Nadar Saraswathi College of Engineering & Technology

NSCET E-LEARNING PRESENTATION

LISTEN … LEARN… LEAD…

Computer Science Engineering

II YEAR / IV SEMESTER

CS8491-Computer Architecture

B.Sri Chithra Devi,M.E
Assistant Professor
Nadar Saraswathi College of Engineering & Technology,
Vadapudupatti, Annanji (po), Theni – 625531.

UNIT-V

MEMORY HIERARCHY & I/O SYSTEMS

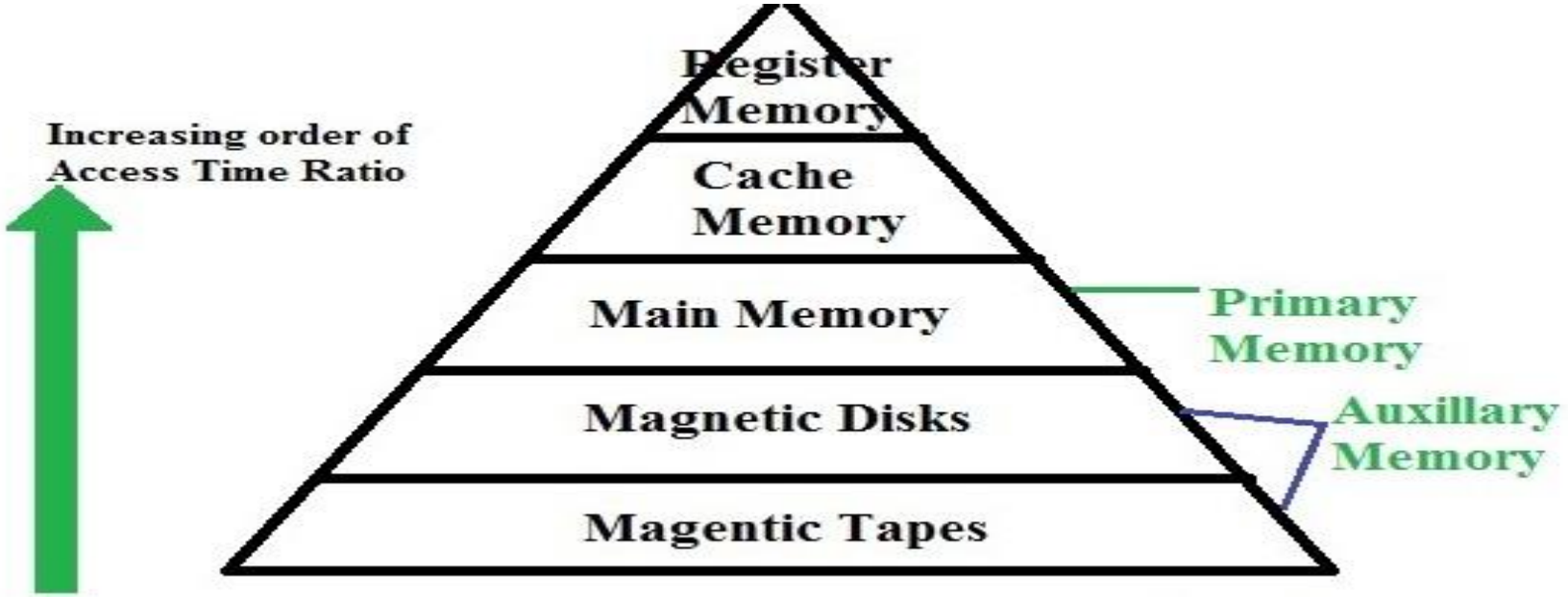
MEMORY HIERARCHY & I/O SYSTEMS

- Memory Hierarchy.
- memory technologies.
- cache memory .
- measuring and improving cache performance
- virtual memory.
- TLB‘s ,Accessing I/O Devices .
- Interrupts
- Direct Memory Access
- Bus structure
- Bus operation
- Arbitration
- Interface circuits ,USB.

Lecture1-Memory Hierarchy

- The **memory hierarchy design** in a computer system mainly includes different storage devices.
- Most of the computers were inbuilt with extra storage to run more powerfully beyond the main memory capacity.
- The following **memory hierarchy diagram** is a hierarchical pyramid for computer memory.
- The designing of the memory hierarchy is divided into two types such as primary (Internal) memory and secondary (External) memory

Lecture1-Memory Hierarchy



Lecture1-Memory Hierarchy

Primary Memory

- The primary memory is also known as internal memory, and this is accessible by the processor straightly. This memory includes main, cache, as well as CPU registers.

Secondary Memory

- The secondary memory is also known as external memory, and this is accessible by the processor through an input/output module. This memory includes an optical disk, magnetic disk, and magnetic tape.

Lecture1-Memory Hierarchy

Characteristics of Memory Hierarchy

The memory hierarchy characteristics mainly include the following.

- Performance
- Ability
- Access Time
- Cost per bit

Lecture1-Memory Hierarchy

Memory Hierarchy Design

The memory hierarchy in computers mainly includes the following.

