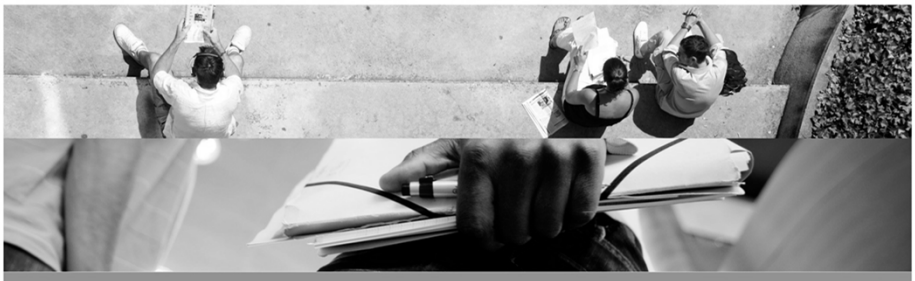


ARCHITECTURE DES SYSTEMES A MICROPROCESSEUR

Séance 3 :Gestion des périphériques d'entrées sorties

membre de UNIVERSITÉ DE LYON

CPE LYON ÉCOLE SUPÉRIEURE DE CHIMIE PHYSIQUE ÉLECTRONIQUE DE LYON

N.ABOUCHI

Objectifs du cours

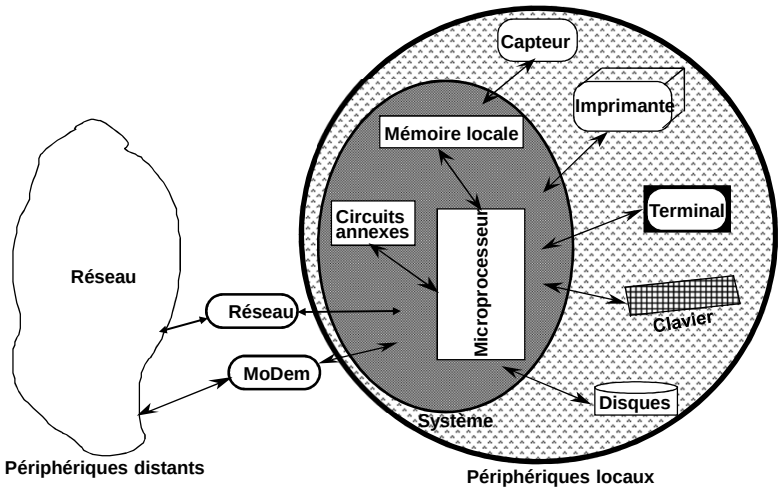
CPE LYON ÉCOLE SUPÉRIEURE DE CHIMIE PHYSIQUE ÉLECTRONIQUE DE LYON

Connaître les principes des échanges.

Connaître les différents modes d'échanges.

Exemple : la communication série.

Relations possibles entre le système et les periphériques



3

Quelques exemples d'entrées sorties

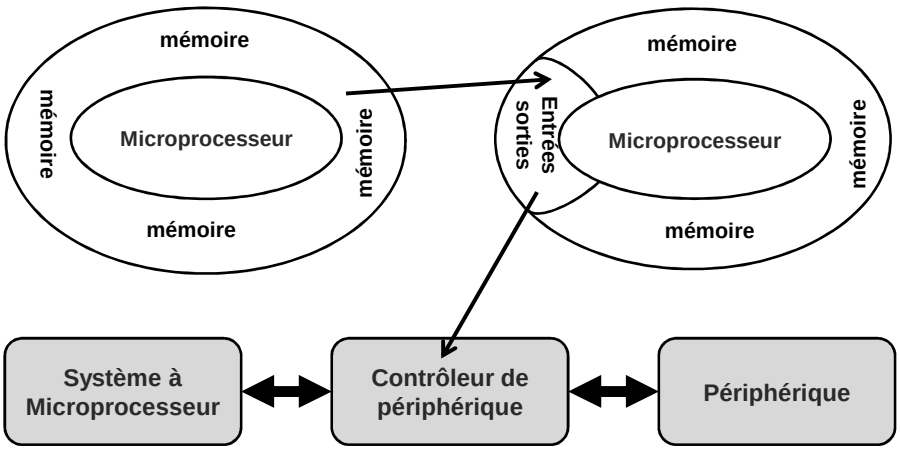
- ✓ Clavier (entrée)
- ✓ Souris (entrée)
- ✓ Imprimante Laser (sortie)
- ✓ CD-Rom (entrée ou sauvegarde)
- ✓ Disque Dur (sauvegarde)
- ✓ Ecran graphique (sortie)
- ✓ Capteurs (entrées)
- ✓ Actionneurs (sorties)
- ✓ etc.

