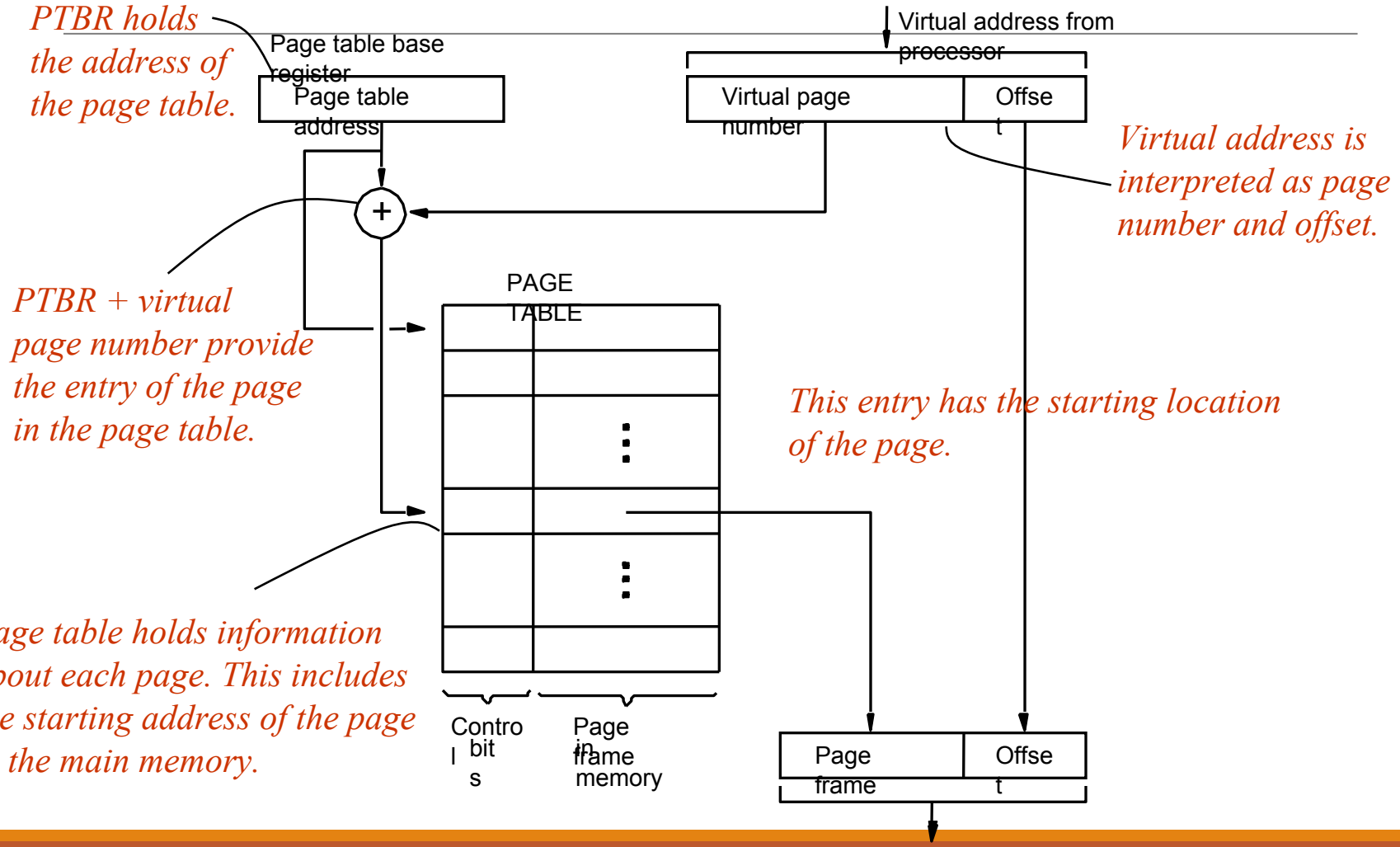
Virtual page number generated by the processor is added to the contents of the page table base register.

- This provides the address of the corresponding entry in the page table.

The contents of this location in the page table give the starting address of the page if the page is currently in the main memory.

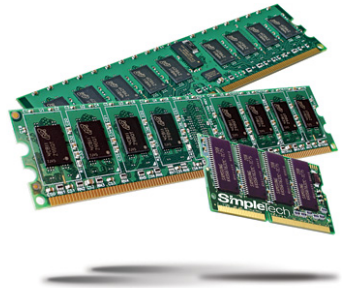


Address Translation (conte



Address Translation (contd..)

- Page table entry for a page also includes some control bits which describe the status of the page while it is in the main memory.
- One bit indicates the validity of the page.
  - Indicates whether the page is actually loaded into the main memory.
  - Allows the operating system to invalidate the page without actually removing it.
- One bit indicates whether the page has been modified during its residency in the main memory.
  - This bit determines whether the page should be written back to the disk when it is removed from the main memory.
  - Similar to the dirty or modified bit in case of cache memory.



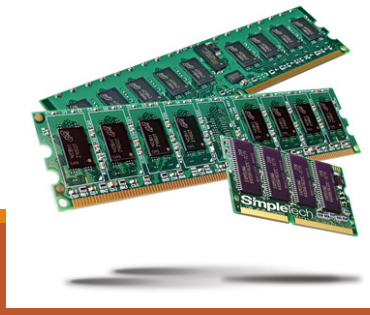
Address Translation (contd..)

Other control bits for various other types of restrictions that may be imposed.

- For example, a program may only have read permission for a page, but not write or modify permissions.

Address Translation (contd..)

