CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - I

Sub:	Computer Organization and Arm Microcontroller						Code:	21EC52
Date:	18/12/2023	Duration:	90 mins	Max Marks:	50	Sem: 3rd	Branch:	ECE

Answer Any FIVE FULL Questions

		Marks	OBE CO	OBE RBT
1.	a) Define Bus. Explain single bus structure in an architecture. b) Differentiate hardware and architecture of a computer	[8] [2]	CO1 CO1	L2 L1
2.	Explain different functional units of a digital computer with the help of a diagram.	[10]	CO1	L2
3.	a) Write an ALP using ARM to add two 64 bit numbers. b) How to improve the performance of a computer?	[6] [4]	CO1 CO1	L3 L3
4.	a) Define Exception. Explain two kinds of exception. b) With a diagram explain daisy chaining technique.	[6] [4]	CO1 CO1	L2 L1,L2
5.	With a neat sketch, Explain a method for handling interrupts from multiple devices.	[10]	CO1	L3
6.	With a neat diagram, explain DMA controller.	[10]	CO1	L3

1 a)

Bus Structures

The individual parts of a computer need to be connected in an organized way to increase the speed of operation. When a word of data is transferred between units, all its bits are transferred in parallel that is bits are transferred simultaneously over many wires or lines, one bit per line. A group of lines that serves as a connecting path for several devices is called a bus. The simplest way to inter connect functional units is to use a single bus as shown in Fig 1.3.

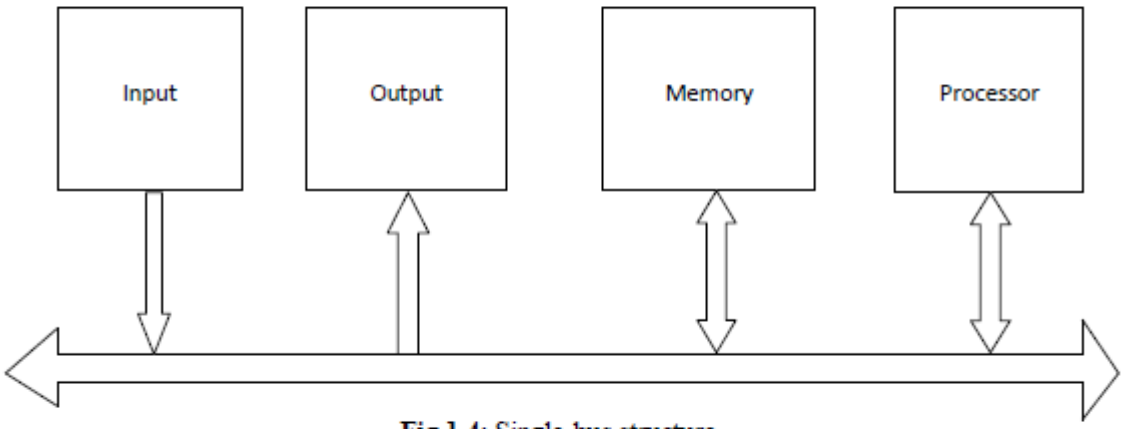

Fig 1.4: Single-bus structure

The main virtue of single-bus structure is its low cost and its flexibility for attaching peripheral devices. Systems that contain multiple buses achieve more concurrency in operation by allowing two or more transfers to be carried out at the same time. This leads to better performance but at increased cost.

A common approach is to include buffer registers with the devices to hold the information during transfers.

1 b) Computer hardware consists of electronic circuits, displays, magnetic and optical storage media, electromechanical equipment, and communication facilities.

Computer architecture encompasses the specification of an instruction set and the hardware units that implement the instructions.

2.

Functional Units

A computer consists of five functionally independent main parts: input, memory, arithmetic and logic, output and control units as shown in fig 1.1.