Le processeur doit gérer des périphériques de rôles différents et en particulier de vitesses différentes !

Utilisation de contrôleurs de périphérique

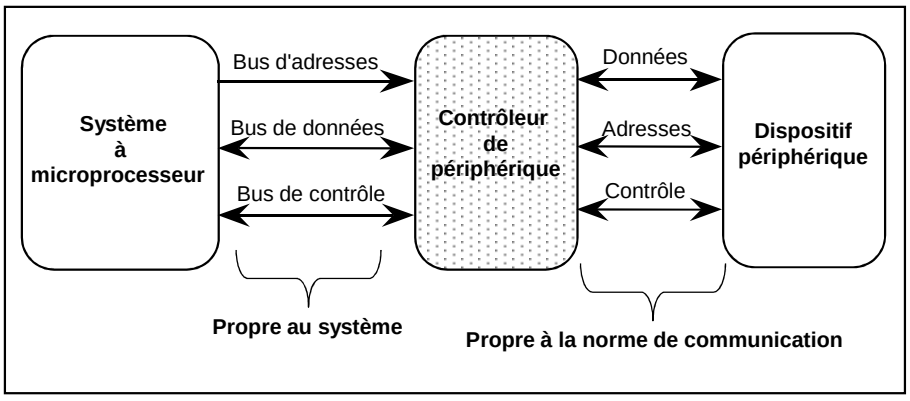
4

Principe du contrôleur de périphérique



5

Rôle du contrôleur de périphérique



Exemples : RS232, CENTRONIC, SCSI, IEEE388, USB, PCMCA, I2C, etc.

6

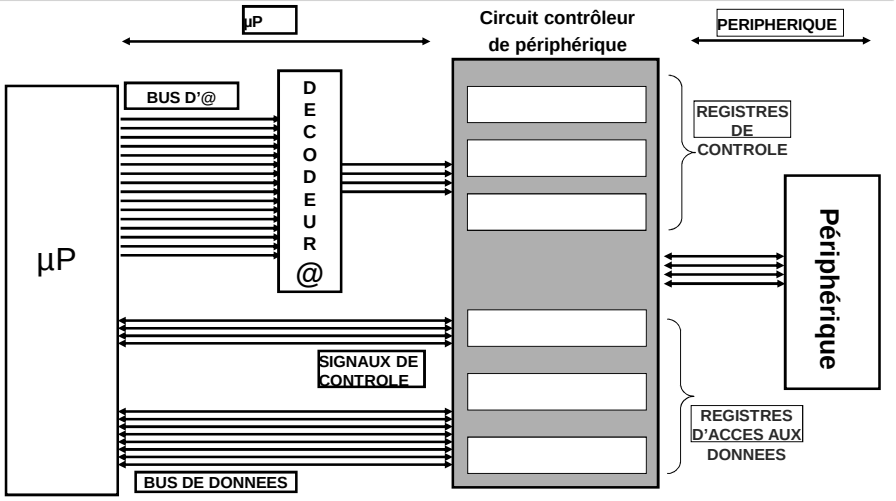
Principe de l'architecture général d'un contrôleur de périphérique

Chaque périphérique est « piloté » par un contrôleur qui lui est spécifique :

- ✓ contient souvent son propre microprocesseur,
- ✓ ses registres,
- ✓ sa mémoire tampon,
- ✓ s'occupe des commandes détaillées du périphérique :
  - gestion des incidents,
  - détection d'erreurs,
  - conversion de format,
  - etc.

7

Architecture général d'un circuit contrôleur de périphérique



8

Dialogue Processeur Contrôleur

Rappel : une partie de la mémoire du microprocesseur est réservée aux entrées sorties,

(**lire/écrire dans cette zone = commander le périphérique**)

Chaque E/S est « implantée » en mémoire

L'échange entre le système et le contrôleur s'effectue grâce aux registres du contrôleur contenant :

- les données,
- l'état du contrôleur,
- les commandes à effectuer.

