A) Discuss the two ---- multiple interrupts.

Disabling Interrupts :

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

Interrupt Service Routine (ISR):

Priorities assigned to the different types of interrupts. Interrupt 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.

B) Discuss the types of exchanges ..... and I/O modules.

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 internal (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 ~~following~~ 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 interrupt 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. It also receives.

I) Discuss the QuickPath Interconnect ~~(QRT~~ (QPI) protocol layers?

QPI PROTOCOL LAYERS:

QPI is defined as four-layer protocol architecture, encomposing the following layers.

1) PHYSICAL:

Consists 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.

ii) LINK :

Responsible for ~~tig~~ reliable transmission and flow control. It unit of transfer is an 80-bit Flit.

iii) ROUNTING :

Provides the framework for directing ~~Packets~~ Packets through the fabric.

~~iv~~ PROTOCOL :

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

8) Discuss the physical and logical architecture of PCIe in detail?

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.

PCIe links from the chipset may attach to the following kinds of devices that implement PCIe;

SWITCH :

The switch manages multiple PCIe streams.

PCIe ENDPOINT :

An I/O device or controller that implement PCIe, such as a Gigabit ethernet switch, a graphics or video controller, disk interface, or a communications controller.

LEGACY ENDPOINT :

Legacy endpoint Category is intendid 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 BRIGHG :

Allows elder PCI devices to be connected to PCIe-based Systems.

two

A) INSTRUCTION CYCLE:

Fetch Cycle:

The opcode of 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 ~~too~~ loading a word into the MBR, and then down to the IBR, IR and MAR.

Execute Cycle:

The control circuitry interprets the opcode and executes the instruction by sending out the appropriate control signal to cause data to be moved or an operation to be performed by the ALU.

8) Instruction Cycle state diagram?

The States in instruction Cycle Can be discribed as follows.

- **Instruction Address Calculation (iac):** Determine the address of the next instruction to be executed. usually this involves adding a fixed number to address of the previous instruction.

- **Instruction Fetch (If):** Read instruction from its memory location into the ~~processor~~ 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 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.

E) Classes of interrupts ?

1) **PROGRAMS :**
It is generated by some condition that occurs as a result of an instruction execution such as ~~or~~ arthmetic overflow division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space.

2) **TIMER :**
It is generated by a timer within the processor. This allows the operating System to perform certain function on a regular basis.

3) **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.

4) **Hardware Failure :**
It is generated by a failure such as power failure or memory parity error.

D) 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 from about fifty to hundreds of separate lines. These lines can be classified into three functional Groups; data, address and Control lines.

i) **DATA LINES :**
The data lines provide a path for moving data among system modules. These lines, collectively are called the data bus.

ii) **ADDRESS LINES :**
The address lines are used to designate the source or destination of the data on the data bus. The width of address bus determines the maximum possible memory capacity of the system.

iii) 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 includes;

Memory write, Memory read, I/O write, I/O read, Transfer ACK, Bus request, Bus grant, interrupt request, Interrupt ACK, clock and Reset.

A) Programing in hardware and ~~programing~~ Programing in Software.

PROGRAMING IN HARDWARE:

The "Program" is in the form of hardware and termed as hardware program. Suppose we construct a general-purpose configuration of arthmetic and logic functions. This set of hardware will perform various functions on data depending on control

Signals applied to the hardware. In the original case of customized hardware, the system accepts data and produces results.

PROGRAMING IN SOFTWARE:

The new method of programing which is sequence of codes or instructions is called softer programming. In this method, Programming is much easier, Instead of rewiring the hardware for each ~~view~~ 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.

B) Program flow of Control without interrupt and with interrupt.

In the interrupt cycle, the processor checks to see if any interrupts have occured, 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.

C) Disabled interrupt and nested interrupt processing.

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.

Nested Interrupt :

A nested interrupt is to allow an 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~~ routine is still executing, at t = 15 a communication interrupt occurs.

Question # 4

~~(A)~~ Part (A)

Solution :

Memory (contents in hex): 300: 3005;
301: 5940; 302: 7005 Step 1
305 -> IR; Step 2: 3 -> AC;
Step 3: 5940 -> IR; Step 4: 3+2 = 5
-> AC Step 5: 7006 -> IR
Step 6 AC -> Device 6

(B)

1(a) The PC Contain 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 of 1940 in hexadecimal) is loaded into the MBR, and the PC in 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 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.

(c) 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 1941) 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.

(c)

Ans (a) $224 = 16$ 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.

(c) The program counter must be 24 bits. Typically, a 32-bit microprocessor will have a 32-bit external address bus and a 32-bit 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 opcode (called the opcode register) then it will have to be 8 bits long.

(D)

Ans) Clock cycle = 1 = 125 ns @ 8MHz

Bus cycle = 4 × 125 ns = 500 ns

2 bytes transfered every 500 ns

thus transfer rate

4MB/sec.

Doubling frequency may mean adopting a new chip manufacturing technology (assuming each instruction will have the same number of clock cycles) doubling the external data bus means wider (maybe newer) on-chip data bus drivers/latches and modifications to the bus control logic. In the first case, the speed of memory chips will also need to double (roughly) not to

slow down the microprocessor in the second case, the 'word length' of the memory will have to double.

(E)

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

(b) Suppose we do 100 transfer of operands and instructions of which 50 are one byte long and 50 are two bytes long. The 8-bit microprocessor takes $50 + (2 \times 16 - 50) = 150$ bus cycles for the transfer. The 16-bit microprocessors require $50 + 50 = 100$ bus cycles. Thus, the data transfer rate differ by a factor of 1.5.

(F)

Ans) A bus cycle take 0.25 μs, so a memory cycle take 1 μs. If both operands are even-aligned, it takes 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.

(G)

Ans) Consider a mix of 100 instruction and operands. On average, they consist of 20% 32 bit items, 40% 16 bit items, and 40% 8-bit items. The number of bus cycles required for 16 bit microprocessor is (2×20)+40+40 = 120. For the 32-bit microprocessor, the number required is 100. This amounts to an improvement of 20/120 or about 17%.