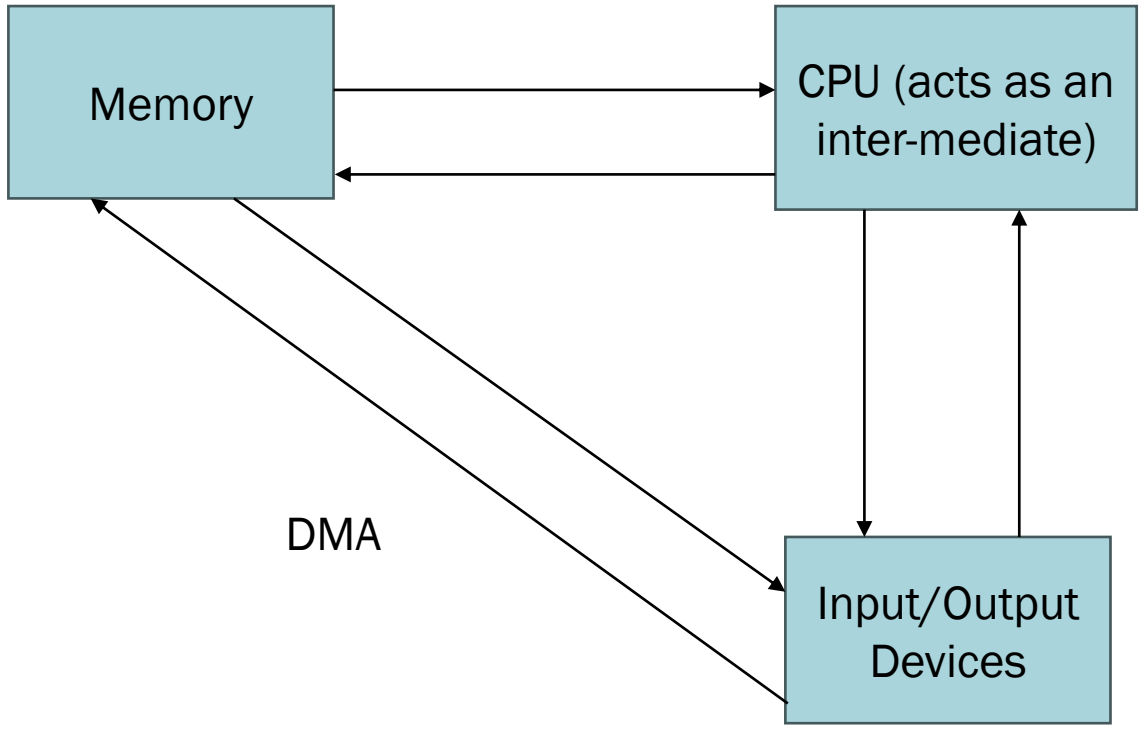
Modes of Transfer (Computer Architecture and Microprocessors)

Introduction

- Binary information received from an external device is stored in memory for later processing.
- Information transferred from the CPU into an external device originates in the memory unit.
- The CPU merely executes the I/O instructions and may accept the data temporarily, but the ultimate source or destination is the memory unit.

