
Master of Computer Applications

CAPOL403R01: Computer Organization & Architecture

Unit V: Lecture 6
IO Channels and Processors

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

The Evolution of the I/O Function

- The CPU directly controls a peripheral device
 - This is seen in simple microprocessor-controlled devices
- A controller or I/O module is added
 - The CPU uses programmed I/O. With this step, the CPU becomes somewhat free from the specific details of external device interfaces
- Interrupt initiated I/O
 - Interrupts are employed
 - The CPU need not spend time waiting for an I/O operation to be performed
 - Efficiency is enhanced

The Evolution of the I/O Function

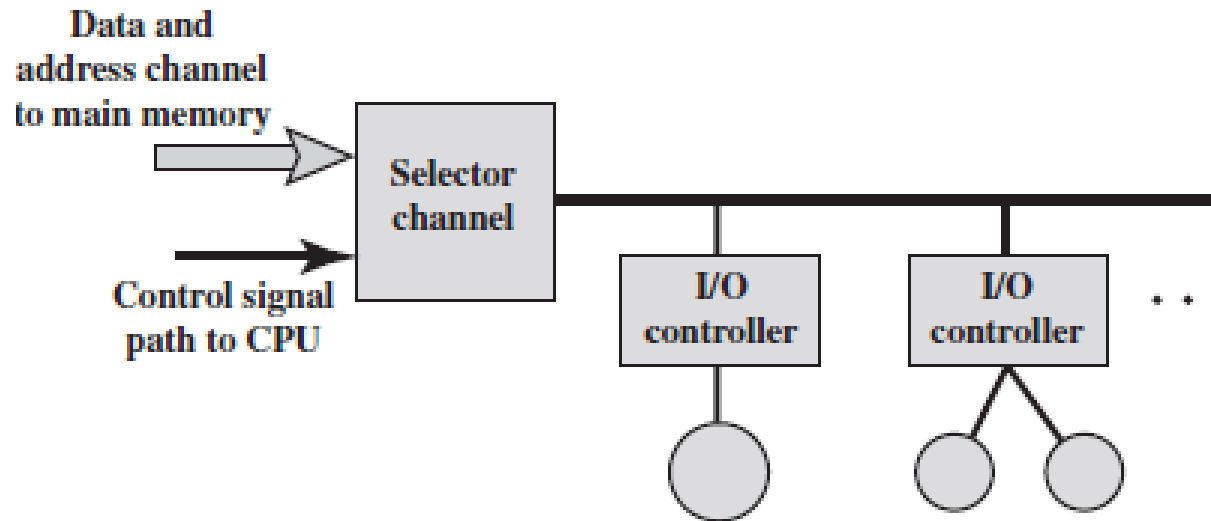
- The I/O module is given direct access to memory via DMA
 - The processor is involved in the process only during the beginning and end of the transfer.
- The I/O module is enhanced to become a processor in its own right, with a specialized instruction set tailored for I/O
 - The CPU directs the I/O processor to execute an I/O program in memory
 - The I/O processor fetches and executes these instructions without CPU intervention
 - This allows the CPU to specify a sequence of I/O activities and to be interrupted only when the entire sequence has been performed.
 - This I/O controller is called as IO Channel

The Evolution of the I/O Function

- The I/O module has a local memory and a computer in its own right
 - With this architecture, a large set of I/O devices can be controlled with minimal CPU involvement
 - A common use for such an architecture has been to control communication with interactive terminals.
 - The **I/O processor** takes care of most of the tasks involved in controlling the terminals.

Types of I/O Channel

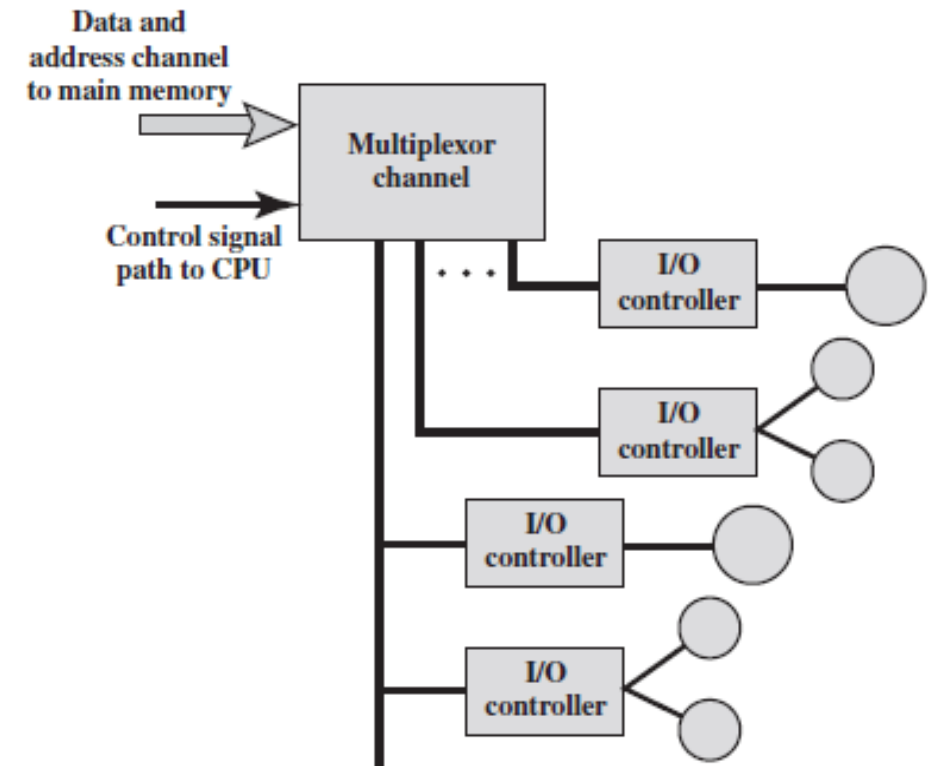
- Selector Channel
 - It controls multiple high-speed devices
 - The I/O channel selects one device and effects the data transfer at any time
 - Each device, or a small set of devices, is handled by a *controller* or I/O module



Types of I/O Channel

- Multiplexor Channel

- It controls multiple devices at the same time
- For low-speed devices, a *byte multiplexor* accepts or transmits characters as fast as possible to multiple devices
- For example, the resultant character stream from three devices with different rates and individual streams $A_1A_2A_3A_4$ c, $B_1B_2B_3B_4$ c, and $C_1C_2C_3C_4$ c might be $A_1B_1C_1A_2C_2A_3B_2C_3A_4$, and so on.



Thank you