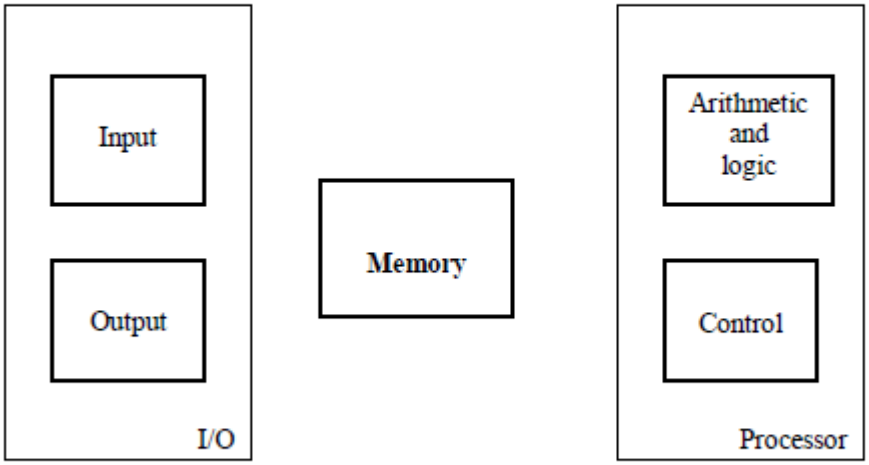

Fig 1.1 Basic functional units of a computer

The input unit accepts coded information from human operators, from electromechanical devices such as keyboards or from other computers over digital communication line. The information received is either stored in the computer memory for later reference or immediately used by the arithmetic and logic circuitry to perform the desired operations. The processing steps are determined by the program stored in the memory. Finally the results are shown on the output unit. All of these actions are co-ordinated by the control unit. We refer to the arithmetic and logic circuits in conjunction to the control circuits as the processor and input and output units are referred to as input-output (I/O) unit.

Instructions or *machine instructions* are explicit commands that

- Govern the transfer of information within a computer as well as between the computers and its I/O devices.
- Specify the arithmetic and logic operations to be performed.

A list of instructions that perform a task is called *program*. The computer is completely controlled by a stored program except for a possible external interruption by an operator or by I/O devices connected to the machine. Data is used to mean any digital information. Each number, character or instruction is encoded as a string of binary digits known as bits each having one of two possible values 0 or 1.

Input Unit

Computers accept the coded information through input units which read the data. The well known input device is Keyboard. When a key is pressed, the corresponding letter or digit is automatically translated into its corresponding binary code and transmitted over a cable to either the memory or processor.

Memory Unit

The memory unit is used to store program and data. There are two classes of storage known as primary and secondary.

Primary memory is a fast storage that operates at electronic speeds. Programs are stored in the memory while they are executed. The memory contains large number of semiconductor storage cells each capable of storing one bit but instead are processed as groups of fixed size called words. The memory is organized so that a word can be stored or retrieved in one basic operation. A distinct address is associated to each word in the memory. Addresses are numbers that identify successive locations.

Programs must reside in the memory during execution. Instructions and data can be read out or written into the memory under the control of the processor. Memory in which any location can be reached in short and fixed amount of time after specifying its address is called random-access memory (RAM). The small, fast RAM units are called *caches*.

The additional cheaper secondary storage is used when large amount of data and many programs have to be stored particularly for information that is accessed infrequently.

Arithmetic and Logic Unit (ALU)

Any arithmetic and logic operation is initiated by bringing the required operands into the processor where the operation is performed by the ALU. When the operands are brought into the processor they are stored in high speed storage elements called *registers*. Access time to register is faster than access time to the fastest cache unit in the memory hierarchy. The control and the arithmetic logic units are many times faster than any other devices connected to a computer system.

Output Unit

The output unit is a counterpart of input unit. Its function is used the processed results to the outside world. The most familiar example of such a device is a printer.

Control Unit

The control unit is a well defined physically separate unit that interacts with other parts of the machine. The control unit sends the control signals to other units and senses their states. Timing signals are generated by the control circuits that determine when a given action is to take place. Data transfer between memory and processor is also controlled unit through timing signals. A large set of control lines (wires) carries the signals used for timing and synchronization of events in all units.

3 a)

```
        AREA ADD64, CODE, READONLY
        EXPORT __main
__main
ENTRY
        LDR R0, = Value1
```

```
                LDR R1, [R0]
                LDR R2, [R0, #4]

                LDR R0, = Value2
                LDR R3, [R0]
                LDR R4, [R0,#4]

                ADDS R6, R2, R4
                ADCS R5, R1, R3

                MOV R8, #0
                ADC R7, R8,#0

                LDR R0, = Result
                STR R7, [R0]
                STR R5, [R0, #4]
                STR R6, [R0, #8]
STOP    B       STOP


Value1  DCD 0X1AAAAAAA, 0XBBBBBBBB          ;
Value2  DCD 0XCCCCCCCC, 0XDDDDDDDD          ;
;Value1 DCD 0XFFFFFFFF, 0XFFFFFFFF     ;
;Value2 DCD 0XFFFFFFFF, 0XFFFFFFFF     ;

                AREA data64, DATA, READWRITE
Result  DCD 0X0
                END
```

