USN □□□□□□□□□□

Improvement Test – Nov. 2017

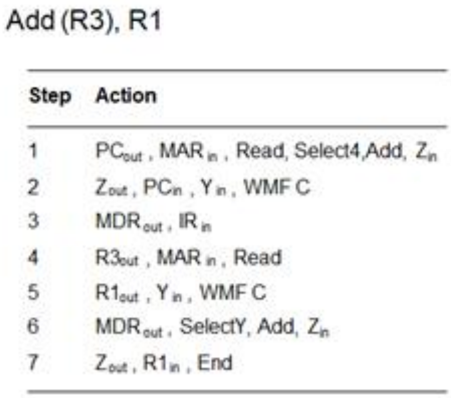
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sub: | Computer Organization | | | | Sub Code: | 15CS34 | Branch: | CSE, ISE |
| Date: | 20/11/2017 | Duration: | 90 mins | Max Marks: 50 | Sem/Sec: | 3 (A,B,C) | | OBE |

Answer **FOUR** FULL questions selecting AT LEAST ONE question **FROM EACH PART** — MARKS — CO — RBT

## PART A

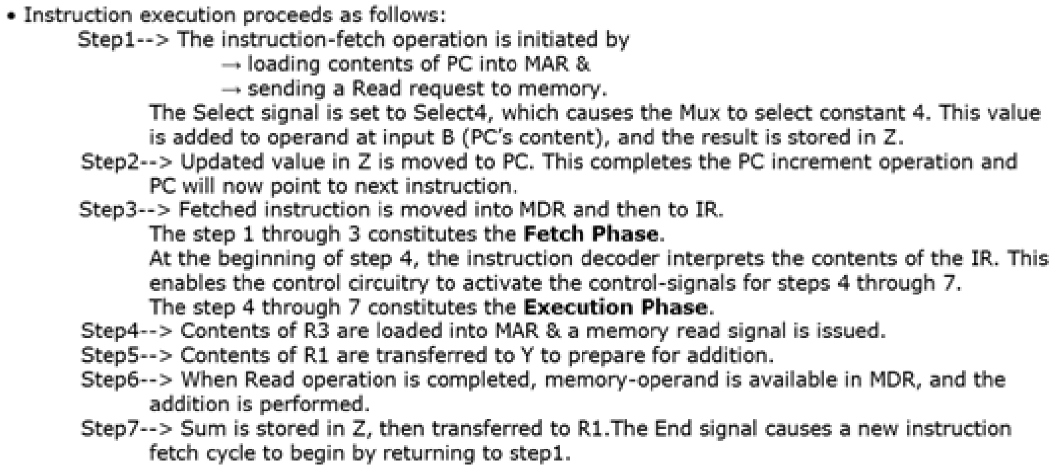
1 (a) Give the control sequence for execution of complete instruction ADD (R3), R1. [10] CO3 L3

(steps 7M and Explanation 3M)



Add (R3), R1

| Step | Action |
|---|---|
| 1 | $PC_{out}$ , $MAR_{in}$ , Read, Select4,Add, $Z_{in}$ |
| 2 | $Z_{out}$ , $PC_{in}$ , $Y_{in}$ , WMF C |
| 3 | $MDR_{out}$ , $IR_{in}$ |
| 4 | $R3_{out}$ , $MAR_{in}$ , Read |
| 5 | $R1_{out}$ , $Y_{in}$ , WMF C |
| 6 | $MDR_{out}$ , SelectY, Add, $Z_{in}$ |
| 7 | $Z_{out}$ , $R1_{in}$ , End |



- Instruction execution proceeds as follows:

Step1--> The instruction-fetch operation is initiated by

→ loading contents of PC into MAR &

→ sending a Read request to memory.

The Select signal is set to Select4, which causes the Mux to select constant 4. This value is added to operand at input B (PC's content), and the result is stored in Z.

Step2--> Updated value in Z is moved to PC. This completes the PC increment operation and PC will now point to next instruction.

Step3--> Fetched instruction is moved into MDR and then to IR.

The step 1 through 3 constitutes the **Fetch Phase**.

At the beginning of step 4, the instruction decoder interprets the contents of the IR. This enables the control circuitry to activate the control-signals for steps 4 through 7.

The step 4 through 7 constitutes the **Execution Phase**.

Step4--> Contents of R3 are loaded into MAR & a memory read signal is issued.

Step5--> Contents of R1 are transferred to Y to prepare for addition.

Step6--> When Read operation is completed, memory-operand is available in MDR, and the addition is performed.

Step7--> Sum is stored in Z, then transferred to R1.The End signal causes a new instruction fetch cycle to begin by returning to step1.

**OR**

2 (a) Explain the block diagram of a typical embedded processor chip. [4] CO4 L2

(Diagram 7M and Explanation 3M)