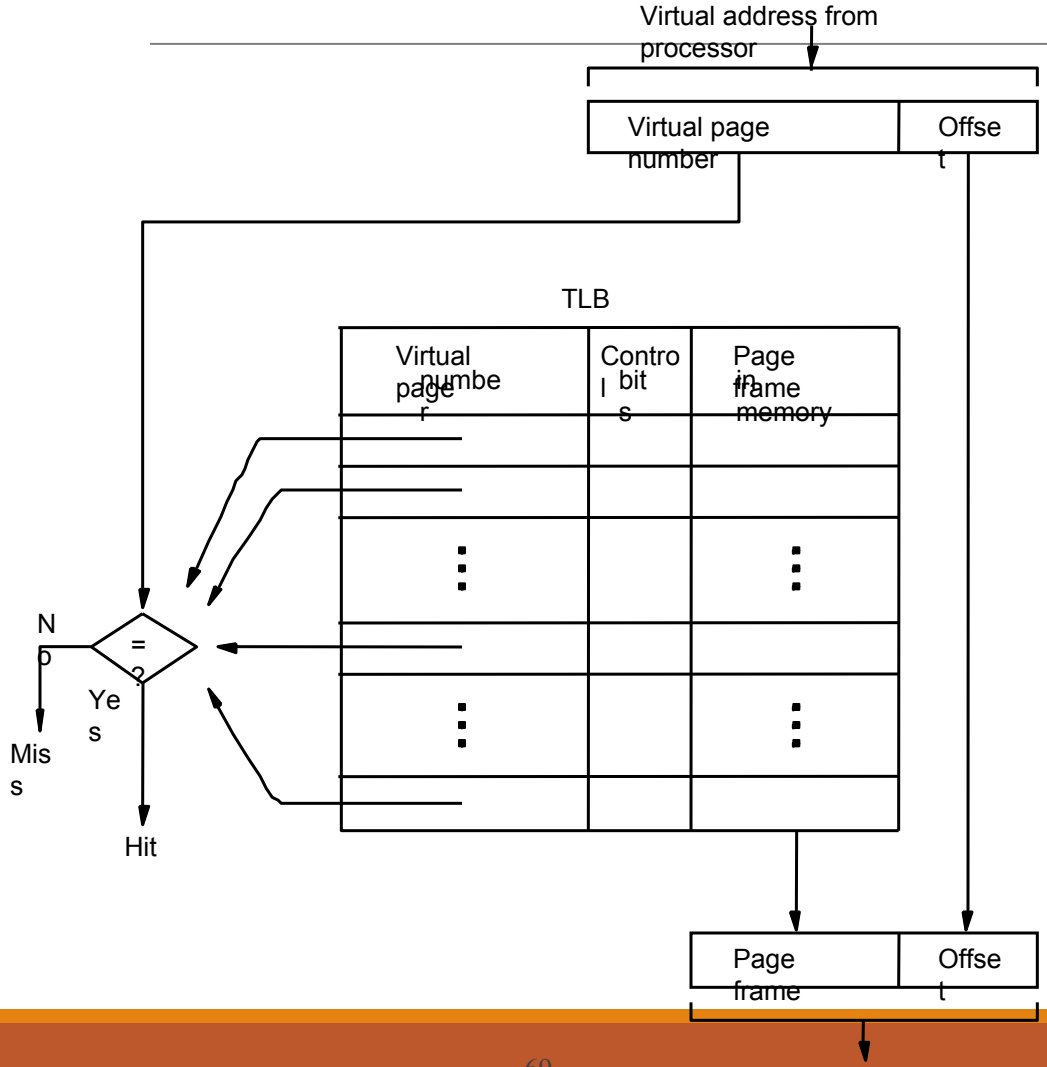
- Where should the page table be located?
- Recall that the page table is used by the MMU for every read and write access to the memory.
  - Ideal location for the page table is within the MMU.
- Page table is quite large.
- MMU is implemented as part of the processor chip.
- Impossible to include a complete page table on the chip.
- Page table is kept in the main memory.
- A copy of a small portion of the page table can be accommodated within the MMU.
  - Portion consists of page table entries that correspond to the most recently accessed pages.



Address Translation (contd..)

- A small cache called as Translation Lookaside Buffer (TLB) is included in the MMU.
  - TLB holds page table entries of the most recently accessed pages.
- Recall that cache memory holds most recently accessed blocks from the main memory.
  - Operation of the TLB and page table in the main memory is similar to the operation of the cache and main memory.
- Page table entry for a page includes:
  - Address of the page frame where the page resides in the main memory.
  - Some control bits.
- In addition to the above for each page, TLB must hold the virtual page number for each page.

Address Translation (contd..)



Associative-mapped TLB

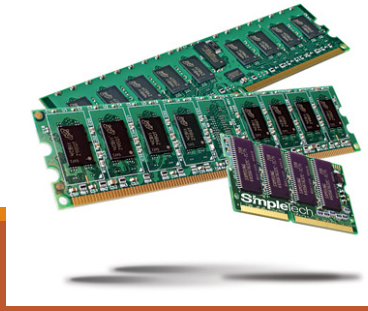
High-order bits of the virtual address generated by the processor select the virtual page.
These bits are compared to the virtual page numbers in the TLB.
If there is a match, a hit occurs and the corresponding address of the page frame is read.
If there is no match, a miss occurs and the page table within the main memory must be consulted.
Set-associative mapped TLBs are found in commercial processors.

Address Translation (contd..)

- How to keep the entries of the TLB coherent with the contents of the page table in the main memory?
- Operating system may change the contents of the page table in the main memory.
  - Simultaneously it must also invalidate the corresponding entries in the TLB.
- A control bit is provided in the TLB to invalidate an entry.
- If an entry is invalidated, then the TLB gets the information for that entry from the page table.
  - Follows the same process that it would follow if the entry is not found in the TLB or if a “miss” occurs.

