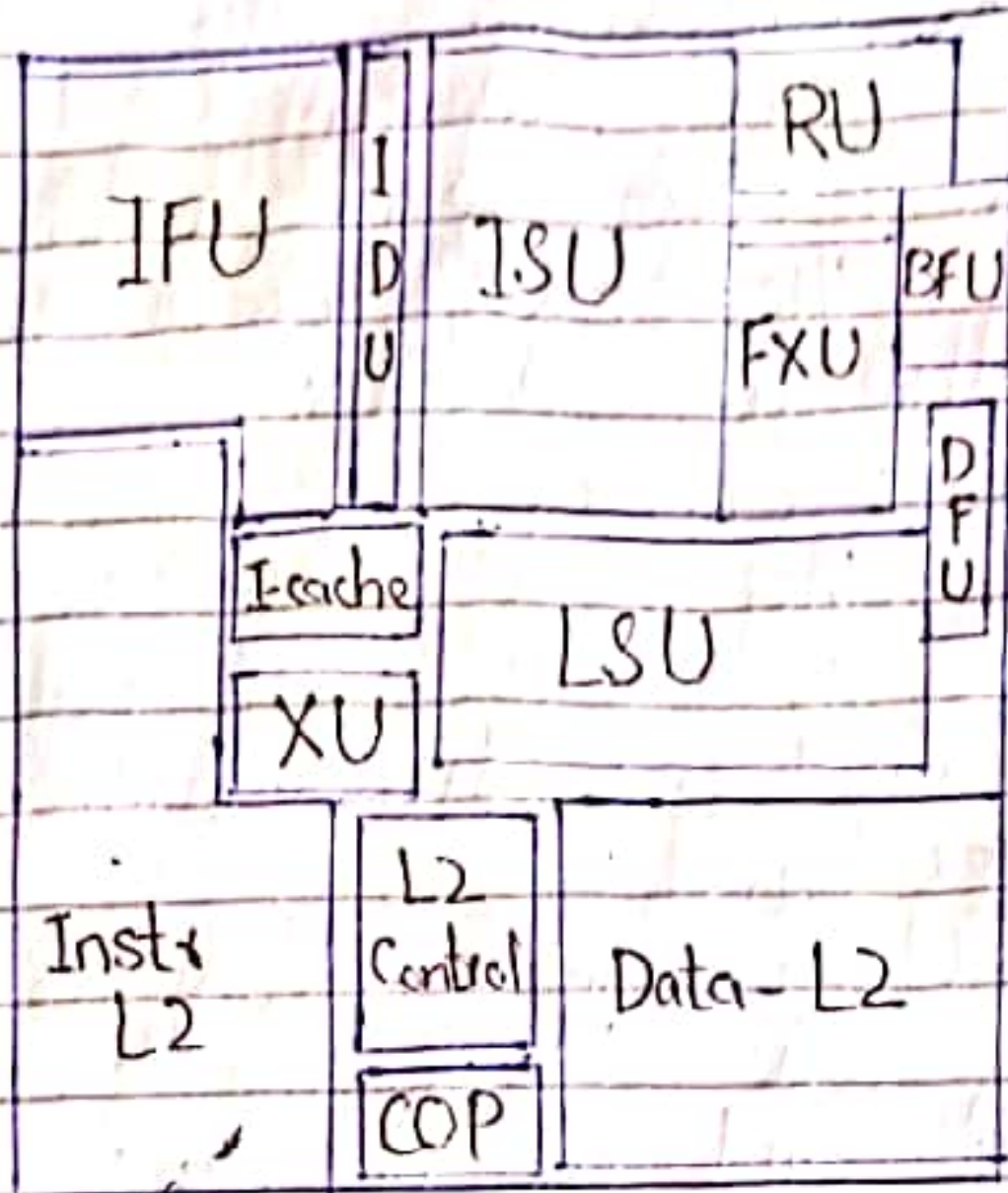


Computer Architecture

Q1:

Ans: (A)



Function of each Subarea:-

• ISU: (instruction sequence unit)

Determine the sequence in which instructions are executed in what is referred to as a superscalar architecture.