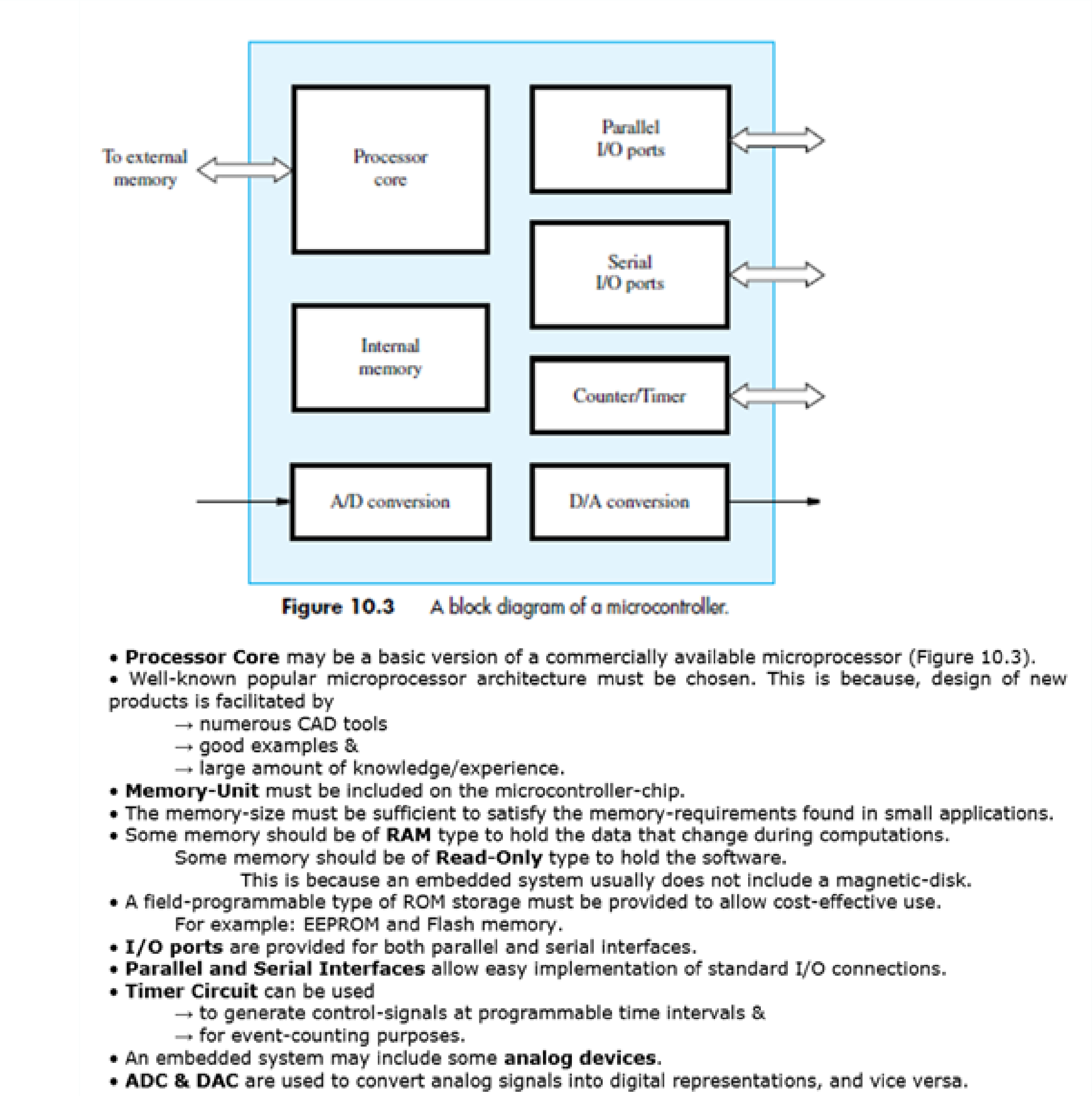

To external memory
Processor core
Parallel I/O ports
Serial I/O ports
Internal memory
Counter/Timer
A/D conversion
D/A conversion

**Figure 10.3** A block diagram of a microcontroller.

- **Processor Core** may be a basic version of a commercially available microprocessor (Figure 10.3).
- Well-known popular microprocessor architecture must be chosen. This is because, design of new products is facilitated by
  - → numerous CAD tools
  - → good examples &
  - → large amount of knowledge/experience.
- **Memory-Unit** must be included on the microcontroller-chip.
- The memory-size must be sufficient to satisfy the memory-requirements found in small applications.
- Some memory should be of **RAM** type to hold the data that change during computations.
  Some memory should be of **Read-Only** type to hold the software.
  This is because an embedded system usually does not include a magnetic-disk.
- A field-programmable type of ROM storage must be provided to allow cost-effective use.
  For example: EEPROM and Flash memory.
- **I/O ports** are provided for both parallel and serial interfaces.
- **Parallel and Serial Interfaces** allow easy implementation of standard I/O connections.
- **Timer Circuit** can be used
  - → to generate control-signals at programmable time intervals &
  - → for event-counting purposes.
- An embedded system may include some **analog devices**.
- **ADC & DAC** are used to convert analog signals into digital representations, and vice versa.

(b) Explain three different structures of general-purpose multiprocessors. [6]

(Three types 3x2M)

CO2 L2

**1. UMA (Uniform Memory Access) Multiprocessor**

- An interconnection-network permits n processors to access k memories (Figure 12.2). Thus, any of the processors can access any of the memories.
- The interconnection-network may introduce network-delay between
  1) Processor &
  2) Memory.
- A system which has the same network-latency for all accesses from the processors to the memory-modules is called a **UMA Multiprocessor**.
- Although the latency is uniform, it may be large for a network that connects
  → many processors &
  → many memory-modules.
- For better performance, it is desirable to place a memory-module close to each processor.
- **Disadvantage:**
  ➢ Interconnection-networks with very short delays are costly and complex to implement.

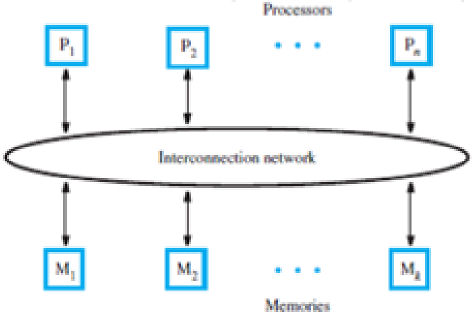


**Figure 12.2** A UMA multiprocessor.

**2. NUMA (Non-Uniform Memory Access) Multiprocessors**

- Memory-modules are attached directly to the processors (Figure 12.3).
- The network-latency is avoided when a processor makes a request to access its local memory.
- However, a request to access a remote-memory-module must pass through the network.
- Because of the difference in latencies for accessing local and remote portions of the shared memory, systems of this type are called **NUMA** multiprocessors.
- **Advantage:**
  ➢ A high computation rate is achieved in all processors
- **Disadvantage:**
  ➢ The remote accesses take considerably longer than accesses to the local memory.

