:: ASSIGNMENT # 3 ::

ID: 11533

Name: Ashir Ali Khan

Subject: Computer Architecture

Teacher: Sir Muhammad Amin

Iqra National University

ASSIGNMENT : 3 :-

QUESTION : 1 :-

Paet : A :-

Answer :-

Disabling Interrupts :-

Processor has the ability to and will ignore the specific interrupts. Those interrupts remain pending and will be checked after the processor has enabled interrupts.

Interrupt service routine (ISR) :-

Priorities assigned to the different types of interrupt. Interrupts service routines with higher priorities can interrupt ones with lower priority, in which case the ISR with the lower priority is put on the stack until that ISR is completed.

X———X

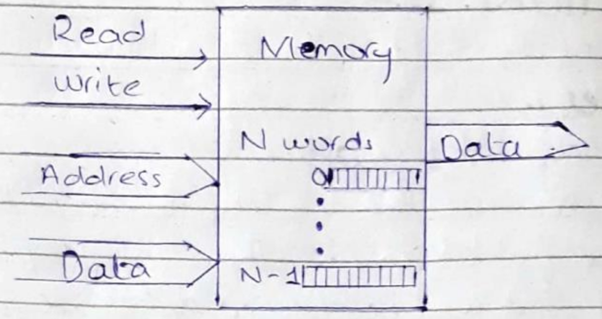
Part : B :-

Answer :-

Memory :-

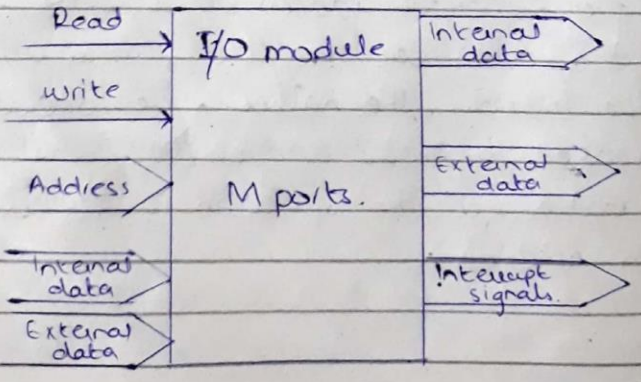
Typically, a memory module will consist of N words of equal length. Each word is assigned a unique numerical address (0, 1, c, N-1)

A word of data can be read from or written into the memory. The nature of the operation is indicated by read and write control signals. The location for the operation is specified by an address.



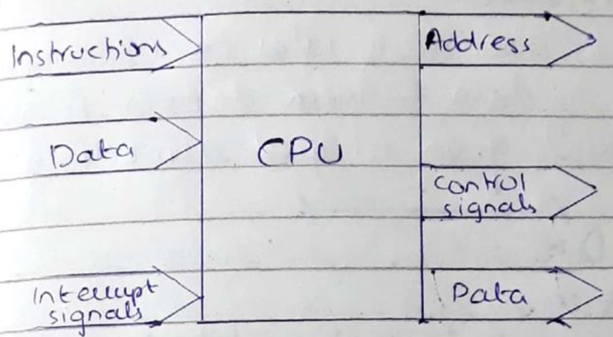
I/O module:-

From an internals (to the computer system) point of view, I/O is functionally similar to memory. There are two operations; read and write. Further, an I/O module may control more than one external device. We can refer to each of the interfaces to an external device as a port and give each a unique address (e.g, 0, 1, c, M-1). In addition, there are external data paths for the input and output of data with an external device. Finally, an I/O module may be able to send interrupts signals to the processor.



Processor :-

The processor reads in instructions and data, writes out data after processing, and uses control signals to control the overall operation of the system. ~~B~~ It also receives interrupt signals.



Part : C :-

Answer :-

Quick path Interconnect (QPI) protocol layers :-

QPI is defined as a four-layer protocol architecture,[3] encompassing the following layers.

