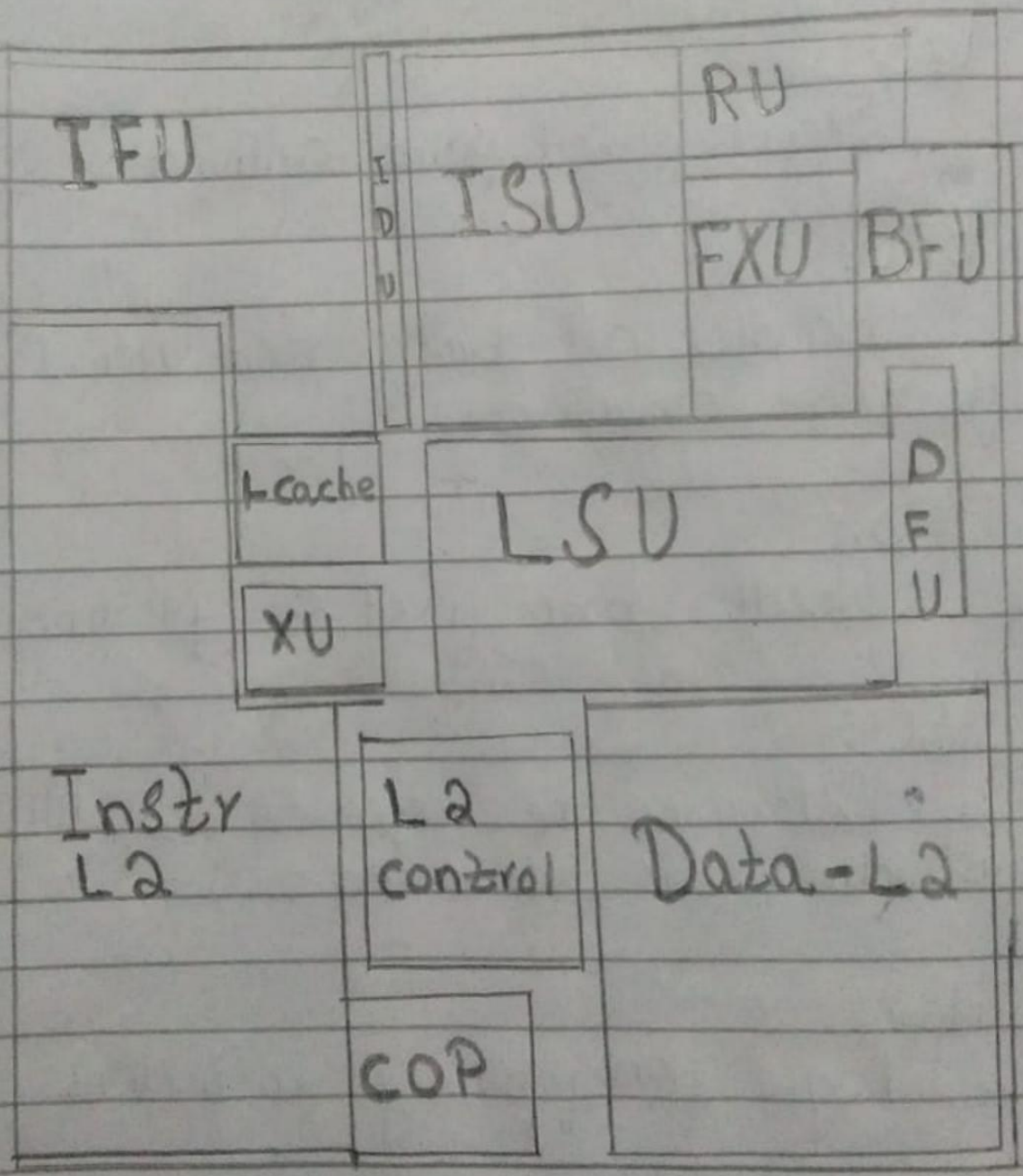
Name Farkhanda Jabeen.

Id 14725

Subject Computer Architecture

Semester. Bs(cs) 4th



IDU:-

IDU is fed from the IFU buffers

LSU:-
Responsible for handling all types of operand.

~~X~~U:-
Translate logical Address into physical Addresses in main memory.