• IFU: (instruction fetch unit)

Logic for fetching instructions.

• IDU: (instruction decode unit)

The IDU is fed from the IFU buffers, and is responsible for the parsing and decoding of all z/Architecture operation codes.

• LSU: (Load-store unit)

It is responsible for handling all types of operand accesses of

all lengths, modes, and formats as defined in the z/Architecture.

• XU:- (translation unit)

This unit translates logical addresses from instructions into physical addresses in main memory.

• FXU:- (Fixed-point unit)

The FXU executes fixed-point arithmetic operations.

• BFU:- (binary floating-point unit)

The BPU handles all binary and hexadecimal floating-point operations, as well as fixed-point multiplication operations.

• DFU:- (decimal floating-point unit)

The DFU handles both fixed-point and floating-point operations on numbers that are stored as decimal digits.

• RU:- (recovery unit)

The RU keeps a copy of the complete state of the system that includes all registers, collects hardware fault signals etc.

• COP:- (dedicated co-processor)

The COP is responsible for the data compression and encryption function for each core.

• I-cache:-

This is a 64-KB L1 instruction cache, allowing IFU to prefetch instructions before they are needed.

• L2 Control:

This is a control logic that manages the traffic through the two L2 caches.

• Data-L2:

A 1-MB L2 data cache for all memory traffic other than instructions.

• Insts-L2:

A 1-MB L2 instruction cache.

Ans: (B) IAS Operation:

The IAS operates by repetitively performing an instruction cycle. Each instruction cycle consists of two subcycles.

1. Fetch Cycle:

During fetch cycle, the opcode of the next instruction is loaded into the IR and the address portion is loaded into the MAR. This instruction may be taken from the IBR, or it can be obtained from memory by loading a word into the MBR, and then down to the IBR, IR and MAR.

2. Execute Cycle:

Control circuitry interprets the opcode and executes the instruction by sending out the appropriate control signals to cause

data to be moved or an operation to be performed by the ALU.

Ans(C): Embedded System:-

The term embedded system refers to the use of electronics and software within a product, as opposed to a general-purpose computer, such as a laptop or desktop system. Today, many devices that use electric power have an embedded computing system.

Different embedded systems used in everyday life are;

Cell phones, digital cameras, video cameras, calculators, microwave ovens, home security systems, washing machines, lighting systems, printers etc.

Ans(D) Different desktop applications that require the great power of contemporary microprocessor based system are:

- Image processing
- Three-dimensional rendering
- Speech recognition
- Videoconferencing
- Multimedia authoring
- Voice and video annotation of files
- Simulation modeling

Ans: (E) The techniques used in contemporary processors to increase speed are following;

(*) Pipelining:

Pipelining enables a processor to work simultaneously on multiple instructions by performing a different phase for each of the multiple instructions at the same time.

(*) Branch prediction:

Branch prediction potentially increase the amount of work available for the processor to execute.

(*) Superscalar Execution:

This is the ability to issue more than one instruction in every processor clock cycle. In effect, multiple parallel pipelines are used.

(*) Data flow analysis:

The processor analyzes which instructions are dependent on each other's result, or data, to create an optimized schedule of instructions.

(*) Speculative execution:

This enables the processor to keep its execution engines as busy as possible by

executing instructions that are likely to be needed.

Ans.(F) The problems created due to increase in clock speed and logic density of the processor are:

• Power:

As the density of logic and the clock speed on a chip increase, so does the power density. The difficulty of dissipating the heat generated on high-density, high speed chips is becoming a serious design issue.

• RC delay:

The speed at which electrons can flow on a chip between transistors is limited by the resistance and capacitance of the metal wires connecting them. Specifically, delay increases as the RC product increases.

