SHEME OF VALUATION

(Scoring Indicators)

Revision: 2015 **Course Code: 3131**

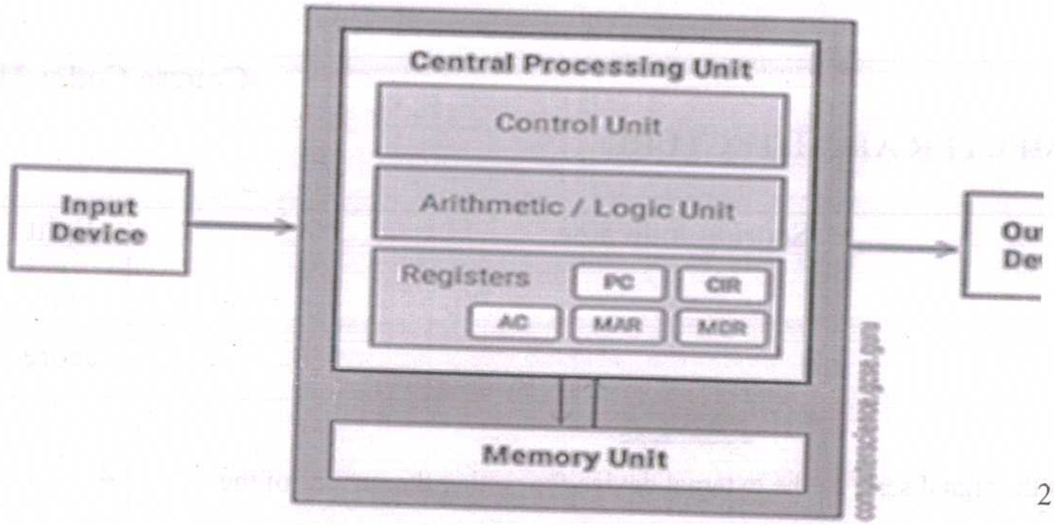
Course Title: COMPUTER ARCHITECTURE

Qst. No	Scoring Indicator	Split up score	Sub Total	Total
	PART A			
I(i)	An interrupt is the signal send by the external device for getting the service of the processor.	2	2	

I(ii)
I(iii)
	Control and timing , Processor communication, Device communication ,data buffering, error detection(any two)	2
		2
			2
			2	

I(iv)	The organization of data on platter surface in a concentric set of rings,called tracks.			10
I(v)	Data are transferred to and from the disk in sectors There are typically hundreds of sectors per track	2 2	2 2	
	instruction pipelining is a technique for implementing instruction level parallelism within a single processor			
	a microinstruction program that controls the functions of a central processing unit or peripheral controller of a computer.			
	PART B			
II(i)		2 fig+4 expl	6	

The Von Neumann Machine



2*3 6

Functional Units :A computer consists of 5 main parts. □ Input □ Memory □ Arithmetic and logic □ Output □ Control Units

This figure 2.3 reveals that both the control unit and the ALU contain storage locations ,called registers, defined as follows:

- Memory buffer register (MBR): Contains a word to be stored in memory or sent to the I/O unit, or is used to receive a word from memory or from the I/O unit.
- Memory address register (MAR): Specifies the address in memory of the word to be written from or read into the MBR.
- Instruction register (IR): Contains the 8-bit opcode instruction being executed.
- Instruction buffer register (IBR): Employed to hold temporarily the right-hand instruction from a word in memory.
- Program counter (PC): Contains the address of the next instruction-pair to be fetched from memory.
- Accumulator (AC) and multiplier quotient (MQ): Employed to hold temporarily operands and results of ALU operations.

6 6

30

II(2) Advanced DRAM types are SDRAM, DDR-DRAM, and RDRAM

SDRAM Unlike the traditional DRAM, which is asynchronous, the SDRAM exchanges data with the processor synchronized to an external clock signal and running at the full speed of the processor/memory bus without imposing wait states. With synchronous access, the DRAM moves data in and out under control of the system clock. The processor or other master issues the instruction and address information,

3 fig+3 expl 6

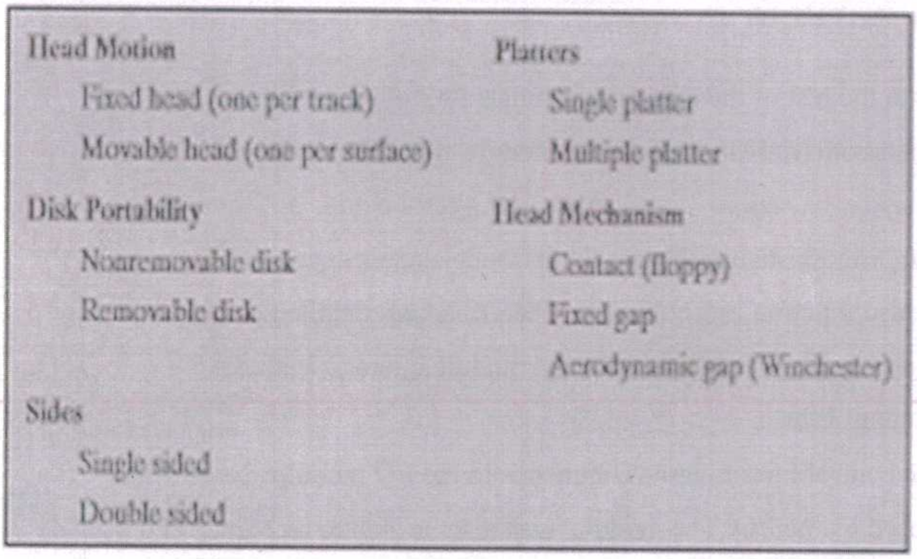
which is latched by the DRAM.

Rambus DRAM RDRAM chips are vertical packages, with all pins on one side. The chip exchanges data with the processor over 28 wires no more than 12 centimeters long. The bus can address up to 320 RDRAM chips and is rated at 1.6 Gbps.

The RDRAM configuration consists of a controller and a number of RDRAM modules connected via a common bus. The controller is at one end of the configuration, and the far end of the bus is a parallel termination of the bus lines. The bus includes 18 data lines (16 actual data,two parity) cycling at twice the clock rate; that is, 1 bit is sent at the leading and following edge of each clock signal. This results in a signal rate on each data line of 800 Mbps.

3

Table 6.1 Physical Characteristics of Disk Systems



Head Motion	Platters
Fixed head (one per track)	Single platter
Movable head (one per surface)	Multiple platter
Disk Portability	Head Mechanism
Nonremovable disk	Contact (floppy)
Removable disk	Fixed gap
	Aerodynamic gap (Winchester)
Sides	
Single sided	
Double sided	

6 6

6

6

II(v)

4.

