Name: Manahil Khan
ID: 14525

Subject: Computer architecture
Semester: 4th
program: BS(CS)
instructor: Sir M. Amir
Date: 26/April/20

Q NO#1

Give answer to each of the following.

Part (a)

A: Draw the IBM zenterprize EC12 core layout & explain the function of each sub-area

ANS:- ISU (instruction Sequence unit):-
Determine the sequence in which instruction are executed in what is reffered to as a superscalar architecture.

* IFU (instruction fetch unit):-
logic for fetching instruction.

* IDU (instruction decode unit):-
The IDU is fed from the Ifu buffers & is responsible for the parsing & decoding of all z/architecture operation code.

* LSU (Load Store unit):-
its responsible for handling all types of operand accesses of all length, modes, & formats as defined in all z/architecture.

* XU (translation unit):-
This unit translate logical addresses from instruction into physical addresses in main memory. it contain TLB used to speed up memory access.

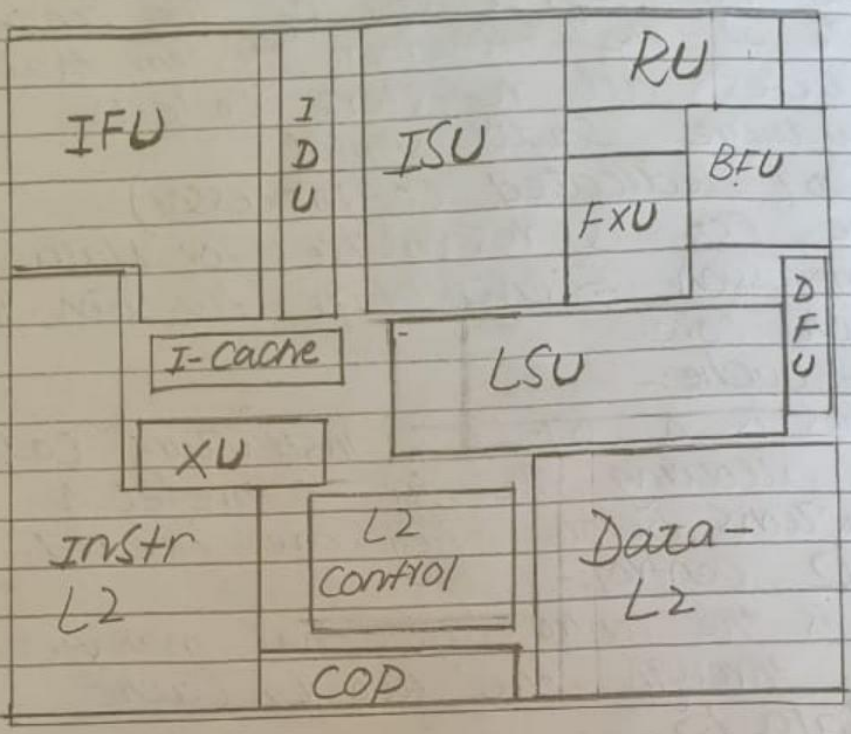
* FXU (fixed point unit):-
The fxu execute fixed-point arthimatic operation.

* BFU (binary floating unit):-
The Bfu handles all binary & hexadecimal floating operations, as well as fixed-point multiplication operations.

* DFU (decimal floating point unit):-
The dfu handles both fixed point & floating point operation on numbers that are stored as decimal digits.

* RU (recovery unit):-
The RU keep a copy of the complete state of the diagram system that includes all registers collects hardware fault signal.
* Cop (dedicated co-processor):-
The Cop is responsible for data compression & encryption function for each core.
I - Cache:-
This is a 64-KB L1 instruction code cache allowing the ifu to prefecch instructions before they are needed.
* L2 control:-
This is the control logic that manages the traffic through the two L2 cache.
* Data L2
A - 1MB L2 data cache for all memory traffic other than instructions.
* instr - L2:-
A1 - MB L2 instruction cache.

Diagram on the next page


IFU
IDU
ISU
RU
BFU
FXU
DFU
I-Cache
LSU
XU
Instr L2
L2 control
Data-L2
COP

Part (b)

b:- Discuss the IAS operation in detail.