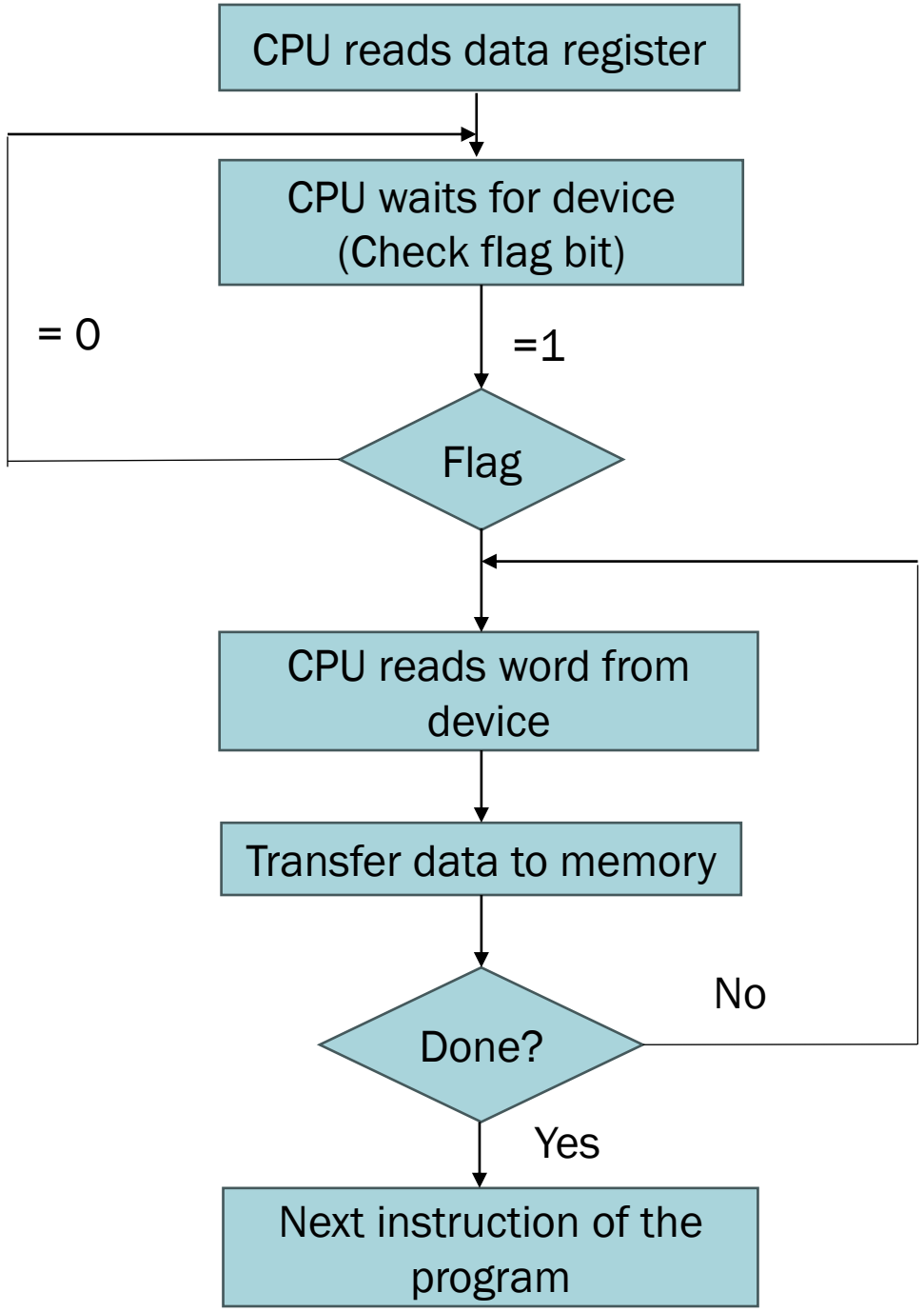


Different modes of data transfer

- Programmed I/O
- Interrupt Initiated I/O
- Direct Memory Access (DMA)