6

3fig+3 expl

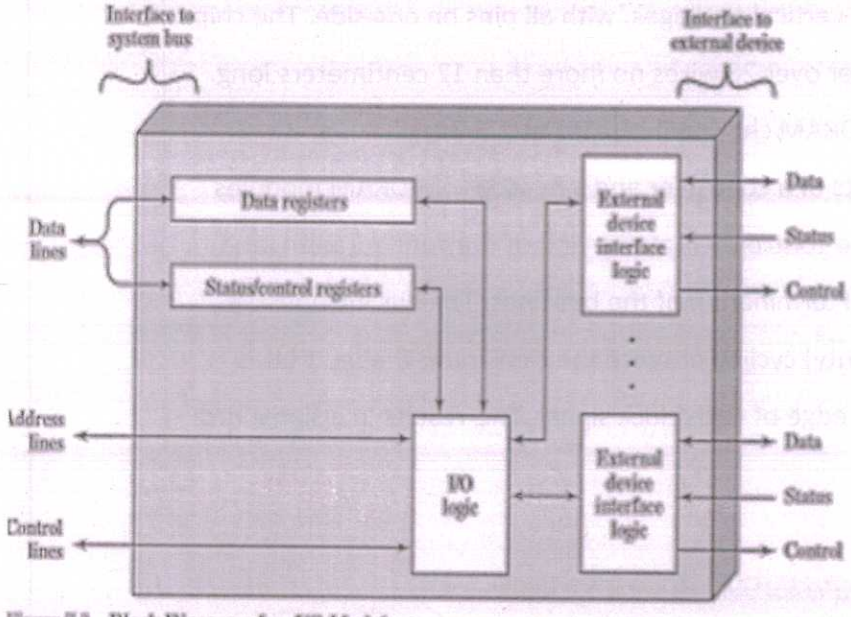


Figure 7.3 Block Diagram of an I/O Module

The module connects to the rest of the computer through a set of signal lines (e.g., system bus lines).Data transferred to and from the module are buffered in one or more data registers

There may also be one or more status registers that provide current status information

A status register may also function as a control register, to accept detailed control information from the processor .The logic within the module interacts with the processor via a set of control lines.

The processor uses the control lines to issue commands to the I/O module. Some of the control lines may be used by the I/O .The module must also be able to recognize and generate addresses associated with the devices it controls .Each I/O module has a unique address or, if it controls more than one external device, a unique set of addresses .Finally, the I/O module contains logic specific to the interface with each device that it controls. An I/O module functions to allow the processor to view a wide range of devices in a simple-minded way.

5

6

3fig+3expl

User-Visible Registers

A user-visible register is one that may be referenced by means of the machine language that the processor executes. We can characterize these in the following categories:

- General purpose
- Data
- Address
- Condition codes

General-purpose registers can be assigned to a variety of functions by the programmer,

II(vii)	any general-purpose register can contain the operand for any opcode. Data registers may be used only to hold data and cannot be employed in the calculation of an operand address Address registers may themselves be somewhat general purpose ,or they may be devoted to a particular addressing mode. .User visible registers ,in turn, exchange data with the MBR. Many processor designs include a register or set of registers, often known as the program status word (PSW),that contain status information. The PSW typically contains condition codes plus other status information. Common fields or flags include the following: • Sign: Contains the sign bit of the result of the last arithmetic operation. • Zero: Set when the result is 0. • Carry: Set if an operation resulted in a carry (addition) into or borrow (subtraction) out of a high-order bit. • Equal: Set if a logical compare result is equality. • Overflow: Used to indicate arithmetic overflow. The block diagram of control unit consist of • Clock: This is how the control unit “keeps time. “The control unit causes one micro-operation (or a set of simultaneous micro-operations) to be performed for each clock pulse. • Instruction register: The opcode and addressing mode of the current instruction are used to determine which micro-operations to perform during the execute cycle. • Flags: These are needed by the control unit to determine the status of the processor and the outcome of previous ALU operations. For example, for the increment-and-skip-if-zero (ISZ) instruction, the control unit will increment the PC if the zero flag is set. • Control signals from control bus: The control bus portion of the system bus provides signals to the control			

6

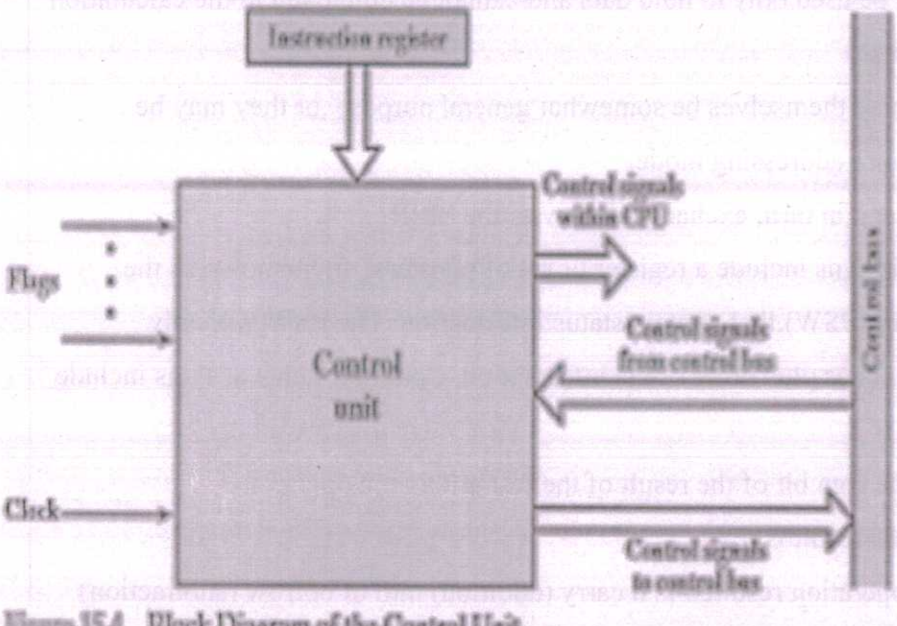

Instruction register
Control signals within CPU
Flags
Control signals from control bus
Control bus
Control unit
Clock
Control signals to control bus

Figure 15.4 Block Diagram of the Control Unit

the operation of a computer, in executing a program, consists of a sequence of instruction cycles, with one machine instruction per cycle. Of course, each instruction cycle is made up of a number of smaller units. is fetch, indirect, execute, and interrupt each cycle is a combination of smaller known as micro-operations. the execution of a program consists of the sequential execution of instructions. Each instruction is executed during an instruction cycle made up of shorter sub cycles (e.g. Fetch, indirect, execute, interrupt).The execution of each sub cycle involves one or more shorter operations, that is, micro-operations. Micro-operations are the functional, or atomic, operations of a processor.

