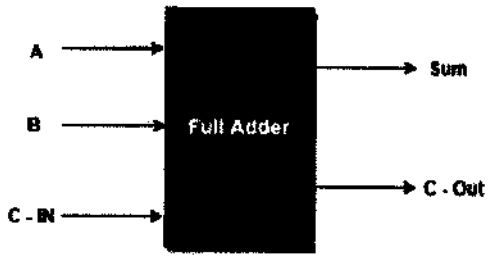
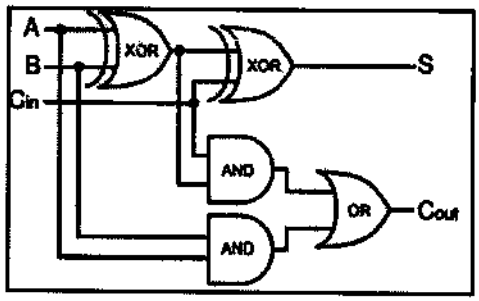
F.Y.B.Sc COMPUTER SCIENCE SEMESTER 1 EXAMINATION, NOVEMBER 2018
COMPUTER ORGANIZATION AND DESIGN SOLUTION SET (Regular)

Q. P. Code: 59462

Note: - Any point other than solution set which is relevant to the question; as per evaluator's knowledge; can be subject for marks.			
Q.1		**Attempt All (Each of 5 marks)**	
a)		**Multiple Choice Questions.**	**(15M)**
	i)	In decimal number system base is ___ a) 8 b) 2 c) **10** d) 16	**(5M)**
	ii)	RISC stands for _____ a)**Reduced instruction set computer.** b) Reduced instruction set component. c) Reference instruction set computer. d) Reduced in set computer.	
	iii)	If one of the input to an OR gate is high its output will be ___ a) Medium b) **High** c) Low	
	iv)	A _____ used in microcomputers to temporarily store data being transmitted to or from a peripheral device. a) **Data register** b) MBR c) Index Register d) MDR	
	v)	Assembly is called a _____ programming language a) **low-level** b) high level c) binary	
b)		**Fill in the blanks** **(false ,true, adder ,peripheral devices, half adder ,mnemonic, memory buffer register , memory bus register)**	**(5M)**
	i)	The output of AND gate is **true** only when all the inputs are true.	
	ii)	An **adder** is a device that can add two binary digits	
	iii)	Assembly language uses a **mnemonic** to represent each low-level machine instruction or opcode.	
	iv)	MBR stands for **memory buffer register.**	
	v)	Input or output devices that are connected to computer are called **peripheral devices.**	
c)		**Short Answers**	
	i)	**Define fan-in.** Fan-in is the number of inputs a logic gate can handle. Using logic gates with higher fan-in will help reducing the depth of a logic circuit.	**(5M)**
	ii)	**What is read and write operation.** Reading data from memory and Writing data to memory sections, we considered the process of loading and storing data from the perspective of the microprocessor and in terms of signals on control wires and the buses.	
	iii)	**Define stack.** In computing, a *stack* is a data structure used to store a collection of objects. Individual items can be added and stored in a *stack* using a push operation. Objects can be retrieved using a pop operation, which removes an item from the *stack*	

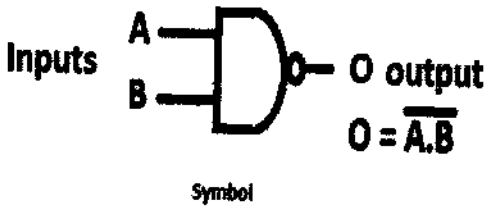
Q. P. Code: 59462

	iv)	**Find the equivalent decimal number for octal number 143** Steps of conversion **(1/2 marks)** Decimal number :-99 **(1/2 marks)**	
	v)	**What are ALU and CU?** An *arithmetic logic unit* (*ALU*) is a combinational digital electronic circuit that performs arithmetic and bitwise operations on integer binary numbers. The control unit (CU) is a component of a **computer's** central processing unit (CPU) that directs the operation of the processor. It tells the **computer's** memory, arithmetic/logic unit and input and output devices how to respond to a program's instructions.	
Q.2		**Attempt the following (Any THREE) (Each of 5Marks)**	**(15M)**
	a)	**Design full adder circuit.** Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.  So, we can implement a full adder circuit with the help of two half adder circuits. At first, half adder will be used to add A and B to produce a partial Sum and a second half adder logic can be used to add C-IN to the Sum produced by the first half adder to get the final S output.  *Full Adder Logic Circuit*	5
	b)	**Explain the concept of universal gate.** A universal logic gate is a logic gate that can be used to construct all other logic	

gates. **NAND and NOR** Gates are called Universal Gates because all the other gates can be created by using these gates

