NAME: SYED TAIMOOR SHAH

ID : 14832

SUBJECT: COMPUTER ARCHITECTURE

DEPARTMENT: BS CS 4TH

SUBMITTED TO: SIR AMIN

Date : 15/05/2020

ASSIGNMENT: 3

Question No 1

A: Discuss the two approaches for dealing with multiple interrupts?

Ans: (1) The first is to disable interrupts while an interrupt is being processed. A disable interrupt simply means that the processor can and will ignore that interrupt request signal. The draw back to the preceding approach is that it does not take into account relative priority or time-critical needs.

(2) A second approach is to define priorities for interrupts and to allow an interrupt of higher priority to cause a lower-priority interrupt handler to be itself interrupted.

(8) Discuss the type of exchanges that are needed by indicating the major form of input and output for processor, memory and I/O modules?

Ans: The types of exchanges that are needed by indicating the major form of I/O for processor, memory and I/O modules are:

- **Memory to processor:**
  The processor reads an instruction from memory.
- **Processor to memory:**
  The processor writes a unit of data to memory.
- **I/O to processor:**
  The processor reads data from an I/O device via an I/O module.
- **Processor to I/O:**
  The processor send data to the I/O devices.

I/O to or from memory:

For these two cases, an I/O module is allowed to exchange data directly with memory, without going through the processor using direct memory access.

(C) Discuss the QuickPath Interconnect (QPI) protocol layer?

Ans QPI Protocol Layers:

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

(i) 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.

(ii) **Link:**

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

(iii) **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 number of Flits.

(D) Discuss the physical and logical architecture of PCIe in detail:

Ans 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 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 PCIe-based systems.

Question No 2

Write a short note on each of the following:

Ans Instruction Cycle:

The processing required for a single instruction is called an instruction cycle. Using the simplified two-step description, the instruction cycle is depicted. The two steps are referred to as the fetch cycle and execute cycle. Program execution halts only if the machine is turned off, some of unrecoverable errors occurs, or a program instruction that halts the computer is encountered.

(B) Instruction cycle state diagram:

Ans The states in instruction cycle can be described 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.

★ **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.

(ii

★ **Data Operation (do):** Perform the ine program

operation indicated in the instruction.

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

(C) Classes of Interrupts:

Ans (i) 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

(ii) **Timer:-**

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

(iii) **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.

(iv) **Hardware Failure:**

It is generated by a failure such as

power failure or memory parity error.

(D) Bus interconnection Scheme:

Ans 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 include: Memory write, Memory read, I/O write, I/O read, Transfer ACK, Bus request, Bus grant interrupt request, Interrupt ACK, Clock and Reset.

Question No 3

A Programming in hardware and programming in software.

Ans: Programming in Software:

The "program" in the form of hardware is termed as hardware 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 signals 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 programing which is 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~~, in fact an instruction, and part of the hardware interpets each instruction and generates control signals.

(B) Program flow of control without interrupt and with interrupt.

Ans 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.

(E) Disabled interrupt and nested interrupt processing.

Ans 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 systems stack and execution continues at the printer interrupt service routine (ISR). While this routine is still executing, at t = 15 a communication interrupt occurs.

Question No 4

(A)

Sol: Memory (Contents in hex): 300: 3005;
301: 5940; 302: 7006 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

(B)

Sol:

(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 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 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

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 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.

(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 but 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)

Sol: Clock cycle =

1 = 125 ns @ MHz

Bus cycle = 4 x 125 ns = 500 ns

2 bytes transferred every 500 ns

thus transfer rate

4 MB/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 microprocesor 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 tranfers 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 microprocessor requires $50 + 50 = 100$ bus cycles. Thus, the data transfer rate differ by a factor of 1.5.

(F) Ans:

A bus cycle take 0.25 ns, so a memory cycle take 1 ns. If both operands are evenaligned, it takes 2 ns to fetch the two operands. If one is odd-aligned, the time required is 3 ns. If both are odd-aligned, the time required is 4 ns.

(G) Ans:

Consider a mix of 100 instructions 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 (2x20) + 40 + 40 = 120. For the 32-bit microprocessor, the number required is 100. This amounts to an improvement of 20/120 or about 17%.