

Chapitre 5 : Les interfaces d'entrées/sorties

V.1 Définition

Une interface d'entrées/sorties est un circuit permettant au Cpu de communiquer avec l'environnement extérieur (périphérique) : clavier, écran, imprimante, processus industriel etc...

Les interfaces d'E/S sont connectées au microprocesseur à travers les bus d'adresses de données et de commandes.

Un circuit d'E/S possède des registres pour gérer les échanges avec les périphériques :

- Registre de configuration
- Registre de données

A chaque registre est assignée une adresse : le microprocesseur accède à un port d'E/S en spécifiant l'adresse de l'un de ses registres.

Intel a développé plusieurs circuits intégré de contrôle de périphérique conçus pour soutenir la famille de processeurs 80x86 tels que:

- Le 8255A Programmable Peripheral Interface (PPI),
- Le 8259 Programmable Interrupt Controller (PIC),
- Le 8253/54 Programmable Interval Timer (PIT),
- Le 8237 Programmable DMA Controller.

V.2 L'interface parallèle 8255

Le rôle d'une interface parallèle est de transférer des données du microprocesseur vers un périphérique et l'inverse en parallèle. Le 8255 est une interface parallèle programmable, elle peut être configurée en entrée et/ou en sortie par programme.

Le 8255 fournit 24 lignes d'E/S qui peuvent être organisées en trois ports d'E/S (A, B, et C) de 8 bits chacun. De plus le 8255 peut fonctionner selon 3 modes: mode 0, mode1 ou mode2.

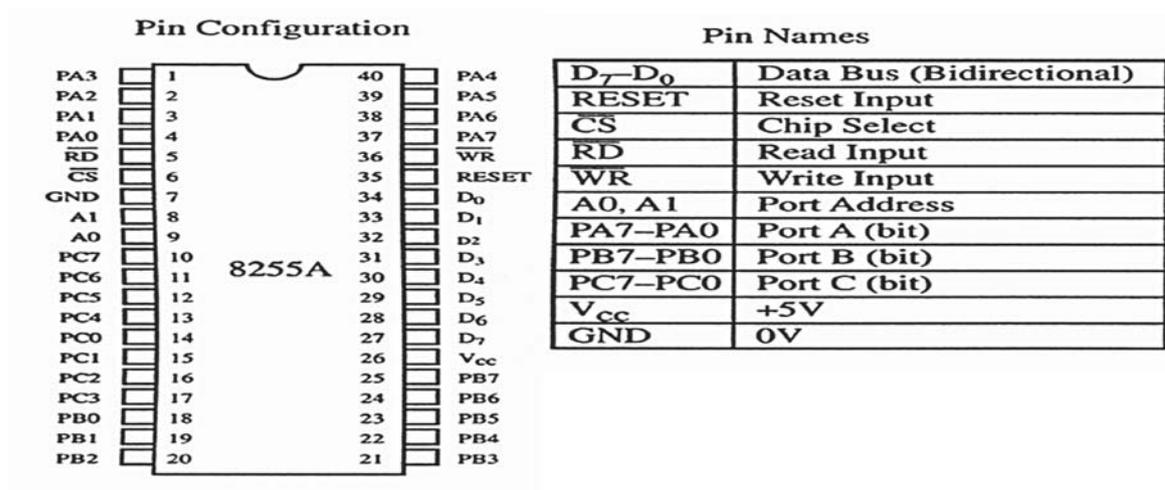


Figure V.1: configuration des broches du 8255

Tableau V.1: Sélection des ports de 8255

A_1	A_0	\overline{RD}	\overline{WR}	\overline{CS}	
0	0	0	1	0	<i>Input operation (READ)</i>
0	1	0	1	0	Port A → data bus
1	0	0	1	0	Port B → data bus
					Port C → data bus
					<i>Output operation (WRITE)</i>
0	0	1	0	0	Data bus → port A
0	1	1	0	0	Data bus → port B
1	0	1	0	0	Data bus → port C
1	1	1	0	0	Data bus → control
X	X	X	X	1	Data bus tristate
1	1	0	1	0	Illegal condition
X	X	1	1	0	Data bus tristate