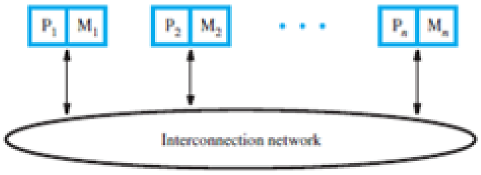


**Figure 12.3** A NUMA multiprocessor.

**3. Distributed Memory Systems**

- All memory-modules serve as private memories for processors that are directly connected to them.
- A processor cannot access a remote-memory without the cooperation of the remote-processor.
- This cooperation takes place in the form of messages exchanged by the processors.
- Such systems are often called **Distributed-Memory Systems** (Figure 12.4).

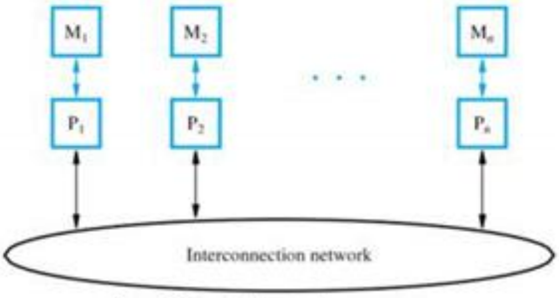


**Figure 12.4** A distributed memory system.

**PART B**

3 (a) With a figure, explain single bus organization of datapath inside a processor. [10] CO3 L2

(Diagram 7M, Explanation 3M)

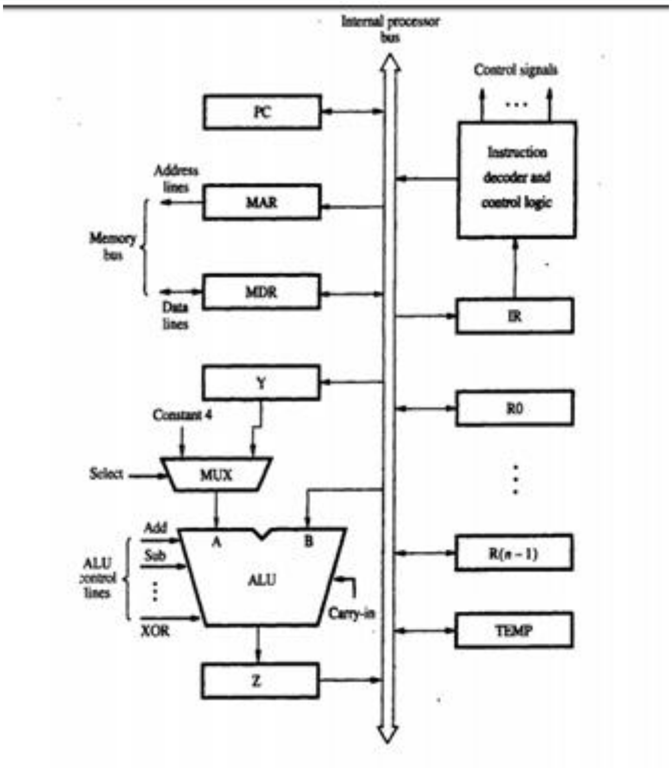


**Figure 7.1** Single-bus organization of the datapath inside a processor.

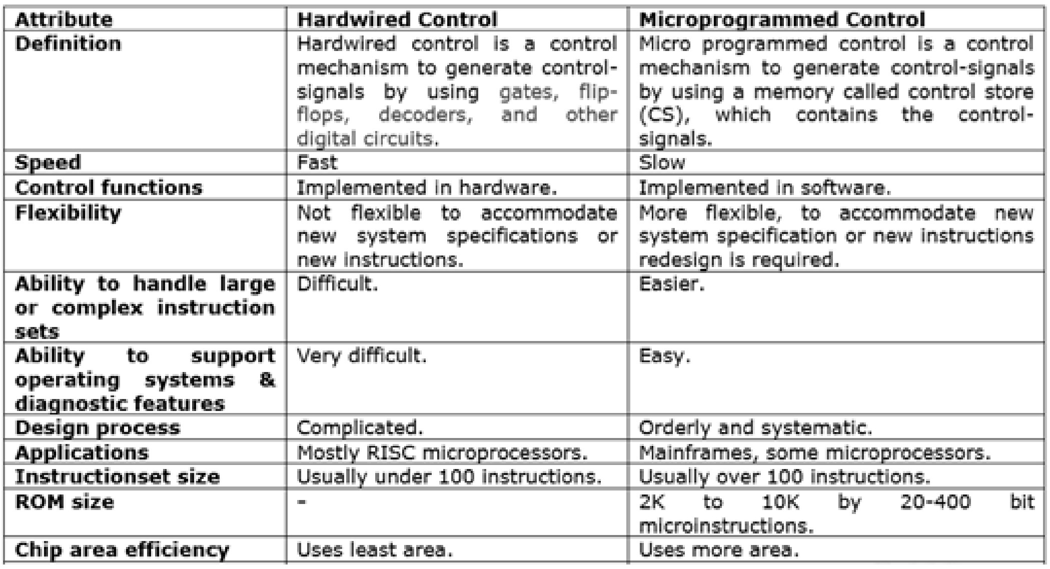
**SINGLE BUS ORGANIZATION**

- ALU and all the registers are interconnected via a **Single Common Bus** (Figure 7.1).
- Data & address lines of the external memory-bus is connected to the internal processor-bus via MDR & MAR respectively. (MDR→ Memory Data Register, MAR → Memory Address Register).
- **MDR** has 2 inputs and 2 outputs. Data may be loaded
  → into MDR either from memory-bus (external) or
  → from processor-bus (internal).
- **MAR**'s input is connected to internal-bus;
  MAR's output is connected to external-bus.
- **Instruction Decoder & Control Unit** is responsible for
  → issuing the control-signals to all the units inside the processor.
  → implementing the actions specified by the instruction (loaded in the IR).
- Register R0 through R(n-1) are the **Processor Registers**.
  The programmer can access these registers for general-purpose use.
- Only processor can access 3 registers **Y**, **Z** & **Temp** for temporary storage during program-execution.
  The programmer cannot access these 3 registers.
- In **ALU**, 1) 'A' input gets the operand from the output of the multiplexer (MUX).
  2) 'B' input gets the operand directly from the processor-bus.
- There are 2 options provided for 'A' input of the ALU.
- MUX is used to select one of the 2 inputs.
- **MUX** selects either
  → output of Y or
  → constant-value 4( which is used to increment PC content).