9

Principales méthodes d'échange

1. **Entrées-sorties programmées :** basée sur la synchronisation logicielle à l'initiative du processeur.
2. **Echange par interruption :** basée sur la synchronisation matérielle à la demande du périphérique.
3. **Accès direct à la mémoire (DMA) :** échange transparent pour le processeur entre un organe périphérique et la mémoire du système.

La synchronisation reste le problème le plus délicat à résoudre dans une opération d'échange entre deux systèmes.

10

Echanges programmés sans synchronisation

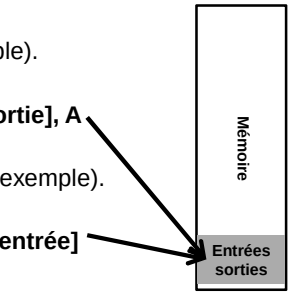
L'échange, **basé sur l'initiative du microprocesseur,** se réduit à **une simple instruction de lecture ou d'écriture à l'adresse du registre de données couplée au périphérique** (exactement comme la mémoire système).

- Pour un envoi vers l'extérieur (la liaison série par exemple).

MOV [adresse du périphérique de sortie], A

- Pour une lecture depuis l'extérieur (liaison parallèle par exemple).

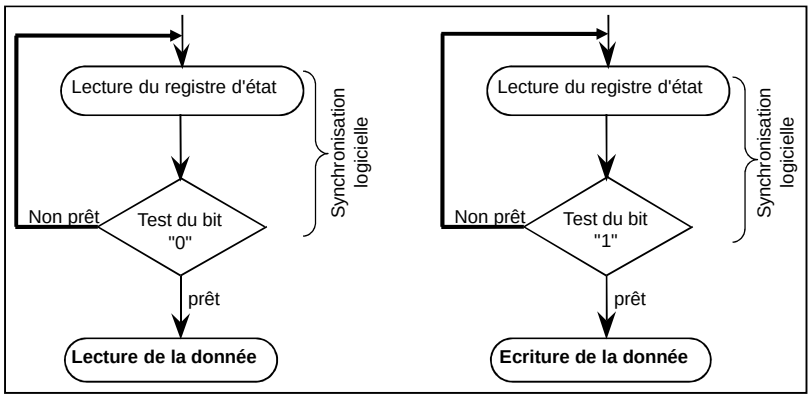
MOV A, [adresse du périphérique d'entrée]



11

Echanges programmés avec Synchronisation logicielle

Cette technique, **basée sur l'initiative du microprocesseur,** consiste à lire le registre d'état du contrôleur d'entrée-sortie et d'agir en conséquence. On parle de **SCRUTATION**



12

Les échanges par interruption

Ce mode d'échange est principalement **basé sur l'initiative du périphérique**. Grâce à ce mécanisme d'interruption, le dipositif périphérique prend l'initiative de l'échange.

Une interruption est un évènement qui provoque l'arrêt du programme en cours et le branchement du microprocesseur à un sous-programme particulier dit de "traitement de l'interruption".

L'adresse du sous-programme d'interruption est donnée par ce qui est appelé **"vecteur d'interruption".** Un vecteur définit l'adresse du sous-programme d'interruption. Une interruption peut être déclenchée soit par **matériel**, soit par **logiciel**.

L'interruption peut etre masquable (notion de priorité) ou non Masquable.

13

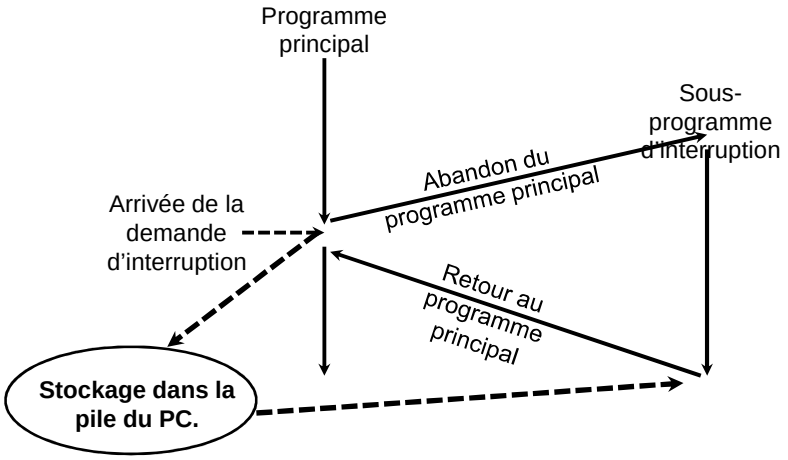
Prise en charge d'une interruption par le µP