Address Translation (contd..)

- What happens if a program generates an access to a page that is not in the main memory?
- In this case, a page fault is said to occur.
  - Whole page must be brought into the main memory from the disk, before the execution can proceed.
- Upon detecting a page fault by the MMU, following actions occur:
  - MMU asks the operating system to intervene by raising an exception.
  - Processing of the active task which caused the page fault is interrupted.
  - Control is transferred to the operating system.
  - Operating system copies the requested page from secondary storage to the main memory.
  - Once the page is copied, control is returned to the task which was interrupted.



Address Translation (contd..)

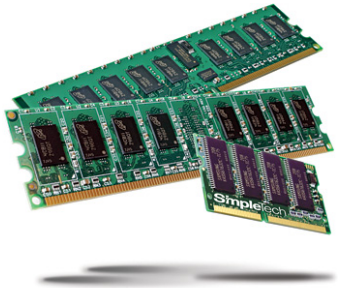
Servicing of a page fault requires transferring the requested page from secondary storage to the main memory.

This transfer may incur a long delay.

While the page is being transferred the operating system may:

- Suspend the execution of the task that caused the page fault.
- Begin execution of another task whose pages are in the main memory.

Enables efficient use of the processor.



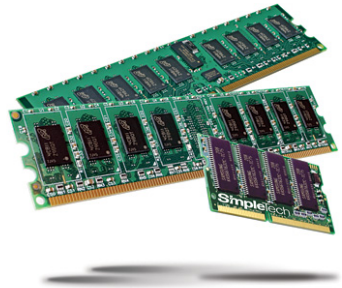
Address Translation (contd..)

How to ensure that the interrupted task can continue correctly when it resumes execution?

There are two possibilities:

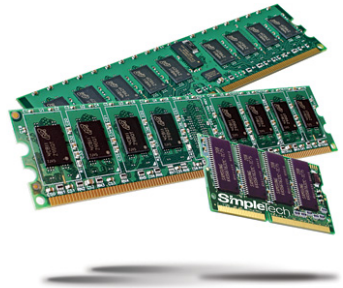
- Execution of the interrupted task must continue from the point where it was interrupted.
- The instruction must be restarted.

Which specific option is followed depends on the design of the processor.



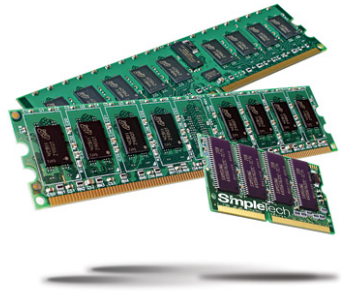
Address Translation (contd..)

- When a new page is to be brought into the main memory from secondary storage, the main memory may be full.
  - Some page from the main memory must be replaced with this new page.
- How to choose which page to replace?
  - This is similar to the replacement that occurs when the cache is full.
  - The principle of locality of reference (?) can also be applied here.
  - A replacement strategy similar to LRU can be applied.
- Since the size of the main memory is relatively larger compared to cache, a relatively large amount of programs and data can be held in the main memory.
  - Minimizes the frequency of transfers between secondary storage and main memory.



Address Translation (contd..)

- A page may be modified during its residency in the main memory.
- When should the page be written back to the secondary storage?
- Recall that we encountered a similar problem in the context of cache and main memory:
  - Write-through protocol(?)
  - Write-back protocol(?)
- Write-through protocol cannot be used, since it will incur a long delay each time a small amount of data is written to the disk.



Input and Output Organization: Data Transfer Techniques

- The I/O subsystem of a computer provides an efficient mode of communication between the central system and the outside environment.
- It handles all the input-output operations of the computer system.

Input and Output Organization: Data Transfer Techniques

- **Peripheral Devices:**
  - Input or output devices that are connected to computer are called **peripheral devices.**
  - These devices are designed to read information into or out of the memory unit upon command from the CPU and are considered to be the part of computer system. These devices are also called **peripherals.**

  For example: Keyboards, display units and printers are common peripheral devices.

- There are three types of peripherals:
  - **Input peripherals :** Allows user input, from the outside world to the computer. Example: Keyboard, Mouse etc.
  - **Output peripherals:** Allows information output, from the computer to the outside world. Example: Printer, Monitor etc.
  - **Input-Output peripherals:** Allows both input (from outside world to computer) as well as, output(from computer to the outside world). Example: Touch screen

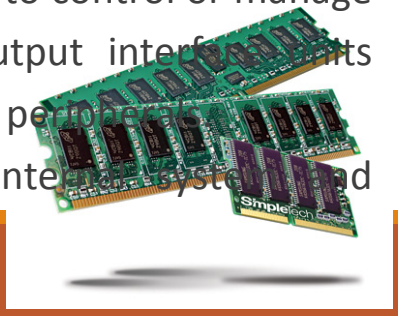
Input and Output Organization: Data Transfer Techniques

- **Interfaces:**

  Interface is a shared boundary between two separate components of the computer system which can be used to attach two or more components to the system for communication purposes.

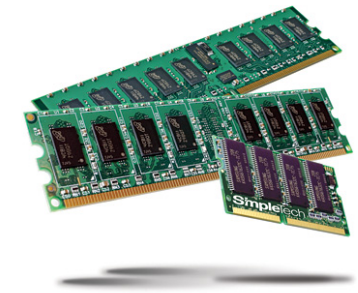
- There are two types of interface:
  - **CPU Inteface**
  - **I/O Interface**

- **Input-Output Interface:**
  - Peripherals connected to a computer need special communication links for interfacing with CPU.
  - It is a special hardware components between the CPU and peripherals to control or manage the input-output transfers. These components are called input-output interface units because they provide communication links between processor bus and peripherals.
  - They provide a method for transferring information between internal system and input-output devices.