FXU:-
Execute fixed point arithmetic operation.

BFU:-
Handles all binary and hexadecimal floating operations.

DFU:-
Handles both fixed & floating operations.

ISU:-
Determine the sequence in which istruction are executed.

IFU:-
Logic for fetching instructions.

RU:-
Keep a copy of the complete state of System.

Cop:-
Responsible for Data Compression & encryption.

I-Cache
64-KB L1 instruction code.

L2 Control:-
Manages the trafic through L2 Cache.

Data L2:-
A1-MB L2 data Cache for all memory trafic.

Instr-L2
A1-MB L2 instruction Cache.

Part (b)

IAS Operation:-

The IAS operates by repetitively performing an instruction cycle.

Each instruction cycle consist of the two subcycles.

i) Fetch cycle
ii) Execute cycle

Fetch cycle:-

During fetch cycle the op code of the next instruction is loaded into the IR & the address portion is loaded into the MAR.

Execute cycle:-

Control circuitry interpret the op code & executes the instruction by sending out the appropriate control signals ~~the~~ to cause data to be moved.

Part (c)

Part (C)

Embedded System:-

The embedded system is a Computer System - a combination of computer processor, computer memory, and Input/ output peripheral.

Examples:-

* Digital cameras.
* Cell phone
* Computer.

Part (D)

Desktop Applications:

* Image processing
* Multimedia authoring.
* Simulation modeling
* Video conferencing
* Videoice and Video annotation of files.
* Three dimentional rendering.

Part (E)

Techniques.

The techniques used in contemporary processors to increase speeds are following.

* Pipelining
* Branch Prediction
* Data flow analysis
* Speculative execution.
* Superscalar execution.

Part (F)
Problems due Increase in Clock Speed:-
following are the problems.

Power:-
When density of logic and clock speed on a chip increase, the power density increases also dissipated the heart.

RC delay:-
The speed of electron at which electrons can flow b/w transistors is limited by resistance of the metal wires, Specifically delay increases at the RC product increases.

Memory latency:-
Memory access speed (latency) and transfer speed (throughout) lag processor speeds.

Part (G)

Amdahl's Law.

The speedup using paralled processor with N processors that fully exploits the paralled portion of the programs as follows.

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

$$= T(1-f) + TF(T(1-f) + TF/N = 1(1-f) + f/N.$$

Part (H)

Mullicore, MIC and GPUs

Mullicore :-

* The use of multiple processor on the same chip provides the potential to increase performance without increasing the clock rate.

* With two processors larger caches are justified.

* As caches became larger it made performance sense to create two and then three levels of cache on a chip.

MIC :-

Leap in a performance as well as the challenges in developing software to exploit such a large number of cores

* The multicore and MIC strategy involves a homogenous collection of general purpose processors on a single chip.

GPUs :-

* Core designed to perform parallel operations on graphics data.

* Traditionally found on a plug in graphics card, it is used to encode and render 2D & 3D graphics as well as process video.

* Used as vector processors for a variety of applications that require repetitive computations.

Part (I)

Quickpath interconnect (QPI) protocol layers:-

In this layer the packet is defined as the unit of transfer, one key function performed at this level is cache coherency protocol, which deals with making sure that main memory values held in multiple cache are consistant.

Part (J)

Physical & Logical architecture of PCIe:-

A root complex device, also refered to as the chip set or a host bridge, connects the processor and the memory subsystem to the PCI express switch - fabric comprising one or more PCIe and PCI E switch devices.

Write a detailed note on each of the following.

A. Main structural components of computer:-

There are four main structural components.

Similar identical instruction set:-

* CPU:-

Control the operation of the computer & performs its data processing functions, often simply reffered to as processor.

* Main memory:-

Stores data.

* Input/output:-

Moves the data between the computer & its external movement.

* System Interconnection:-

Some mechanism that provides for communication among CPU, main memory & I/O.

b- Key characteristic of a planned computer family:

* 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. It means program can move up.

* 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 lover to higher family members.

* Increasing memory size:-
The memory size increase from lower to higher.

* Increasing cost:-
At a given point in a time the cost of a system increases in going from lower to higher family members.

c. Stored program canter:

A fundamental design approach first implemented in the IAS computer is known as the first stored program concept. This idea is usually attributed to the mathematician john von neumann.

