AMEY THAKUR **SE-Comps B-50**

Practice Test COA

Class: SE B - 50 **Date: 04/05/2020** **Time: 10 AM to 11AM**

50	Q1) Explain cache consistency and coherency with suitable examples .Also give methods to maintain cache consistency? Q2) Explain programmed I/O and Interrupt Driven I/O

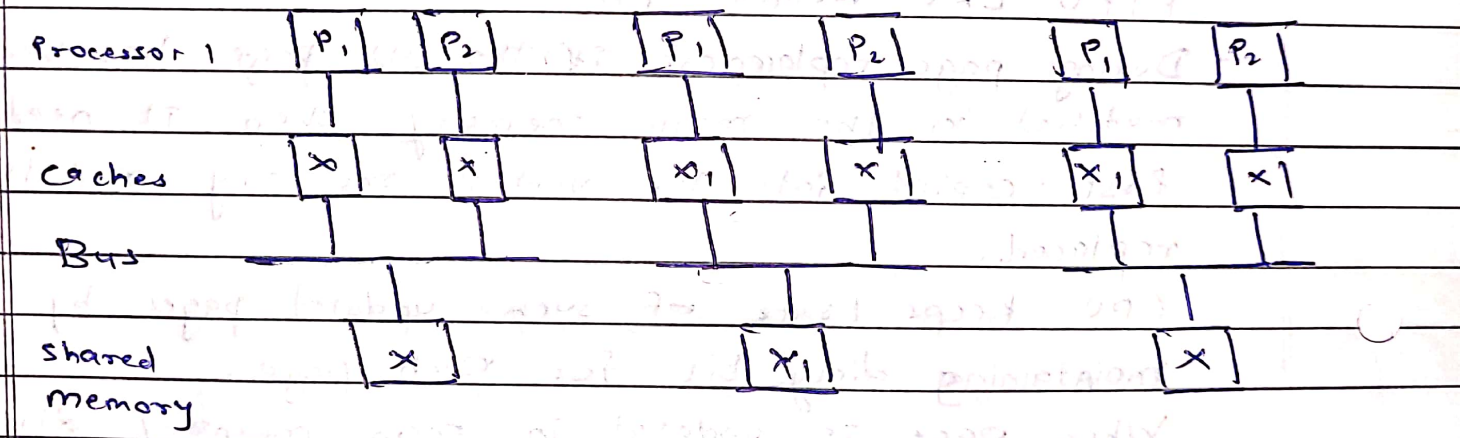
Amey Thakur
B - 50

Q1.

Ans:

The Cache Coherence Problem

- In a multiprocessor system, data inconsistency may occur among adjacent levels or within the same level of memory hierarchy.
- For ex- the cache and the main memory may have inconsistent copies of the same object.
- As multiple processor operate in parallel and independently multiple caches may possess different copies of the same memory block, this creates cache coherence problem. Cache coherence schemes help to avoid this problem by maintaining a uniform state for each cached block of data.



Before update | write - through | Write - back

- Let x be an element of shared data which has been referenced by 2 processors, P_1 & P_2.
- In the beginning, 3 copies of x are consistent. If the processor P_1 writes a new data x_1 into the cache, by using write-through policy, the same copy will be written immediately into the shared memory.

Name: Amey Thakur Class: Div: B Roll No: 50

Subject: Topic: Date: Page No:

- In this case, inconsistency occurs between cache memory and the main memory.
- When a write-back policy is used, the main memory will be updated when the modified data in the cache is replaced or invalidated.
- In general, there are 3 sources of inconsistency problem
  - → Sharing of writable data
  - → Process migration
  - → I/O Activity
- Maintaining cache coherency is a problem in a multiprocessor system when the processor contain local cache memory.
  Data inconsistency between different caches easily occurs in the system.

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Q2.

Ans:

Interrup Driven I.O

- Interrupt Driven IO overcomes the disadvantage of programmed IO. i.e. CPU waiting for IO device.
- This disadvantage is overcome by CPU not repeatedly checking for the device being ready or not instead the IO module interrupts when ready.
- The sequence of operation for interrupt driven IO.

(1) CPU issues the read command to IO device.
(2) IO module gets data from peripheral while CPU does other work.
(3) Once the IO module completes the data transfer from IO device, it interrupts CPU.
(4) On getting the interrupt, CPU requests data from IO module.
(5) IO module transfers the data to CPU.