- Registers
- Cache Memory
- Main Memory
- Magnetic Disks
- Magnetic Tape

Lecture1-Memory Hierarchy

Advantages of Memory Hierarchy

The need for a memory hierarchy includes the following.

- Memory distributing is simple and economical
- Removes external destruction
- Data can be spread all over
- Permits demand paging & pre-paging
- Swapping will be more proficient

Lecture 2-Memory technologies

- The term memory hierarchy is used in computer architecture when discussing performance issues in computer architectural design, algorithm predictions, and the lower level programming constructs such as involving locality of reference.
- A "memory hierarchy" in computer storage distinguishes each level in the "hierarchy" by response time.
- Since response time, complexity, and capacity are related, the levels may also be distinguished by the controlling technology.

Lecture 2-Memory technologies

- Memory is categorized into volatile and non-volatile memories, with the former requiring constant power ON of the system to maintain data storage.
- Furthermore, a typical computer system provides a hierarchy of different times of memories for data storage.

Lecture 2-Memory technologies

- There are four major storage levels
- Internal – Processor registers and cache.
- Main – the system RAM and controller cards.
- On-line mass storage – Secondary storage.
- Off-line bulk storage – Tertiary and Off-line storage.

Lecture 2-Memory technologies

Internal or cache:

- Cache is the fastest accessible memory of a computer system.
- Its access speed is in the order of a few nanoseconds.
- It is volatile and expensive, so the typical cache size is in the order of megabytes.

Lecture 2-Memory technologies

Main memory :

- Main memory is arguably the most used memory.
- The main memory is reasonably fast, with access speed around 100 nanoseconds.
- It also offers larger capacity at a lower cost.
- Typical main memory is in the order of 10 GB.
- However, the main memory is volatile.

Lecture 2-Memory technologies

RAM:

- Random Access Memories are volatile in nature.
- As soon as the computer is switched off, the contents of memory are also lost.

ROM:

- Read only memories are non volatile in nature.
- The storage is permanent, but it is read only memory.
- We cannot store new information in ROM.

Lecture 2-Memory technologies

Several types of ROM

PROM: Programmable Read Only Memory

it can be programmed once as per user requirements.

EPROM: Erasable Programmable Read Only Memory

- the contents of the memory can be erased and store new data into the memory.
- In this case, we have to erase whole information.

Lecture 2-Memory technologies

EEPROM: Electrically Erasable Programmable Read Only Memory

- in this type of memory the contents of a particular location can be changed without effecting the contents of other location.

Online mass storage (or) secondary storage (tb):

- Secondary storage refers to non-volatile data storage units that are external to the computer system. Hard drives and solid state drives are examples of secondary storage.
- They offer very large storage capacity in the order of terabytes at very low cost.

Lecture 2-Memory technologies

- Therefore, database servers typically have an array of secondary storage devices with data stored distributedly and redundantly across these devices.
- Modern hard drives have access speed in the order of a few milliseconds.

Lecture 2-Memory technologies

Memory technologies

- There are four primary technologies used today in memory hierarchies. Main memory is implemented from DRAM (dynamic random access memory), while levels closer to the processor (caches) use SRAM (static random access memory).
- DRAM is less costly per bit than SRAM, although it is substantially slower.
- The price difference arises because DRAM uses significantly less area per bit of memory, and DRAMs thus have larger capacity for the same amount of silicon

Lecture 2-Memory technologies

- The third technology is flash memory.
- This non-volatile memory is the secondary memory in Personal Mobile Devices.
- The fourth technology, used to implement the largest and slowest level in the hierarchy in servers, is magnetic disk.

SRAM Technology

- SRAMs are simply integrated circuits that are memory arrays with (usually) a single access port that can provide either a read or a write.

Lecture 2-Memory technologies

- SRAMs have a fixed access time to any datum, though the read and write access times may differ.
- SRAMs don’t need to refresh and so the access time is very close to the cycle time.
- SRAMs typically use six to eight transistors per bit to prevent the information from being disturbed when read.
- SRAM needs only minimal power to retain the charge in standby mode.
- In the past, most PCs and server systems used separate SRAM chips for either their primary, secondary, or even tertiary caches.

Lecture 2-Memory technologies

DRAM Technology

- In a SRAM, as long as power is applied, the value can be kept indefinitely. In a dynamic RAM (DRAM), the value kept in a cell is stored as a charge in a capacitor.
- A single transistor is then used to access this stored charge, either to read the value or to overwrite the charge stored there.
- DRAMs use only a single transistor per bit of storage, they are much denser and cheaper per bit than SRAM.

Lecture 2-Memory technologies

- As DRAMs store the charge on a capacitor, it cannot be kept indefinitely and must periodically be refreshed. That is why this memory structure is called dynamic, as opposed to the static storage in an SRAM cell.
- To refresh the cell, we merely read its contents and write it back. The charge can be kept for several milliseconds.
- If every bit had to be read out of the DRAM and then written back individually, we would constantly be refreshing the DRAM, leaving no time for accessing it.

Lecture 2-Memory technologies

- DRAMs use a two-level decoding structure, and this allows us to refresh an entire row (which shares a word line) with a read cycle followed immediately by a write cycle.