Physical :-

Consist of the actual wires carrying the signals, as well as circuitry and logic to support ancillary features required in the transmission and receipt of the 1s and 0s. The unit of transfer at the physical layer is 20 bits, which is called a phit (physical unit).

Link :-

Responsible for reliable transmission and flow control. The link layer's unit of transfer is an 80-bit Flit (Flow control unit).

Routing:-

Provides the framework for directing packets through the fabric.

Protocol:-

The high-level set of rules for exchanging packets of data between devices. A packet is comprised of an integral numbers of Flits.

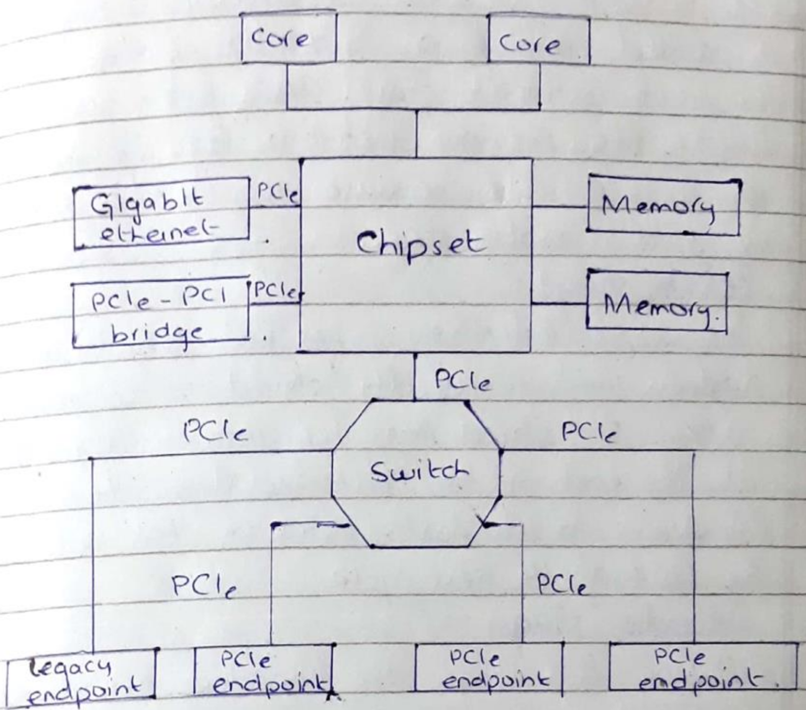
x———x

Part : D:-

Answer :-

Physical and logical architecture of PCIe:-

A root complex device, also referred to as a chipset or a host bridge, 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 devices, to deal with difference in data rates between I/O controllers and memory and processor components. The root complex also translates between PCIe transactio formats and the processor and memory signal and control requirements. The chipset will typically support multiple PCIe ports, some of which attach directly to a PCIe device, and one or more that attach to a switch that manages multiple PCIe streams.



X———————X

Question : 2 :-

Part : A :-

Answer:

Instruction Cycle :-

The instruction is the cycle that the central processing unit (CPU) follows from boot-up until the computer has shut down in order to process instructions. It is composed of three main stages: the fetch stage, the decode stage, and the execute stage. The instruction cycle is executed sequentially, each instruction being processed before the next one is started. In most modern CPUs, the instruction cycles are instead executed concurrently, and often in parallel, through as instruction pipeline: the next instruction starts being processed before the previous instruction has finished, which is possible because the cycle is broken up into seperate steps.

Fetch stage:

The next instruction is fetched from the memory address that is currently stored in the program counter and stored into the instruction register. At the end of the fetch operation, the PC points to the next instruction that will be read at the next cycle.

Decode stage :-

During this stage, the encoded instruction presented in the instruction register is interpreted by decoder.

Read the effective address :-

In the case of a memory instruction (direct or indirect), the execution phase will be during the next clock pulse. If the instruction has an indirect address, the effective address is read from main memory, and any required data is fetched from main memory to be processed and then placed in to data register (clock pulse: T_3). If the instruction is direct, nothing is done during this clock pulse. If this is an I/O instruction or a register instruction, the operation is performed during the clock pulse.