- When CPU gets an interrupt, it performs the operations →

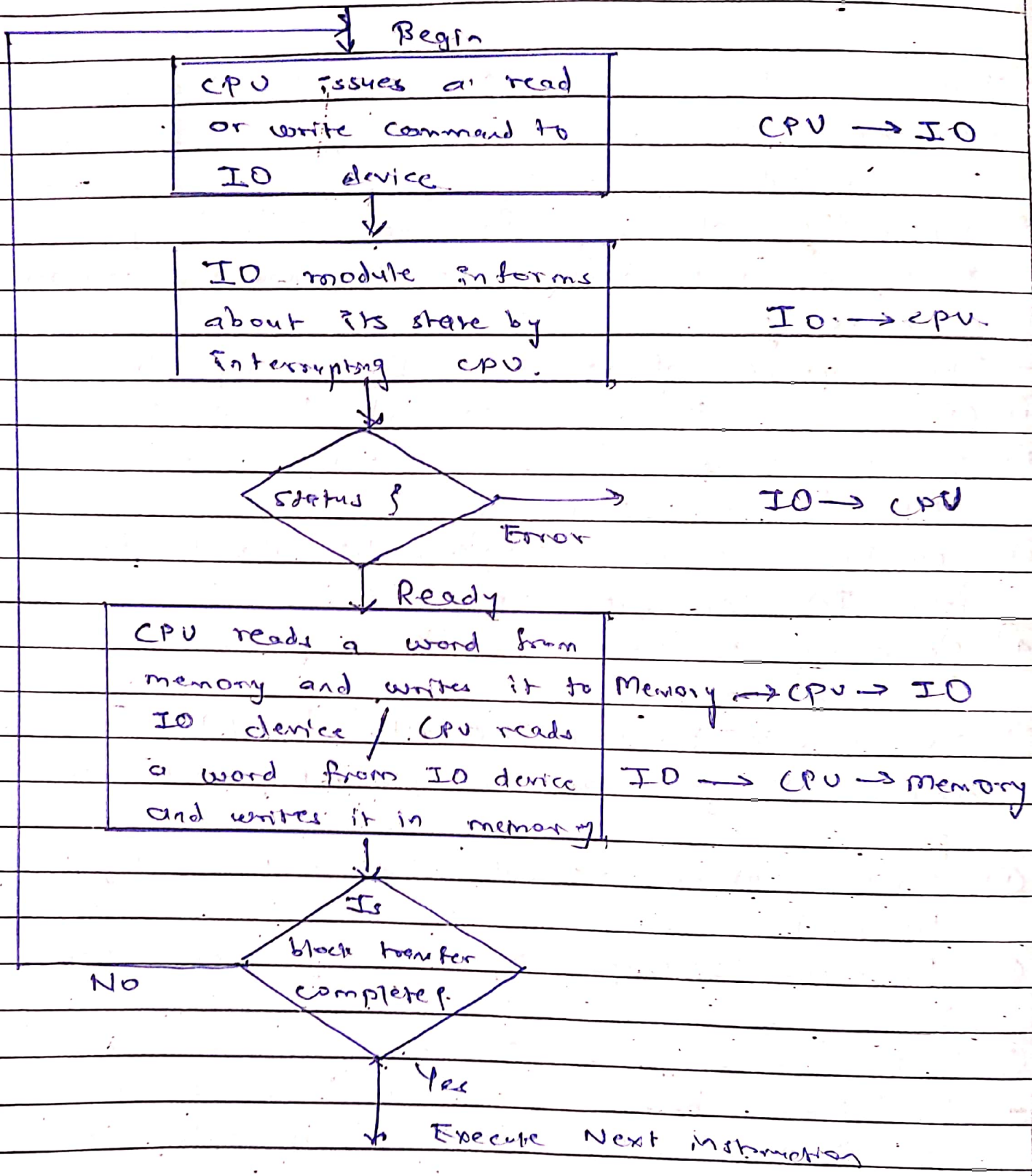
(1) Save context. i.e the contents of the register on the stack.
(2) Processes interrupt by executing the corresponding ISR.
(3) Restore the register context from the stack.

- IC 8259 has 8 interrupt lines and is used as IO module when interrupt driven IO is used.

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

- Transferring a block of data using interrupt driven IO.



Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Programmed IO

- In programmed IO method of interfacing, CPU has direct control over IO
- The processor checks the status of the devices and issues read / write command and then transfers data
During the data transfer, CPU waits for IO module to complete operation and hence this system wails the CPU time
- That's why Interrupt Driven IO is Syperior than programmed IO.

- In programmed IO, sequence of operations to be carried out →

(1) CPU requests for IO operation
(2) IO module performs the said operation
(3) IO module updates the status bits
(4) CPU checks these status bits periodically. Neither the IO module cannot inform CPU directly nor can IO module interrupt CPU.
(5) CPU may wait for operation to complete or may continue the operations later

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

- Transferring a block of data using programmed IO