Suite à une demande d'interruption par un périphérique :

- ✓ le microprocesseur termine l'exécution de l'instruction en cours,
- ✓ range le contenu des principaux registres sur la pile, le pointeur de code et le registre d'état en particulier,
- ✓ émet, si les interruptions sont autorisées, un accusé de réception à la demande d'interruption (acquittement de l'interruption) pour indiquer au circuit d'E/S que la demande d'interruption est acceptée,
- ✓ abandonne l'exécution du programme en cours et va exécuter un sous-programme de service de l'interruption (ISR : Interrupt Service Routine);
- ✓ après l'exécution de l'ISR, les registres sont restaurés à partir de la pile et le microprocesseur reprend l'exécution du programme qu'il avait abandonné.

14

Prise en charge d'une interruption par le µP



Prise en charge d'une interruption par le µP

Différence entre les instructions

CALL et JUMP

Interruptions sont vectorisées.

Lorsqu'une interruption est acceptée, le microprocesseur a besoin de connaître l'adresse du sous-programme de service correspondant à cette interruption. Pour cela, la source d'interruption place sur le bus de données un numéro indiquant la nature de cette interruption.

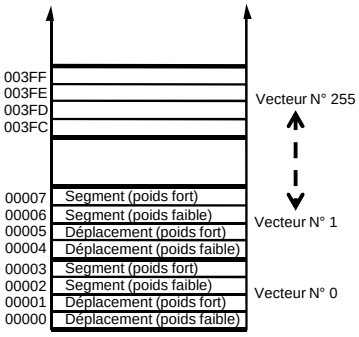
Le microprocesseur utilise ce numéro pour rechercher dans une table stockée en mémoire centrale (table des vecteurs d'interruptions) l'adresse du sous-programme d'interruption à exécuter. Chaque élément de cette table s'appelle un vecteur d'interruption.

L'emplacement du sous programme d'interruption peut donc être placé n'importe où dans la mémoire, il suffit de spécifier le vecteur d'interruption lui correspondant.

Cas du microprocesseur 8086

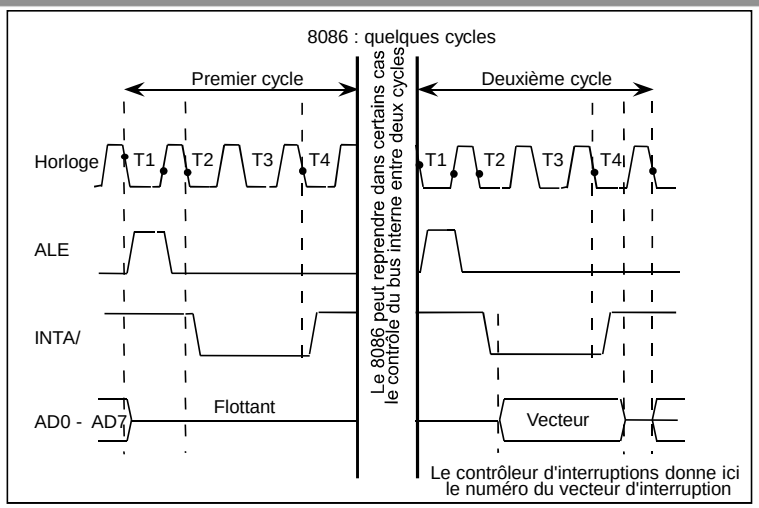
Le 8086 dispose de 256 sources d'interruptions possibles :

- L'interruption masquable **INTR** : quand une demande sur l'entrée INTR est acceptée par le CPU, un signal externe, appelé reconnaissance d'interruption (INTA, pour Interrupt Acknoledge), est généré.
- L'interruption Non Masquable **NMI** : une transition ascendante sur cette entrée provoque une interruption, non masquable par le logiciel, effectuant le branchement au sous-programme dont l'adresse est définie par le vecteur N°2.