Execute stage :-

The control unit of the CPU passes the decoded information as a sequence of control signals to the relevant function units of CPU to perform the actions required by the instruction, such as reading values from registers, passing them to the ALU to perform mathematical or logic functions on them, and writing the result back to a register. If the ALU is involved, it sends a condition signal back to the CU. The result generated by the operation is stored in the main memory or sent to an output device. Based on the feedback from the ALU, the PC may be updated to a different address from which the next instruction will be fetched.

x——————x

Part: B

Answer:-

⇒ 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. for example, if each instruction is 16 bits long and memory is organized into 16-bit words, then add 1 to the previous address. If instead, memory is organized as individually addressable 8-bit bytes, then add 2 to the previous address

⇒ Instruction Fetch (IF): Read instruction from its memory location into the processor.

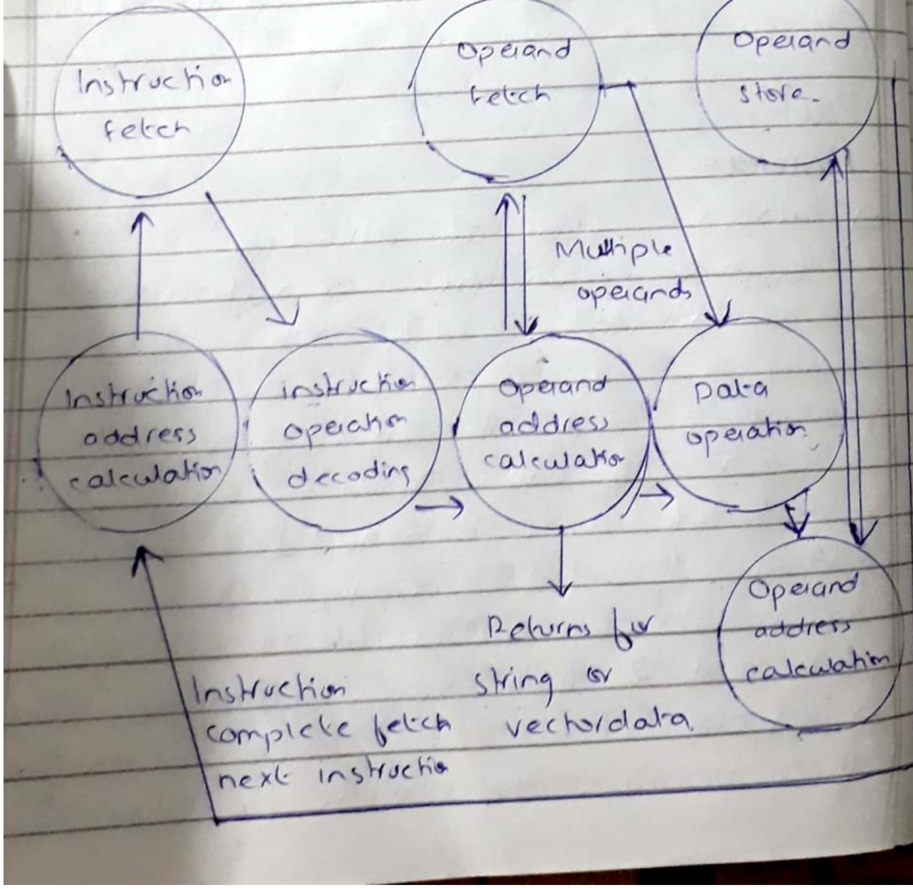
⇒ Instruction operation decoding (Iod): Analyze instruction to determine type of operation to be performed and operand(s) to be used.

⇒ Operand address calculation (oac): If the operation involve 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 operation (dO): Perform the operation indicated in the instruction.

→ Operand store (OS): Write the result into memory or out to I/O.


Instruction fetch
Operand fetch
Operand store
Multiple operands
Instruction address calculation
instruction operation decoding
Operand address calculation
Data operation
Operand address calculation
Returns for string or vector data
Instruction complete fetch next instruction

Part: C :-

Answer:-

classes of interrupts..