• Memory latency and throughput:

Memory access speed (latency) and transfer speed (throughput) lag processor speeds.

Ans.(G) Consider a program running on a single processor such that a fraction $(1-f)$ of the execution time involves code that is inherently sequential,

and a fraction f that involves codes that is infinitely parallelizable with no scheduling overhead. Let T be the total execution time of the program using a single processor. Then the Speedup using a parallel processor with N processors that fully exploits the parallel portion of the program is as follows:

$$\text{Speedup} = \frac{\text{Time to execute program on a single processor}}{\text{Time to execute program on } N \text{ parallel processors}}$$

$$\text{Speedup} = \frac{T(1-f) + Tf}{T(1-f) + \frac{Tf}{N}} = \frac{1}{(1-f) + \frac{f}{N}}$$

Ans(H) Multicore :-

- The use of multiple processors on the same chip, also referred to as multiple cores or multicore, provides the potential to increase performance without increasing the clock rate.
- If the software can support the effective use of multiple processors, then doubling the number of processors almost double performance.
- Two core chips were quickly followed by four-core chips, then 8, then 16,

and so on.

MIC:-

- The leap in performance as well as the challenges in developing software to exploit such a large number of cores has led to the introduction of new term called "many integrated core (MIC)".
- The multicore and MIC strategy involves a homogeneous collection of general-purpose processors on a single chip.

GPUs :- (GPGPU)

- A GPU is a core designed to perform parallel operations on graphics data. It is found on a plug-in graphics card.
- It is used to encode and render 2D and 3D graphics as well as process video.
- GPUs perform parallel operations on multiple sets of data, they are increasingly being used as vector processors for a variety of applications that require repetitive computations.

Ans: (1) QuickPath Interconnect (QPI) Protocol Layer:-
(*) In this layer, the packet is defined as the unit of transfer.

* The packet contents definition is standardized with some flexibility allowed to meet differing market segment requirements.

* One key function performed at this layer is a cache coherency protocol, which deals with making sure that main memory values held in multiple caches are consistent.

Ans: (J) Physical and Logical Architecture of PCIe :-

→ Root Complex:

• It is also called "chipset" or a "host bridge", which connects the processor and memory subsystem to the PCI Express switch fabric. Comprising one or more PCIe and PCIe switch devices.

• The root complex acts as a buffering device, to deal with difference in data rates between I/O controllers and memory and processors components.

PCIe links from the chipset may attach to the following kinds of devices that implements PCIe:

→ Switch:-

The switch manages multiple PCIe streams.

→ PCIe endpoint:-

An I/O device or controller that implements PCIe. Such as a Gigabit ethernet switch, a graphics, disk interface etc.

→ Legacy endpoint:-

Legacy endpoint category is intended for existing designs that have been migrated to PCI Express, and it allows legacy behaviors such as use of I/O space and locked transactions.

→ PCIe/PCI bridge:-

Allows older PCI devices to be connected to the PCIe-based systems.

Q2:

Ans: (A) Structural Components of Computer:-

There are four main structural components of computer:

1. Central Processing Unit (CPU):-

It controls the operation of the computer and performs its data processing functions; often simply referred to as "processor".

2. Main Memory:-

It stores data.

3. I/O:-

It moves data between the computer and its external environment.

4. System Interconnection:-

Some mechanism that provides for communication among CPU, main memory, and I/O. A common example of system interconnection is by means of a "system bus".

Ans:(B) The characteristics of computer family are as follows:

• Similar or identical instruction set:-

In some cases, the lower end of the family has an instruction set that is a subset of that of the top end of the family. This means that programs can move up but not down.

• Similar or identical operating system:-

The same basic operating system is available for all family members.

• Increasing Speed:-

The rate of instruction execution increases in going from lower to higher family members.

• Increasing number of I/O ports:-

The number of I/O ports increases in going from lower to

higher family members.