Lecture 2-Memory technologies

Flash Memory

- Flash memory is a type of electrically erasable programmable read-only memory (EEPROM).
- Flash memory stores information in an array of memory cells made from floating-gate transistors.
- The floating gate may be conductive (typically polysilicon in most kinds of flash memory) or non-conductive (as in SONOS flash memory).
- In traditional single-level cell (SLC) devices, each cell stores only one bit of information.

Lecture 2-Memory technologies

- Some newer flash memory, known as multi-level cell (MLC) devices, including triple-level cell (TLC) devices, can store more than one bit per cell by choosing between multiple levels of electrical charge to apply to the floating gates of its cells.
- Embedded devices also need non-volatile storage, but premiums placed on space and power normally lead to the use of Flash memory instead of magnetic recording.
- Flash memory is also used as a rewritable ROM in embedded systems, typically to allow software to be upgraded without having to replace chips.

Lecture 2-Memory technologies

Disk Memory or Magnetic Memory

- Magnetic hard disk consists of a collection of platters, which rotate on a spindle at 5400 to 15,000 revolutions per minute.
- The metal platters are covered with magnetic recording material on both sides, similar to the material found on a cassette or videotape.
- Magnetic hard disk consists of a collection of platters, which rotate on a spindle at 5400 to 15,000 revolutions per minute.
- The metal platters are covered with magnetic recording material on both sides, similar to the material found on a cassette or videotape.

Lecture 2-Memory technologies

- To read and write information on a hard disk, a movable arm containing a small electromagnetic coil called a read-write head is located just above each surface.
- The entire drive is permanently sealed to control the environment inside the drive, which, in turn, allows the disk heads to be much closer to the drive surface.
- Each disk surface is divided into concentric circles, called tracks. There are typically tens of thousands of tracks per surface.

Lecture 2-Memory technologies

- Each track is in turn divided into sectors that contain the information; each track may have thousands of sectors.
- Sectors are typically 512 to 4096 bytes in size.
- The sequence recorded on the magnetic media is a sector number, a gap, the information for that sector in including error correction code a gap, the sector number of the next sector, and so on.

Lecture 3-Cache Memory

- 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.

Lecture 3-Cache Memory

- Cache memory is based on the property of computer programs known as “locality of reference”.
- Prefetch the data into cache before the processor needs it. Needs to predict processor future access requirement. [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.

Lecture 3-Cache Memory

- 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.

Lecture 3-Cache Memory

- Use of a cache memory
- 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”.

Lecture 3-Cache Memory

- 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.

Lecture 3-Cache Memory

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- 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 the block is replaced - This is write-back or copy-back protocol.

Lecture 3-Cache Memory

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.

Lecture 3-Cache Memory

- 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 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.

Lecture 3-Cache Memory

Mapping Functions

- The process of moving the data from main memory to cache memory is known as mapping. 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.

Lecture 3-Cache Memory

Three mapping functions:

- Direct mapping.
- Associative mapping.
- Set-associative mapping.