7

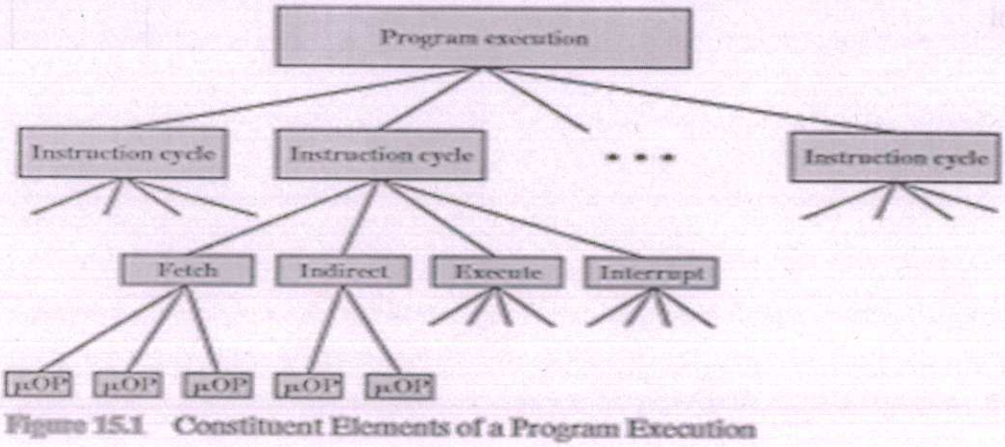
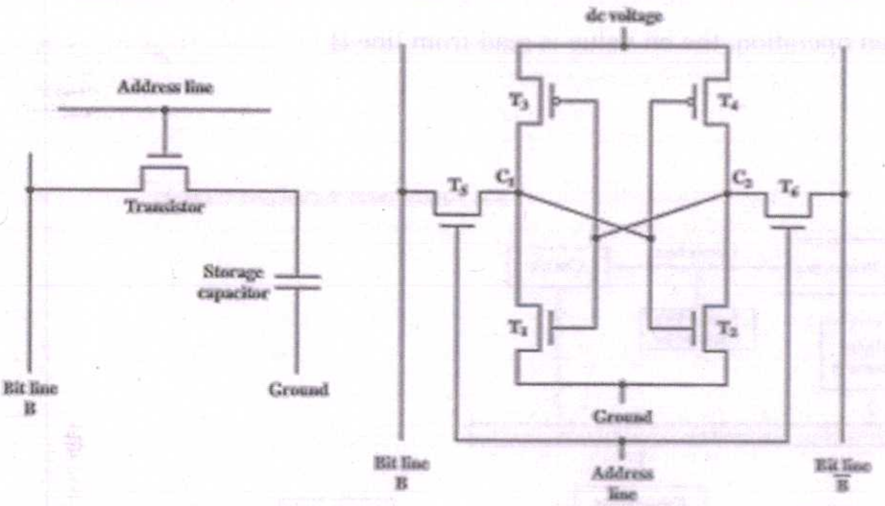
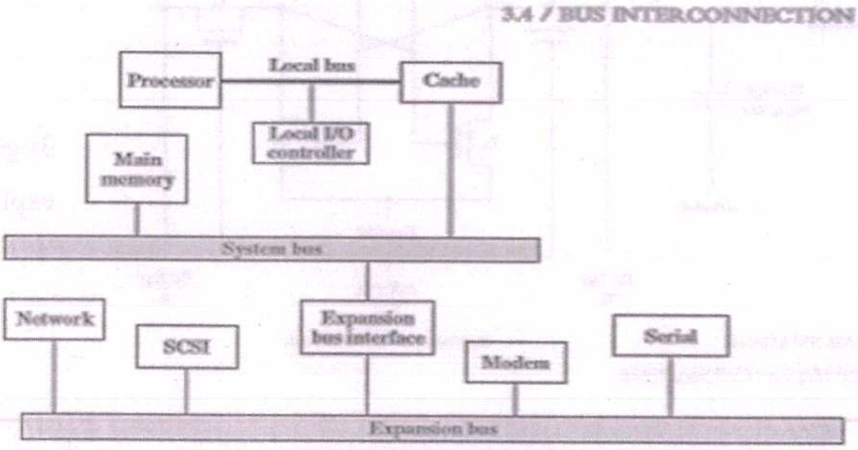


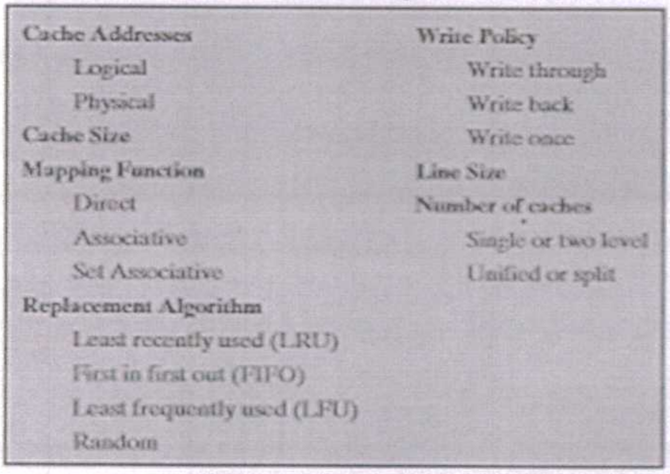
Figure 15.1 Constituent Elements of a Program Execution

III(a)	**PART C** DRAM and SRAM  Address line Transistor Storage capacitor Bit line B Ground dc voltage T3 T4 C1 C2 T5 T6 T1 T2 Ground Bit line B Address line Bit line $\overline{B}$ (a) Dynamic RAM (DRAM) cell (b) Static RAM (SRAM) cell Figure 5.2 Typical Memory Cell Structures A dynamic RAM (DRAM) is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as a binary 1 or 0. Because capacitors have a natural tendency to discharge, dynamic RAMs require periodic charge refreshing to maintain data storage. an individual cell that stores 1 bit. The address line is activated when the bit value from this cell is to be read or written. The transistor acts as a switch that is closed (allowing current to flow) if a voltage is applied to the address line and open (no current flows) if no voltage is present on the address line. For the write operation, a voltage signal is applied to the bit line; a high voltage represents 1, and a low voltage represents 0.A signal is then applied to the address line, allowing a charge to be transferred to the capacitor. For the read operation, when the address line is selected, the transistor turns on and the charge stored on the capacitor is fed out onto a bit line and to a sense amplifier. The sense amplifier compares the capacitor voltage to a reference value and determines if the cell contains a logic 1 or a logic 0.	3fig+5 expl	8	15
III(b	SRAM: Four transistors (T1,T2,T3,T4) are cross connected in an arrangement that	2 fig	7	

)	produces a stable logic state. In logic state 1,point C1 is high and point C2 is low;in this state,T1 and T4 are off and T2 and T3 are on.1 In logic state 0,point C1 is low and point C2 is high;in this state,T1 and T4 are on and T2 and T3 are off.Both states are stable as long as the direct current (dc) voltage is applied.Unlike the DRAM,no refresh is needed to retain data. As in the DRAM, the SRAM address line is used to open or close a switch. The address line controls two transistors (T5 and T6).When a signal is applied to this line, the two transistors are switched on, allowing a read or write operation. For a write operation,the desired bit value is applied to line B,while its complement is applied to line .This forces the four transistors (T1,T 2,T 3,T 4) into the proper state. For a read operation, the bit value is read from line B	5	
IV(a)	 3.4 / BUS INTERCONNECTION Processor Local bus Cache Local I/O controller Main memory System bus Network SCSI Expansion bus interface Modem Serial Expansion bus (a) Traditional bus architecture	4fig+4	8
	more devices attached to the bus,the greater the bus length and hence the greater the propagation delay. The bus may become a bottleneck as the aggregate data transfer demand approaches the capacity of the bus. Accordingly ,most computer systems use multiple buses, generally laid out in a hierarchy. There is a local bus that connects the processor to a cache memory and that may support one or more local devices. The cache memory controller connects the cache not only to this local bus, but to a system bus to which are attached all of the main memory modules .the use of a cache structure insulates the processor from a requirement to access main memory frequently .Hence ,main memory can be moved off of the local bus onto a system bus.In this way,I/O transfers to and from the main memory across the system bus do not interfere with the processor's activity.	3 fig+4 expl	7

IV(b)

Table 4.2 Elements of Cache Design



Cache Addresses	Write Policy
Logical	Write through
Physical	Write back
Cache Size	Write once
Mapping Function	Line Size
Direct	Number of caches
Associative	Single or two level
Set Associative	Unified or split
Replacement Algorithm	
Least recently used (LRU)	
First in first out (FIFO)	
Least frequently used (LFU)	
Random	

8

Cache addresses: When virtual addresses are used, the system designer may choose to place the cache between the processor and the MMU or between the MMU and main memory A logical cache, also known as a virtual cache, stores data using virtual addresses. The processor accesses the cache directly, without going through the MMU. A physical cache stores data using main memory physical addresses. 4*2

DIRECT MAPPING The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line.