• Increasing memory size:

The size of main memory increases in going from lower to higher family members.

• Increasing Cost:

At a given point in time, the cost of a system increases in going from lower to higher family members.

Ans(C) Stored Program Computer:

A fundamental design approach first implemented in the IAS computer is known as the "Stored-program concept". This idea is usually attributed to the mathematician John von Neumann.

The first publication of the idea was in a 1945 proposal by von Neumann for a new computer, the EDVAC (Electronic Discrete Variable Computer).

In 1946, von Neumann and his colleagues began the design of a new stored-program computer, referred to as the "IAS computer", at the Princeton Institute for Advanced Studies.

It consists of:

→ A main memory, which stores both data and instructions.

→ An arithmetic and logic unit (ALU) capable of operating on binary data.

Ans: (D) Moore's Law:-

- The famous Moore's law, which was proposed by Gordon Moore, co-founder of Intel, in 1965.
- Moore observed that the number of transistors that could be put on a single chip was doubling every year, and correctly predicted that this pace would continue into the near future.
- The consequences of Moore's law are;
 1. The cost of computer logic and memory circuitry has fallen at a dramatic rate.
 2. The computer becomes smaller, making it more convenient to place in a variety of environments.
 3. There is a reduction in power requirements.
 4. The interconnections on the integrated circuit are much more reliable than solder connections.

Ans: (E) Instruction Cycle State Diagram:-

The states in instruction cycle state diagram are follows;

■ Instruction address calculation (IAC):-

Determine the address of the next instruction to be executed. Usually, this involves adding a fixed number to the address of the previous instruction.

■ Instruction fetch (IF):-

Read instruction from its memory location into the processor.

■ Instruction operation decoding (IOD):-

Analyze instruction to determine type of operation to be performed and operand(s) to be used.

■ Operand address calculation (OAC):-

If the operation involves reference to an operand in memory or available via I/O, then determine the address of the operand.

■ Operand fetch (OF):-

Fetch the operand from memory or read it in from I/O.

■ Data Operand (DO):-

Perform the operation indicated in the instruction.

■ Operand Store (OS):-

Write the result into memory or out to I/O.

Ans. (F) Classes of Interrupts:

(*) Program:

It is generated by some condition that occurs as a result of an instruction execution. Such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space.

(*) Timer:

It is generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.

(*) I/O:

It is generated by an I/O Controller, to signal normal completion of an operation, request service from the processor, or to signal a variety of error conditions.

(*) Hardware Failure:

It is generated by a failure such as power failure or memory parity error.

3. (G) Bus Interconnection Scheme:

The most common computer interconnection structures are based on the use of one or more system buses.

A system bus consists, typically of about fifty to hundreds of separate lines. The lines can be classified into three functional groups: data, address, and control lines.

→ Data Lines:

The data lines provide a path for moving data among system modules. These lines, collectively, are called the "data bus".

The data bus may consist of 32, 64, 128, or even more separate lines, the number of lines being referred to as "width of data bus".

→ Address Lines:

The address lines are used to designate the source or destination of the data on the data bus. The width of the address bus determines the maximum possible memory capacity of the system.

→ Control Lines:

The control lines are used to control the access to and the use of the data and address lines. Because the data and address lines are shared by all components, there must be a means of controlling their use. Typical control lines include:

Memory write, memory read, I/O write, I/O read, reset etc.

Q3:

Ans: (A) Computer Architecture:

Computer architecture refers to those attributes of a system visible to a programmer or, put another way, those attributes that have a direct impact on the logical execution of a program. A term that is often used interchangeably with computer architecture is "instruction set architecture (ISA)".

Computer Organization:

Computer organization refers to the operational units and their interconnections that realize the architectural specifications.

Examples of architectural attributes include the instruction set, the number of bits used to represent various data types (e.g. numbers, characters), I/O mechanisms, and techniques for addressing memory.