- An instruction is executed by performing one or more of the following operations:
  1) Transfer a word of data from one register to another or to the ALU.
  2) Perform arithmetic or a logic operation and store the result in a register.
  3) Fetch the contents of a given memory-location and load them into a register.
  4) Store a word of data from a register into a given memory-location.
- **Disadvantage:** Only one data-word can be transferred over the bus in a clock cycle.
  **Solution:** Provide multiple internal-paths. Multiple paths allow several data-transfers to take place in parallel.

**OR**

| | | | | |
|---|---|---|---|---|
| 4 (a) | Explain the differences between Hardwired and Micro-programmed control. | [10] | CO3 | L2 |



| Attribute | Hardwired Control | Microprogrammed Control |
|---|---|---|
| Definition | Hardwired control is a control mechanism to generate control-signals by using gates, flip-flops, decoders, and other digital circuits. | Micro programmed control is a control mechanism to generate control-signals by using a memory called control store (CS), which contains the control-signals. |
| Speed | Fast | Slow |
| Control functions | Implemented in hardware. | Implemented in software. |
| Flexibility | Not flexible to accommodate new system specifications or new instructions. | More flexible, to accommodate new system specification or new instructions redesign is required. |
| Ability to handle large or complex instruction sets | Difficult. | Easier. |
| Ability to support operating systems & diagnostic features | Very difficult. | Easy. |
| Design process | Complicated. | Orderly and systematic. |
| Applications | Mostly RISC microprocessors. | Mainframes, some microprocessors. |
| Instructionset size | Usually under 100 instructions. | Usually over 100 instructions. |
| ROM size | - | 2K to 10K by 20-400 bit microinstructions. |
| Chip area efficiency | Uses least area. | Uses more area. |

## PART C

5 (a) Explain the organization of Microwave oven with a neat block diagram. [10] CO4 L2

(Diagram 7M, Explanation 3M)

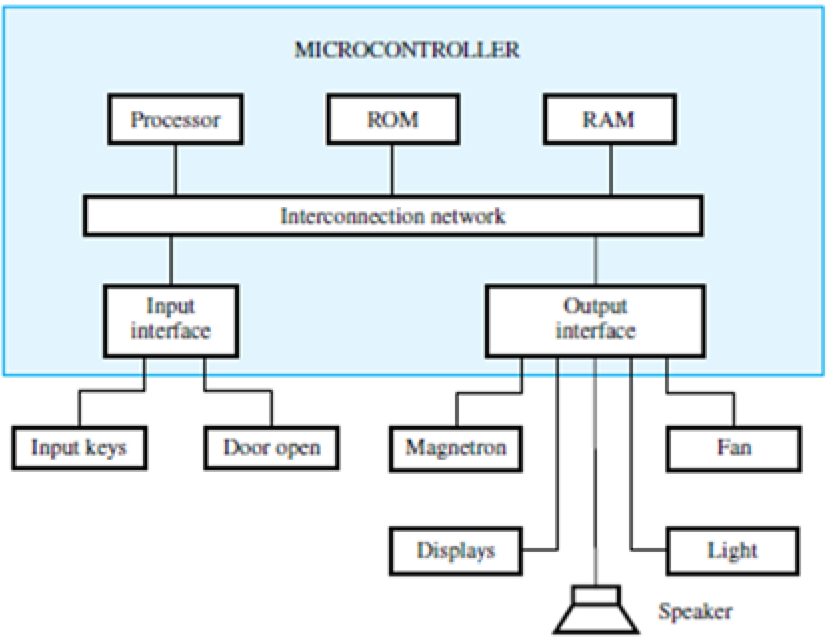
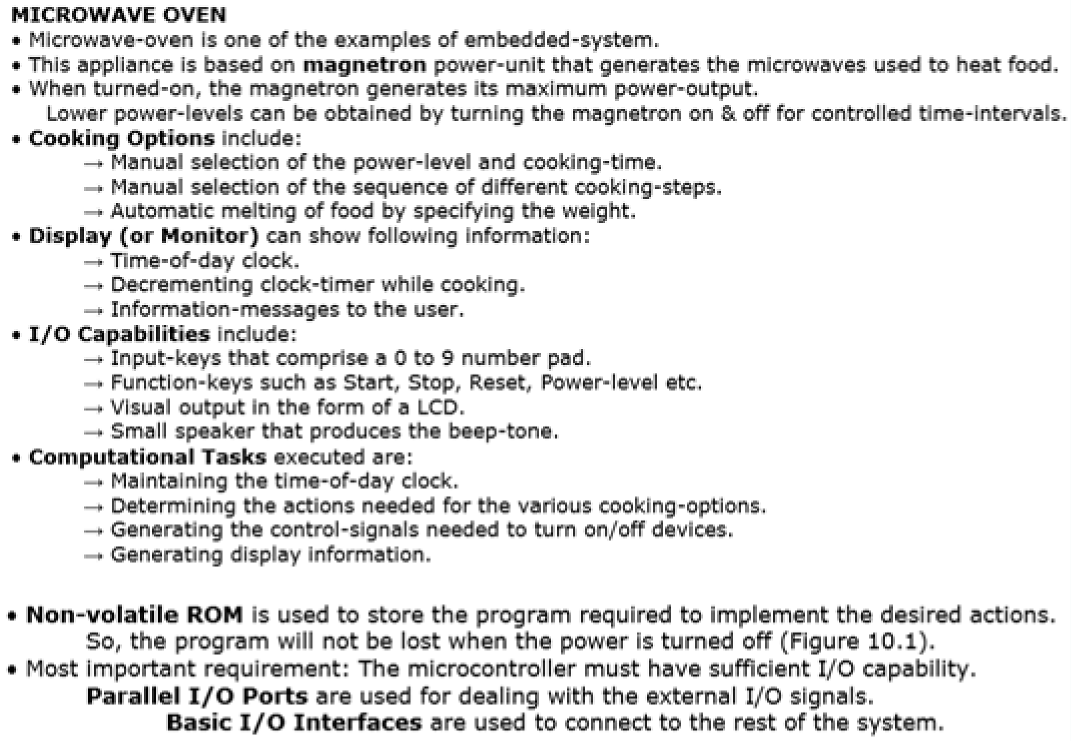