The IAS operates by respectively performing an instruction cycle. Each instruction cycle consist of two sub-cycle.

* **Fetch cycle:-**

The opcode of the next instruction is loaded into the IR & 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 & then down to the IBR, IR & MAR.

* **Execute cycle:-**

The control circuitry interprets the opcode & executes the instruction by sending out the appropriate control signals to cause data to be moved or a operation to be performed by the ALU.

part(C)

C:- What is embeded System? list different embeded System used in everyday life.

The term embeded System refers to the use of electronics & Software within a product as oppose to a general purpose Computer, Such as laptop or desktop System.

* Embeded System are computing Systems but they can range from having no user interface to complex graphical user interface.

* **Examples:-**

Examples of embeded Systems in daily life are cell phones, digital camera, video camera, calculators, microwave ovens, washing machines, lighting systems, printers, various authomatic System etc.

part (D)

D:- Discuss different desktop application that revuise the great power of contemporary microprocessor-based system.

Ans) Different desktop application that revuise the great power of contemporary microprocessor based system are.

* image processing
* Three-dimensional rendering
* Speech recoginations
* video conferencing
* Multimedia authoring
* voice & video annotation of files
* Simulation modeling.

part (E)

E:- Discuss the technique used in contemporary processor to increase speed?

The technique used in contemporary processor to increase speed are following.

* **pipeling:-**

pipeling enables a processor to work simultaneusly on multiple instruction 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 avaiable for process to execute.

* Superscalar execution:-
This is the ability to issue more than one instruction in every processor clock cycle an effects multiple parallel pipline are used.

* Data flow analysis:-
The processor analyzes which instruction are dependent on each other's results or data, to create an optomized schedule of instructions.

* Speculative execution:-
This enables the processor to keep its execution engines as busy as possible by executing instruction that are likely to be needed.

Data (f)

f:. Discuss the problem created due to increase in clock speed & logic density of processor.

problem created due to increase in clock speed & logic density of the processor are.

* power
As the density of logic & the clock speed on a chip increase, so the power density increases & also dissipated the heat.

* RC delay:-
The speed at which electron can flow on a chip between transistors

is limited by the resistance & capacitence of the metal wires connecting them specifically, delay increases as the RC product increases.

* Memory latency:-
Memory access speed (latency) & transfer speed (throughput) lag processor speeds.

Part (G)

g:- Discuss the speedup of a program using multiple processors compared to a single processor using Amdahl's law?

The speedup using the parallel processor with N processor that fully exploits the parallel portion of the program as follows.

Speedup = Time to execute program on a single processor / Time to execute program on N parallel processors.

Part (H)

H: Discuss the multicore MIC & GPGPU in detail.

* Multicore:-
The use of multi processors on the same chip provides the potential to increase performance without increasing the clock rate.

Strategy is to use two simpler processor on the chip rather than one more complex processor.
* With two processor larger cache are justified.
* As cache became larger it made performance sense to create two & then three levels of cache on a chip.

* MIC

Leap in performance as well as the challenges in developing software to exploit such a larger number of cores.
* The multicore & MIC strategy involve a homogenous collection of general purpose processor on a single chip.

GPUS:-

* Core design to perform parallel operation on graphics data.
* Traditionally found a plug in graphics card. it's used to encode & render 2D & 3D graphics as well as process video.
* Used a vector processor for a variety of application that require computation.

PART (I)

Q- Discuss the quick path interconnect (QPI) protocols layer.

* QPI protocol layer:-

In this layer, the packet is defined as the unit of transfer. One key function performed at this level is a cache coherency protocol, which deals with making sure that main memory value held in multiple cache are consistent. A typical data packet payupload is a block of data being sent to or from a cache.

Part (J)

J: Discuss the physical & logical architecture of PCIe in detail.

A root complex device, also referred to as a Chipset or a host bridge, connects the processors & memory subsystem to the PCI Express Switch ~~compt~~ fabric comprising one or more PCIe & PCIe switch devices.

* PCIe links from the chipset may attach to the following kind of device that implent PCIe:

* Switch:-
The switch manage multiple PCIe streams.