NAND Gate

The NAND gate is a digital logic gate with 'n' i/ps and one o/p, that performs the operation of the AND gate followed by the operation of the NOT gate.NAND gate is designed by combining the AND and NOT gates. If the input of the NAND gate high, then the output of the gate will be low.The symbol and truth table of the NAND gate with two inputs is shown below.



Symbol

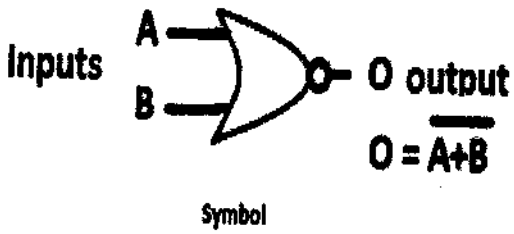
Inputs		Output
A	B	O
0	0	1
0	1	1
1	0	1
1	1	0

Truth table

NAND Gate and Its Truth Table

NOR Gate

The NOR gate is a digital logic gate with n inputs and one output, that performs the operation of the OR gate followed by the NOT gate. NOR gate is designed by combining the OR and NOT gate. When any one of the i/ps of the NOR gate is true, then the output of the NOR gate will be false. The symbol and truth table of the NOR gate with truth table is shown below.



Symbol

Inputs		Output
A	B	O
0	0	1
0	1	0
1	0	0
1	1	0

Truth table

c) **With suitable example explain Octal number system.**

The **octal** numeral system, or **oct** for short, is the base-8 number system, and uses the digits 0 to 7. Octal numerals can be made from binary numerals by grouping consecutive binary digits into groups of three (starting from the right). For example, the binary representation for decimal 74 is 1001010. Two zeroes can be added at the left: (00)1 001 010, corresponding the octal digits 1 1 2, yielding the octal representation 112.

In the decimal system each decimal place is a power of ten. For example:

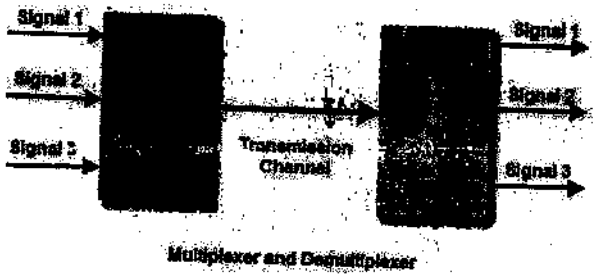
$74_{10} = 7 \times 10^1 + 4 \times 10^0$

In the octal system each place is a power of eight. For example:

$112_8 = 1 \times 8^2 + 1 \times 8^1 + 2 \times 8^0$

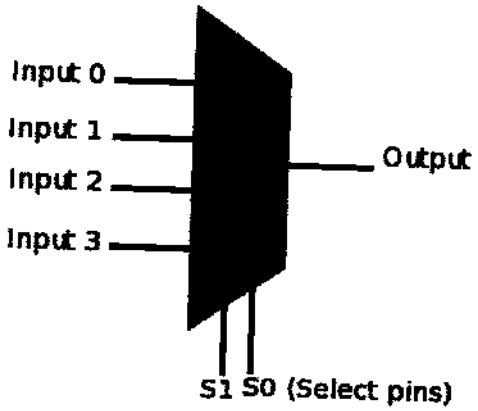
By performing the calculation above in the familiar decimal system we see why 112 in octal is equal to 64+8+2 = 74 in decimal.

d) **Compare multiplexer and demuliplexer.**



Multiplexer and Demultiplexer

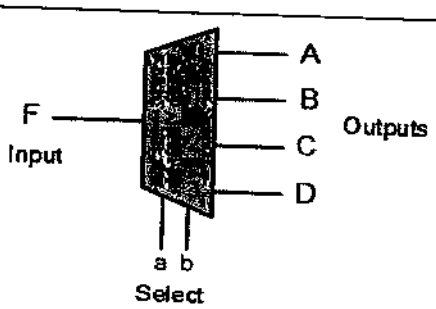
Multiplexer is a device that has multiple inputs and a single line output. The select lines determine which input is connected to the output, and also to increase the amount of data that can be sent over a network within certain time. It is also called a data selector.