Program:

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:-

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

I/O:-

Generated by an I/O controller, to signal normal completion of a operation or to signal a variety of error conditions.

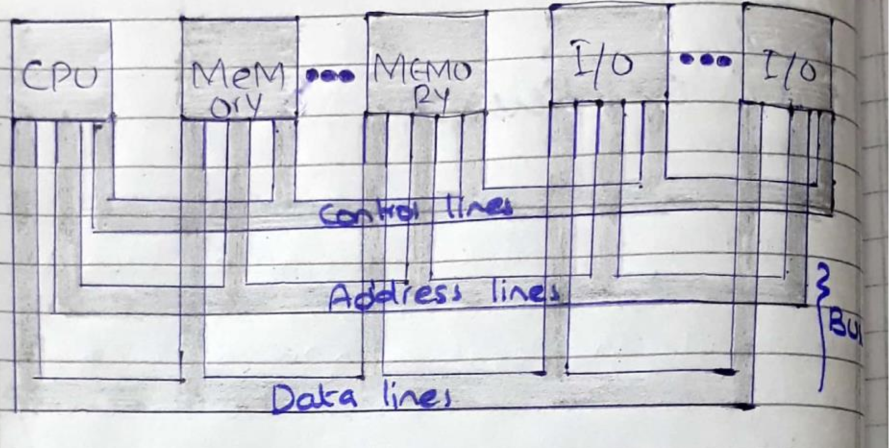
Generated by a failure such as power failure or memory parity error.

X———————X

Part: D:-

Answer:-

Bus Interconnection Scheme:-



Bus Interconnection SCHEME

A bus is a communication pathway connecting two or more devices. A key characteristic of a bus is that it is a shared transmission medium. Multiple devices connect to the bus and a signal transmitted by any one device is available for reception by all other devices attached to the bus. If two devices transmit during the same time period, their signals all overlap and become gabled. Thus, only one device at a time can successfully transmit. Typically, a bus consist of multiple communicate pathways, or lines. Each line is capable of transmitting signals representing binary 1 and binary 0. An 8-bit unit of data can be transmitted over eight bus lines. A bus that connect major computer components (processor, memory, I/O) is called a system bus.

Data lines :-

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

Address lines :-

The address lines are used to designate the source or destination of the data on the data bus. for example, on an 8-bit address bus, address 01111111 and below might reference locations in a memory module (module 0) with 128 words of memory, and address 10000000 and above refers to devices attached to an I/O module (module 1).

Control lines :-

The control lines are used to control the access to and the use of the data and address lines. Control signals transmit both command and timing information among system modules. Timing signals indicate the validity of data and address information. Command signals specify ~~the~~ operations to be performed. Typical control lines include:

- **Memory Write :-**
  Causes data on the bus to be written into the addressed location.
- **Memory read :-**
  causes data from the addressed location to be placed on the bus

- I/O write :-
  Causes data on the bus to be output to the addressed I/O port.
- I/O read :-
  Causes data from the addressed I/O port to be placed on the bus.
- Transfer ACK :-
  Indicates that data have been accepted from or placed on the bus.
- Bus request :-
  Indicates that a module needs to gain control of the bus.
- Bus grant :-
  Indicates that a requesting module has been granted control of the bus.
- Interrupt request :-
  Indicates that an interrupt is pending.
- Interrupt ACK :-
  Acknowledges that the pending interrupt has been recognized.
- Clock :-
  Is used to synchronize operations.
- Reset :-
  Initializes all modules.