* PCIe endpoint:-
An I/O device or controller that implement PCIe such as Gigabit ethernet switch a graphics or video controller disk interface, or a communication controller.

* Legacy endpoint:-
Legacy endpoint category is intended for existing designs that have been migrated to PCI Express, it allow legacy behaviours such as use of I/O space & locked transactions.

* PCIe/PCI bridge:- Allows older PCI devices to be connected to PCIe based systems.

Q NO #2

Write a detail note on each of the following.

part (A)

A:- Main structual component of a computer.

There are four main structural components.

* cpu:- Controls the operation & performs its data processing function, often simply referred to as processor
* Main memory
  Stores data
* I/O
  Moves data between the computer & its external environment
* System interconnection:-
  Some mechanism that provides for communication among cpu main memory & I/O

part (b)

B:- Key characteristics of a planned computer family:-

The characteristic of a family are as follow

* Similalar 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 mean that program can move up but not down.
* Similalar or identical operating system:-
The same basic operating system is avaiable for all family members.
* increasing speed:-
The rate of instruction execution increases in going from lower to higher family members.
* increasing numbers 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 is going from lower to higher family members.
* increasing cost:-
At a given point in a time the cost of a system increases in going from lower to higher family members.

part (C)

C:- Stored program computer:-

* A fundamental design approch first implemented in the IAS computer is known as the stored-program concept. This idea is usually attributed to the mathematician john von Neuman The first publication of the idea was in a 1945 proposal by von Neuman for a new computer the EDVAC (Electronic discrete variable computer). In 1946, von Neuman & his colleagues began the design of a new stored program computer, as reffered to ~~IAS~~ the IAS computer, at the princeton institute for advanced studies. it consist of.
* A main memory which store both data & instructions.
* An aathimatic & logic unit (ALU) capable of operating on binary data.

part (D)

D:- Moore's law:-

The famous moore's law which was propounded by gorden moore cofounder of intel in 1965 (MOORES). Moore observed that the number of transistors that could be put on a single chip was doubling every year. The place slowed to a doubling every 18 months in the 1970's but has

Sustained that rate ever since.
The consequence of moore's law are profound.
1:- The cost of computer logic & memory circuitry has fallen at a dramatic rate.
2:- Because logic & memory elements are placed closer together on more densely packed chips, the electrical path length is shortned, increasing operating speed.
3:- The computer become smaller, making it more convenient to place in a variety of enviroments.
4:- There is a reduction in power requirements.
5:- With more circuitry on each chip, there are fewer interchip connection.

Part (E)

E:- instruction cycle state diagram:-
The state in instruction cycle can be described as follow.

* instruction address calculation (IAC):-
Determine the address of the next instruction to be executed, usually this involves adding a fix number the address of the previous instruction.
* instruction fetch (If)
Read instruction from its memory location into the processor.
* instruction operating decoding:-
Analyze instruction to determine

the type of operation to be performed & 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 form I/O.
* Data operation (DO):- perform the operation indicated in the instruction.
* Operand store (OS):- write the results into memory or out o/p.

part (f)

f:- Classes of interupts:-
its generated by some condition that occur as a result of an instruction execution such as arthimatic overflow, division by zero, attempt to execute an illegal machine instruction or reference outside a user allowed memory space.

* Timer

its generated by a timer within a processor. this allows the operating system to perform certain function

on a regular basis.

* I/O:-
it's 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 condition.

Part (g)

g:- Bus interconnection scheme:-
The most common computer interconnection structure are based on the use of one or more system buses.
* The lines can be classified into three functional groups, data, address, & control lines.
* Data lines:-
The data lines provide a path for moving data among system module these lines collectively are called data bus.
* Address lines:-
The address lines are used to designate the source of destination of the data on the data bus. The width of the address bus determines the maximum possible memory capcity of the system.
* Control lines:-
The control lines are used to control the access to & the used of the

data & address lines. Because the address & data lines are shared by all components, there must be a means of controling their use typical control lines include memory write, memory read, I/O write, I/O read transfer ACK, bus request, bus grant, interrupt request interrupt ACK, clock & request.