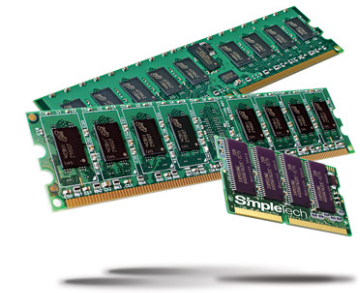
Modes of I/O Data Transfer

- Data transfer between the central unit and I/O devices can be handled in generally three types of modes which are given below:
  - **Programmed I/O**
  - **Interrupt Initiated I/O**
  - **Direct Memory Access**



Programmed I/O

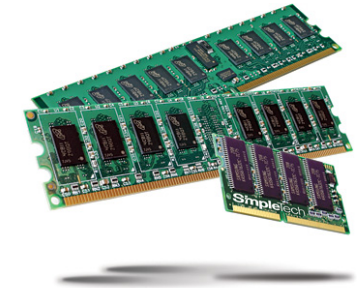
- Programmed I/O instructions are the result of I/O instructions written in computer program. Each data item transfer is initiated by the instruction in the program.
- Usually the program controls data transfer to and from CPU and peripheral. Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.



Interrupt Initiated I/O

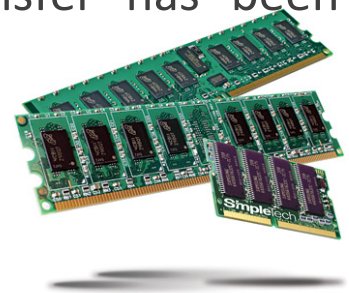
- In the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is time consuming process because it keeps the processor busy needlessly.

- This problem can be overcome by using interrupt initiated I/O. In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt. After receiving the interrupt signal, the CPU stops the task which it is processing and service the I/O transfer and then returns back to its previous processing task.

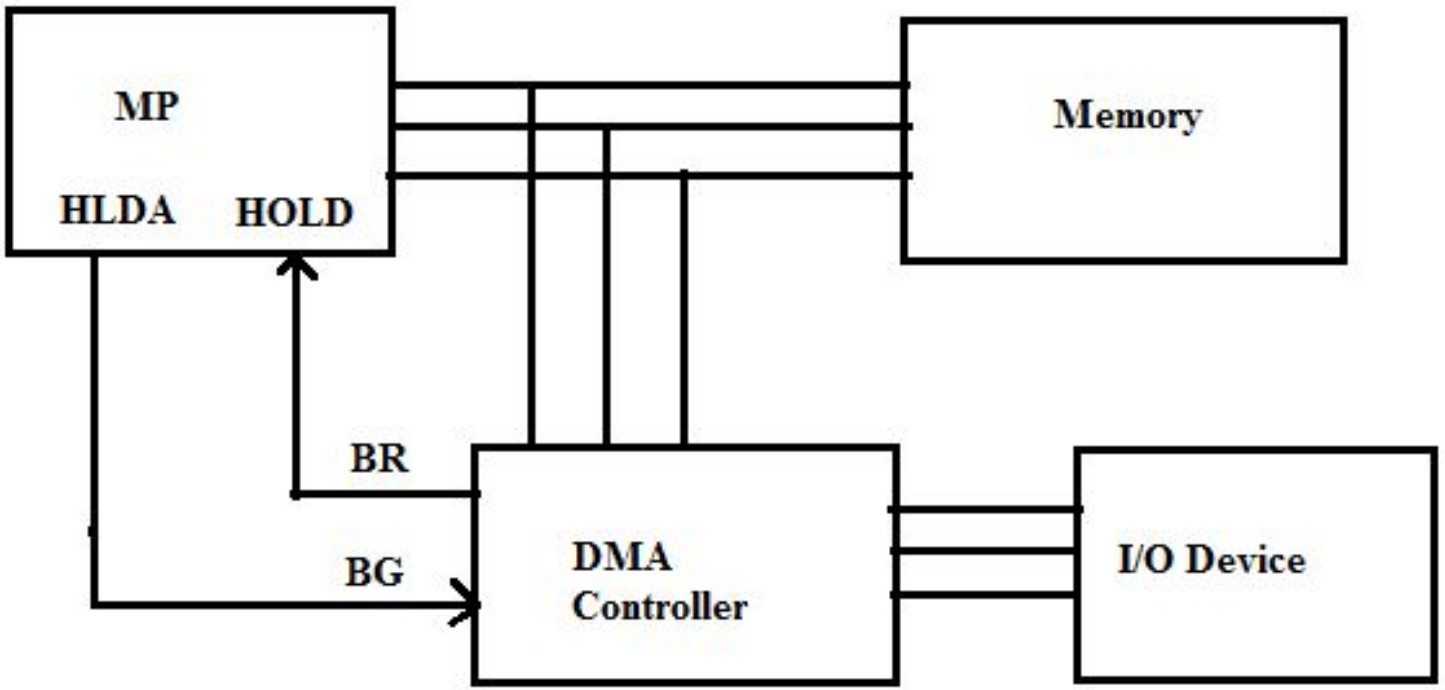


Direct Memory Access (DMA)

- Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This technique is known as **DMA.**
- In this, the interface transfer data to and from the memory through memory bus. A DMA controller manages to transfer data between peripherals and memory unit.
- Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc. It is also used for intra chip data transfer in multicore processors. In DMA, CPU would initiate the transfer, do other operations while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed.