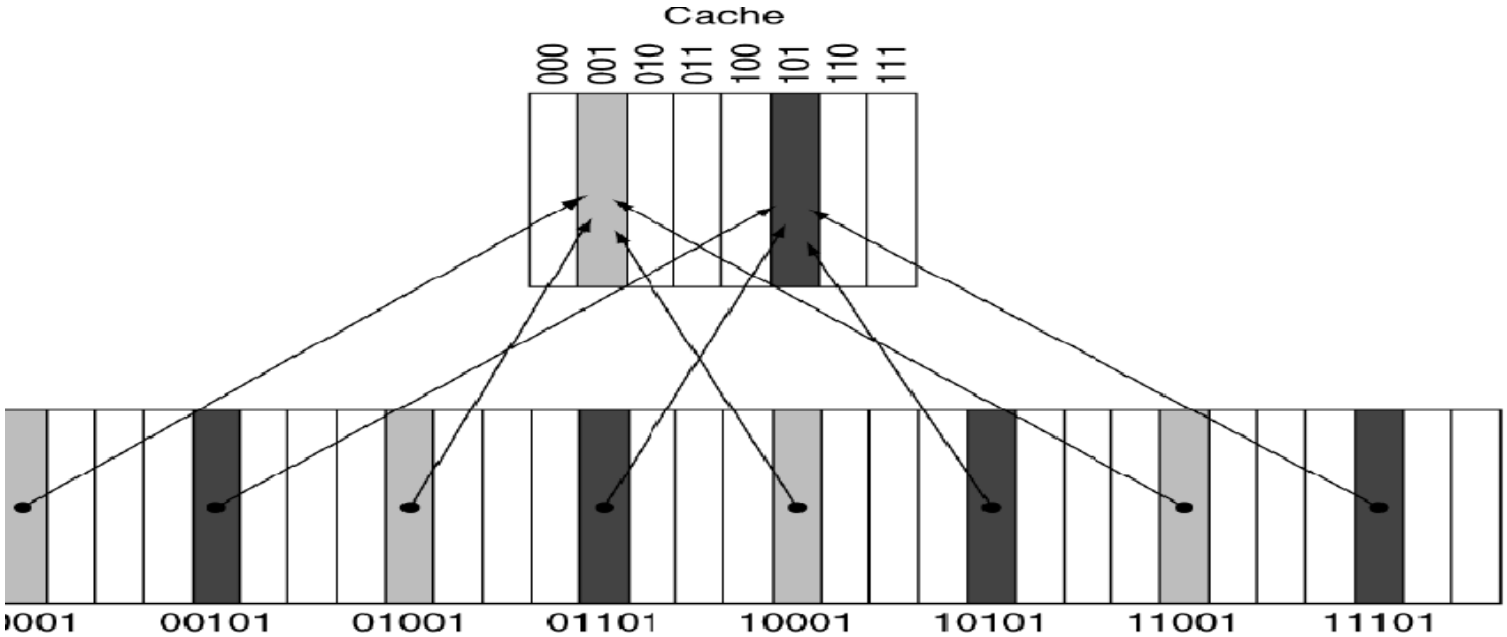
Direct Mapping

- Block j of the main memory maps to j modulo 128 of the cache. 0 maps to 0, 129 maps to Block 1, 129, 257 are stored in cache block 1 & so on.
- More than one memory block is mapped onto the same position in the cache.

Lecture 3-Cache Memory

- 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.

Lecture 3-Cache Memory



Lecture 3-Cache Memory

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.

Lecture 3-Cache Memory

Associative mapping

- 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.

Lecture 3-Cache Memory

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.

Lecture 3-Cache Memory

- 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.
- The number of blocks per set is decided according to the requirement of the application used.

Lecture 3-Cache Memory

- Cache that has k blocks per set is referred to as a k-way set associative cache.
- A control bit called valid bit must be provided for each block.

Advantage:

- less expensive.

Replacement Algorithms

- Difficult to determine which blocks to kick out.
- Least Recently Used (LRU) block.

Lecture 3-Cache Memory

- The cache controller tracks references to all blocks as computation proceeds.
- Increase / clear track counters when a hit/miss occurs.
- For Associative & Set-Associative Cache.

First In First Out (FIFO)

- Least Recently Used (LRU)
- Distinguish an Empty location from a Full one
- Valid Bit

Lecture 4-Measuring and Improving cache performance

- 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.

Ways to improve the performance:

- Memory interleaving.
- Improving hit ratio and reducing miss penalty.
- Having caches on the processor chip.
- Write buffer.
- Prefetching.
- Look-up free cache

Lecture 4-Measuring and Improving cache performance

Memory 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.

Lecture 4-Measuring and Improving cache performance

Methods of address layouts// Methods to achieve interleaving

- Successful access to data in a cache is called hit.
- The number of hits stated as a fraction of all attempted accesses is called the hit rate.
- Miss rate is the number of misses stated as a fraction of attempted accesses.
- The extra time needed to bring the desired information into the cache is called the miss penalty.
- Hit rate can be improved by increasing block size, while keeping cache size constant

Lecture 4-Measuring and Improving cache performance

Methods of address layouts// Methods to achieve interleaving

- 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 [one for data and another for instruction are normally used.

Lecture 4-Measuring and Improving cache performance

- Average access time in a system with 2 levels of caches is

  T ave = h1c1+(1-h1)h2c2+(1-h1)(1-h2)M
- Combined cache: increases hit ratio.
- Separate cache: increases parallelism.

Write buffer

- Write-through
- Write-back:
- Prefetching
- Lockup-Free Cache

Lecture 4-Measuring and Improving cache performance

The steps to be taken on an instruction cache miss

- Send the original PC value (current PC – 4) to the memory.
- Instruct main memory to perform a read and wait for the memory to complete its access.
- Write the cache entry, putting the data from memory in the data portion of the entry,writing the upper bits of the address (from the ALU) into the tag field, and turning the valid bit on.
- Restart the instruction execution at the first step, which will refetch the instruction, this time finding it in the cache.

Lecture 5-Virtual Memory

The main memory can act as a “cache” for the secondary storage, this technique *is* called virtual memory.

Need for Virtual Memory

- To allow efficient and safe sharing of memory among multiple programs.
- Virtual memory role is to protect and ensuring that a program can only read and write the portions of main memory that have been assigned to it.Virtual memory implements the translation of a program’s address space to physical addresses. This translation process enforces protection of a program’s address space

Lecture 5-Virtual Memory

User program

- Virtual memory allow a single user program to exceed the size of primary memory.
- Programmers divided programs into pieces and then identified the pieces that were mutually exclusive.
- User program control ensures that program never access overlays that are not loaded and these overlays never exceeds the allocated size of the memory.

Lecture 5-Virtual Memory

Page, Page Fault, Address Translation

- A virtual memory block is called a *page, and a virtual memory miss* is called a ***page fault.***
- The processor produces a **virtual address, which is translated by a combination of** hardware and soft ware to a *physical address,* which inturn *can be used to access main* memory. This process is called *address mapping or* ***address translation.***

Lecture 5-Virtual Memory

To Design a virtual memory system

- Pages should be large enough to reduce high access time. Sizes from 4 KiB to 16 KiB are typical today (depends on computer types).
- The primary technique used here is to allow fully associative placement of pages in memory.
- Page faults can be handled in software because the overhead will be small compared to the disk access time. (it reduces miss penalty)
- Write-through will not work for virtual memory, since writes take too long. Instead, virtual memory systems use write-back.