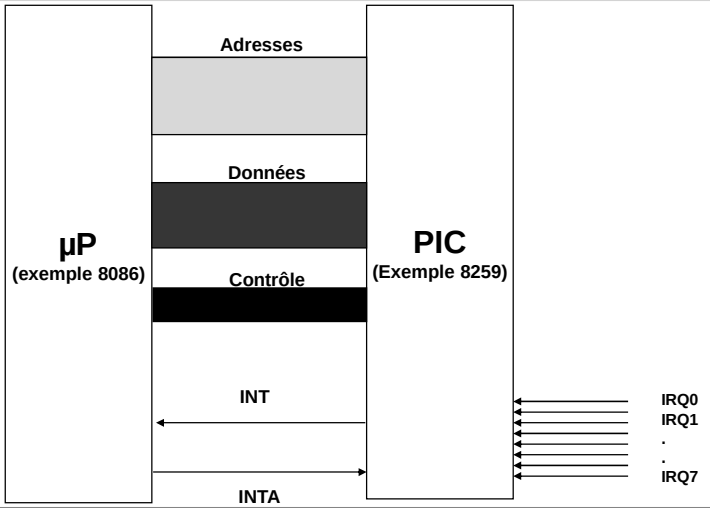


- *(La remise à l'état initial **RESET** : le RESET a pour effet d'initialiser le pointeur de code à sa valeur de démarrage).*

Cas du microprocesseur 8086
(Acquitement d'une interruption)



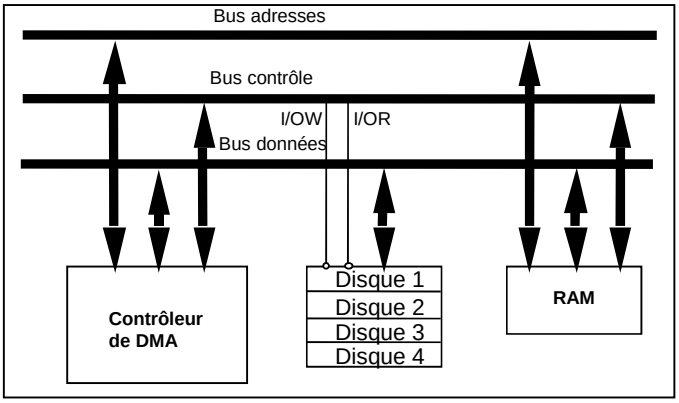
Gestion d'interruptions multiples (Utilisation d'un PIC)



20

Principe du DMA
Accès Direct à la Mémoire du sytème

L'opération de DMA consiste à remplacer un processus de de transfert **logiciel** par un processeur de transfert **matériel. La rapidité de transfert n'est linmitée que par la vitesse de la mémoire.**



Exemple : transmission série

Types de communication séries

communication en tension :
un niveau "1" (appelé MARK) est donné par une tension (souvent moins V), un niveau "0" (appelé SPACE) est donné par une tension (souvent plus V).

communication en courant :
- La boucle simple courant indique un état "1" par un courant (généralement 20 mA) et un état "0" par l'absence de courant.
- La boucle double courant définit les états "0" et "1" par le sens dans lequel circule le courant. (l'absence de courant indique un incident).

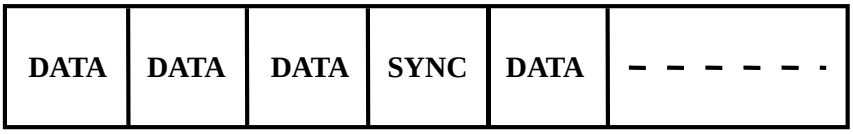
communication par Modem:
Les informations sont converties en signaux analogiques qui peuvent être **modulés** en amplitude, en fréquence ou en phase et qui sont envoyés sur la ligne de transmission. A la réception, ces signaux sont **démodulés** et reconvertis en signaux logiques.

23

Transmision série synchrone

L'émetteur et le récepteur sont synchronisés. Ils utilisent **la même horloge ou des horloges synchronisées en fréquence et en phase**. La transmission synchrone se distingue essentiellement par :

- l'apparition de **mots (ou de ligne) de synchronisation**
- une émission continue,

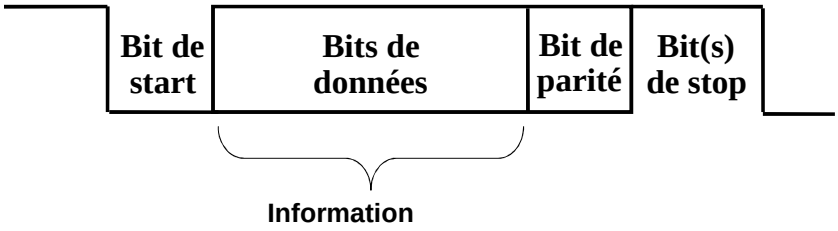


Exemple : Le bus SPI (Serial Peripheral Interface), c'est une liaison série synchrone de 3 fils, 2 Mbps. (Le maître est le seul qui peut initier un dialogue avec l'esclave. Plusieurs esclaves sont possibles).