**Figure 10.1** A block diagram of a microwave oven.



**MICROWAVE OVEN**

- Microwave-oven is one of the examples of embedded-system.
- This appliance is based on **magnetron** power-unit that generates the microwaves used to heat food.
- When turned-on, the magnetron generates its maximum power-output.
  Lower power-levels can be obtained by turning the magnetron on & off for controlled time-intervals.
- **Cooking Options** include:
  → Manual selection of the power-level and cooking-time.
  → Manual selection of the sequence of different cooking-steps.
  → Automatic melting of food by specifying the weight.
- **Display (or Monitor)** can show following information:
  → Time-of-day clock.
  → Decrementing clock-timer while cooking.
  → Information-messages to the user.
- **I/O Capabilities** include:
  → Input-keys that comprise a 0 to 9 number pad.
  → Function-keys such as Start, Stop, Reset, Power-level etc.
  → Visual output in the form of a LCD.
  → Small speaker that produces the beep-tone.
- **Computational Tasks** executed are:
  → Maintaining the time-of-day clock.
  → Determining the actions needed for the various cooking-options.
  → Generating the control-signals needed to turn on/off devices.
  → Generating display information.

- **Non-volatile ROM** is used to store the program required to implement the desired actions.
  So, the program will not be lost when the power is turned off (Figure 10.1).
- Most important requirement: The microcontroller must have sufficient I/O capability.
  **Parallel I/O Ports** are used for dealing with the external I/O signals.
  **Basic I/O Interfaces** are used to connect to the rest of the system.

**OR**

6 (a) Briefly explain the block diagram of digital camera. [10]
(Diagram 7M, Explanation 3M)

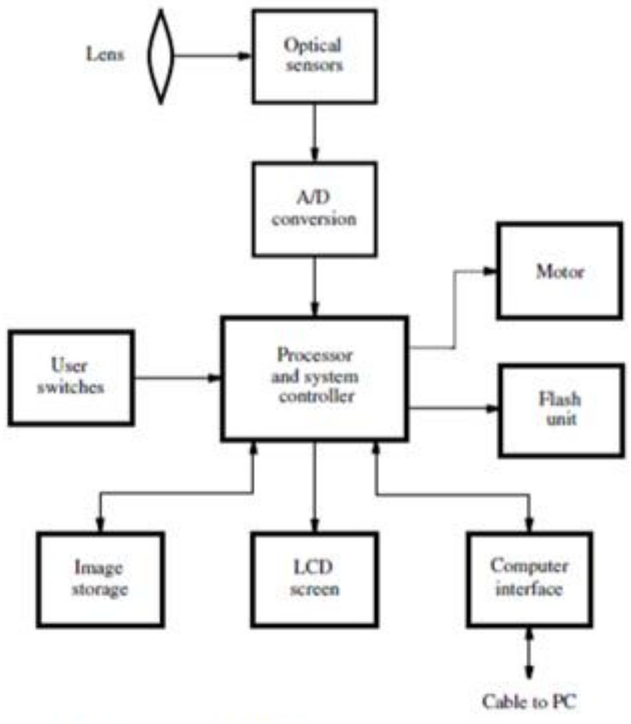
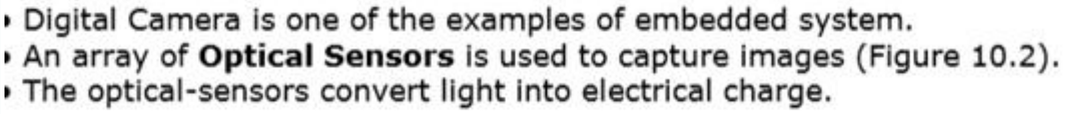
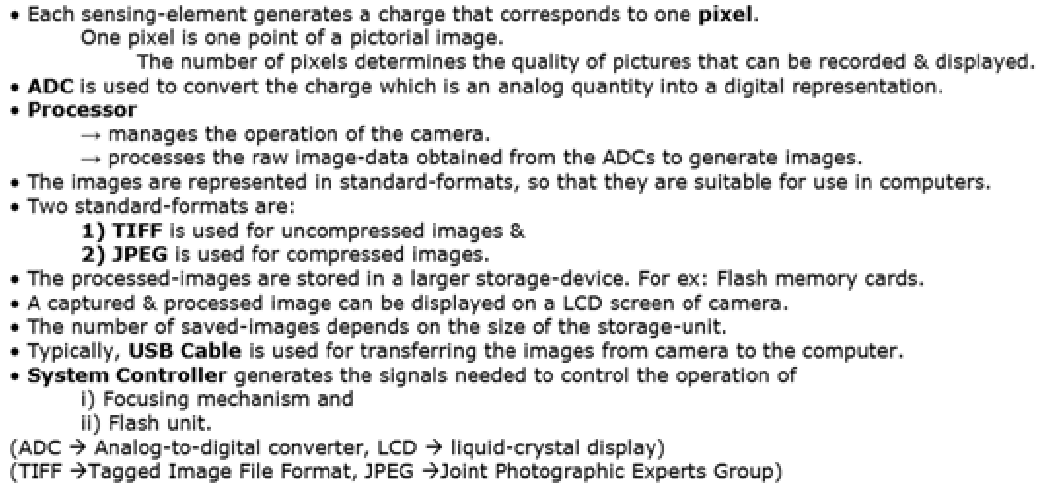


**Figure 10.2** A simplified block diagram of a digital camera.



- Digital Camera is one of the examples of embedded system.
- An array of **Optical Sensors** is used to capture images (Figure 10.2).
- The optical-sensors convert light into electrical charge.

