Master of Computer Applications

CAPOL403R01: Computer Organization & Architecture

Unit V: Lecture 3: Part 1
Interrupt Driven IO

Dr. D. MURALIDHARAN
School of Computing
SASTRA Deemed to be University

Interrupt driven IO

- Drawback of programmed I/O
  - The processor has to wait for the completion of data transfer without doing any useful task
  - It must interrogate the status of the I/O module repeatedly
  - So, the level of the performance of the entire system is severely degraded
- A solution
  - After instructing the IO module what to do, the processor may execute some other program
  - The I/O module will interrupt the processor for its service when it is ready to exchange data with the processor
  - The processor then executes the data transfer and then resumes its former processing

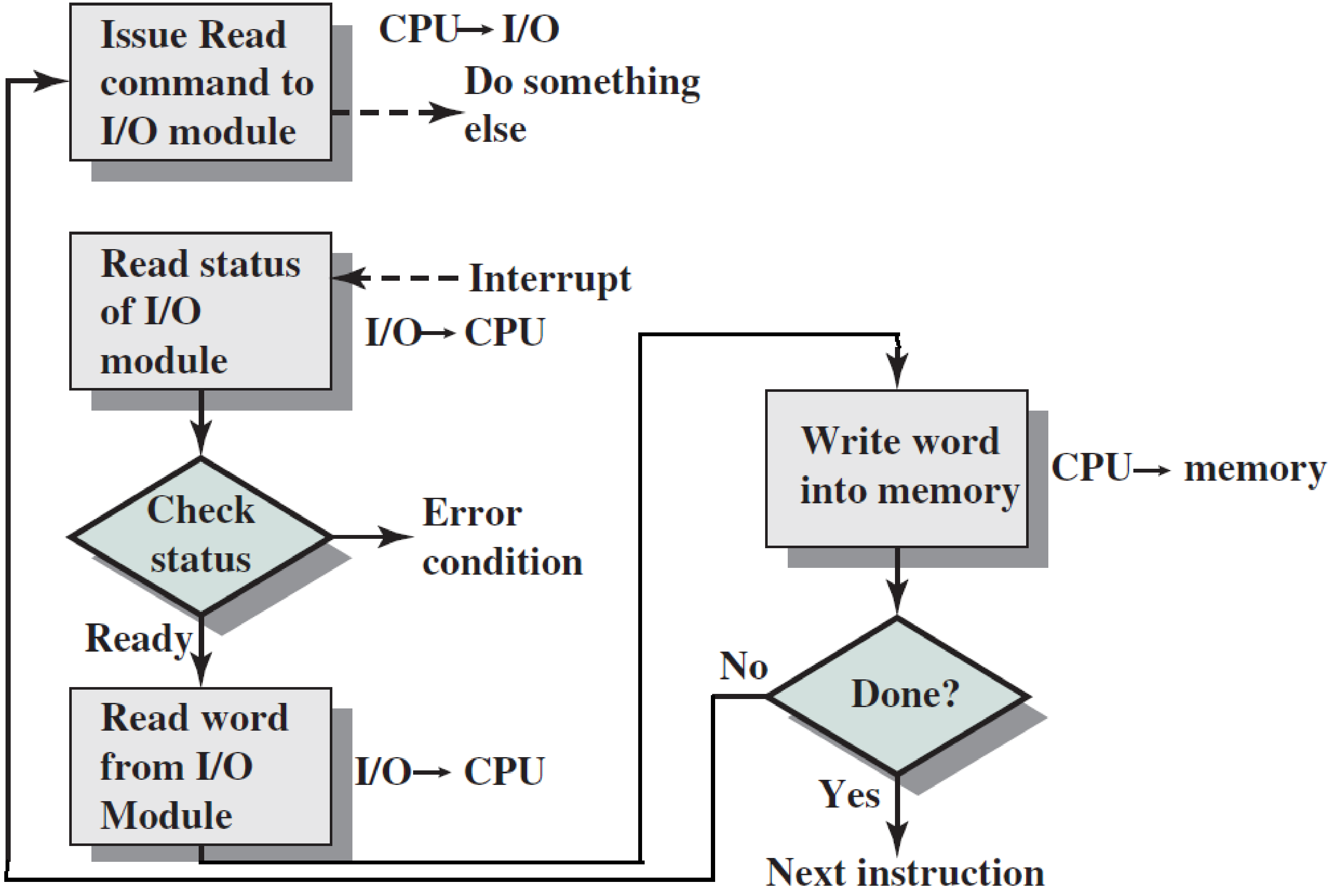
Actions of IO module

- IO module gets READ command from the processor
- IO module proceeds to read the data from the peripheral
- Save the data in the data register of the IO
- Inform this to the processor by raising the interrupt through a control line
- IO module waits for the processor to take the data