24

Transmision série asynchrone

Autorise l'utilisation d'horloges distinctes à l'émission et à la réception si les fréquences de ces horloges sont proches. La transmission asynchrone est caractérisée par un certain nombre de **bits qui complètent la donnée** à transmettre.



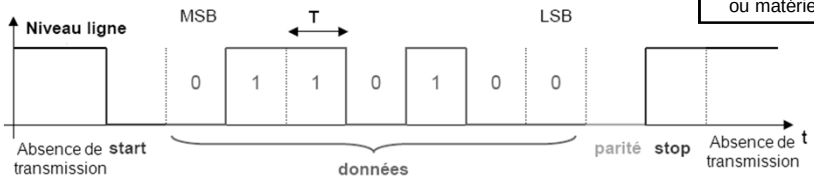
Fonctionnement de la transmision série asynchrone

En l'absence de transmission, la liaison est au repos, **donc au niveau haut**, afin de détecter une éventuelle coupure sur le support de transmission.

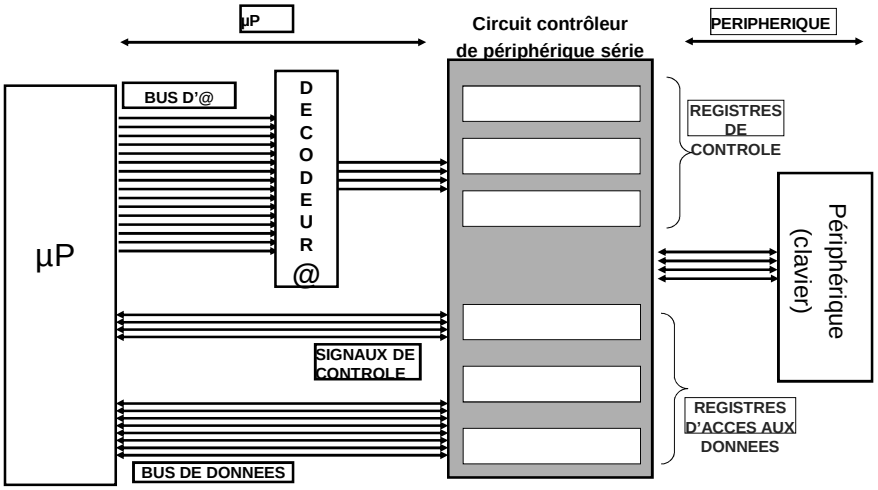
Etapes de la transmission :

1. L'émetteur positionne la ligne à l'état bas : c'est le **bit de START**.
2. Les bits sont transmis les un après les autres, **en commençant par le bit de poids fort**.
3. Le **bit de parité** est éventuellement transmis.
4. L'émetteur positionne la ligne à l'état haut : c'est le **bit de stop**.

Le contrôle de flux peut être réalisé de manière logiciel ou matériel.