Q NO #3

~~Writing~~

Differentiate each of the following:

part (a)

A:- computer organization & 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:-

refers to the operational units & their interconnection that realize the architectural

Sepcification e.g example of architectural attributes include the instruction set the number of bits used to represent various data types (e.g numbers, characters) I/O mechanism & technique for addressing memory

part (b)

b:- Risc & CISC

The current x86 offering represents the result of decades of design effort on complex instruction set computers (CISC). The x86 incorporates the sophisticated design principal once found only on mainframes & SuperComputers & ~~as~~ Serves as an excellent example of CISC design.

* An alternative approch to processor design is reduce instruction set computer (RISC) The ARM architecture is used in a wide variety of embeded system & is one of the most powerful & best design RISC based system on the market.

part (c)

c:- Microprocessor & microcontroller:-

* Microprocessor:-

chip include registers an ALU & some sort of control unit or instruction processing logic. A

transistors increased it became possible to increased it ultimately to add memory & more than one processor.

* Microcontroller:-

is a single chip that contain the processor non volatile memory for input & output (RAM) a clock an I/O control unit. the processor portion of the microcontroller has a much lower silicon area than other microprocessor & much higher energy efficiency.

part (D)

D:- Cortex-A, Cortex-R, Cortex-M

CORtex-A

. The cortex A & Cortex-A50 are application processor intended for mobile device such as smartphones & eBook readers, as well as consumer device such as digital tv & home gateways (e.g DSL & cable internet modems) these processor run at higher clock frequency (over 1 GHz) & support a memory mangement unit (MMU).

* CORtex-R

The Cortex-R is designed to support real time application in which the timing of events need to be controlled with rapid response to events. They can run at a fairly high clock frequency

(e.g 200MHz to 800MHz) & have very low response latency.

* Cortex-M Series processor have been developed primarily for the microcontroller domain where the need for fast, highly deterministic interrupt mangment is coupled with the desire for extremly low count & lowest possible power consumption.

part (E)

E:- program flow of control without interupt & with interupt.

~~A disabled interupt simply means that the processor can & will ignore that interupt request ;. Sorry SH~~

* In the interupt cycle, the processor checks to see if any interupts have occured, indicated by the presence of an interupt signal.
* If no interupts are pending, the processor proceeds to the fetch cycle & fetches the next instruction of the current program.

part (f)

f:- Disabled interupt & nested interrupt processing.

A disabled interupt simply means that the processor can & will ignore that interupt request signal. if an interupt occur during this time it generally remain pending & will be checked by the processor after the processor has enabled interupts.

* A nested interrupts is to allow an interrupts 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 interrupts occurs, user information is placed on the system stack & execution continues at the printer interrupt service routine (ISR). while this routine is still executing at t = 15 a communication interrupt occurs.

---

part (G)

G. programing in hardware & programing in software

* programing in hardware:-

The program is in the form of hardware & term a hardware program. Suppose we ~~term~~ a ~~hardware~~ we construct a general purpose configuration of arthimatic & logic function, this set of hardware will perform various function on data

depending on control signals applied to the hardware. In orginal case of customized hardware the system accept data & produce result.

* Programming in Software:-
This new method of programing which is a sequence of codes or instruction is called Software. In this method programing is much easier, instead of rewiring the hardware for each new program all we need to do is provide a new sequence of code. each code is in effect an instruction & part of hardware interprets each instruction & generate control signal.

Q NO#4

Solve each of the following

Part(a)

A:- Given the memory contents of the IAS computer show below.

Address	contenents
08A	010FA210FB
08B	010FA0F08D
08C	020FA210FB

a) Show the assembly language code for the program, starting at address 08A.

1:- Here is a simple way to understand this problem:
Contents are divided up into two 5-bit instruction LH & RH.
LH instruction = 010FA
Opcode = 01
Address = 0FA
RH instruction = 210FB
Opcode = 21
Address = 0FB
Since this in hexadecimal form you have to convert the number to binary form: (Use the IAS instruction set)
LH instruction:
01 = 00000001 = Load M(x)
M(x) refer to the memory address location 0FA.
The first 5 bits of 08A should read - load M(0FA)
RH instruction:
21 = 00100001 = STOR M(x)
M(x) refer to the memory address location 0FB
The second 5 bits of 08A should read - STOR M(0FB)
Finally the assembly language code for 08A 010FA210FB is
LOAD M(0FA)
STOR M(0FB)
2:- Here is a simply way to understand this problem.