Lecture 6-TLB

- Page tables are stored in main memory, Memory access by a program can take at least twice as long.
- One memory access to obtain the physical Address
- Second access to get the data.
- To improve performance, rely on locality of reference i.e., when a virtual page is translated, it will be referenced again(due to temporal and spatial locality).
- So, special address translation cache is used and is traditionally referred to as a **translation-lookaside buffer (TLB) or translation cache**

Lecture 6-TLB

- TLB is accessed instead of the page table on every reference, the TLB will need to include other status bits, such as the dirty and the reference bits
- When a miss in the TLB occurs, we must determine whether it is a page fault or merely TLB miss.
- If the page exists in memory, then the TLB miss indicates only that the translation is missing, so the processor can handle the TLB miss by loading the translation from the page table into the TLB.
- If the page is not present in memory, then the TLB miss indicates a true page fault. **translation-lookaside buffer (TLB) or translation cache**

Lecture 6-TLB

Details about TLB

1. TLB size: 16–512 entries
2. Block size: 1–2 page table entries (typically 4–8 bytes each)
3. Hit time: 0.5–1 clock cycle
4. Miss penalty: 10–100 clock
5. cycles Miss rate: 0.01%–1%

To obtain lower miss rate

- TLB uses small, fully associative mapping (lower miss rate).
- Handling a TLB miss or a page fault requires using the exception mechanism to interrupt the active process, transferring control to the operating system, and later resuming execution of the interrupted process.

Lecture 6-TLB

Details about TLB

1. TLB size: 16–512 entries
2. Block size: 1–2 page table entries (typically 4–8 bytes each)
3. Hit time: 0.5–1 clock cycle
4. Miss penalty: 10–100 clock
5. cycles Miss rate: 0.01%–1%

To obtain lower miss rate

- TLB uses small, fully associative mapping (lower miss rate).
- Handling a TLB miss or a page fault requires using the exception mechanism to interrupt the active process, transferring control to the operating system, and later resuming execution of the interrupted process.

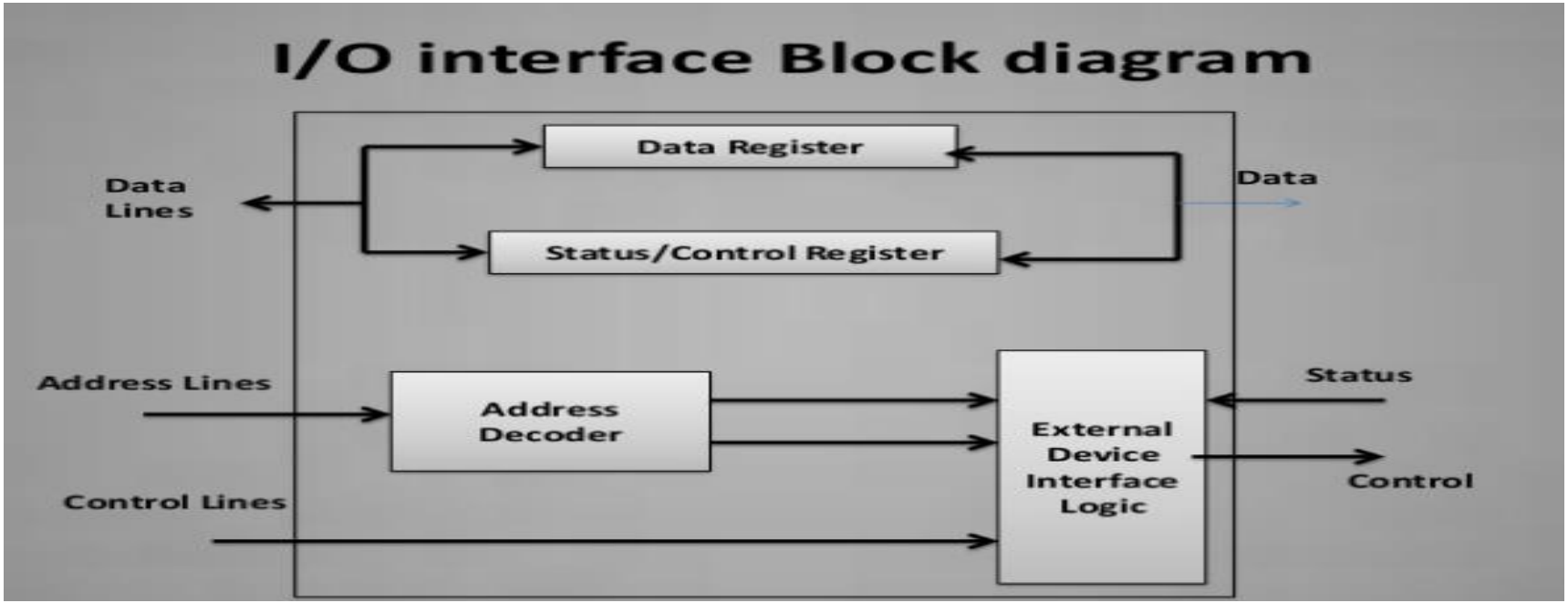
Lecture 8-Accessing I/O Devices

- I/O devices accessed through I/o interface.