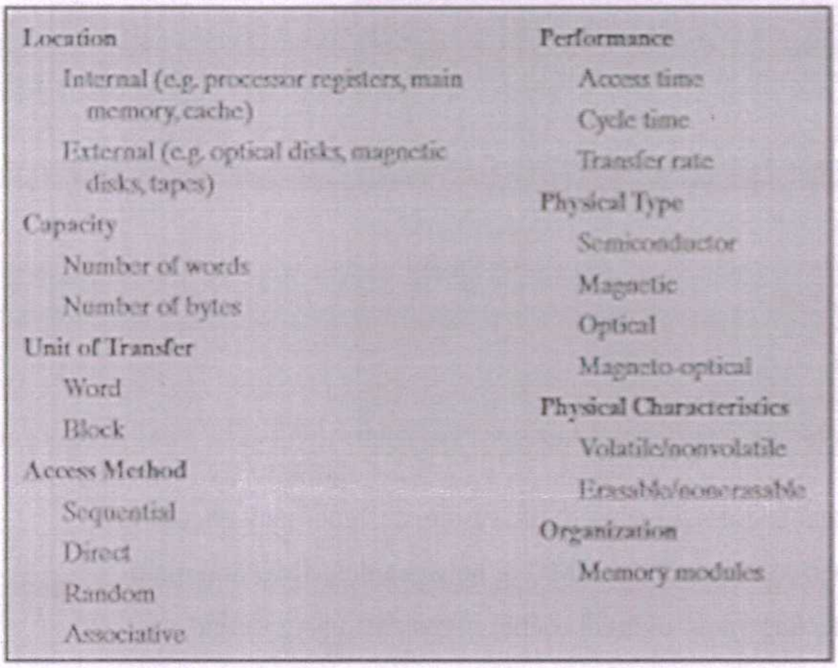
V(a) ASSOCIATIVE MAPPING Associative mapping overcomes the disadvantage of direct mapping by permitting each main memory block to be loaded into any line of the cache

SET-ASSOCIATIVE MAPPING Set-associative mapping is a compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages.

write through-.Using this technique, all write operations are made to main memory as well as to the cache.

An alternative technique, known as write back, minimizes memory writes.With write back,updates are made only in the cache.When an update occurs, a dirty bit,or use bit,associated with the line is set.Then,when a block is replaced,it is written back to main memory

Table 4.1 Key Characteristics of Computer Memory Systems



Location
- Internal (e.g. processor registers, main memory, cache)
- External (e.g. optical disks, magnetic disks, tapes)

Capacity
- Number of words
- Number of bytes

Unit of Transfer
- Word
- Block

Access Method
- Sequential
- Direct
- Random
- Associative

Performance
- Access time
- Cycle time
- Transfer rate

Physical Type
- Semiconductor
- Magnetic
- Optical
- Magneto-optical

Physical Characteristics
- Volatile/nonvolatile
- Erasable/nonerasable

Organization
- Memory modules

1. RAID is a set of physical disk drives viewed by the operating system as a single logical drive.

2. Data are distributed across the physical drives of an array in a scheme known as striping,

3. Redundant disk capacity is used to store parity information, which guarantees data recoverability in case of a disk failure. 7

Raid increases performances, reliability and fault tolerance.

RAID Level 0 2fig+5

it does not include redundancy to improve performance.

few applications, such as some on supercomputers in which performance and capacity are primary concerns and low cost is more important than improved reliability uses this level.

For RAID 0,the user and system data are distributed across all of the disks in the array. This has a notable advantage over the use of a single large disk:

The data are striped across the available disks.

RAID Level 1(mirroring)-require 2n disks

redundancy is achieved.

redundancy is achieved by duplicating all the data.As Figure 6.8b shows,data striping is used,as in RAID 0.But in this case,each logical strip is mapped to two separate physical disks so that every disk in the array has a mirror disk that contains the same data.in RAID 1

V(b) 1. A read request can be serviced by either of the two disks that contains the requested 8 data,

2. A write request requires that both corresponding strips be updated,but this can be done in parallel.

3fig+5

RAID Level 2 -Redundant via Hamming code-require n+m disk

RAID levels 2 and 3 make use of a parallel access technique. In a parallel access array, all member disks participate in the execution of every I/O request. Typically, the spindles of the individual drives are synchronized so that each disk head is in the same position on each disk at any given time.

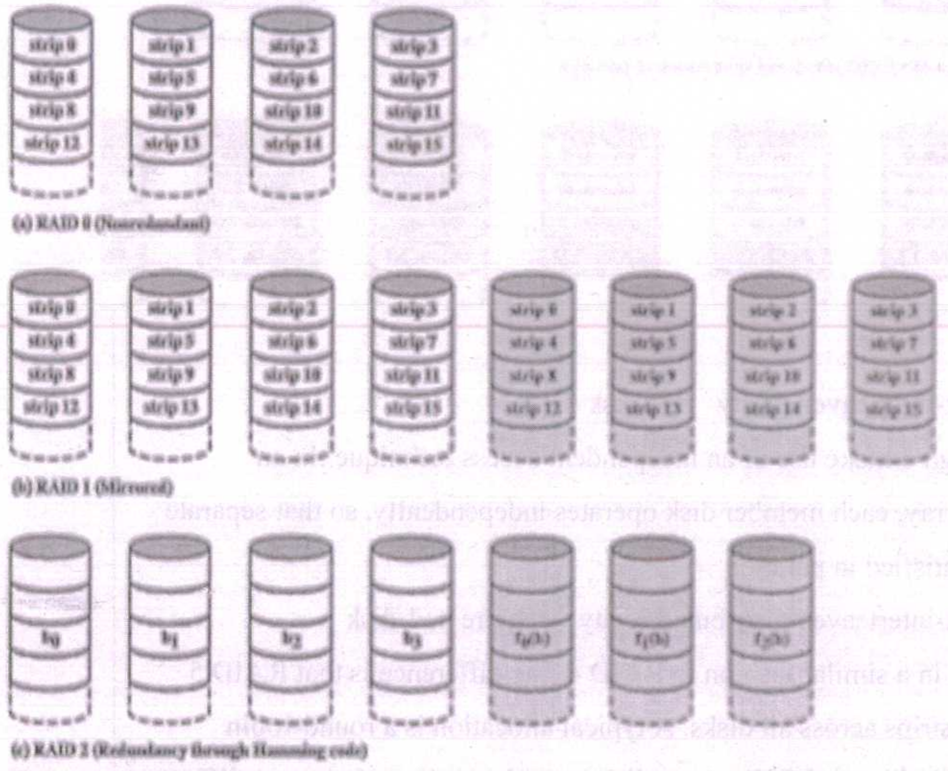
In the case of RAID 2 and 3,the strips are very small, often as small as a single byte or word

RAID Level 3 -Bit-interleaved parity-require n+1 disk

RAID 3 is organized in a similar fashion to RAID 2.The difference is that RAID 3 requires only a single redundant disk, no matter how large the disk array

6.2 / RAID 197

VI(a)



(a) RAID 0 (Nonredundant)

