Section: 2nd Year Engineer in Computer Science

Chapter 3

Input/Output Modes

Pr. Ch. Bencheriet

2024/2025

Plan

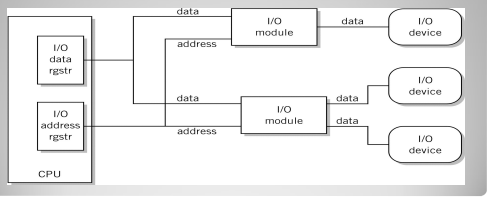
1. Introduction

- It is the CPU that initiates all input or output operations.
- How can input/output operations be managed without degrading the performance of the entire machine?
- Three main methods of managing input/output operations are distinguished:

1. Programmed Input/Output (Polling).
2. Direct Connection with Interrupt.
3. Direct Memory Access (DMA).

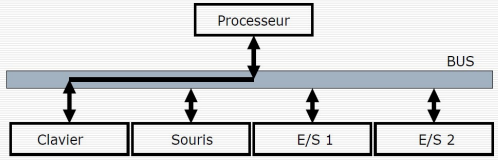
2. Programmed Input/Output (polling)

- The simplest model in which the I/O controller is connected to a pair of I/O registers (data and address) in the CPU via a bus.
- The central processor is fully utilized to control and manage exchanges with the device.



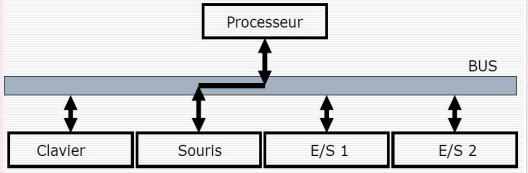
2. Programmed Input/Output (polling)

✓The CPU polls the controller at regular intervals to check the status of an I/O operation.



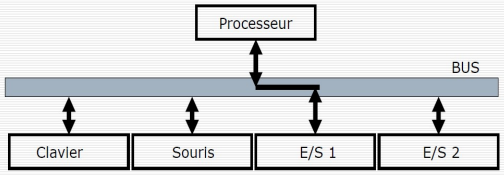
2. Programmed Input/Output (polling)

✓The CPU polls the controller at regular intervals to check the status of an I/O operation.



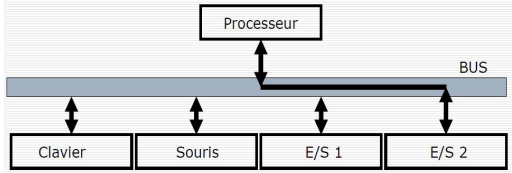
2. Programmed Input/Output (polling)

✓The CPU polls the controller at regular intervals to check the status of an I/O operation.



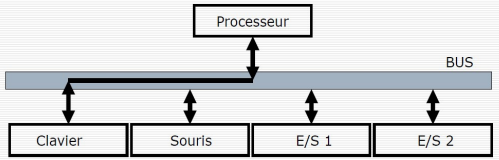
2. Programmed Input/Output (polling)

✓The CPU polls the controller at regular intervals to check the status of an I/O operation.



2. Programmed Input/Output (polling)

✓The CPU polls the controller at regular intervals to check the status of an I/O operation.



2. Programmed Input/Output (polling)

Remarques

- In practice, there are several devices connected to the CPU (with the ability to address 100 I/O operations).
- Communication is very slow.
- This polling wastes a significant amount of CPU time.
- Usage example: Computer keyboard.

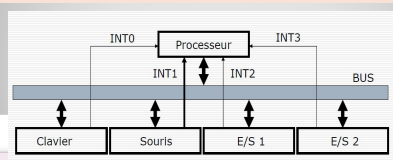
3. Direct Connection with Interrupt

Interrupt?

An interrupt is an asynchronous signal that can be issued by any external device to the processor, interrupting the current task of the processor to force the execution of a program that handles the cause of the interrupt (the interrupt service routine).