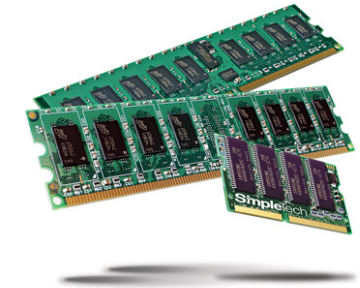
Direct Memory Access



Input/Output Processor

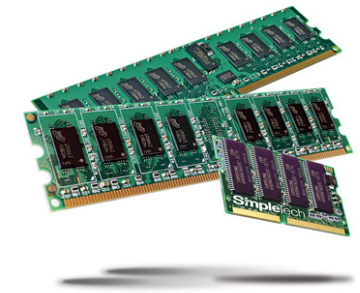
- An **Input-Output Processor (IOP)** is a processor with direct memory access capability. In this, the computer system is divided into a memory unit and number of processors.

- Each IOP controls and manage the input-output tasks. The IOP is similar to CPU except that it handles only the details of I/O processing. The IOP can fetch and execute its own instructions. These IOP instructions are designed to manage I/O transfers only.



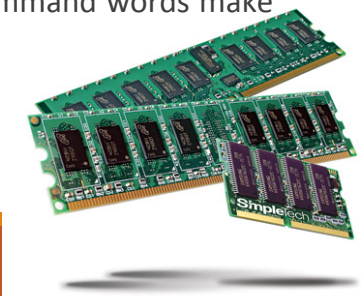
Input/Output Processor

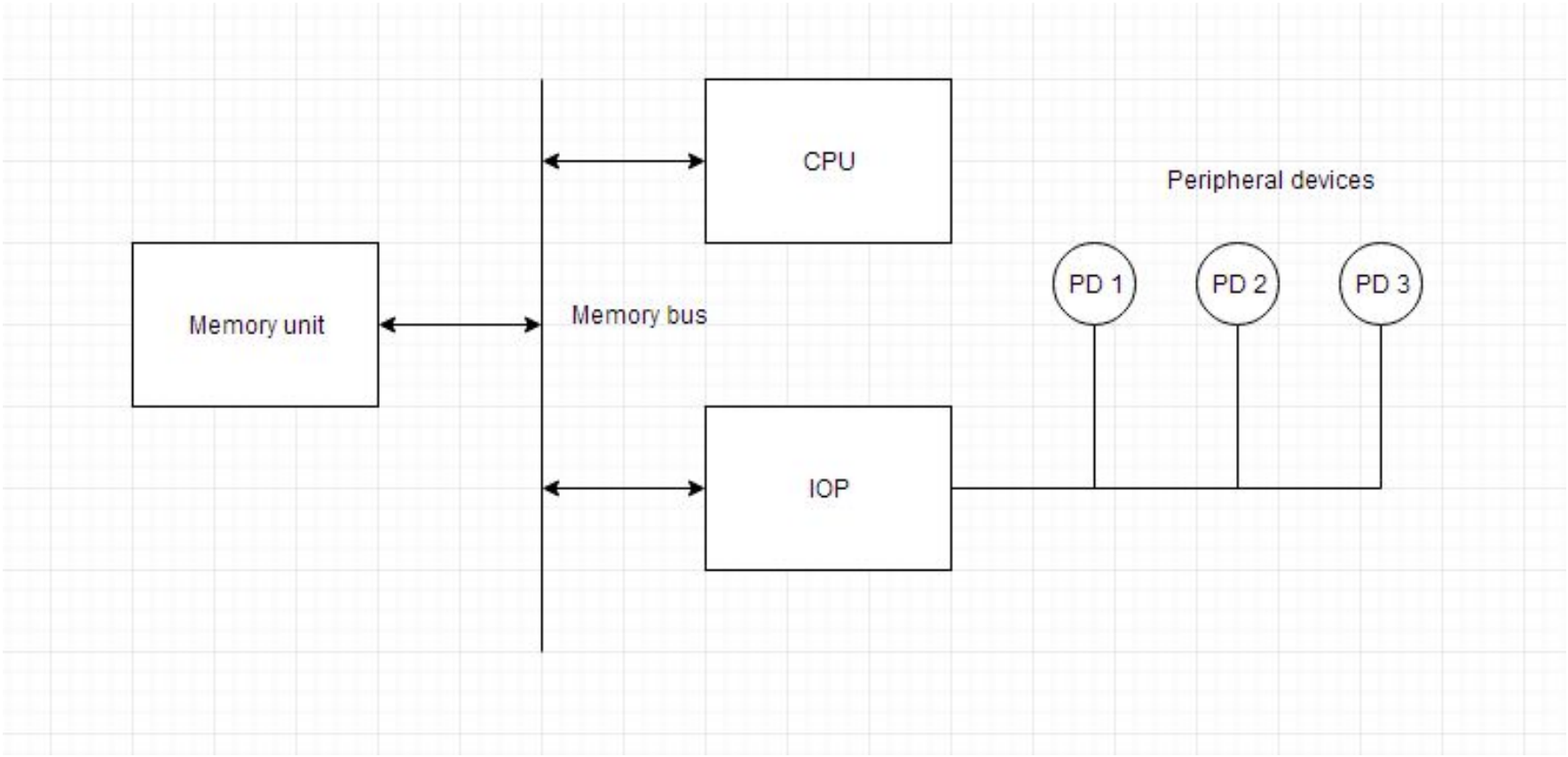
- Below is a block diagram of a computer along with various I/O Processors. The memory unit occupies the central position and can communicate with each processor.
- The CPU processes the data required for solving the computational tasks. The IOP provides a path for transfer of data between peripherals and memory. The CPU assigns the task of initiating the I/O program.
- The IOP operates independent from CPU and transfer data between peripherals and memory.