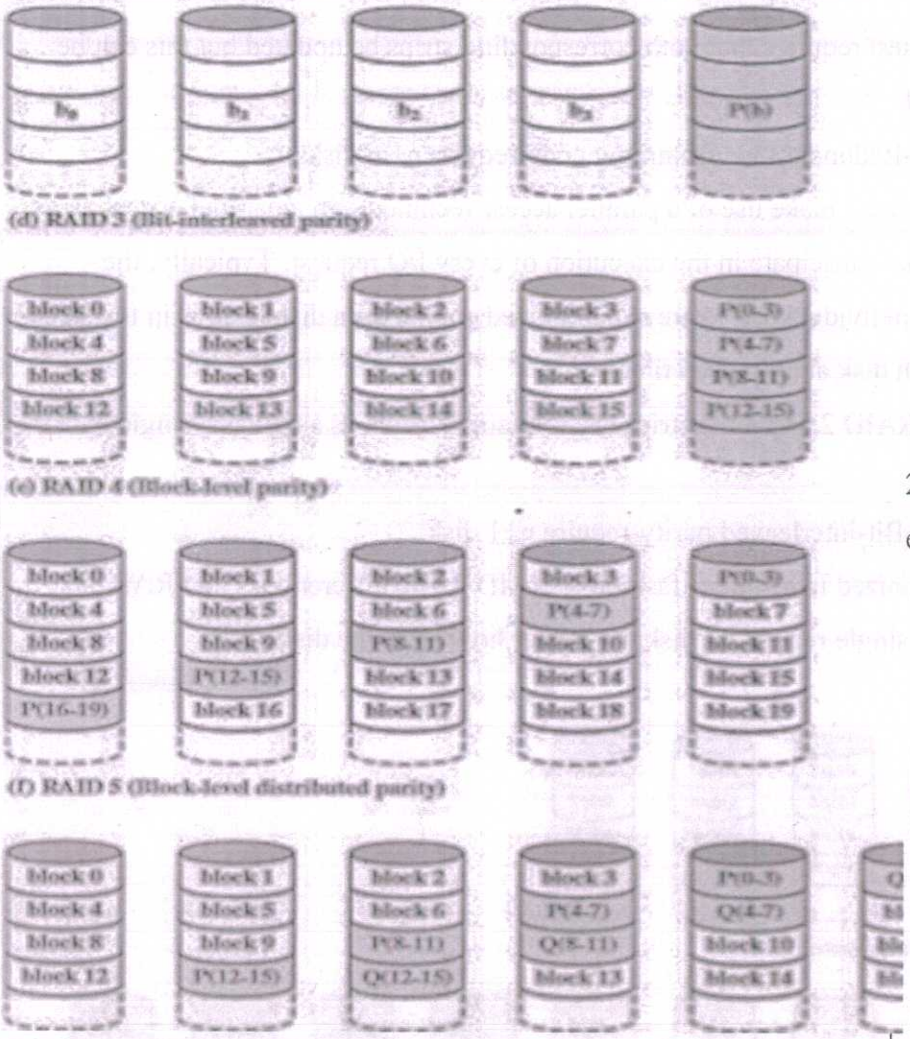
(b) RAID 1 (Mirrored)

(c) RAID 2 (Redundancy through Hamming code)

Figure 6.8 RAID Levels

Fig2+5 expl

VI(b)



(d) RAID 3 (Bit-interleaved parity)

(e) RAID 4 (Block-level parity)

(f) RAID 5 (Block-level distributed parity)

2 fig+6 expl

RAID Level 4- Block-interleaved parity n+1 disk

RAID levels 4 through 6 make use of an independent access technique .In an independent access array, each member disk operates independently, so that separate I/O requests can be satisfied in parallel.

RAID Level 5- Block-interleaved distributed parity ,require n+1 disk

RAID 5 is organized in a similar fashion to RAID 4.The difference is that RAID 5 distributes the parity strips across all disks. A typical allocation is a round-robin scheme, as illustrated in Figure 6.8f.For an n-disk array,the parity strip is on a different disk for the first n stripes,and the pattern then repeats.

RAID Level 6: - Block-interleaved dual distributed parity -n+2disk

In the RAID 6 scheme,two different parity calculations are carried out and stored in separafailure occurs to any disk. user data require N disks consists of N +2 disks

3 fig+4expl

VII(b)

15

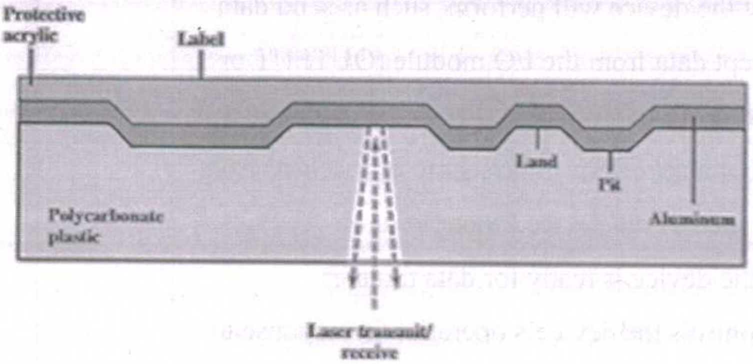


Figure 6.10 CD Operation

The disk is formed from a resin, such as polycarbonate. Digitally recorded information (either music or computer data) is imprinted as a series of microscopic pits on the surface of the polycarbonate. This is done,first of all, with a finely focused ,high-intensity laser to create a master disk. The master is used ,in turn, to make a die to stamp out copies onto polycarbonate.The pitted surface is then coated with a highly reflective surface, usually aluminum or gold. This shiny surface is protected against dust and scratches by a top coat of clear acrylic. Finally,a label can be silk screened onto the acrylic.

Information is retrieved from a CD or CD-ROM by a low-powered laser housed in an optical-disk player, or drive unit. The laser shines through the clear polycarbonate while a motor spins the disk past it (Figure 6.10).The intensity of the reflected light of the laser changes as it encounters a pit. Specifically, if the laser beam falls on a pit, which has a somewhat rough surface, the light scatters and a low intensity is reflected back to the source. The areas between pits are called lands .A land is a smooth surface, which reflects back at higher intensity. The change between pits and lands is detected by a photosensor and converted into a digital signal.The sensor tests the surface at regular intervals.The beginning or end of a pit represents a 1;when no change in elevation occurs between intervals,a 0 is recorded.

VII
b)

VII(a)

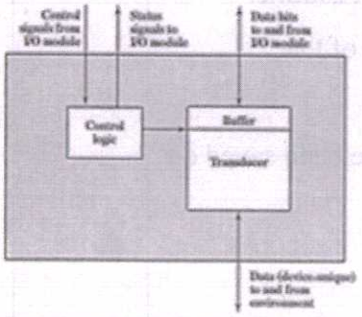

Control signals from I/O module
Status signals to I/O module
Data bits to and from I/O module
Control logic
Buffer
Transducer
Data (device-unique) to and from environment

Figure 7.2 Block Diagram of an External Device

3fig*2+
11 exp

15

The interface to the I/O module is in the form of control, data, and status signals.

Control signals determine the function that the device will perform, such as send data to the I/O module (INPUT or READ), accept data from the I/O module (OUTPUT or WRITE), report status, or perform some control function particular to the device (e.g., position a disk head). Data are in the form of a set of bits to be sent to or received from the I/O module.Status signals indicate the state of the device.Examples are READY/NOT-READY to show whether the device is ready for data transfer.

Control logic associated with the device controls the device's operation in response to direction from the I/O module.