Contents are divided up into two 5 Bits instruction LH & RH.
LH instruction = 010FA
opcode = 01
address = 0FA
RH instruction = 0F08D
opcode = 0F
address = 08D
Since this is in hexadecimal from you have to convert the number to binary form: (use the IAS instruction set)
LH instruction:
01 = 00000001 = LOAD M(a)
M(a) refers to the memory address location 0FA
The first 5 bits of 08B should read - LOAD M(0FA)
RH instruction:
0F = 00001111 = jump+M(08D, 0:19) refers to the memory address location 08D.
The second 5 bits of 08B should read - jump+M (08D, 0:19)
Finally the assembly language code for 08B 010FA0F08D is
LOAD M(0FA)
Jump+M(08D, 0:19)
3 Hex is a Simple way to understand this problem:
contents are divided up into two 5 bit instructions LH & RH

LH instruction = 020FA
opcode = 02
address = 0FA
RH instruction = 210FB
opcode = 21
address = 0FB
Since this in ~~form~~ hexa-decimal form you have to convert the numbers to binary form: (Use the IAS instruction set)
LH instruction:
02 = 00000010 = LOAD - M(x)
M(x) refers to the memory address location 0FA
The first 5 bits of 08C should read - LOAD - M(0FA)
RH instruction:
21 = 00100001 = STOR M(x)
M(x) refers to the memory address location 0FB
The second 5 bits of 08C should read - STORM(0FB)
Finally the assembly language code for 08C 020FA210FB is
LOAD - M(0FA)
STOR M(0FB)

b) Explain what this's program does?
1: In 08A address, the M(0FA) transfer to the accumulator & transfer contents of accumulator to memory location 0FB.
2: In 08B address, the M(0FA) transfer to the accumulator & take next instruction from left half of M(08D).
3: In 08C address, the -M(0FA) transfer to

the accumulator & tranfer contents of accumalator to memory location OFB.

Part (b)

(a) OP code = 00000001
operand = 0000000000010

(b) In the begining, the cpu have to fetch the instruction from the memory, Then the instruction will include the address of the data which is required to load. Through the execution time, the memory will be accessed in that time to load the data contents which is located at that address for a total of two trips to memory.

Part (c)

C:- A benchmark program is run on a 60MHz processor. The executed program consist of 104,000 instruction execution, with the instruction mix & clock cycle count given below. Determine the effective CPI MIPS rate & execution time for this program.

Instruction type	instruction count	cycle per instruction
integar arthimatic	46,000	1
Data transfer	33,000	2
Floating point	16,000	2
control transfer	9000	2

* Effective CPI:-

$CPI = (1*46000)+(2*33000)+(2*16000)+(2*9000)/100$

$CPI = 162000/100$

$CPI = 1620$

* MIPS rate

MIPS rate = 60MHz/1620*106

MIPS rate = 60*106/1620*106

MIPS rate = 60/1620

MIPS rate = 0.037

* Execution time:

$T = IC/(MIPS * 106)$

$T = 104000/(0.37 * 106)$

$T = 104000/37*103$

$T = 2811 * 10-3$

$T = 2.811$ sec

ANS

part (D)

D:- Consider the example in Section 2.5 for the calculation of average CPI & MIPS rate, which yieded the result of CPI = 2.24 & MIPS rate = 178. Now assume that the program can be executed in eight parallel tasks or threads with roughly equal numbers of instructions executed in each task. Execution is on an 8-core system with each core system (processor) having the same performance as the single processor orginally used. Coordination & synchronization between the parts add an extra 25,000 instruction execution to each task. Assume the same instruction mix as in the example for each task, but increase the CPI for memory reference with cache miss to 12 cycles due to contention for memory.

a) Determine the average CPI.
b) Determine the corresponding MIPS rate.
c) Calculate the Speedup factor.
d) Compare the actual Speedup factor with the theoratical Speedup factor determined by Amdahl's law.