The operation of the bus is as follows. If one module wishes to send data to another, it must do two things. To obtain the use of the bus, and to transfer data via the bus. If one module wishes to request data from another module, it must obtain the use of the bus, and transfer a request to the other module over the appropriate control and address lines. It must then wait for that second module to send the data.

x———x

Question: 3 :-

Part : A:-

Answer:-

Differentiate b/w Programming in hardware and programming in software :-

Programming in hardware	Programming in Software.
1) Programming in hardware means configuring a small set of basic logic components specifically for a particular computation.	1) Programming in software means supplying a specific set of control signals to a a general-purpose hardware,
2) Hardware programming languages are concurrent in nature and executed a piece of code in parallel	2) A software programming language are sequential in nature and executed a piece of code sequentially.
3) Hardware programming languages are used to find the timing Delay of the circuit	3) Software languages cannot be used for to find the Timing Delay of the circuit.
4) Some of the examples of HDL (Hardware Description languages)	4) Some examples of software languages are C, C++.

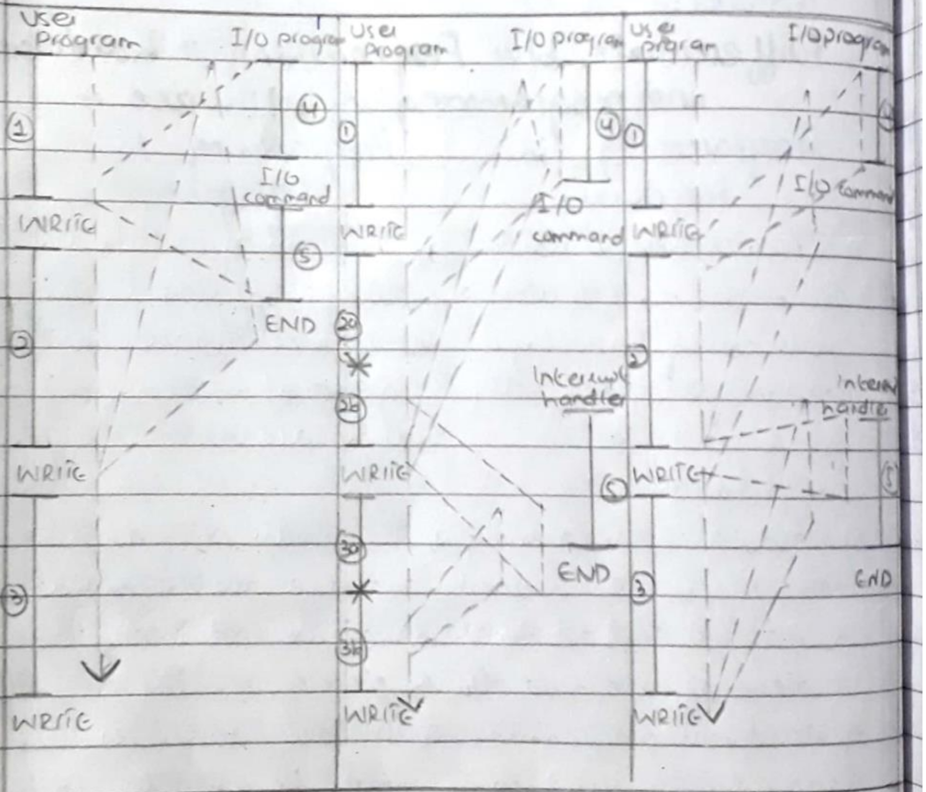
which are used as hardware language are VHDL, Verilog, System Verilog.

x————————x

Part, B:-

Answer:-

Program flow control:-



(a) No interrupt (b) interrupt; short I/O wait. (c) Interrupt; long I/O wait

1, 2 and 3 - code segments refer to sequences of instruction that do not involve I/O.

The WRITE calls are calls to an I/O program that is a system utility and that will perform the actual I/O operation.

The I/O consists of three sections:

4-A sequence of instructions which prepare for the operation.

I/O command - the actual I/O command.