The transducer converts data from electrical to other forms of energy during output and from other forms to electrical during input.Typically, a buffer is associated with the transducer to temporarily hold data being transferred between the I/O module and the external environment

VI (b) ,The I/O module will then interrupt the processor to request service when it is ready to exchange data with the processor.The processor thenexecutesthedatatransfer,asbefore,andthenresumesitsformerprocessing. Let us consider how this works,

first from the point of view of the I/O module. For input,the I/O module receives a READ command from the processor.The I/O module then proceeds to read data from an associated peripheral.Once the data are in the module's data register, the module signals an interrupt to the processor over a control line.The module then waits until its data are requested by the processor.When the request is made,the module places its data on the data bus and is then ready for another I/O operation

1. The device issues an interrupt signal to the processor. 2. The processor finishes execution of the current instruction before responding to the interrupt,

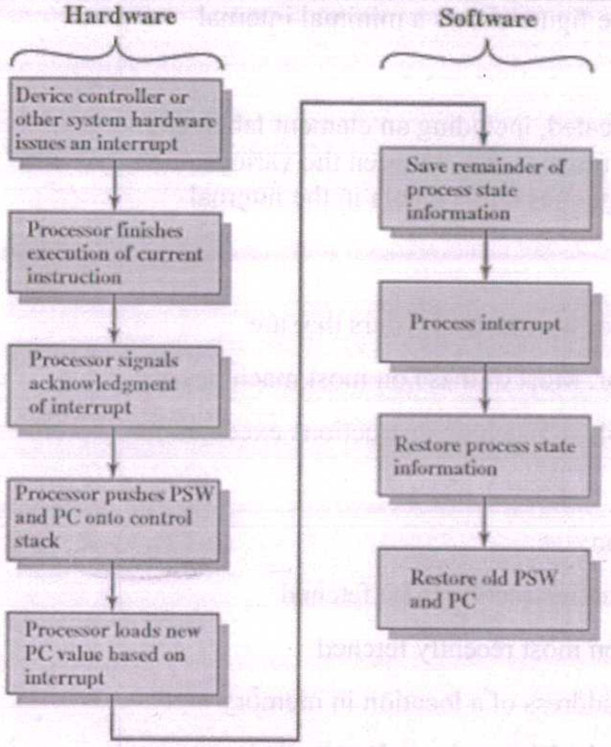
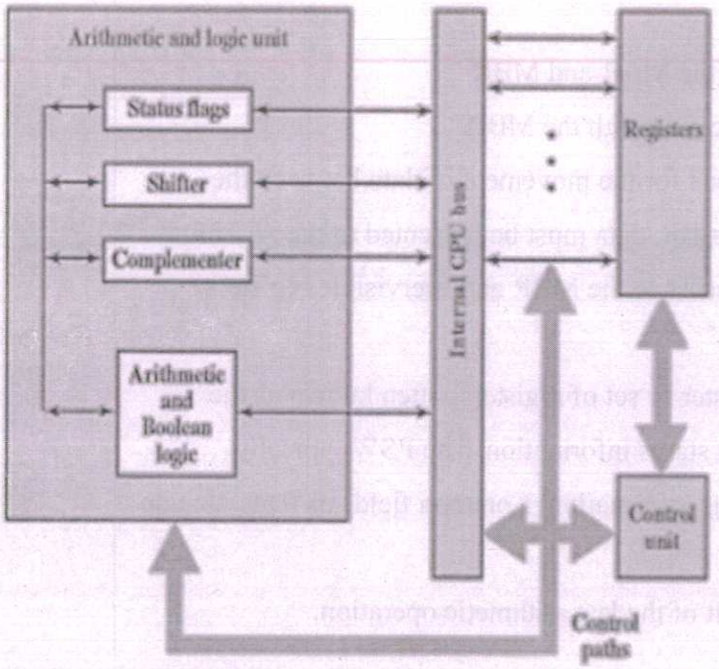
3. The processor tests for an interrupt,determines that there is one,and sends an acknowledgment signal to the device that issued the interrupt

4. The processor now needs to prepare to transfer control to the interrupt routine.

5. The processor now loads the program counter with the entry location of the interrupt-handling program that will respond to this interrupt.

6 the program counter and PSW relating to the interrupted program have been saved on the system stack.

7. The interrupt handler next processes the interrupt.

	 Hardware Software Device controller or other system hardware issues an interrupt Processor finishes execution of current instruction Processor signals acknowledgment of interrupt Processor pushes PSW and PC onto control stack Processor loads new PC value based on interrupt Save remainder of process state information Process interrupt Restore process state information Restore old PSW and PC **Figure 7.6** Simple Interrupt Processing  Arithmetic and logic unit Status flags Shifter Complementer Arithmetic and Boolean logic Internal CPU bus Registers Control unit Control paths **Figure 12.2** Internal Structure of the CPU	Fig3*2= 6 9 expl	15	
IX(a	Indicating its connection to the rest of the system via the system bus. the major components of the processor are an arithmetic and logic unit (ALU) and a control unit (CU).The ALU does the actual computation or processing of data.The control unit controls the movement of data and instructions into and out of the processor and			

)	controls the operation of the ALU. In addition, the figure shows a minimal internal memory, consisting of a set of storage .The data transfer and logic control paths are indicated, including an element labeled internal processor bus. This element is needed to transfer data between the various registers and the ALU because the ALU in fact operates only on data in the internal processor memory Control and Status Registers. There are a variety of processor registers that are employed to control the operation of the processor. Most of these,on most machines,are not visible to the user.Some of them may be visible to machine instructions executed in a control or operating system mode. Four registers are essential to instruction execution: • Program counter (PC): Contains the address of an instruction to be fetched • Instruction register (IR): Contains the instruction most recently fetched • Memory address register (MAR):Contains the address of a location in memory • Memory buffer register (MBR): Contains a word of data to be written to memory or the word most recently ,the processor updates the PC after each instruction fetch so that the PC always points to the next instruction to be executed .The fetched instruction is loaded into an IR, where the opcode and operand specifiers are analyzed. Data are exchanged with memory using the MAR and MBR . User visible registers ,in turn,exchange data with the MBR. The four registers just mentioned are used for the movement of data between the processor and memory. Within the processor, data must be presented to the ALU for processing.The ALU may have direct access to the MBR and uservisible registers. psw . Many processor designs include a register or set of registers, often known as the program status word (PSW),that contain status information. The PSW typically contains condition codes plus other status information. Common fields or flags include the following: • Sign: Contains the sign bit of the result of the last arithmetic operation. • Zero: Set when the result is 0. • Carry: Set if an operation resulted in a carry (addition) into or borrow (subtraction) out of a high-order bit. • Equal: Set if a logical compare result is equality. • Overflow: Used to indicate arithmetic overflow. • Interrupt Enable/Disable: Used to enable or disable interrupts.	10	15	

- Supervisor: Indicates whether the processor is executing in supervisor or user mode.

X(a) Certain privileged instructions can be executed only in supervisor mode.

VIII(a) Data Flow :

The exact sequence of events during an instruction cycle depends on the design of the processor

a processor that employs a memory address register (MAR), a memory buffer register (MBR), a program counter (PC), and an instruction register (IR). During the fetch cycle, an instruction is read from memory. The PC contains the address of the next instruction to be fetched. This address is moved to the MAR and placed on the address bus. The control unit requests a memory read, and the result is placed on the data bus and copied into the MBR and then moved to the IR. Meanwhile, the PC is incremented by 1,preparatory for the next fetch.