Requirements of Input Interface

- CPU communication.
- Device Communication.
- Data Buffering.
- Control and timing.
- Error detection.

Lecture 8-Accessing I/O Devices



Lecture 8-Accessing I/O Devices

I/O Devices can be interface in to a computer in two ways.They are

Memory Mapped I/O:

- No need of special i/o instructions.
- Memory related instructions are used for I/O related operations

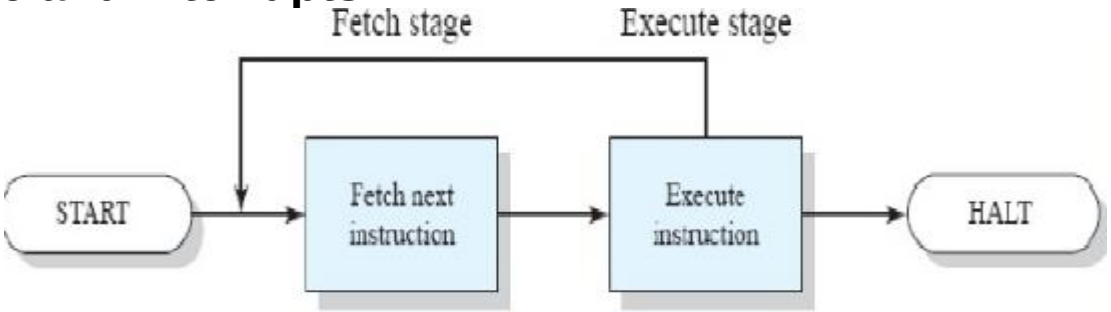
I/o Mapped I/o

- Memory related instructions do not work here.
- Processor use these mode only for I/O read,I/O write.

Lecture 9-Interrupt

- A suspension of a process such as 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.

Instruction cycle and interrupts



Lecture 9-Interrupt

Classes of Interrupt

- Program
- Timer
- I/o
- Hardware Failure

Handling of interrupt

- Most modern general purpose microprocessors handle the interrupts the same way. When a hardware interrupt occurs the CPU stops executing the instructions that it was executing and jumps to a location in memory that either contains the interrupt handling code or an instruction branching to the interrupt handling code.

Lecture 9-Interrupt

I/O communication Techniques

Three techniques are possible for I/O operations

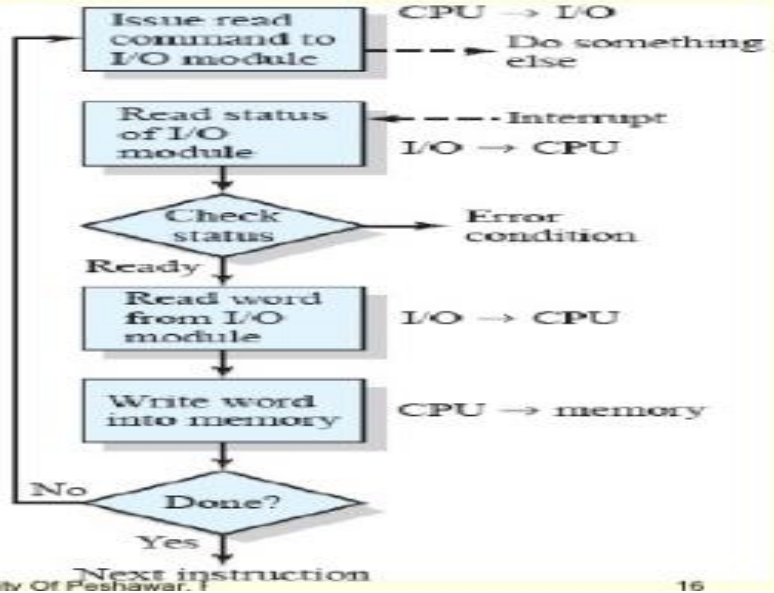
- Programmed I/o
- Interrupt driven I/o
- Direct Memory Access(DMA)

Lecture 9-Interrupt

Clip slide

Interrupt-Driven I/O

- Similar to direct I/O but processor not required to poll device.
- I/O module will interrupt CPU for data exchange when ready