Input/Output Processor

- The communication between the IOP and the devices is similar to the program control method of transfer. And the communication with the memory is similar to the direct memory access method.
- In large scale computers, each processor is independent of other processors and any processor can initiate the operation.
- The CPU can act as master and the IOP act as slave processor. The CPU assigns the task of initiating operations but it is the IOP, who executes the instructions, and not the CPU. CPU instructions provide operations to start an I/O transfer. The IOP asks for CPU through interrupt.
- Instructions that are read from memory by an IOP are also called commands to distinguish them from instructions that are read by CPU. Commands are prepared by programmers and are stored in memory. Command words make the program for IOP. CPU informs the IOP where to find the commands in memory.

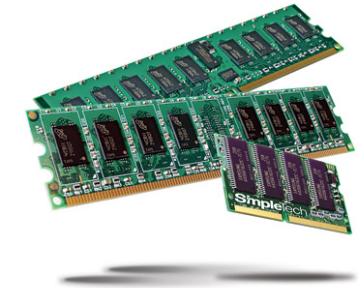




Input/Output Processor

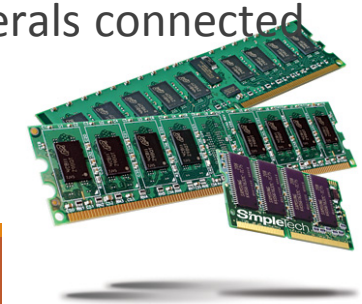
Need for Input/Output Processor

- Input Output Interface provides a method for transferring information between internal storage and external I/O devices.
- Peripherals connected to a computer need special communication links for interfacing them with the central processing unit.
- The purpose of communication link is to resolve the differences that exist between the central computer and each peripheral.



Need for Input/Output Processor

- The Major Differences are:-
  - Peripherals are electromechanically and electromagnetic devices and CPU and memory are electronic devices. Therefore, a conversion of signal values may be needed.
  - The data transfer rate of peripherals is usually slower than the transfer rate of CPU and consequently, a synchronization mechanism may be needed.
  - Data codes and formats in the peripherals differ from the word format in the CPU and memory.
  - The operating modes of peripherals are different from each other and must be
  - controlled so as not to disturb the operation of other peripherals connected to the CPU.



The Memory System

MEMORY MANAGEMENT

Memory management

Operating system is concerned with transferring programs and data between secondary storage and main memory.

Operating system needs memory routines in addition to the other routines.

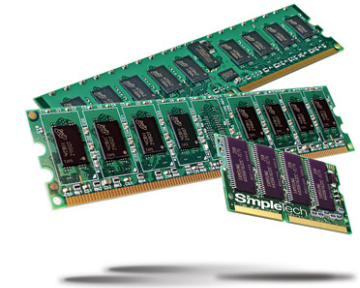
Operating system routines are assembled into a virtual address space called system space.

System space is separate from the space in which user application programs reside.

- This is user space.

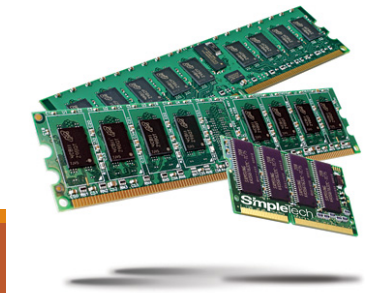
Virtual address space is divided into one

system space + several user spaces.



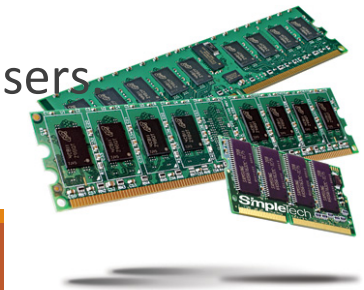
Memory management (contd..)

- Recall that the Memory Management Unit (MMU) translates logical or virtual addresses into physical addresses.
- MMU uses the contents of the page table base register to determine the address of the page table to be used in the translation.
  - Changing the contents of the page table base register can enable us to use a different page table, and switch from one space to another.
- At any given time, the page table base register can point to one page table.
  - Thus, only one page table can be used in the translation process at a given time.
  - Pages belonging to only one space are accessible at any given time.



Memory management (contd..)

- When multiple, independent user programs coexist in the main memory, how to ensure that one program does not modify/destroy the contents of the other?
- Processor usually has two states of operation:
  - Supervisor state.
  - User state.
- Supervisor state:
  - Operating system routines are executed.
- User state:
  - User programs are executed.
  - Certain privileged instructions cannot be executed in user state.
  - These privileged instructions include the ones which change page table base register.
  - Prevents one user from accessing the space of other users.



Memory Mapped I/O

There is no specific input or output instructions

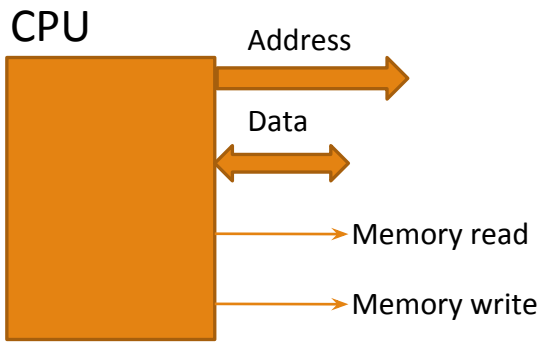
The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words.

Each interface is organized as set of registers(read & write in normal address space).