CO4 L2



- Each sensing-element generates a charge that corresponds to one **pixel**.
  One pixel is one point of a pictorial image.
  The number of pixels determines the quality of pictures that can be recorded & displayed.
- **ADC** is used to convert the charge which is an analog quantity into a digital representation.
- **Processor**
  → manages the operation of the camera.
  → processes the raw image-data obtained from the ADCs to generate images.
- The images are represented in standard-formats, so that they are suitable for use in computers.
- Two standard-formats are:
  **1) TIFF** is used for uncompressed images &
  **2) JPEG** is used for compressed images.
- The processed-images are stored in a larger storage-device. For ex: Flash memory cards.
- A captured & processed image can be displayed on a LCD screen of camera.
- The number of saved-images depends on the size of the storage-unit.
- Typically, **USB Cable** is used for transferring the images from camera to the computer.
- **System Controller** generates the signals needed to control the operation of
  i) Focusing mechanism and
  ii) Flash unit.

(ADC → Analog-to-digital converter, LCD → liquid-crystal display)
(TIFF →Tagged Image File Format, JPEG →Joint Photographic Experts Group)

## PART D

7(a) Explain three types of mapping functions for cache memory. [15] CO5 L2

(Direct mapping-5M, Associative Mapping- 5M, Set-associative mapping- 5M)

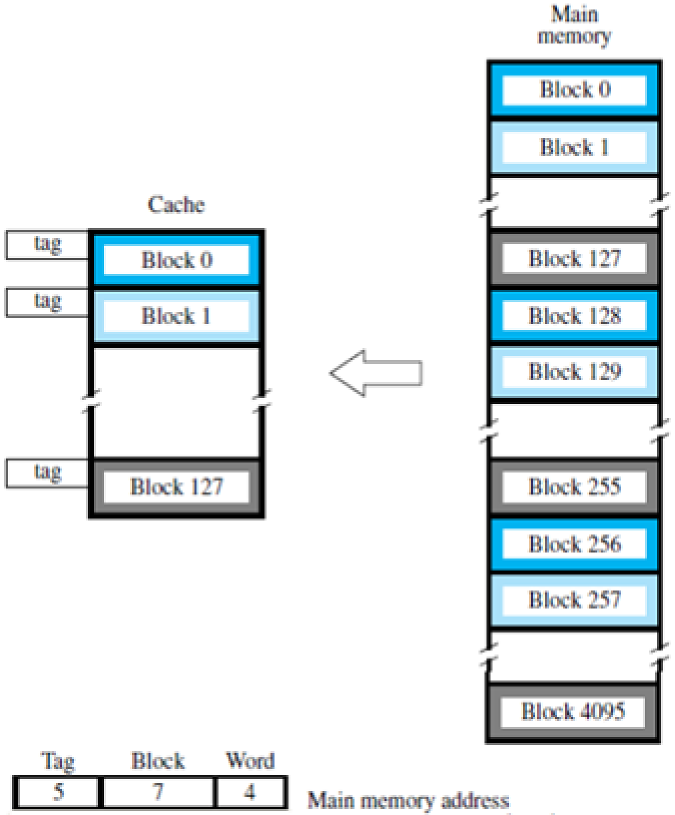


**Figure 8.16** Direct-mapped cache.

**DIRECT MAPPING**

This technique is easy to implement but not very flexible.

Block j of the main memory maps onto j modulo 128 of the cache. For example, whenever **one of** the main memory blocks 0, 128, 256, …. Is loaded in the cache, it is stored in cache block 0 . Main memory blocks 1, 129, 257, …. are stored in cache block 1 (one at a time), and so on. Contention may occur for a single cache block required by multiple memory blocks. E.g when for program execution both memory block 1 and 129 are required but cache block 1 can only store one memory block. To resolve this, new blocks are allowed to overwrite the currently resident block.

From example,

4096 memory blocks need to be mapped to 128 cache blocks. i.e, each cache block identified 32 memory blocks(4096/128).