5-A sequence of instruction which complete the operation.

x———x

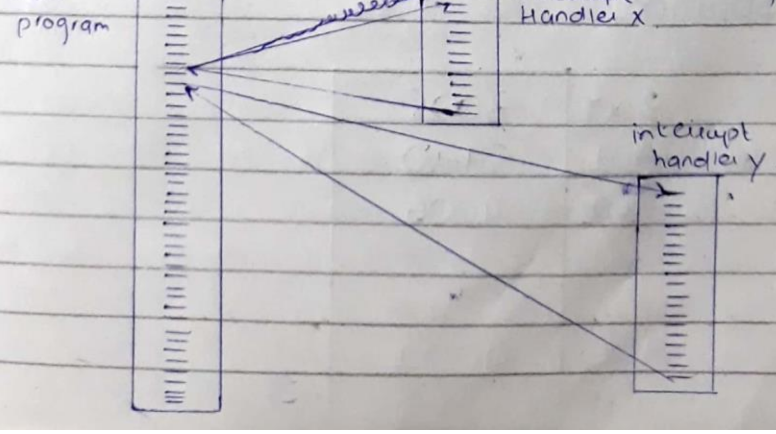
Part : C:-

Answer:-

Disabled Interrupt Processing and nested interrupt processing:-

Disabled Interrupt processing:-

- ⇒ Handle and service individual interrupts squentially.
- ⇒ High interrupt latency
- ⇒ Relatively easy to implement and debug.
- ⇒ Not suitable for complex embedded system



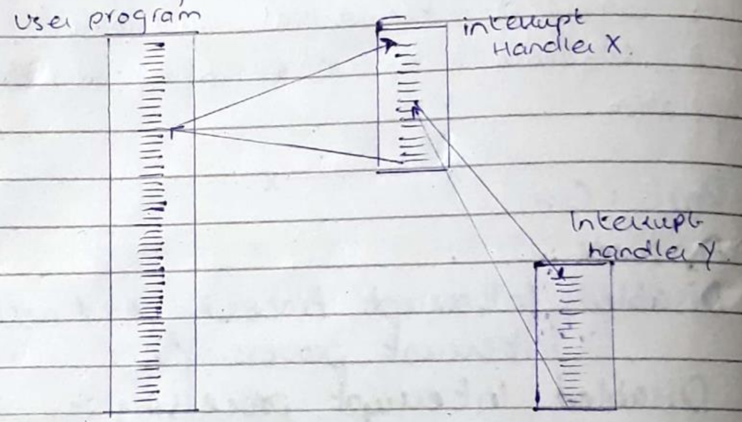
Nested Interrupt processing :-

⇒ Handle multiple interrupts without a priority assignment.

⇒ Medium or high interrupt latency

⇒ Enables interrupts before the servicing of an individuals interrupt is complete.

⇒ No prioritization, so low priority interrupts can block higher priority interrupts



Question : 4 :-

Part : A :-

Answer :-

Solution :-

300	3005
301	5940
302	7006
.	
.	
940	0002
941	

We will assume that the memory as the previous table :
300 : 3005; 301 : 5940; 302 : 7006.
Therefore, the steps will be as the following

Step 1 : 3005 → IR
Step 2 : 3 → AC
Step 3 : 5940 → IR.
Step 4 : 3+2 = 5 → AC
Step 5 : 7006 → IR
Step 6 : AC → Device 6.

X———X

Part : B :-

Answer :-

Step : 1 :-

a) The PC contains 300, the address of the first instruction. This value is loaded in to 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.

Step : 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

Step: 3 :-

a) The value in the PC (301) is loaded in to 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.

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

Step: 4 :-

a) The addless 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.

Step: 5 :-

a) The value in the PC (302) is loaded in to 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.

Step: 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.

Part: C :-

Answer:-

(A) what is the maximum directly addressable memory capacity (in bytes)?

$$2^{(32-8)} = 2^{(24)}$$

$= 16,777,216$ bytes $= 16$MB (8 bits = 1 byte for he opcode)

(B) Discuss the impact on the system speed if the microprocessor bus has:

(b.1) 32-bit local address bus and a 16-bits local data bus, or.