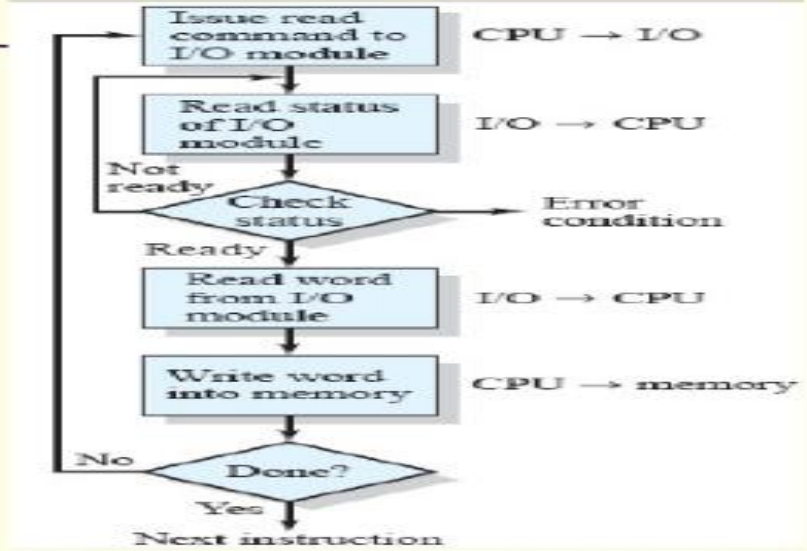


Interrupts By: Irfanullah University Of Peshawar. I 16

Lecture 9-Interrupt

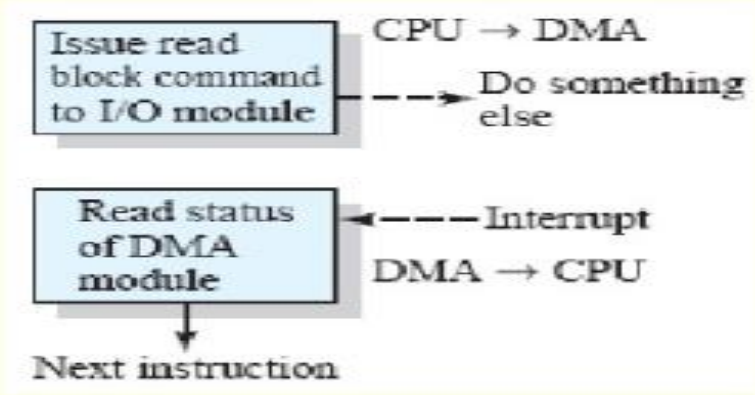
Programmed I/O

- CPU while executing a program encounters an I/O instruction
- CPU issues I/O command to I/O module
- I/O module performs the requested action & set status registers
- CPU is responsible to check status registers periodically to see if I/O operation is complete. **SO**
- No Interrupt to alert the processor



Lecture 10-DMA

- I/O exchanges occur directly with memory
  - Requires DMA module on system bus
  - Capable of mimicking CPU and taking over control of system from CPU
  - DMA will use bus when
    - Processor does not require it OR
    - Must force processor to suspend operation temporarily– called cycle stealing
- An interrupt is sent when the task is complete
- The processor is only involved at the beginning and end of the transfer



Lecture 11-Bus structure and operation

- A bus is a subsystem that is used to connect computer components and transfer data between them. For example, an internal bus connects computer internals to the motherboard.
- A bus may be parallel or serial. Parallel buses transmit data across multiple wires. Serial buses transmit data in bit-serial format.

Lecture 11-Bus structure and operation

- Parallel bus standards include advanced technology attachment (ATA) or small computer system interface (SCSI) for printer or hard drive devices. Serial bus standards include universal serial bus (USB), FireWire or serial ATA with a daisy-chain topology or hub design for devices, keyboards or modem devices.

Lecture 11-Bus structure and operation

Computer Bus types

- System bus
- Internal bus
- External Bus
- Expansion bus

Lecture 11-Bus structure and operation

Bus speeds

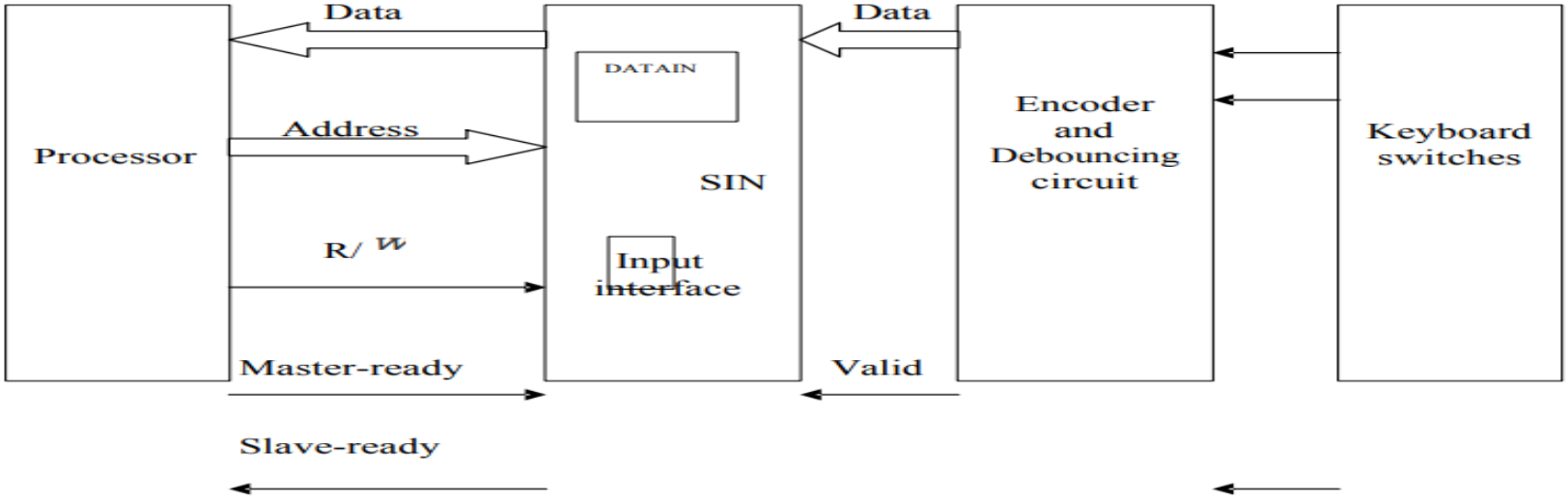
- A computer or device's **bus speed** is measured in MHz, e.g., an FSB may operate at a frequency of 100 MHz. The throughput of a bus is measured in bits per second or megabytes per second.