- When the processor requests the data, IO module places it in the data lines
- Current data transfer is completed, (waits for the next data transfer)

Actions of the processor

- The processor issues READ command
- Assuming the processor is working on several different programs, it suspends this program and starts to execute another program
- At the end of each instruction cycle, checks for interrupts
- When the interrupt from the IO module occurs, it saves the current status and processes the interrupt
  - In this example, the processor reads the data from the data register of IO
- Processor restores the status of the program and proceeds to execute

Reading a block of data

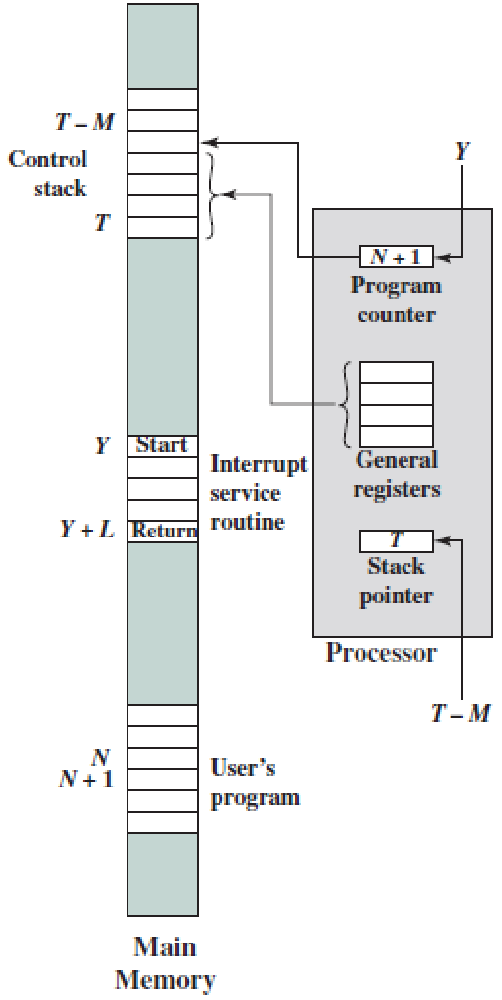


Interrupt processing

1. Interrupt signal is issued by the IO device once the IO operation is completed
2. The processor finishes the current instruction
3. The processor checks for the interrupt,
   - It finds the interrupt
   - It sends an acknowledgement signal to the device
   - This acknowledgement enables the device to remove its interrupt signal
4. The processor does some ‘household activities’
   - It saves the current status on stack (PSW contents)
   - It saves the PC on stack