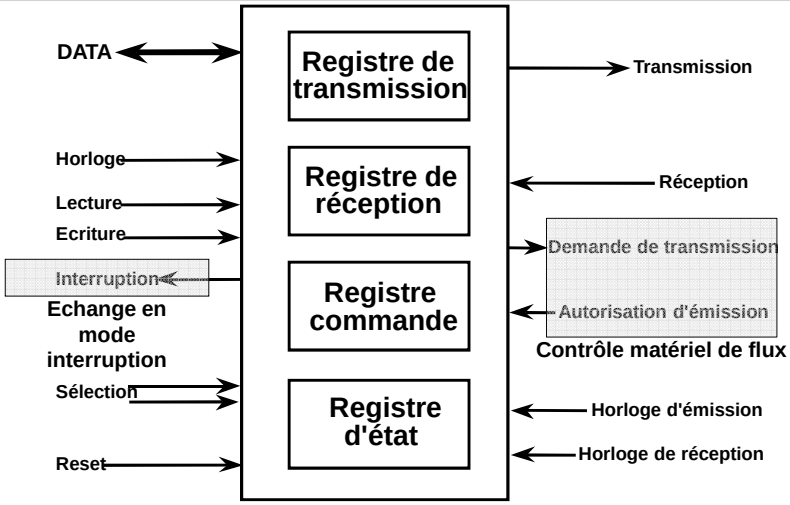


Principe de mise en œuvre



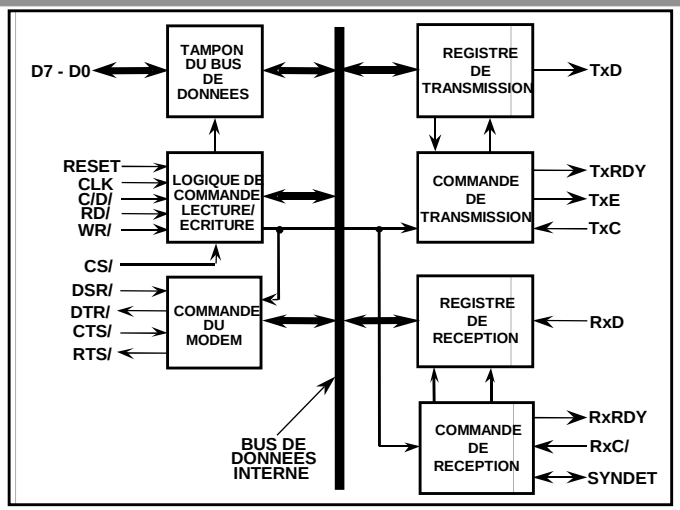
27

Structure d'une interface série UART

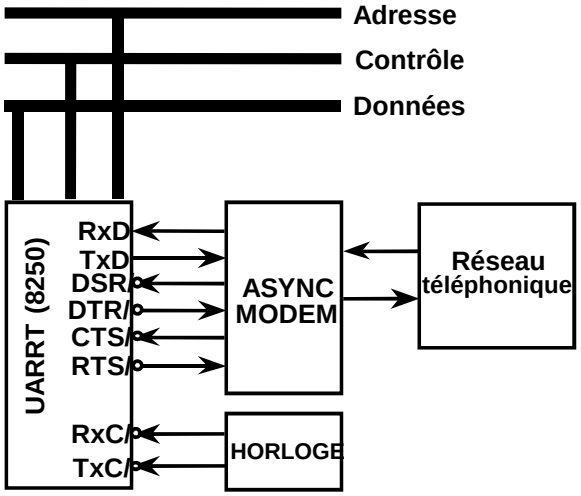


28

Exemple : Le 8250/8251 de Intel



Connexion à un MoDem



Avez-vous des questions ?

Questions de révisions

- Le mot périphérique sous entend :
  - Clavier, souris, carte réseau, carte graphique.
  - Imprimante, écran, machine industrielle.
  - Liaison série, liaison parallèle, liaison USB, liaison I2C, liaison internet.
- Le contrôleur de périphérique :
  - Sert à décoder les adresses et à générer les CS ‘Chip Select’ des circuits périphériques.
  - Permet d’adapter les bus (adresses, données et contrôle) du microprocesseur aux bus du dispositif périphérique,
  - Permet la synchronisation de l’échange entre le microprocesseur et le dispositif périphérique.
  - Fait partie du dispositif périphérique.
  - Fait partie du système à microprocesseur.
- Les méthodes d’échanges entre le système à microprocesseur et l’extérieur sont :
  - L’échange programmé et l’accès direct à la mémoire.
  - L’échange par interruption et la liaison série.
  - L’échange par liaison parallèle, l’échange par liaison série et l’échange par interruption.
  - L’échange programmé, l’échange par interruption et l’accès direct à la mémoire.

Questions de révisions

- Le vecteur d'interruption désigne :
  - L'espace mémoire réservé pour le sous programme d'interruption.
  - L'adresse du sous programme d'interruption,
  - Le sous programme d'interruption.
- Quelles sont les principales différences entre les modes de transmission synchrone et asynchrone ?
- Quel est le débit en caractères (7 bits de données + le bit de parité) d'une ligne de 4800 bits/seconde dans chacun des modes de transmission synchrone et asynchrone avec un bit de START et un bit de STOP ?
- Représenter l'allure du signal émis dans le cas de la transmission du caractère « A » en mode asynchrone avec un contrôle de parité impaire ?