3b)

Basic Performance Equation

Let T be the processor time required to execute a program that has been prepared by some high level language. The compiler generates machine level object program that corresponds to source program. Assume that complete execution of the program requires the execution of N machine language instructions. Suppose that the average number of basic steps needed to execute one machine instruction is S, where each basic step is completed in one clock cycle. If the clock rate is R cycles per second, the program execution time is given by *basic performance equation*.

$$T = \frac{N \times S}{R}$$

To achieve high performance, the value of T must be reduced which can be done by reducing N and S, and increasing R. The value of N is reduced if the source program is compiled in fewer machine instructions. The value of S is reduced if instructions have a smaller number of basic steps to perform or if the execution of instructions are overlapped. Using a higher-frequency clock increases the value of R which means the time required to complete a basic execution step is reduced.

4 a)

Exceptions and interrupts are unexpected events that disrupt the normal flow of instruction execution. An exception is an unexpected event from within the processor. An interrupt is an unexpected event from outside the processor. You are to implement exception and interrupt handling in your multicycle CPU design. External interrupts come from input input (I/O) devices, from a timing device, from a circuit monitoring the power supply, or from any other external source. Examples that cause external interrupts are l/0 device requesting transfer of data, l/0 device finished transfer of data, elapsed time of an event, or power failure. Internal interrupts arise from illegal or erroneous use of an instruction or data. Internal interrupts are also called traps. Examples of interrupts caused by internal error conditions are register overflow, attempt to divide by zero, an invalid operation code, stack overflow, and protection violation. External and internal interrupts are initiated from signals that occur in the hardware of the CPU. A software interrupt is initiated by executing an instruction

4 b)

If several devices share one interrupt request line, some other mechanism is needed. When several devices raises interrupt request and $\overline{INTR}$ line is activated, the processor responds by setting the INTA line to 1. The signal is received by device 1. Device 1 passes the signal onto device 2 only if it does not require any service. If device 1 has pending request for interrupt, it blocks the INTA signal and proceeds to put its identification code on to data lines. In daisy chain the device that is electrically closest to the processor has the highest priority.

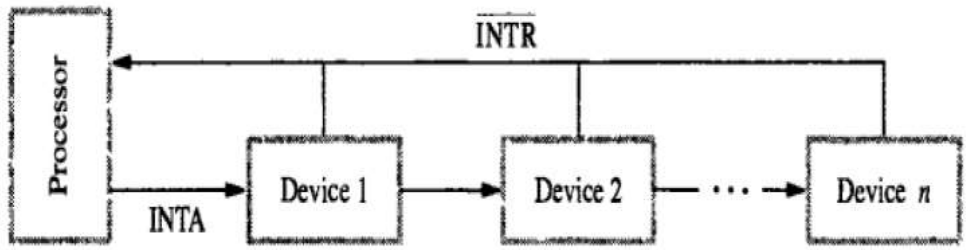

Fig 7: Daisy chain

5.

Handling Multiple devices

The information needed to determine whether a device is requesting an interrupt is available in its status register. When a device raises an interrupt request, one of the bits of the status register is set to 1 which we call IRQ bit.

KIRQ, DIRQ are the interrupt request bits for keyboard and display. The polling scheme ha the disadvantage that the time spent interrogating the IRQ bits of all the devices that may not be requesting any service. An alternate approach is to use vectored interrupts.

Vectored Interrupts

A device requesting an interrupt can identify itself by sending special code to the processor over the bus. The code supplied by the device represents the starting address of the interrupt service routine. The code length is 4 to 8 bits. The processor reads this address called the interrupt vector and stores it in to the PC. The interrupt vector may also include a new value for a processor status register.

The interrupted device must wait to put on the bus only when the processor is ready to receive it. When the processor is ready to receive the vector interrupt code, it activates the interrupt acknowledge line INTA. The I/O device responds by sending its interrupt vector code and turning off INTR signal.

Interrupt Nesting

I/O devices should be organized in a priority structure. An interrupt request from a high priority should be accepted while the processor is serving another request from the lower priority device.

We can assign priority level to the processor that can be changed under program control. The priority level of the processor is the priority of the program that is currently being executed. The processor accepts interrupts from devices that have priorities higher than its own.