Examples:

- Unexpected input
- Abnormal situation
- Illegal instructions

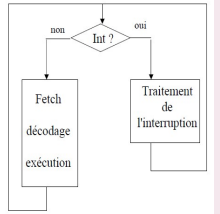


3. Direct Connection with Interrupt

The CPU is informed by the controller that a task needs to be performed via the sending of an interrupt request:

Then:

1. Stop the current program;
2. Save the machine state;
3. Execute the interrupt service routine;
4. Restore the machine state;
5. Resume the interrupted program.



3. Direct Connection with Interrupt

- The device uses the interrupt mechanism to signal that it is ready.
- All devices signal an event to the processor via a single interrupt line by setting the corresponding signal.
- To handle interrupts, the operating system has a set of interrupt management programs.
- When an interrupt is received, the currently running program is stopped in favor of the interrupt service routine.

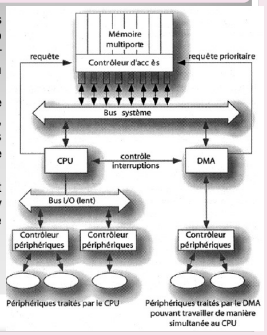
4. Direct Memory Access (DMA)

Problem?
The interrupt mechanism is very efficient, but the time used by the processor for the interrupt service routine and the driver should not be too significant.

Solutions:
- Decrease the number of interrupts.
- Load data from main memory without using the central processor, using Direct Memory Access (DMA).

4. Direct Memory Access (DMA)

Direct Memory Access (DMA) devices are added to many computers to allow the transfer of a large number of words without intervention from the CPU.
- DMA is connected between a device controller and the system bus, enabling the device to access memory without going through the CPU.
- It is a specialized controller that transfers data between memory and/or an I/O component while the CPU handles other tasks.



4. Direct Memory Access (DMA)

Constitution of DMA

The hardware components include:
- An address register
- A count register
- A control register (read or write)
- A buffer area for storing data
- An active component, processor-like

4. Direct Memory Access (DMA)

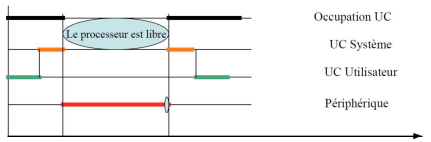
Operation
- DMA takes full responsibility for transferring a block of data.
- The CPU initiates the transfer by providing:
  1. The identification of the concerned device,
  2. The direction of the transfer,
  3. The memory address of the first word to be transferred, and the number of words involved in the transfer.
- When the transfer is complete, DMA signals to the CPU that the operation is finished via an interrupt.

4. Direct Memory Access (DMA)

Transfer Process

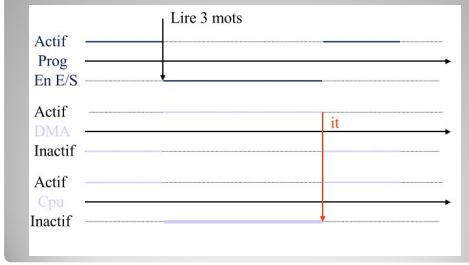
Initialization: The processor programs the DMA controller with the transfer details.
Direct Memory Access: The DMA controller directly accesses memory without going through the processor.
Interrupt: An interrupt is generated to inform the processor that the transfer is complete.

During the entire I/O operation, the central processor is free.

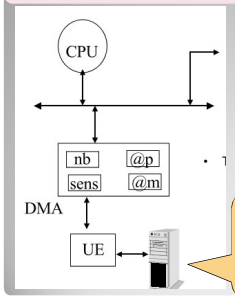


4. Direct Memory Access (DMA)

Input/Output with DMA and Interrupt



4. Direct Memory Access (DMA)



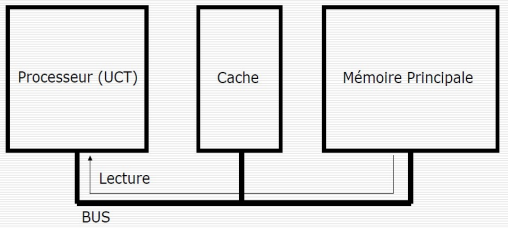
Sous_Programme d'It_DMA :
Vérification transfert OK
RTI
Programme principal :
Initialiser DMA :
nb = nombre d'octets à transférer
@m = adresse premier octet en MC
@p = adresse périphérique
sens = écriture;
Lancer DMA;
attendre la fin de l'entrée/sortie

I/O Processing

1. The processor initializes the DMA.
2. The DMA performs the input/output; the processor is inactive.
3. The DMA signals the end of the input/output via an interrupt.