A 32-bit local address bus and a 16-bit local data bus. Instruction and data transfer would take three bus cycle each, one for the address and two for the data. Since if the address bus is 32 bits, the whole address can be transfered to memory at once and decoded there; however, since the data bus is only 16 bits, it will require 2 bus cycles to fetch the 32-bit instruction or operand.

(b.2) 16-bit local address bus and a 16-bit local data bus.

A 16-bit local address bus and a 16-bit local data bus. Instruction and data transfer would take four bus cycles each, two for the address and two for the data. Therefore, that will have the processor perform two transmission in order to send to memory the whole 32-bit address ; this will require more complex memory interface control to latch the two halves of the address before it performs an access to it. In addition to this two-steps address issue, since the data bus is also 16-bits, the micrpprocessor will need 2 bus cycles to fetch the 32-bit instruction or operand.

(c) **How many bits are needed for the program counter and the instruction registers**

For the PC needs 24 bits (24-bit addresses), and for the IR needs 32-bits (32-bit addresses).

x———x

Part : D :-

Answer :-

Since, minimum bus cycle duration = 4 clock
and bus clock.
= 8 MHz.

Then, maximum bus cycle rate = 8M/4
= 2M/s

Data transfered per bus cycle = 16 bit = 2 bytes

Data transfe ratepe second = bus cycle rate x data per bus cycle = 2M x 2
= 4 Mbytes/sec.

To increase its performance:

- By doubling the frequency, it may mean adopting a new chip manufacturing technology (assuming each instruction will have the same number of clock cycles).

- By doubling the external data bus, that means wider (may be never) or chip-data bus drivers/latches and modifications to the bus control logic.

Therefore, in the first situation the speed of the memory chips will need to double, not to slow down the microprocessor. Regarding the second situation, the word length of the memory will must double to the able to send/receive 32-bit quantities.

X———X

Part: E:-

Answer:-

(A):- Through a single bus cycle, the 8-bit microprocessor transfer one byte while the 16-bit microprocessor transfer two bytes. The 16-bit microprocessor has twice the data transfer rate

(B) By assuming that we have to perform 50 transfers of operand and instructions which 25 are one byte long and 25 are two bytes long. The 8-bit microprocessor needs 25 + (2 x 25) = 75 bus cycles for the transf. The 16-bit microprocessor needs 25 + 25 = 50 bus cycles. Therefore, the data transfer rates differ by a factor of 1.5.

X———X

Part : F :-

Answer :-

Time taken to fetch the operands: First we need to find the time taken to fetch one operand from one memory location, frequency and bus cycle.

Given clock rate = 4MHz

$$\text{Frequency} = \frac{1}{\text{Clock Rate}}$$

$$= \frac{1}{4\text{MHz}}$$

$$= 0.25\text{Ms}$$

Therefore, frequency or bus cycle is 0.5Ms. Then, memory cycle will take 0.5Ms x 4 = 1Ms. Hence the fetch one operand from memory 1Ms is required

By applying this "If an odd-aligned word is referenced, two memory cycle, each consisting of four bus cycles, are required to transfer the word" from the Question, so we will have 3 cases.

First case :- is if both operands are even-aligned, so the time required is 1x2 = 2Ms to fetch both operands.

Second case:- is if both operands are odd-aligned, so the time required is 1x4 = 4Ms to fetch both operand.

Thirdcase :- is if only one is odd-aligned. so the time required is 1x3 = 3Ms to fetch both operands.

Part. G :-

Solution :-

By assuming that we have a mix of 100 instructions and operands. from the question:

20% of the operand and instructions are 32-bits long, so it is 20 32-bit.

40% of the operand and instructions are 16-bits long, so it is 40 16-bit.

40% of the operand and instructions are only 8bits = 1byte long, so it is 40bytes.

The number of bus cycles needed for the 16-bit microprocessor will equal to:

(20x2)+40+40 = 120 bus cycles.

The number of bus cycles needed for the 32-bit microprocessor will equal to:

20+40+40 = 100 bus cycles.

By calculating the improvement achieved with 32-bit microprocessor to the 16-bit microprocessor will equal to 20/120 = 16.6%.