The processor is in supervisory mode when it is executing the OS routines. It switches to User mode before beginning to execute application programs. The privileged instructions can be executed only while the processor is running in the supervisory mode. A multiple priority scheme can be implemented by using separate interrupt request and interrupt acknowledge lines from each device.

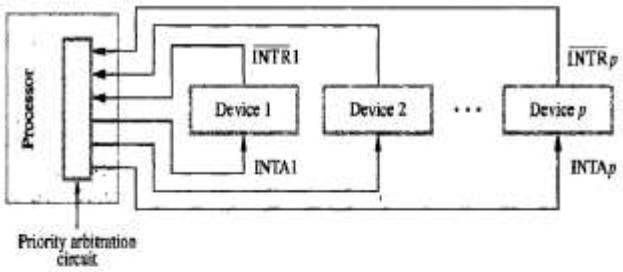

Fig 6: Implementation of interrupt priority using individual interrupt request & acknowledge lines

Simultaneous Requests

If several devices share one interrupt request line, some other mechanism is needed. When several devices raises interrupt request and *INTR* line is activated, the processor responds by setting the INTA line to 1. The signal is received by device 1. Device 1 passes the signal onto device 2 only if it does not require any service. If device 1 has pending request for interrupt, it blocks the INTA signal and proceeds to put its identification code on to data lines. In daisy chain the device that is electrically closest to the processor has the highest priority.

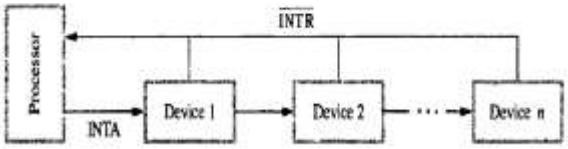

Fig 7: Daisy chain

6.

Direct Memory Access

To transfer large blocks of data at high speeds, an alternate approach is used. A special control unit may be provided to allow transfer of block of data directly between external device and main memory without intervention by processor. This approach is called direct memory access or DMA.

DMA transfers are performed by control circuits that are part of I/O interface called DMA controller. The DMA controller performs functions that would normally be carried out by processor when accessing main memory.

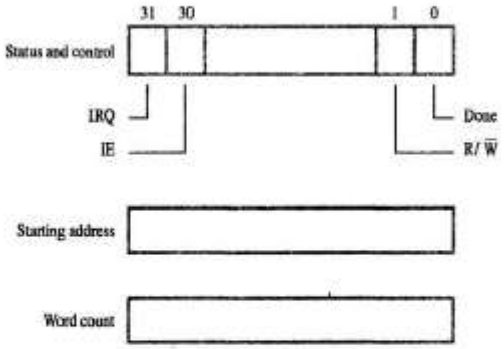

Fig 9: Registers in DMA interface

The R/W bit determine the direction of transfer. When this bit is set to 1 by a program instruction, the controller performs read operation that is it transfers data from memory to I/O device. When transfer is complete, it sets done flag to 1. When IE is1, it causes the controller to raise an interrupt after it has completed transferring block of data. Finally IRQ bit is set to 1 when it has requested interrupt.

Requests from DMA devices are given high priority than processor requests. Among different DMA devices high priority is given to high speed peripherals such as disks, high speed network interface or graphic display device.

The processor originates most memory cycles, the DMA controller is said to steal memory cycles from processor. This technique is called cycle stealing. DMA controller is given access to main memory to transfer a block of data without interruption. This is called as block or burst mode.

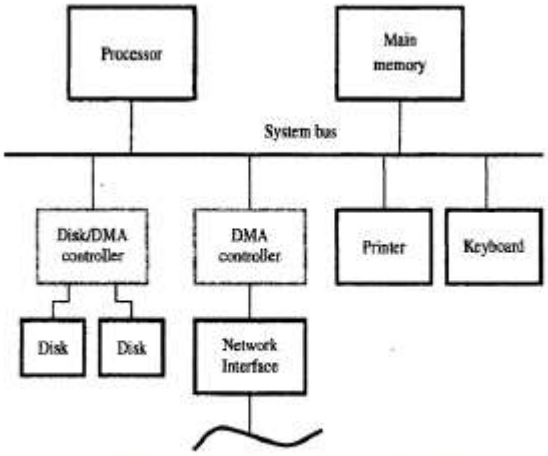

Fig 10: Use of DMA controllers in computer system