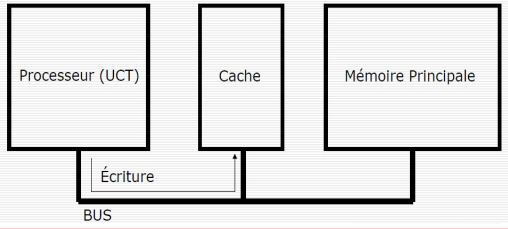
4. Direct Memory Access (DMA)

Without DMA Controller



4. Direct Memory Access (DMA)

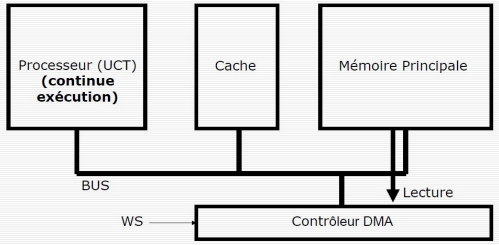
Without DMA Controller



(Read/Write) executed 65,536 times by the processor for a 64K block, leaving a large number of cycles unavailable for performing more useful tasks.

4. Direct Memory Access (DMA)

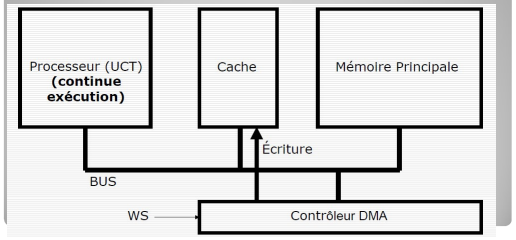
With DMA Controller



4. Direct Memory Access (DMA)

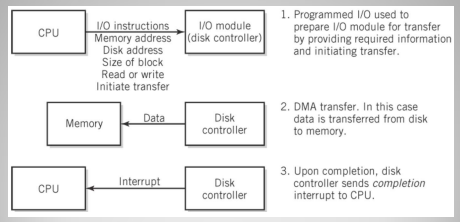
With DMA Controller

(Read/Write) executed 65,536 times by the DMA Controller for a 64K block, freeing a large number of cycles for the CPU to perform more useful tasks.



4. Direct Memory Access (DMA)

Example: Writing/Reading a memory block to/from a disk



DMA I/O Channel

Definition

The DMA I/O channel refers to the pathway or conduit used by the DMA controller to transfer data between memory and peripheral devices.

Operation

The DMA controller has a dedicated I/O channel that acts as a link between the main memory and peripheral devices.
The processor initiates the transfer by programming the DMA controller through this channel.

DMA Channel Architecture

DMA Controller: Supervises and coordinates data transfers between peripherals and memory without constant intervention from the processor.
Configuration Registers: Specific registers programmed by the processor to configure transfer parameters. These registers may include the source address, destination address, amount of data to transfer, etc.
Status Registers: Used to track the current state of the transfer, such as the number of remaining data to transfer, completion status, etc.
Data Bus: The channel uses a data bus to perform the actual transfer of data between memory and peripherals.
Address Bus: Used to specify the source and destination addresses of the data during the transfer. These addresses are often stored in the configuration registers.
Control Logic: Manages the flow of operations. It decides when to initiate a transfer, when to update the status registers, and how to handle interrupts once the transfer is completed.

Selector Channel and Multiplexer Channel

Selector Channel

Controls multiple high-speed I/O modules at all times. It is dedicated to transferring data from a single peripheral to memory.

Multiplexer Channel

Controls multiple I/O modules simultaneously. It allocates bandwidth between each device (typically low-speed) to fetch data in turn.