5

2×3 fig
+1
exp1 -

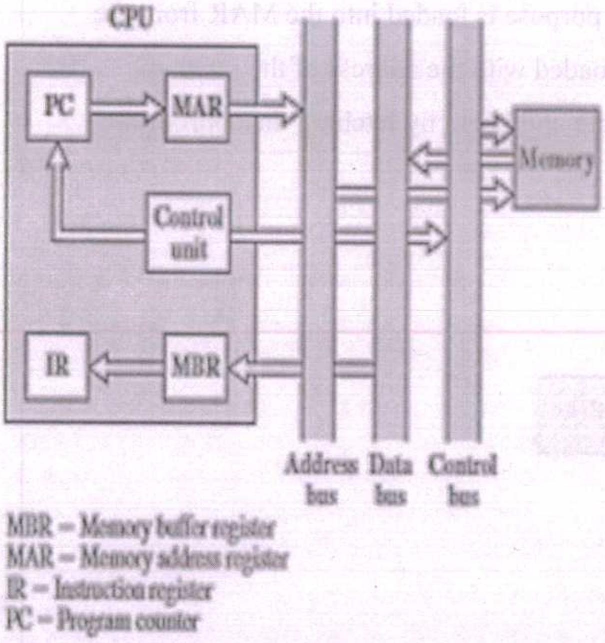

CPU
PC
MAR
Control unit
IR
MBR
Memory
Address bus
Data bus
Control bus
MBR = Memory buffer register
MAR = Memory address register
IR = Instruction register
PC = Program counter

Figure 12.6 Data Flow, Fetch Cycle

Once the fetch cycle is over,the control unit examines the contents of the IR to determine if it contains an operand specifier using indirect addressing. If so, an indirect cycle is performed

The execute cycle: This cycle may involve transferring data among registers, read or write from memory or I/O,and/or the invocation of the ALU

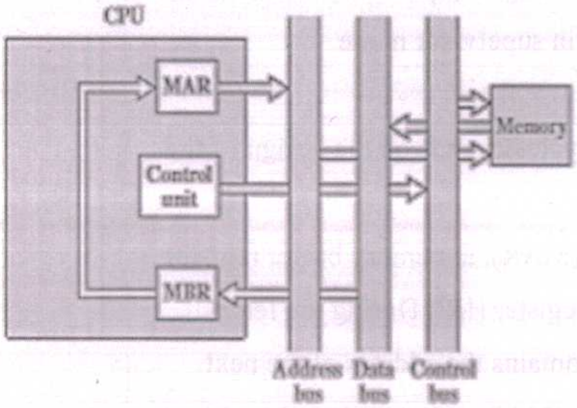


Figure 12.7 Data Flow, Indirect Cycle

the interrupt cycle: is simple and predictable (Figure 12.8).The current contents of the PC must be saved so that the processor can resume normal activity after the interrupt. Thus, the contents of the PC are transferred to the MBR to be written into memory.The special memory location reserved for this purpose is loaded into the MAR from the control unit,be a stack pointer.The PC is loaded with the address of the interrupt routine.As a result,the next instruction cycle will begin by fetching the appropriate instruction

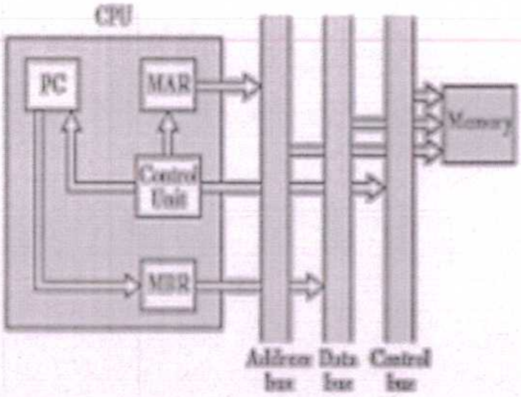


Figure 12.8 Data Flow, Interrupt Cycle

IX

.There are mainly two organisational technic used to implement control unit

1.hardwired implementation

2. Microprogrammed implementation

In a hardwired implementation, the control unit is essentially a state machine circuit.Its input logic signals are transformed into a set of output logic signals,which are the control signals.

8+
7.

It is difficult to design and test such a piece of hardware. Furthermore,the design isrelatively inflexible.For example,it is difficult to change the design if one wishes to add a new machine instruction.mainly used in risc machines.

The key inputs are the instruction register,the clock,flags,and control bus signals.In the case of the flags and control bus signals, each individual bit typically has some meaning (e.g., overflow).The other two inputs,however,are not directly useful to the control unit. First consider the instruction register.The control unit makes use of the opcode and will perform different actions (issue a different combination of control signals) for different instructions.To simplify the control unit logic, there should be a unique logic input for each opcode.This function can be performed by a decoder, which takes an encoded input and produces a single output. In general, a decoder will have n binary inputs and 2n binary outputs. Each of the 2n different input patterns will activate a single unique output.

The clock portion of the control unit issues a repetitive sequence of pulses. This is useful for measuring the duration of micro-operations.

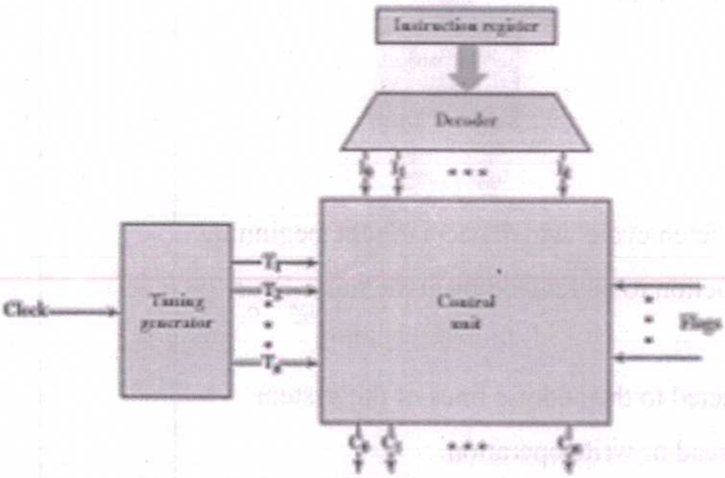
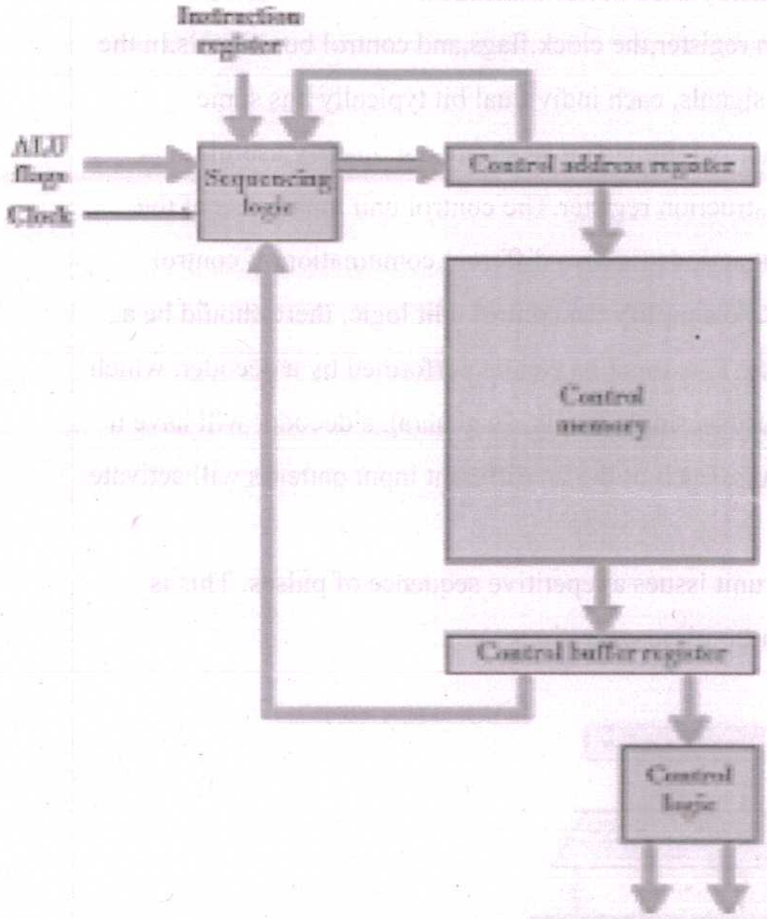


Figure 15.10 Control Unit with Decoded Inputs

An alternative to a hardwired control unit is a microprogrammed control unit, in which the logic of the control unit is specified by a microprogram. A microprogram consists of a sequence of instructions in a microprogramming language. These are very simple instructions that specify micro-operations.