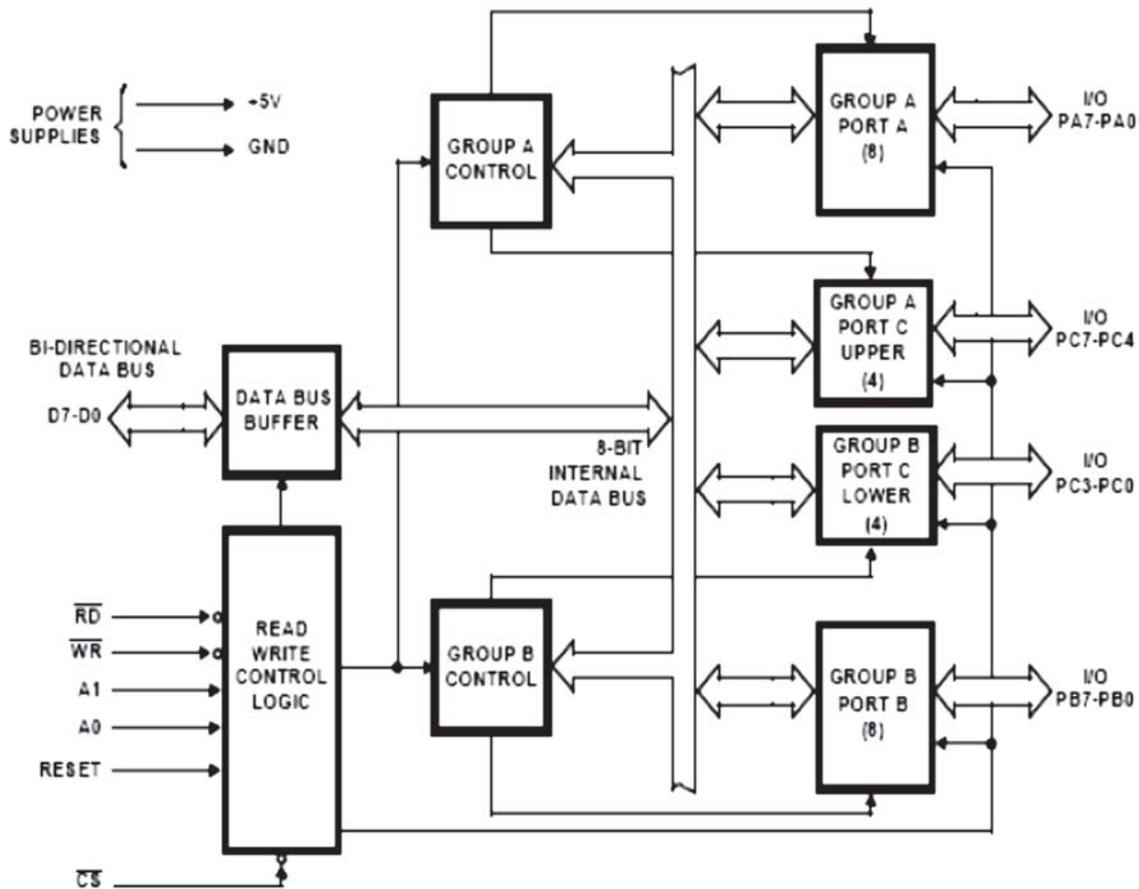


Figure V.2: Schéma fonctionnel du 8255

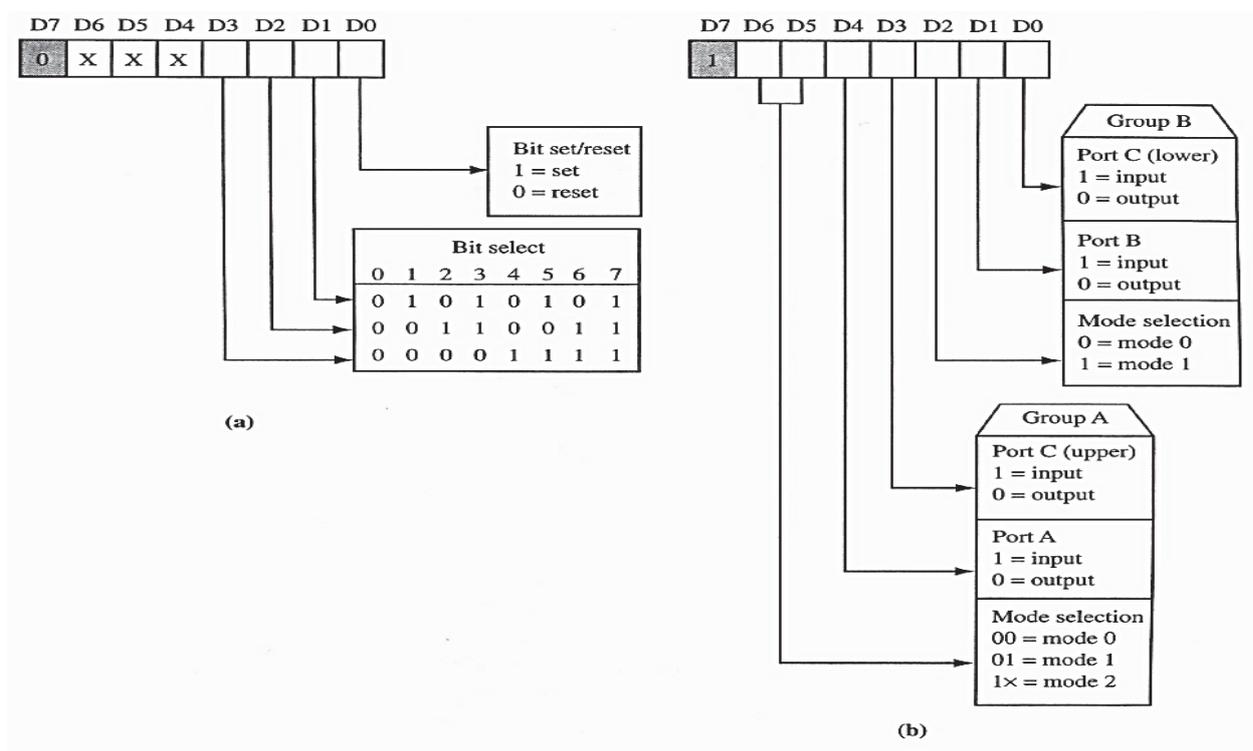


Figure V.3: Structure du registre de commande

La figure précédente montre le fonctionnement du PPI 8255 :

- (a) Lorsque le bit 7 = 0, mise à « 1 » ou « 0 » d'une ligne du PORTC individuellement.
- (b) Lorsque le bit 7 = 1, l'un des modes 0, 1, ou 2 peuvent être programmés.

Mode 0 (E/ S de base): Ports A et B fonctionnent comme entrées ou sorties. Le port C est divisé en deux groupes de 4 bits qui peuvent être configurés comme entrées ou sorties.