In 1946, von neuman & his collaques began the design of a new stored program computer. which consists.

* A main memory which stores both data & instructions.
* An arithmetic and logic unit (ALU) capable of operating on binary data.

d. moore's Law:-

The famous moore's law 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 1970s but has sustained that rate ever since.

* The consequence of moore's law are profound.

* There is a reduction in power requirment
* With more circuitary on each chip there are fewer interchip connections.
* The computer become smaller, making it more convenient to place in a variety of environments.

e. Instruction Cycle State diagram:-

The states in instruction cycle can be described as follow.

* Instruction address Calculation (IAC)

Determine the address of the next instruction to be executed.

* Instruction fetch (If)

Read instruction from its memory location into the processor.

* Instruction operating decoding:-

Determine the type of operation to be performed & operand(s) to be used.

* Operand fetch (of):-

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

* Data operation:-

Perform the operation indicated in the instruction.

* Operand Store (os)

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

f. Classes of interupts:-

It is generated by some condition that occurs as a result of an instruction execution such as automatic overflow, division by zero attempt to execute an illegal machine instruction.

* Timer:-

This allows the operating system to perform certain function on a regular basis.

* I/O:-

Its 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.

G Bus Interconnection Scheme:-

A system bus consists, typically of from about fifty to hundreds of separate lines. The lines canbe classified into three functional groups data, address & control lines.

* Data lines:-

The data lines proride a path for moving data among system module called data bus.

* Address lines :-
The address lines are used to designate the source or destination of the data on the data bus.
* Control lines :-
The control lines are used to control the access to & the used of the data & address lines. Because the address & the data lines are shared by all components.
Control lines include :-
Memory write, I/O write, Bus request, Bus grant, Clock & reset etc -

Differentiate each of the following.

A. Computer organization and Computer architecture.

	Computer architecture	Computer organization.
i)	Architecture describes what the computer does.	Organization describe how it does it.
ii)	Computer architecture deals with function behaviour of computer system.	Computer organization deals with the structural relationship.
iii)	Architecture indicates its hardware.	Organization indicates its performance.

b RISC and CISC

	RISC	CISC
i)	The original microprocessor ISA.	Redesigned ISA that emerge in the early 1980s.
ii)	Instructions can take several clock cycles.	Single cycle instructions.
iii)	More efficient use of RAM.	Heavy use of RAM.
iv)	Large number instructions.	Small number of fixed-length instructions
v)	Composed Adressing modes.	Limited Addressing modes.

C Microprocessor and Micro controller.

	Microprocessor	Microprocessor.
i)	Microprocessor is a heart of Computer System.	Micro controller is a heart of embedded system.
ii)	Since memory and I/O has to be connected externally the circuit becomes large.	Since memory and I/O are present internally, the Circuit is small.
iii)	Cost of the entire system increases.	Cost of entire System is low.
iv)	Used in personal Computers.	Used in washing machine, MP3 players.

D) Corten-A, corten-R, corten-M

D.	Corten-A	Corten-R	Corten-M
Architecture profile	ARMv7-A ARMv8-A	ARMv7-R ARMv8-R	ARMv7-M ARMv8-M
Instruction set	32-bit/64bit	32-bit	32-bit
Interrupts	Software managed	Determined softw- are managed	Hardware managed
Operating system support	Rich os/RTOS	RTOS	RTOS
Example processors	Corten-A7 Corten-A35	Corten-R8 Corten-R52	Corten-M7 Corten-M33

e. Program flow of Control without Interrupt and with interrupt.

Interrupt	Without Interrupt
In the interrupt cycle the processor Check to see if anyone interrupts have occured, indicate by the presence of an interupt signal.	If no interrupts are pending, the processor proceeds to the fetch cycle and fetches the next instruction of the current program.

f. Dissabled interrupt and nested interrupt processing.