Ans: (B) CISC :-

The current x86 offering represents the result of decades of design effort on "Complex instruction set

Computers (CISCs).

The x86 incorporates the sophisticated design principles once found only on mainframes and supercomputers and serves as an excellent example of CISC design.

RISC:

An alternative approach to processor design is the "reduced instruction set" computers (RISC).

The ARM architecture is used in a wide variety of embedded systems and is one of the most powerful and best-designed RISC-based systems on the market. In this section and the next, we provide a brief overview of these two systems.

Ans(C) Microprocessor:-

A microprocessor chips included registers, an ALU, and some sort of control unit or instruction processing logic. As transistor density increased, it became possible to increase the complexity of the instruction set architecture, and ultimately to add memory and more than one processor. Microprocessor chips include multiple cores and substantial amount of cache memory.

Microcontroller:

A microcontroller is a single chip that contains the processor, non-volatile memory for the program (ROM), volatile memory for input and output (RAM), a clock, and an I/O control unit. The processor portion of the microcontroller has a much lower silicon area than other microprocessors and much higher energy efficiency.

Ans (D) Cortex-A:

The cortex-A and cortex-A50 are application processors, intended for mobile devices such as smartphones and eBook readers, as well as consumer devices such as digital TV and home gateways (e.g. DSL and cable Internet modems). These processors can run at higher clock frequency and support a memory management unit (MMU).

Cortex-R:

The cortex-R is designed to support real-time applications, in which the timing of events needs to be controlled with rapid response to events. They can run at a fairly high clock frequency and have very low response latency.

Cortex-M:

Cortex-M series processors have been developed primarily for the microcontroller domain where the need for fast, highly deterministic interrupt management is coupled with the desire for extremely low gate count and lowest possible power consumption.

Ans: (E) (*) In the interrupt cycle, the processor checks to see if any interrupts have occurred, indicated by the presence of an interrupt signal.

(*) If no interrupts are pending, the processor proceeds to the fetch cycle and fetches the next instruction of the current program.

Ans: (F) Disabled Interrupt:

A disabled interrupt simply means that the processor can and will ignore that interrupt request signal. If an interrupt occurs during this time, it generally remains pending and will be checked by the processor after the processor has enabled interrupts. When a user program is executing and an interrupt occurs, interrupts are disabled immediately.

Nested Interrupt:

Nested interrupt is to allow interrupt of higher priority to cause a lower-priority interrupt handler to be itself interrupted. A user program begins at $t = 0$. At $t = 10$, a printer interrupt occurs; user information is placed on the system "stack" and execution continues at the printer "interrupt service routine (ISR)". While this routine is still executing at $t = 15$, a "communication interrupt" occurs.

Ans(G) Programming in Hardware:

The "program" is in the form of hardware and is termed as "hardwired program."

Suppose we construct a general-purpose configuration of arithmetic and logic functions. This set of hardware will perform various functions on data depending on control signal applied to the hardware. In the original case of customized hardware, the system accepts data and produces results.

Programming in Software:

The new method of programming which is a sequence

of codes or instructions is called software programming.

In this method, programming is much easier. Instead of rewiring the hardware for each new program, all we need to do is provide a new sequence of codes. Each code is, in effect, an instruction, and part of the hardware interprets each instruction and generates control signals.

Q4: (a)
Ans: (A)

1: Here is a simple way to understand this problem:

Contents are divided up into two 8-bit instructions, LH and RH.

LH instruction = 010FA

opcode = 01
address = 0FA

RH instruction = 210FB

opcode = 21
address = 0FB

Since this is in hexadecimal form, you have to convert these numbers to binary form:

LH instruction:

01 = 00000001 = LOAD M(X)
M(X) refers to the memory address
Location OFA

The first 5 bits of 08A should
read - LOAD M(OFA)

RH instruction:

21 = 00100001 = STOR M(X)
M(X) refers to the memory address
Location OFB.