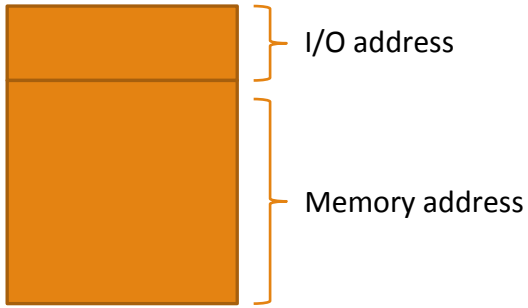
Memory mapped I/O can use memory type instructions to access I/O data.

It allows the computer to use the same instructions for either i/o transfer or for memory transfers.

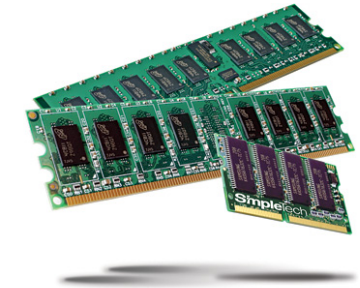
The advantage is that the load and store instructions used for reading and writing from memory can be used to input and output data from I/O registers.



a) CPU Signals

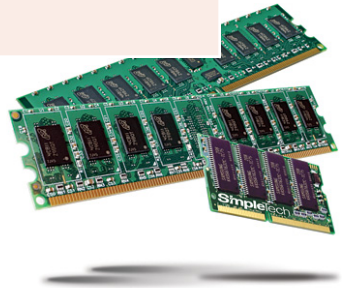


b) Address space division



Difference between Memory mapped I/O and I/O mapped I/O

	Memory Mapped Input/Output	Input/Output Mapped Input/Output
1.	Each port is treated as a memory location.	Each port is treated as an independent unit.
2.	CPU's memory address space is divided between memory and input/output ports.	Separate address spaces for memory and input/output ports.
3.	Single instruction can transfer data between memory and port.	Two instruction are necessary to transfer data between memory and port.
4.	Data transfer is by means of instruction like MOVE.	Each port can be accessed by means of IN or OUT instructions.



Program Controlled I/O

Program controlled I/O is one in which the processor repeatedly checks a status flag to achieve the required synchronization between processor & I/O device.

The processor polls the device.

It is useful in small low speed systems where hardware cost must be minimized.

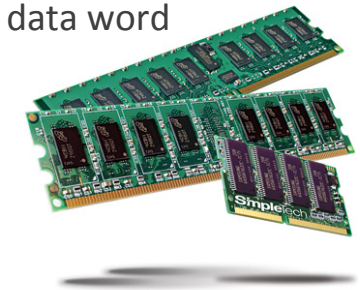
It requires that all input/output operators be executed under the direct control of the CPU.

The transfer is between CPU registers(accumulator) and a buffer register connected to the input/output device.

The i/o device does not have direct access to main memory.

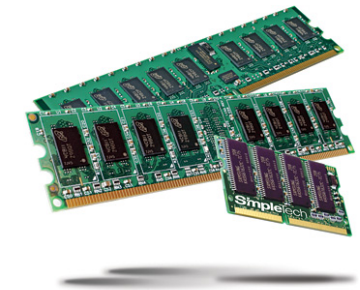
A data transfer from an input/output device to main memory requires the execution of several instructions by the CPU, including an input instruction to transfer a word from the input/output device to the CPU and a store instruction to transfer a word from CPU to main memory.

One or more additional instructions may be needed for address communication and data word counting.



Typical Program Controlle Instructions

Name	Mnemonic
Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Return	RET
Compare	CMP
Test(by ADDing)	TST



Interrupts

A Suspension of a process such as the execution of a computer program, caused by an event external to that process, and performed in such a way that the process can be resumed. A way to improve processor utilization.

Need For Interrupts?

The OS is a reactive program

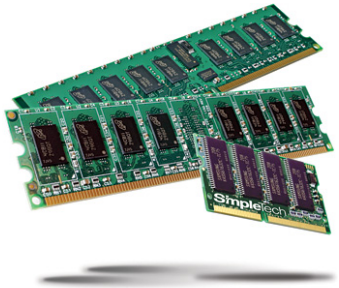
1. When you give some input
2. It will perform computations
3. Produces output BUT
4. Meanwhile you can interact with the system by interrupting the running process

or

5. You can stop and start another process.

This reactiveness is due to interrupts

Modern Operating Systems Are Interrupt driven



Interrupt Hardware

I/O device request an interrupt by activating a bus line called interrupt-request.

A single interrupt request line may be used to serve n devices.

All devices are connected to interrupt request line via switches to ground.

Interrupt Hardware

- Need to be able to handle an unknown number of possible interrupting devices
- INTR = INTR1 or INTR2 or … INTR n

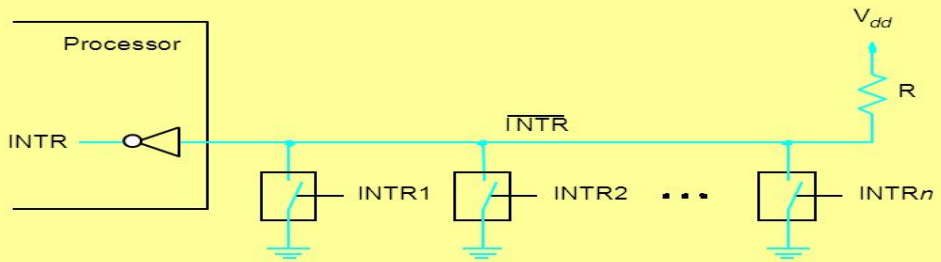
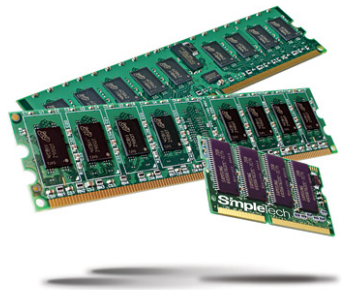


Figure 4.6.An equivalent circuit for an open-drain bus used to implement a common interrupt-request line.