◆ A microprogrammed control unit is a relatively simple logic circuit that is capable of (1) sequencing through microinstructions and (2) generating control signals to execute each microinstruction.

Microinstructions.The design of control unit must include logic for sequencing through micro-operations, for executing microoperations, for interpreting opcodes,and for making decisions based on ALU flags.

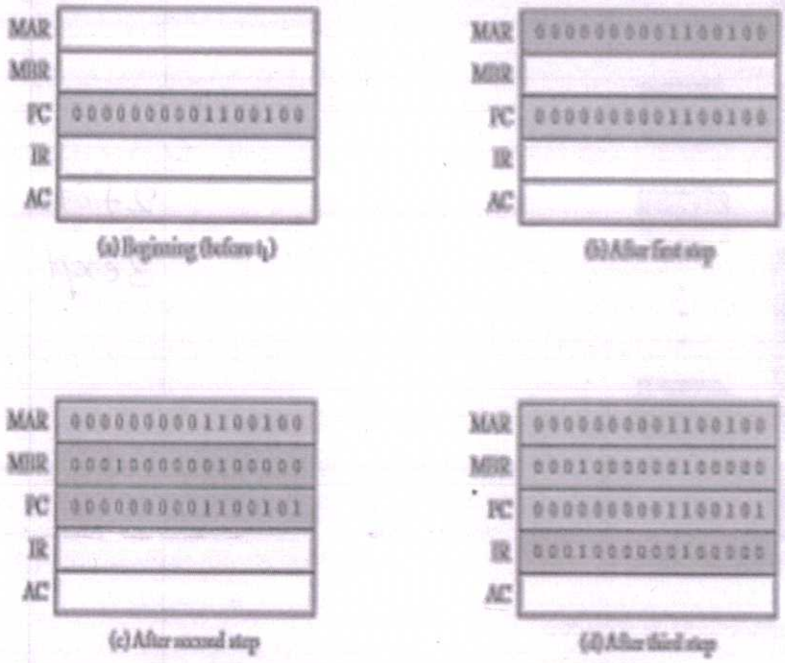


The Fetch Cycle We begin by looking at the fetch cycle, which occurs at the beginning of each instruction cycle and causes an instruction to be fetched from memory. Four registers are involved:

- Memory address register (MAR): Is connected to the address lines of the system bus.It specifies the address in memory for a read or write operation.
- Memory buffer register (MBR):Is connected to the data lines of the system bus. It contains the value to be stored in memory or the last value read from memory.
- Program counter (PC):Holds the address of the next instruction to be fetched.
- Instruction register (IR): Holds the last instruction fetched

X

3 fig + 7 expl.



(a) Beginning (before t1)

(b) After first step

(c) After second step

(d) After third step

Figure 15.2 Sequence of Events, Fetch Cycle

t1: MAR ← (PC)

t2: MBR ← Memory PC ← (PC) + I

t3: IR ← (MBR)

The Indirect Cycle:

t1: MAR ← (IR(Address))

t2: MBR ← Memory

t3: IR(Address) ← (MBR(Address))

The Interrupt Cycle

t1: MBR ← (PC)

t2: MAR ← Save_Address PC ← Routine_Address

t3: Memory ← (MBR

The Execute Cycle

t1: MAR ← (IR(address))

t2: MBR ← Memory

t3: MBR ← (MBR) + 1

t4: Memory ← (MBR) If ((MBR) = 0) then (PC ← (PC) + I).

There are two or more processors .Each processor is self-contained, including a control unit , ALU, registers ,and, typically ,one or more levels of cache. Each processor x(b)

2 fig+
3 expl

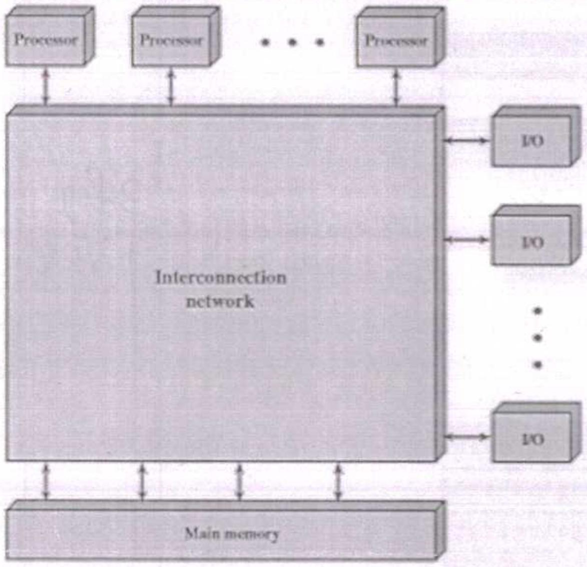


Figure 17.4 Generic Block Diagram of a Tightly Coupled Multiprocessor

17.2 / SYMMETRIC MULTIPROCESSORS

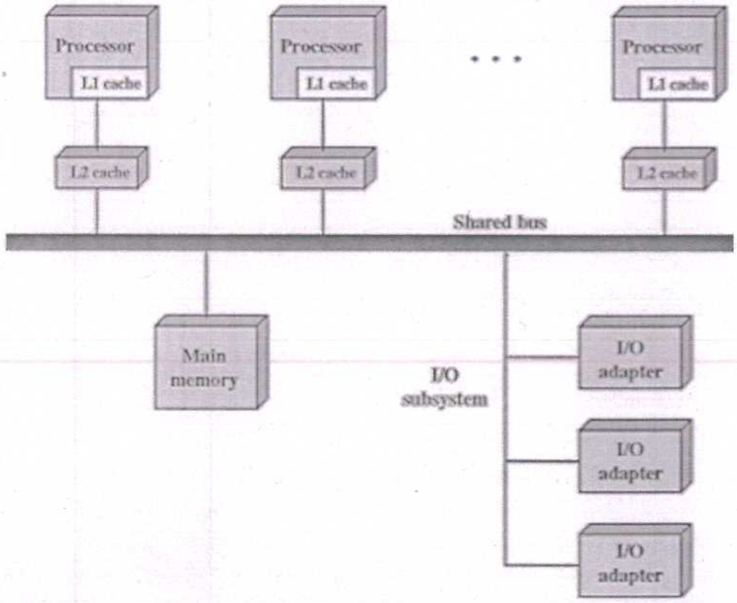


Figure 17.5 Symmetric Multiprocessor Organization

has access to a shared main memory and the I/O devices through some form of interconnection mechanism. The processors can communicate with each other through memory (messages and status information left in common data areas). It may also be possible for processors to exchange signals directly. The memory is often organized so that multiple simultaneous accesses to separate blocks of memory are possible.In some configurations,each processor may also have its own private main memory and I/O channels in addition to the shared resources.