**Main memory address is divided into three parts:**

**Tag** (5 bits): identify which memory block (out of 32 in this case) is currently resident in the cache

**Block** (7 bits): cache block position where the new memory block must be stored

**Word** (4 bits): selects one of the words of the memory block (out of 16 words per block in this case)

**ASSOCIATIVE MAPPING**

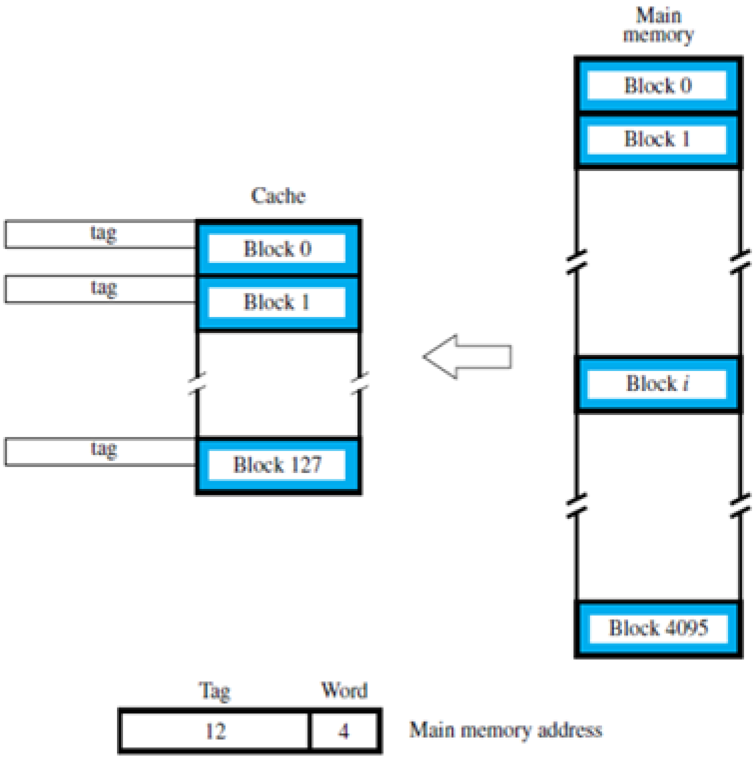


**Figure 8.17** Associative-mapped cache.

- It is more flexible than direct mapping technique but more expensive. Main memory block can be placed into any cache block position.
- Memory address is divided into two fields:
  - Low order 4 bits identify the memory word within a block.
  - High order 12 bits or tag bits identify a memory block when residing 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

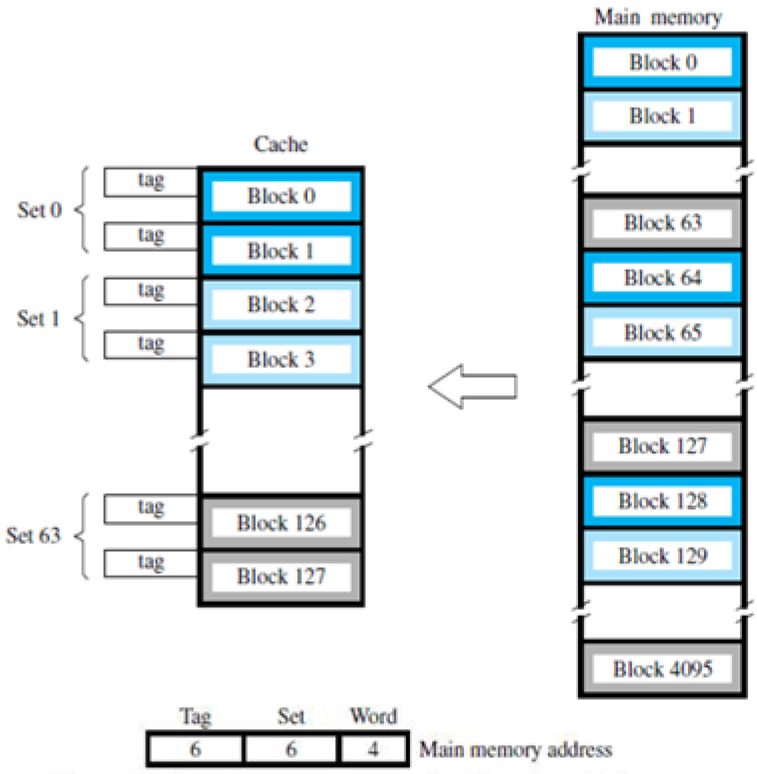


**Figure 8.18** Set-associative-mapped cache with two blocks per set.

It is a **combination** of direct mapping and associative mapping techniques.
Blocks of the cache are grouped into **sets**, and the mapping allows a block of the main memory to reside in **any block of a specific set**.
Contention problem of direct mapping is eased by having a few choices for block placement. Hardware cost is reduced by decreasing the associative search.

(b) Briefly explain replacement algorithms for cache memory. [5]

Example:
Consider cache controller is tracking a set of four blocks in a set associative cache.
A 2-bit counter is used for each block (00=0, 01=1, 10=2, 11=3)

a) Hit occurs:
The counter of the block that is referenced is set to 0. Other block counters having value less than the value of the referenced counter are incremented by 1. Block counters having value greater than the value of referenced counter are remain unchanged.
Initially: 2, 3, 0, 1
Hit occurs for 2
Finally: 0 (after set to 0), 3 (unchanged), 1 (after increment), 2 (after increment)

b) Miss occurs (Set not full):
The counter of the block where new block is loaded from memory is set to 0. All other block counters value is incremented by 1.
Initially: 2, 1, 0, _
Miss occurs
Finally: 3, 2, 1, **0** (new)

c) Miss occurs (Set full):
The block with the highest counter value i.e, 3 is removed. The new block is put into its place and its counter value is set to 0. All other three blocks counter values are incremented by 1.
Initially: 2, 3, 0, 1
Miss occurs
Finally: 3, 0 (replaced), 1, 2

CO5 L2

**OR**

8 (a) With neat diagrams, explain virtual memory in detail. [12]

Techniques that automatically move program and data between main memory and secondary storage when they are required for execution are called virtual-memory techniques.

binary addresses generated by processor are called logical or virtual addresses.

The virtual-address is translated into physical-address by MMU (Memory Management Unit).

• During every memory-cycle, MMU determines whether the addressed-word is in the memory.

If the word is in memory.

Then, the word is accessed and execution proceeds.

Otherwise, a page containing desired word is transferred from disk to memory.

• Using DMA scheme, transfer of data between disk and memory is performed.

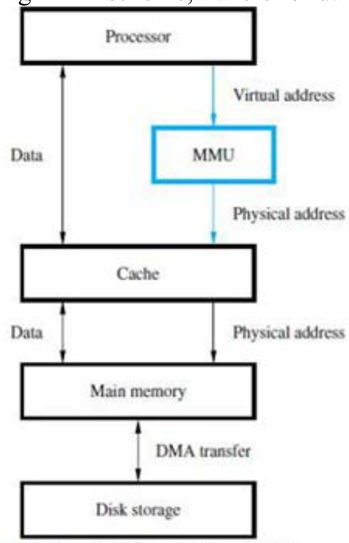


**Figure 8.24** Virtual memory organization.

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.

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.

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.

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.

One bit indicates whether the page has been modified during its residency in the main memory.