Interrupt Hardware Cont….

To request an interrupt, a device closes its associated switch.

If all interrupt-request signals INTR1 to INTRn are inactive, that is, if all switches are open, the voltage on the interrupt request line will equal to Vdd.

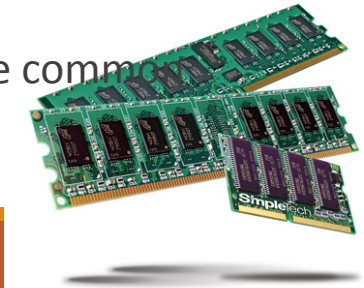
This is an inactivate state of the line.

When a device requests an interrupt by closing its switch, the voltage on the line drops to 0, causing the interrupt- request signal INTR received by the processor to go to 1.

If closing of one (or) more switches that cause the line value to drop to 0, the value of logical OR of the request from individual devices, that is

INTR=INTR1+INTR2+INTR3.....

Use the complement form of INTR to name of the interrupt signal on the comm
line because this signal is active in the low voltage state



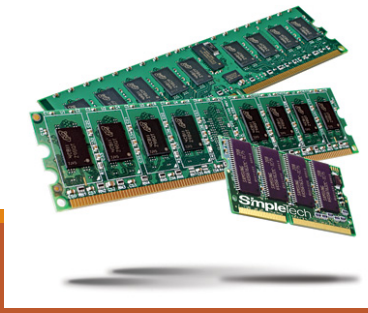
Enabling and Disabling Interrupts

A processor has the facility to enable and disable interrupts as desired.

When a device request the interrupt during the processor service for another interrupt, the result cause the processor enter into the infinite loop.

This can be handled by the following 2 ways:

- The processor ignore the interrupt request line(INTR) until the Interrupt Service Routine(ISR) is completed.
- This can be done by using interrupt-Disable as first instruction and interrupt-Enable as the last instruction.

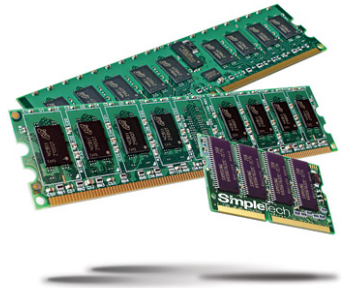


Enabling and Disabling Interrupts Cont…

The second option is processor automatically disable interrupts before starting the execution of the ISR.

The status register PS stored in the stack with PC value.

The processor set this register bit 1 when the interrupt accept and when a return instruction is executed, the contents of the PS are cleared (0)and stored in the stack again.



Handling Multiple Devices

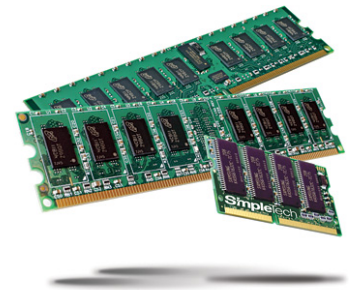
When the number of devices initiating interrupts.

For example, device X may request an interrupt while an interrupt caused by device Y is being serviced.

Hence all the device using the common interrupt line.

Additional information require to identify the device that activated the request.

When the two devices activated the line at the same time, we must break up the tie and chose one the device request among two. Some scheme should be used by the processor.



Handling Multiple Devices Contd…

1. Polling Scheme

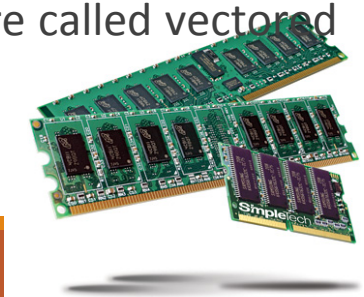
The device that raises the interrupt will set one of the bit (IRQ) in status register to the processor will poll the devices to find which raised an interrupt first.

Disadvantage:

Time spend in interrogating the IRQ bits of the devices that may not be requesting any service.

2. Vectored Interrupts

To reduce the time involved in the polling scheme, a device requesting an interrupt may identify itself directly to the processor. A device can send a special code to the processor over the bus. The code is used to identify the device. If the interrupt produces a CALL to a predetermined memory location, which is the starting address of ISR, then that address is called vectored address and such interrupts are called vectored interrupts.



Handling Multiple Devices Contd…

3. Interrupt priority

When a interrupt arrives from one (or) more devices simultaneously, the processor has to decide which request should be serviced first. • The processor takes this decision with the help of interrupt priorities. • The processor accepts interrupt request having highest priority. • Each request assign a different priority level. • The request received from the interrupt request line are sent to a priority arbitration circuit in the processor. • The request is accepted only if it has a higher priority level than that currently assigned to the processor.

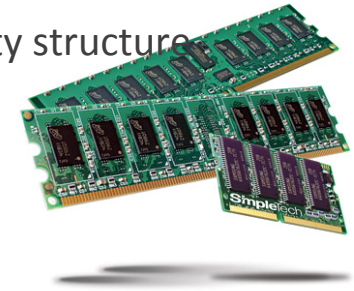
4. Controlling device request

The processor allow only the input / output devices requested(interrupt), that are being used by a given program. • Other devices should not be allowed to generate interrupt requests even though they are ready to transfer the data. • Hence, we need a mechanism in the interface circuits of individual devices to control whether the device is allowed to generate an interrupt request. •

Two mechanism for control request:

1. One is at the device end- interrupt enable bit in the control register(IRQ).

2. Processor end- enable bit in the program status register(PS) or priority structure determine whether a given interrupt request will be accepted.



Thank You