Lecture 12-Interface Circuits

Parallel port

- The hardware components needed for connecting a keyboard to a processor.
- A typical keyboard consists of mechanical switches that are normally open. When a key is pressed, its switch closes and establishes a path for an electrical signal.
- This signal is detected by an encoder circuit that generates the ASCII code for the corresponding character.

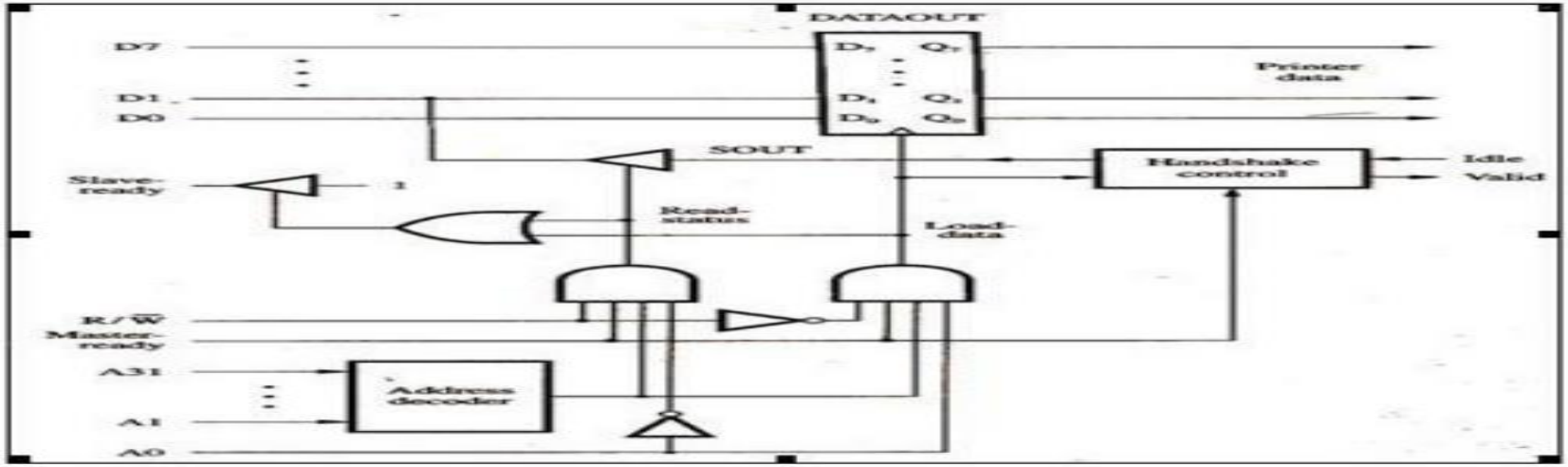
Lecture 12-Interface Circuits



Lecture 12-Interface Circuits

- The printer operates under control of the handshake signals Valid and Idle in a manner similar to the handshake used on the bus with the Master-ready and Slave-ready signals.
- When it is ready to accept a character, the printer asserts its Idle signal. The interface circuit can then place a new character on the data lines and activate the Valid signal.
- In response, the printer starts printing the new character and negates the Idle signal, which in turn causes the interface to deactivate the Valid signal

Lecture 12-Output Interface Circuits



Lecture 13-USB

- A Universal Serial Bus (**USB**) is a common interface that enables communication between devices and a host controller such as a personal **computer** (PC).
- It connects peripheral devices such as digital cameras, mice, keyboards, printers, scanners, media devices, external hard drives and flash drives

Lecture 13-USB

USB Types

- The different versions of USB cables, like USB 2.0 and USB 3.0, are concerned with the functionality and speed of the USB cable; whereas, the type of USB cable (like USB Type A, USB Type B) essentially refers to the physical design of the plugs and ports.

Lecture 13-USB

Port and Receptor

- The slot where one end of the USB cable is attached to the computer (the host) at the back or front of the CPU cabinet is called the port. The electronic device that you wish to charge or transfer data to (say, your smartphone or tablet) is called the receptor.