Multiplexer

De-multiplexer is also a device with one input and multiple output lines. It is used to send a signal to one of the many devices. The main difference between a multiplexer and a de-multiplexer is that a multiplexer takes two or more signals and encodes them on a wire, whereas a de-multiplexer does reverse to what the multiplexer does.

Q. P. Code: 59462



De-multiplexer

e) **With the help of neat diagram explain Shift Register.**
(2 marks diagram, 3 marks explanation)

f) **Convert the binary number to decimal number.**

i) 100101

Steps of conversion

+

Ans:-37 **(1 marks)**

ii) 10001110

Steps of conversion

+

Ans:-142 **(2 marks)**

iii) 10110101

Steps of conversion

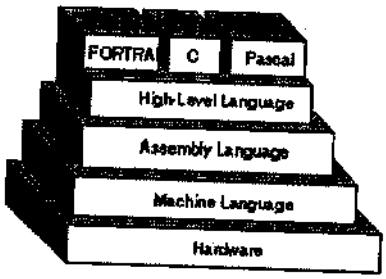
+

Ans:-181 **(2 marks)**

Q.3 **Attempt the following (Any THREE) (Each of 5Marks)** **(15M)**

a) **Compare machine language and assembly language.**

While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers. Programmers, therefore, use either a high-level programming language or an assembly language. An assembly language contains the same instructions as a machine language, but the instructions and variables have names instead of being just numbers.

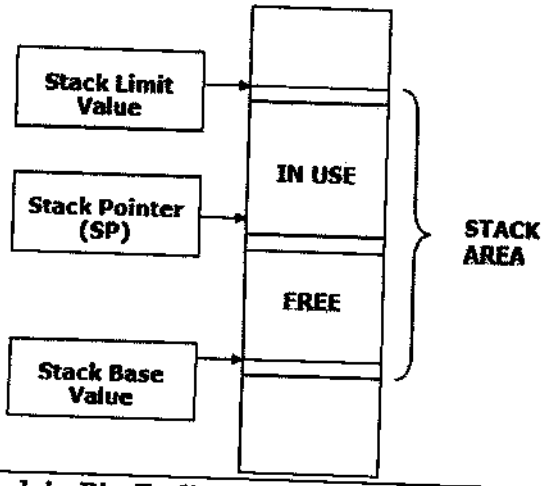


Programs written in high-level languages are translated into assembly language or machine language by a compiler. Assembly language programs are translated

Q. P. Code: 59462

		into machine language by a program called an assembler. Every CPU has its own unique machine language. Programs must be rewritten or recompiled, therefore, to run on different types of computers	
	b)	Explain following assembler directives:- a) INCLUDE **Explanation in one line ,syntax , example(1 marks)** b) ELSE **Explanation in one line ,syntax , example(1 marks)** c) RESET **Explanation in one line ,syntax , example(1 marks)** d) EQU **Explanation in one line ,syntax , example(1 marks)** e) ORG **Explanation in one line ,syntax , example(1 marks)**	
	c)	**Explain characteristics of RISC instruction set.** • Because of the small set of instructions of RISC, high-level language compilers can produce more efficient code. • RISC allows freedom of using the space on microprocessors because of its simplicity. • Instead of using Stack, many RISC processors use the registers for passing arguments and holding the local variables. • RISC functions uses only a few parameters, and the RISC processors cannot use the call instructions, and therefore, use a fixed length instructions which are easy to pipeline. • The speed of the operation can be maximized and the execution time can be minimized. • Very less number of instruction formats (less than four), a few number of instructions (around 150) and a few addressing modes (less than four) are needed.	
	d)	**With the help of neat diagram explain hardware implementation of Stack.** The stack can be used for various purposes like procedure call and return; evaluation of arithmetic expressions .Thus, its implementation depends on the purposes for which it is required. If the programmer intends to use the stack for various operations, then the instruction set will include stack operations, such as Push and Pop. Whereas, if the stack is to be used only by the processor, for purposes such as procedure call and return, then it is not required to include explicit stack-oriented instructions in the instruction set. Figure 1.8 illustrates that irrespective of the use of the stack, some set of memory locations are reserved to store the stack elements. These locations can either be reserved in terms of processor's registers or the physical memory locations.	