33

Exercice : Gestion des entrées sorties (voir annexe)

On souhaite utiliser un système basé sur le microprocesseur CPE8 pour surveiller l'état (ouvert ou fermé) de huit fenêtres d'une salle de cours. Pour cela des capteurs TOR (Tout Ou Rien) sont placés au niveau de chacune des huit fenêtres de la salle à surveiller. Les sorties des capteurs TOR sont reliés au microprocesseur par un circuit 74LS573. Après traitement, les états (ON ou OFF) des fenêtres sont affichés sur 8 LEDs.

Expliquer comment faire communiquer les capteurs TOR et les LEDs avec le microprocesseur.

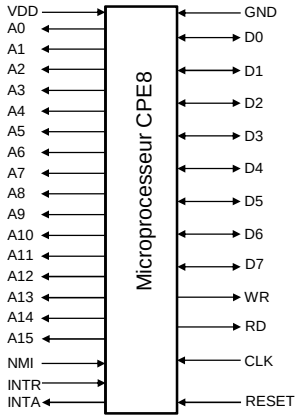
Donner le **schéma électrique** de mise en œuvre des capteurs TOR et des LEDs ?

Donner l'algorithme de **programmation de la lecture des capteurs TOR et de l'affichage de leur état sur les LED ?**

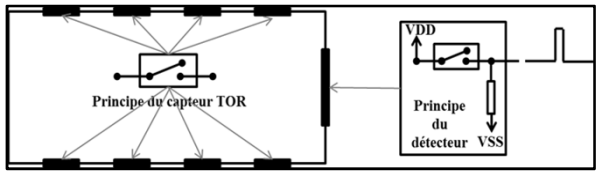
Est-il possible d'envisager un **fonctionnement par interruption ?**

34

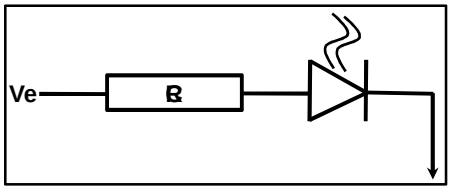
Annexe : CPE8, TOR et LED



Microprocesseur CPE8



Mise en œuvre TOR



Mise en œuvre LED

Annexe : le 74LS573

1. General description

The 74HC573; 74HCT573 is a high-speed Si-gate CMOS device and is pin compatible with Low-power Schottky TTL (LSTTL). It is specified in compliance with JEDEC standard no. 7A.

The 74HC573; 74HCT573 has octal D-type transparent latches featuring separate D-type inputs for each latch and 3-state true outputs for bus-oriented applications. A latch enable (LE) input and an output enable ($\overline{OE}$) input are common to all latches.

When LE is HIGH, data at the Dn inputs enter the latches. In this condition, the latches are transparent, i.e. a latch output changes state each time its corresponding D input changes.

When LE is LOW the latches store the information that was present at the D-inputs a set-up time preceding the HIGH-to-LOW transition of LE. When $\overline{OE}$ is LOW, the contents of the 8 latches are available at the outputs. When $\overline{OE}$ is HIGH, the outputs go to the high-impedance OFF-state. Operation of the $\overline{OE}$ input does not affect the state of the latches.

The 74HC573; 74HCT573 is functionally identical to:

- 74HC563; 74HCT563, but inverted outputs
- 74HC373; 74HCT373, but different pin arrangement

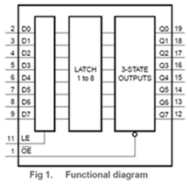


Fig 1. Functional diagram

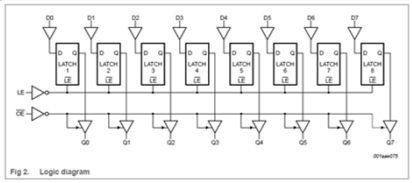


Fig 2. Logic diagram

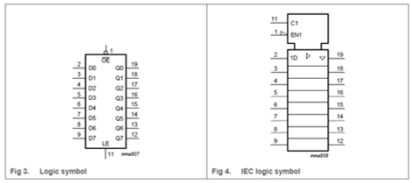


Fig 3. Logic symbol

Fig 4. IEC logic symbol

6. Functional description

Table 3. Function table[1]

Operating mode	Control		Input	Internal latches	Output
	$\overline{OE}$	LE	Dn		On
Enable and read register (transparent mode)	L	H	L	L	L
			H	H	H
Latch and read register	L	L	l	L	L
			h	H	H
Latch register and disable outputs	H	L	l	L	Z
			h	H	Z