The second 5 bits of 08A should
read - STOR M(OFB)

Finally the assembly language code
for 08A 010FA210FB is

LOAD M(OFA)
STOR M(OFB)

2: Here is a simple way to
understand this problem:

Contents are divided up into two
5bit instructions, LH and RH.

LH instruction = 010FA

opcode = 01

address = OFA

RH instruction = 0F08D

OpCode = 0F
address = 08D

Now convert these numbers into binary form:

LH instruction:

01 = 00000001 = LOAD: M(x)
M(x) refers to memory address location 0FA

The first 5 bits of 08B should read - LOAD M(0FA)

RH instruction:

0F = 00001111 = JUMP + M(x, 0:19)
refers to memory address location 08D

The second 5 bits of 08B should read - JUMP + M(08D, 0:19)

Finally the assembly language codes for 08B 010FA0F08D is;

LOAD M(0FA)
JUMP + M(08D, 0:19)

3: Here is a simple way to understand this problem.

Contents are divided into two 8bits instructions, LH and RH.

LH instruction = 020FA

opcode = 02
address = 0FA

RH instruction = 210FB

opcode = 21
address = 0FB

Since this is in hexadecimal form, you have to convert the numbers to binary form:

LH instruction:

02 = 00000010 = LOAD - M(x)
M(x) refers to the memory address location 0FA.

The first 8bits of OBC should read - LOAD - M(0FA)

RH instruction:

21 = 00100001 = STOR M(x)
M(x) refers to the memory address location 0FB.

The second 8bits of OBC should read - STOR M(0FB)

Finally the assembly language code for 08C 020FA210FB is

LOAD - M(0FA)

STOR M(0FB)

- b)
1. In 08A address, the M(0FA) transfer to the accumulator and transfer contents of accumulator to memory location 0FB.
 2. In 08B address, the M(0FA) transfer to the accumulator and take next instruction from left half of M(08D).
 3. In 08C address, the -M(0FA) transfer to the accumulator and transfer contents of accumulator to memory location 0FB.

Q4.

Ans. (B) (x) In an IAS, a machine instruction will be stored with two parts: opcode and operand.

The memory address 2 can be load into the accumulator as follows;

Opcode: 00000001
Operand: 000000000010

(x) Two trips to memory for CPU needed to make to complete this instruction during the instruction cycle. CPU has to make access memory to fetch the instruction which contain the address of the data we want to load.

(C)
Ans: Effective CPI:

$$\text{CPI} = \frac{(1 \times 46000) + (2 \times 33000) + (2 \times 16000) + (2 \times 9000)}{100}$$

$$\text{CPI} = 162000/100$$

$$\boxed{\text{CPI} = 1620}$$

MIPS Rate:

$$\text{MIPS rate} = 60 \text{ MHz} / 1620 \times 10^6$$

$$\text{MIPS rate} = 60 \times 10^6 \text{ Hz} / 1620 \times 10^6$$

$$\text{MIPS rate} = 60 \text{ Hz} / 1620$$

$$\boxed{\text{MIPS rate} = 0.037}$$

Execution Time:

$$T = I_c / (\text{MIPS} \times 10^6)$$

$$T = 1040000 / (0.037 \times 10^6)$$

$$T = 1040000 / 37 \times 10^3$$

$$T = 2811 \times 10^{-3}$$

$$\boxed{T = 2.811 \text{ sec}}$$

Ans: (D) (a). Since we have the same instruction mix, that means the additional instructions for each task could be allocated appropriately between the instruction types. Therefore, the following

table be gotten:

Instruction Type	CPI	Instruction Mix
Arithmetic and logic	1	60%
Load/store with cache hit	2	18%
Branch	4	12%
Memory reference with cache miss	12	10%

The average CPI = $(1 \times 0.6) + (2 \times 0.18) + (4 \times 0.12) + (12 \times 0.1)$

$$\boxed{\text{CPI} = 2.64}$$

Therefore, the CPI has been increased since the time for memory access is also increased.

b. $\text{MIPS} = 400 / 2.64 = 152$

There is a corresponding drop in the MIPS rate.

c. The speedup factor equals to the ratio of the execution times.