Figure 1.8: Typical Stack Organization



e) **Explain Big-Endian and Little-Endian Assignments.**

A load word or store word instruction uses only one memory address. The **lowest address** of the four bytes is used for the address of a block of four contiguous bytes.

How is a 32-bit pattern held in the four bytes of memory? There are 32 bits in the four bytes and 32 bits in the pattern, but a choice has to be made about which byte of memory gets what part of the pattern. There are two ways that computers commonly do this:

Big Endian Byte Order: The **most significant** byte (the "big end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order in the next three bytes in memory.

Little Endian Byte Order: The **least significant** byte (the "little end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order in the next three bytes in memory.

In these definitions, the data, a 32-bit pattern, is regarded as a 32-bit unsigned integer. The "most significant" byte is the one for the largest powers of two: 2^{31}, ..., 2^{24}. The "least significant" byte is the one for the smallest powers of two: 2^{7}, ..., 2^{0}.

For example, say that the 32-bit pattern 0x12345678 is stored at address 0x00400000. The most significant byte is 0x12; the least significant is 0x78.

Within a byte the order of the bits is the same for all computers (no matter how the bytes themselves are arranged).

f) What is function call? Explain its use in ISA.

Explanation of Function call(**2 marks**)

Syntax(**1 marks**) Coding (**1 marks**) Uses (**1 marks**)

Q.4 **Attempt the following (Any THREE) (Each of 5Marks)** **(15M)**

a) How data movement & manipulation operations performed using Data Path.

Explanation (**3 marks**) Diagram (**2 marks**)

b) List and explain different types of peripheral devices.

Q. P. Code: 59462

		List (1 marks) Explanation (4 marks)	
	c)	What is an interrupt? Give example. Explanation (3 marks) Example (2 marks)	
	d)	List and explain with neat diagram main hardware components of Processor. List (1 marks) Diagram (2 marks) Explanation (2 marks)	
	e)	Explain arithmetic, logic & Load instructions with example. Explanation (3 marks) Example (2 marks)	
	f)	Explain Direct Memory Access. **Direct memory access (DMA)** is a feature of computer systems that allows certain hardware subsystems to access main system memory (Random-access memory), independent of the central processing unit (CPU). Without DMA, when the CPU is using programmed input/output, it is typically fully occupied for the entire duration of the read or write operation, and is thus unavailable to perform other work. With DMA, the CPU first initiates the transfer, then it does other operations while the transfer is in progress, and it finally receives an interrupt from the DMA controller when the operation is done. This feature is useful at any time that the CPU cannot keep up with the rate of data transfer, or when the CPU needs to perform work while waiting for a relatively slow I/O data transfer. Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards and sound cards. DMA is also used for intra-chip data transfer in multi-core processors. Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without DMA channels. Similarly, a processing element inside a multi-core processor can transfer data to and from its local memory without occupying its processor time, allowing computation and data transfer to proceed in parallel.	
Q.5		**Attempt the following (Any THREE) (Each of 5Marks)**	
	a)	Explain NOR, Exclusive OR, Exclusive NOR gate with truth tables. Truth table (1 marks) Diagram (2 marks) Explanation (2 marks)	
	b)	Convert decimal number 106 to binary & octal form. **decimal number to binary** Steps of conversion (1 marks) Final Ans (1 marks) **decimal number to octal** Steps of conversion (2 marks) Final Ans (1 marks)	
	c)	List and explain deferent types of Registers. List (1 marks) Explanation (4 marks)	
	d)	With the help of neat diagram Stack frame. Diagram (2 marks) Explanation (3 marks)	
	e)	Explain S-R Flip Flop. Diagram (2 marks) Explanation (3 marks)	

****