Dissabled Interrupt	Nested 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.	To allow an interupt of higher priority to cause a lower-priority interupt handler to be it-self interrupted. A user program being at t=0 At t=10 a printer interrupts occurs, user information is placed on the system stack and execution continues at a printer interrupt service routine while this routine is still executing at t = 15 a Communication interrupts occurs

9. Programming in hardware & programming in Software.

Hardware	Software
* The 'program' is in the form of hardware and is termed a hardware program	The new method of programming which is a sequence of codes or instructions is called software programming.
* 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 signals applied to the hardware. In the original use of customized hardware the system accepts data & produces results.	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 hardware interepts each instruction and part of hardware interepts each instruction and generates control signals.

6/4(A)

Q4) 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
adress = 0FA
RH instruction = 210FB
opcode = 21
adress = 0FB
Since this in hexadecimal form have to convert the number to binary form. (Use the IAS instruction set)
LH instruction:
01 = 00000001 = Load M(n)
M(n) refer to the memory adress location 0FA
The first 5 Bits of 08A should read - load M(0FA)
RH instruction:
21 = 00100001 = STOR M(n)
M(n) refer to the memory adress 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 - 01 0FA
Opcode = 01
adress = 0FA
RH instruction = 0F0 8D
Opcode = 0F
adress = 08D
Since this is in renadechmal from you have to convert the number to binary form: (use the IAS instruction set)
LH instruction:
01 = 00000001 = Load M(n)
M(n) refers to the memory adress location 08D
The first 5 bits of 08B should read - jump + m (08D.0:19)
Finally the assembly language for 08B 01 0FA 0F 08D is Load M (0FA)
Jump + m (08D.0:19)

3 Here is a simple way to understand this problem:
Contents are divided up into 5 bits instruction LH & RH
LH instruction = 020 FA
opcode = 02
adress = OFA
RH instruction = 210FB
opcode = 21
adress = OFB
Since this is in hexadecimal form you have to convert the numbers to binary form.
(use the IAS instruction set)
LH instruction.
02 = 00000010 . load - M(n)
M(n) refers to the memory adress location OFA
The first 5 Bits of 08C should read - load - M(OFA)
RH instruction.
21 = 00100 001 = STOR M(n)
M(n) refers to the memory adress location OFB
The second 5 Bits of 08C should read - STOR M(OFB)
Finally the assembly language code for 08C 020FA21 OFB is
load M(OFA)
STOR M(OFB)

B Explain this programme does.

1 In 08A adress the M(0FA) transfer to the accumalator & transfer contents of accumalator to memory location 0FB.

2 In 08B adress the M(0FA) transfer to the accumulator & take next instruction.
from left half of M(08D)

3 In 08C adress the M(0FA) transfer to the accumulator & & transfer contents of accumalator to memory location 0EB.

Part (b)

(a) opcode = 00000001

operand = 0000000000010

(b) In the beginning, 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.

c.

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:-

$T = Ic/(MIPs*106)$

$T = 104000/0.37*106)$

$T = 104000/37*103$

$T = 2811*10.3$

$T = 2.811\ sec.$

d. 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 & logic	1	60%
Load/store with each hit	2	18%
Branch	4	12%
memory reference with cache miss	12	10%

. The average CPI is $(1 \times 0.6) + (2 \times 0.18) + (4 \times 0.12) + (12 \times 0.1) = 2.64$. 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, $T1 = (2 * 10^6)/(178 * 10^6)$
$= 11ms$.

for the 8 processors, each processor executes 1/8 of the 2 million instructions plus the 25000.

$$T8 = 2 * 10^6 \div 8 + 0.025 * 10^6/1.5 * 10^6$$

$$T8 = 1.8ms$$

Therefore we have.

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

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

$$\text{Speedup} = 6.11$$

d. By depending on the information. It is not obvious how to quantify this effect in Amdahl's equation. Therefore it is supposed that the fraction of code which is parallazable is $f = 1$, then Amdahl's law decreases to speedup $= N = 8$ Therefore the actual speedup $= 4$ is only abat 75% of the theoratical speedup.

Part (e)

e (a) The Pc contain 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 & the PC is incremented. The two steps can be done in parallel (c). (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 (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 & Pc is incremented (c) The value in the MBR is loaded into the IR.

4. (a) 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 the ac & the 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 2941) 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.

5.

Ans A: 2^{24} = 16MBytes.

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

2) The 16 bit 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 & then the second part.

c. The program counter must be at least 32 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 registers) then it will have to be 8 bits long.

g. Part (G)

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