Changes in memory and registers for an interrupt

Interrupt occurs after instruction N

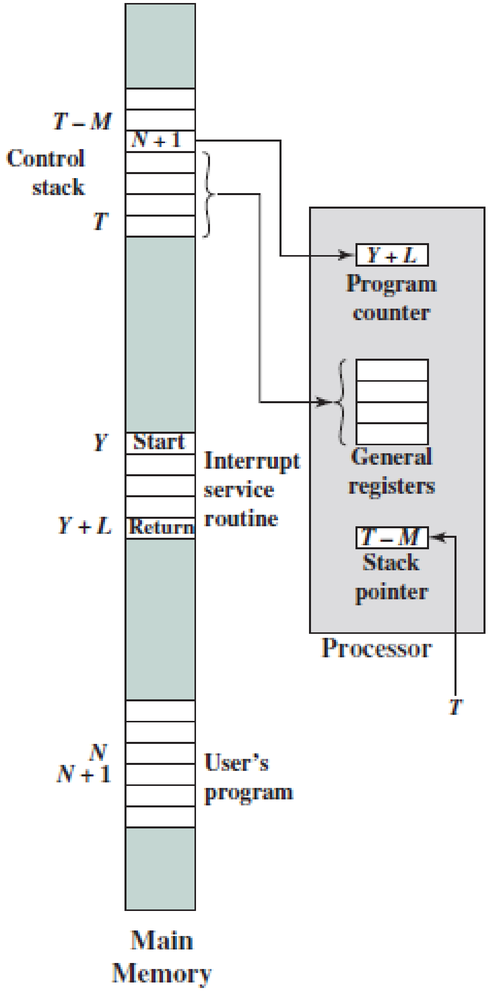


Interrupt processing…

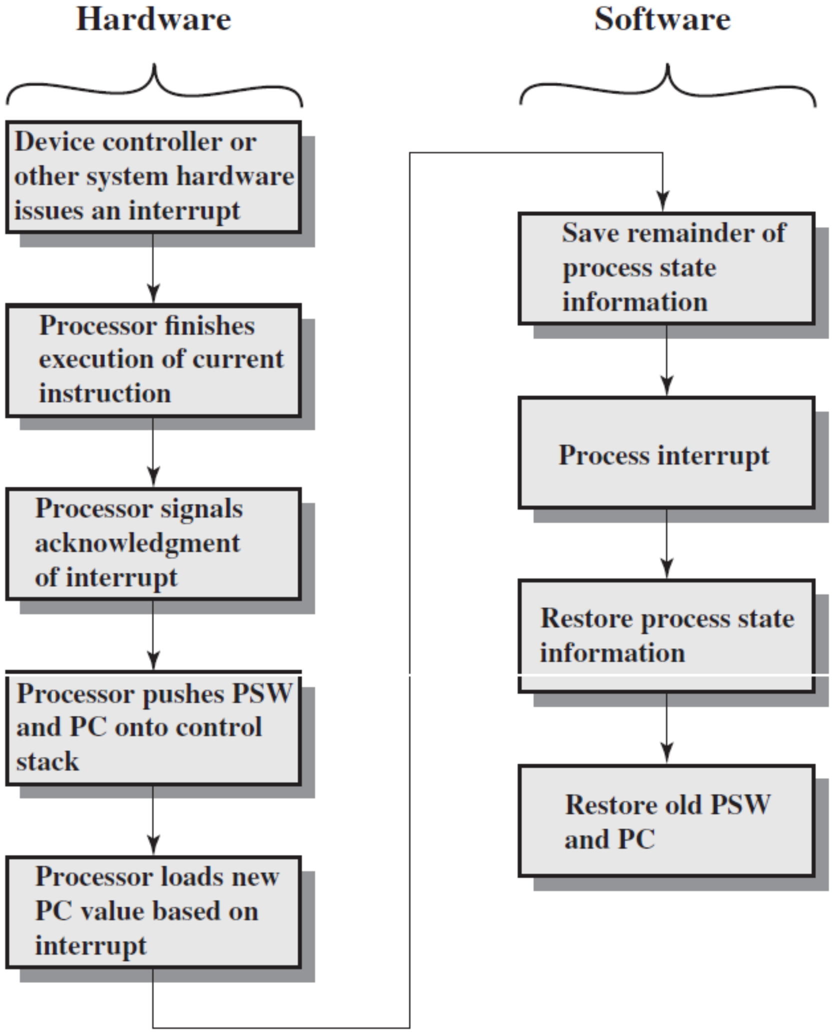
5. The processor loads the first address of the appropriate ISR on PC
6. The interrupt handler saves the appropriate register contents on stack
7. The interrupt handler processes the interrupt.
    - It examines the status information relating to the I/O operation or other event that caused an interrupt
    - It may also involve sending additional commands or acknowledgments to the I/O device.
8. When the interrupt processing is completed, the saved register values are retrieved from the stack & restored in the registers
9. The PSW and the saved processor status are retrieved
    - It enables the processor to resume the suspended program

Changes in memory and registers for an interrupt

Return from interrupt



Interrupt processing flow


Hardware
Software
Device controller or other system hardware issues an interrupt
Processor finishes execution of current instruction
Processor signals acknowledgment of interrupt
Processor pushes PSW and PC onto control stack
Processor loads new PC value based on interrupt
Save remainder of process state information
Process interrupt
Restore process state information
Restore old PSW and PC

Thank you