ANS a) Since we have the same instruction mix, that means the additional instruction for each task could be allocated appropriately between

the instruction types. Therefore the following table be gotten

instruction type	CPI	instruction Mix
Arithmatic & logic	1	60%
load/store with cache hit	2	18%
Branch	4	12%
Memory reference with cache miss	12	10%

The average CPI = (1x0.6) + (2x0.18) + (4x0.12) + (12 x 0.1) = 2.64. Therefore the cpi has been increased since the time for money access is also increased.

b) MIPS - 400/2.64 - 152. There is a corresponding drop in the mips rate.

c) The Speedup factors equals to the ratio of the execution times. The execution time is calculated as the following.

$$T = IC / (MIPS * 106)$$

For the one processor, $T_1 = (2 * 106) / (178 * 106) = 11$ ms.

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

$T8 = 2*106 \div 8 + 0.025 * 106 / 152 * 106$

$T8 = 1.8ms$

Therefor we have

Speedup= Time to execute program on a single processor/time to execute program on N parallel processor.

Speedup = 11/1.8

Speedup = 6.11

* By depending on the information given. its not obvious how to quantity this effect in Amdahl's equation. Therefor its supposed that the fraction of code which is parallable is f=1 then Amdahl's decreases to Speedup = N = 8. Therefore the actual Speedup is only about 75% of the theoritical Speedup.

---

part (E)

E: The program execution of fig 01 is described in the text using steps. Expand this discription to show the use of MAR & MBR.

The pc contain 300, the address of the first instruction. This value is loaded into the MAR. The value in location 300 (which is the instruction with the value 1940 in hexadeeimal) is loaded into the MBR & the pc is incremented the two steps can be done in parallel c. The value in the MBR is loaded into the IR.

2:- 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:- 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 & the pc is incremented C. The value in the MBR is loaded into IR.

4:- The address portion of IR (941) is loaded into the MAR. B. The value in location 941 is loaded into MBR. C. the old value of ac & value of location MBR are added & the result is stored in the ac.

5:- The value in the pc (302) is loaded into the MAR. b. the value in location 302 (which is the instruction with the value 2491) is loaded into the MBR & the pc is incremented C. The value in the MBR is loaded into the IR.

6:- The address portion of the IR (941) is loaded into MAR. b. The value in the ac is loaded into MBR. C. The value in the MBR is stored in location 941.

part (f)

f:- Consider a hypothetical 32-bit microprosser having 32 bit instruction composed of two fields. the first byte contain the opcode & the remainder the immdiate operand address

a) What is the maximum directly addressable memory capcity (in bytes)?

b) Discuss impact on the system speed if the microprocessor bds has:

1: 32 bit local address bus & 16 bit local data bus or

2 16 bit local address bus & a 16 bit local data bus.

c How many bits are needed for the program counter & the instruction register?

ANS: A: 2^{24} = 16 MBytes

b: (1) if the address bus is 32 bits the whole addressed can be transferred at once & decode in the memory. However, because the data bus is only 16-bits, it will reqiuire 2 cycle to fetch a 32-bit instruction or oprend.

2) The 16 bit of 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 address & then second part

C:- The program counter must be at

least 24 bits. Typically a 32-bit microprocessor will have a 32 bit external address bus & a 32 bit program counter. unless on chip segment register 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-bits long. if it will contain only the op code (called the op code register) then it will have to be 8 bits long.

---

part (G)

g:- The intel 8086 is a 16-bit processor similalar in many ways to the 8-bit 8088. The 8086 uses a 16-bit bus that can transfer 2 byte at a time, provided that then lower order byte has an even address, However, the 8086 allow both even & odd aligned word operands. if an odd aligned word is referenced, two memory cycle, each consisting of four bus cycle are required to transfer the word. Consider an instruction on the 8086 that involves two 16-bit operands. How long does it take to fetch the operands? Give the range of possible answer. assume a clocking rate of 4MHz & no

Wait states.

A: A bus cycle take 0.25 us so a memory cycle take 1 us. if both operand are even aligned it take 2 us to fetch the two operands. if one is odd aligned, the time rerquired is 3 us. if both are odd-aligned the time rerquired is 4 us.