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

CO2 L2

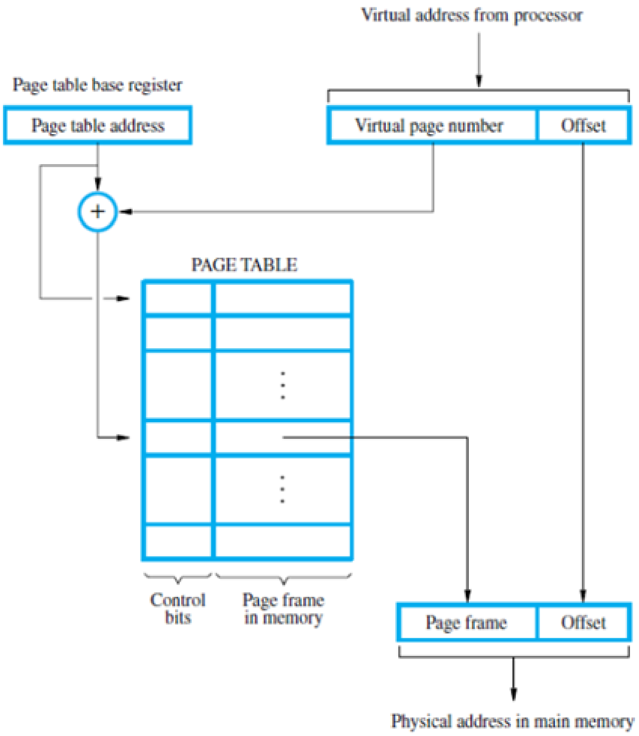
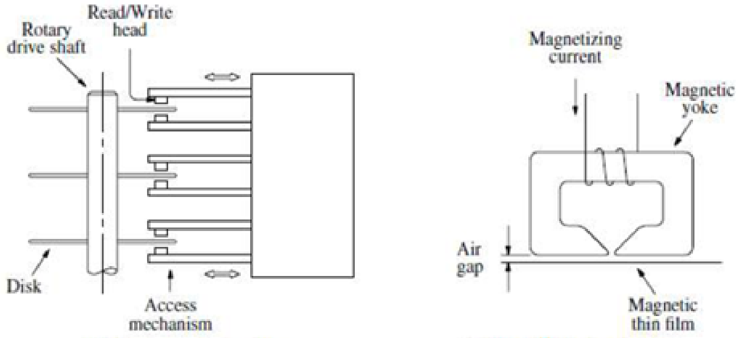


**Figure 8.25** Virtual-memory address translation.

The Page-table is placed in the memory but a copy of small portion of the page-table is located within MMU. This small portion is called TLB (Translation LookAside Buffer).
TLB consists of the page-table entries that corresponds to the most recently accessed pages.
TLB also contains the virtual-address of the entry.
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.
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.
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.

(b) Explain secondary storage of magnetic disk. [8] CO2 L2

- Magnetic Disk system consists of one or more **disk** mounted on a common **spindle**.
- A **thin magnetic film** is deposited on each disk (Figure 8.27).
- Disk is placed in a **rotary-drive** so that magnetized surfaces move in close proximity to R/W heads.
- Each **R/W head** consists of 1) Magnetic Yoke & 2) Magnetizing-Coil.
- Digital information is stored on magnetic film by applying current pulse to the magnetizing-coil.
- Only changes in the magnetic field under the head can be sensed during the Read-operation.
- Therefore, if the binary states 0 & 1 are represented by two opposite states,
  then a voltage is induced in the head only at 0-1 and at 1-0 transition in the bit stream.
- A consecutive of 0's & 1's are determined by using the clock.
- **Manchester Encoding** technique is used to combine the clocking information with data.



(a) Mechanical structure (b) Read/Write head detail

- R/W heads are maintained at small distance from disk-surfaces in order to achieve high bit densities.
- When disk is moving at their steady state, the air pressure develops b/w disk-surfaces & head.
  This air pressure forces the head away from the surface.
- The flexible spring connection between head and its arm mounting permits the head to fly at the desired distance away from the surface.

**Winchester Technology**

- Read/Write heads are placed in a sealed, air-filtered enclosure called the Winchester Technology.

Components of Disk Storage System

The disk system consists of three parts

1. **Disk:** The assembly of disk platters
2. **Drive:** The mechanisms that spin the disk moves the read / write head & disk is called drive
3. **Disk Controller:** The electronic circuit which control the drive is called Disk Controller.

**ORGANIZATION & ACCESSING OF DATA ON A DISK**

- Each surface is divided into concentric **Tracks** (Figure 8.28).
- Each track is divided into **Sectors.**
- The set of corresponding tracks on all surfaces of a stack of disk form a **Logical Cylinder**.

- The data on all tracks of a cylinder can be accessed without moving read / write head.
- The data are accessed by specifying *the surface number, track number and the sector number.*
- The Read/Write-operation start at sector boundaries.
- Data bits are stored serially on each track.

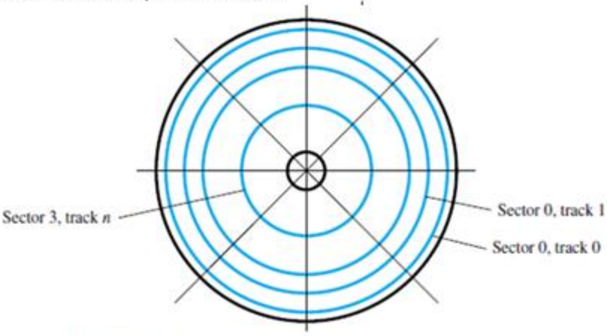


**Figure 8.28** Organization of one surface of a disk.

**1) Seek time:** Time required to move the read/write head to the proper track.
**2) Latency/Rotational Delay:** The amount of time that elapses after head is positioned over the correct track until the starting position of the addressed sector passes under the R/W head.
Seek time + Latency = Disk access time