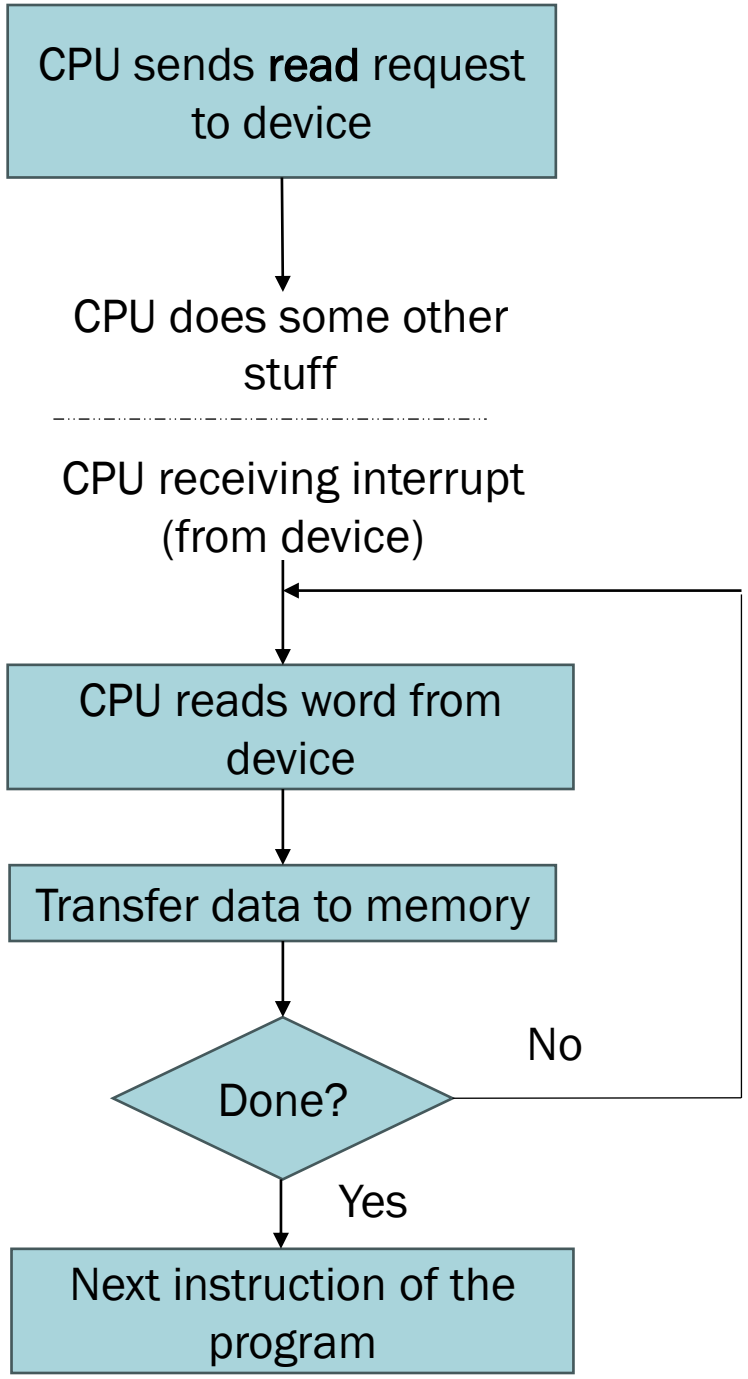
Programmed I/O

- Programmed I/O operations are the result of I/O instructions written in the computer program.
- Each data transfer is initiated by an I/O instruction in the program (to access registers or memory on a device).
- Usually, the data transfer is from a memory and a CPU register.
- This case requires constant monitoring by the CPU and the CPU stays in a loop until the I/O devices indicates that it is ready for data transfer.



Interrupt-Initiated I/O

- In Interrupt-Initiated I/O, instead of continuous monitoring of CPU, interface will be informed to issue an **Interrupt request signal**, when data are available from the device.
- Meanwhile, CPU proceeds to executes some other tasks and the interface keeps monitoring the device.
- When device is ready for data transfer, it generates interrupt request.
- Upon detecting the external interrupt signal, the CPU stops the task it is performing, process the I/O data transfer and resumes the original task it was performing.

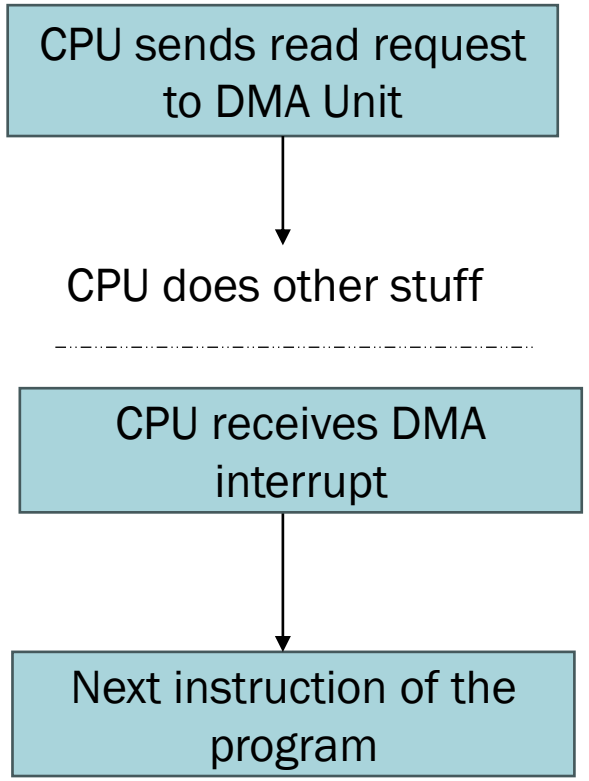


Programmed I/O V/S Interrupt-Initiated I/O

PROGRAMMED I/O	INTERRUPT-INITIATED I/O
Data transfer is initiated by the means of instructions stored in the computer program.	The I/O transfer is initiated by the interrupt command issued to the CPU.
The CPU stays in the loop to know if the device is ready for transfer and has to continuously monitor the peripheral device.	There is no need for the CPU to stay in the loop as the interrupt command interrupts the CPU when the device is ready for data transfer.
This leads to the wastage of CPU cycles as CPU remains busy needlessly and thus the efficiency of system gets reduced.	The CPU cycles are not wasted as CPU continues with other work during this time and hence this method is more efficient.
CPU cannot do any work until the transfer is complete as it has to stay in the loop to continuously monitor the peripheral device.	CPU can do any other work until it is interrupted by the command indicating the readiness of device for data transfer.

DMA(Direct Memory Access)

- To transfer large blocks of data at high speed b/w external device and main memory DMA approach is used effectively.
- DMA allows data transfer between I/O device and main memory with minimal intervention of CPU.
- CPU is involved in beginning and end of the transfer and interrupted only after entire block has been transferred.
- CPU grants I/O interface authority to read from or write to memory without its involvement.
- DMA itself controls data transfer b/w memory and I/O devices.



Advantages of DMA

Reduced CPU overhead	• With DMA, the CPU initiates the transfer but then can continue executing other tasks while the DMA controller handles the data movement independently. This reduces CPU overhead compared to programmed I/O, where the CPU must actively manage each data transfer.
Faster Data Transfer	• DMA transfers data directly between peripheral devices and memory without CPU intervention, resulting in faster data transfer rates. In contrast, programmed I/O requires the CPU to move data between peripherals and memory one byte or word at a time, which is slower.
Improved System Responsiveness	• Since DMA transfers occur independently of the CPU, the system remains more responsive to other tasks. This is particularly beneficial in multitasking environments where the CPU needs to handle multiple concurrent processes.

Thank you

PRESENTED BY:

Kanan Kaura

D2 CSE C-2

URN : 2203845