The execution time is calculated as the following:

$$T_1 = I_c / (MIPS \times 10^6)$$

For one processor, $T_1 = (2 \times 10^6) / (178 \times 10^6)$

$$T_1 = 11 \text{ ms}$$

For 8 processors, each processor executes $1/8$ of the 2 million instructions plus the 25,000.

$$T_8 = \frac{2 \times 10^6 / 8 + 0.025}{157 \times 10^6}$$

$$T_8 = 1.8 \text{ ms}$$

Therefore we have

$$\text{Speedup} = \frac{\text{Time to execute program on a single processor}}{\text{Time to execute program on } N \text{ parallel processors}}$$

$$\text{Speedup} = 11 / 1.8$$

$$\text{Speedup} = 6.11$$

d. By depending on the information given, it is not obvious how to quantify this effect in Amdahl's equation. Therefore, if it is supposed that the fraction of code, which is

parallelizable, is $f = 1$, then Amdahl's law decreases to Speedup = $N = 8$. Therefore, the actual speedup is only about 75% of the theoretical speedup.

Ans (E) Six Steps:

(1). (a) The PC contains 300, the address of the first instruction. This value is loaded into the MAR.

(b) The value in location 300 (which is the instruction with the value 1940 in hexadecimal) is loaded into the MBR, and the PC is incremented. These two steps can be done in parallel.

(c) The value in the MBR is loaded into the IR.

(2). (a) The address portion of the IR (940) is loaded into the MAR.

(b) The value in location 940 is loaded into the MBR.

(c) The value in the MBR is loaded into the AC.

(3). (a) The value in the PC (301) is loaded into the MAR.

(b) The value in location 301 (which is the instruction with the value 5941) is loaded into the MBR, and the PC is incremented.

(4) The value in the MBR is loaded into the IR.

(4). (a) The address portion of the IR (941) is loaded into the MAR.

(b) The value in location 941 is loaded into the MBR.

(c) The old value of the AC and the value of location MBR are added and the result is stored in the AC.

(5). (a) The value in the PC (302) is loaded into the MAR.

(b) The value in location 302 (which is the instruction with the value 2941) is loaded into the MBR, and the PC is incremented.

(c) The value in the MBR is loaded into the IR.

(6). (a) The address portion of the IR (941) is loaded into the MAR.

(b) The value in the AC is loaded into the MBR.

(c) The value in the MBR is stored in location 941.

Ans(F) (a) $224 = 16 \text{ MBytes}$

(b) (1) If the local address bus is 32-bits, the whole address can be

transferred at once and decoded in memory. However, because the data bus is only 16 bits, it will require 2 cycles to fetch a 32-bit instruction or operand.

(2) The 16 bits of the address placed on the address bus can't access the whole memory. Thus a more complex memory interface control is needed to latch the first part of the address and then the second part.

(3) The program counter must be at least 24 bits. Typically, a 32-bit microprocessor will have a 32-bit external address bus and a 32-bit program counter, unless on-chip segment registers are used that may work with a smaller program counter. If the instruction register is to contain the whole instruction, it will have to be 32-bit long; if it will contain only the op code (called the op code register) then it will have to be 8 bits long.

Ans (7)

Q4

Ans (G)

A bus cycle takes $0.25\mu\text{s}$, so a memory cycle takes $1\mu\text{s}$. If both operands are even aligned, it takes $2\mu\text{s}$ to fetch the two operands. If one is odd aligned, the time required is $3\mu\text{s}$. If both are odd aligned, the time required is $4\mu\text{s}$.

{The END}