Mode 1 : Il est utilisé pour le dialogue avec des périphériques nécessitant un asservissement (contrôle). Ports A et B fonctionnent comme des entrées ou sorties comme en mode 0. Port C est utilisé pour le contrôle.

Mode 2: le port A est bidirectionnel (entrée et sortie). Port C est utilisé comme signaux du contrôle. Port B n'est pas utilisé.

Remarque : Ces modes peuvent aussi être mélangés. Par exemple, le port A peut être programmé pour fonctionner en mode 2, tandis que le port B fonctionne en mode 0.

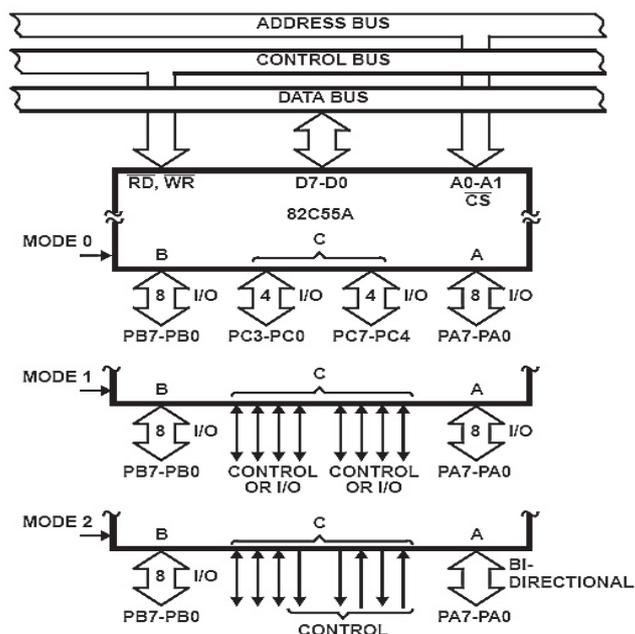


Figure V.4 : Les trois modes de base du 8255

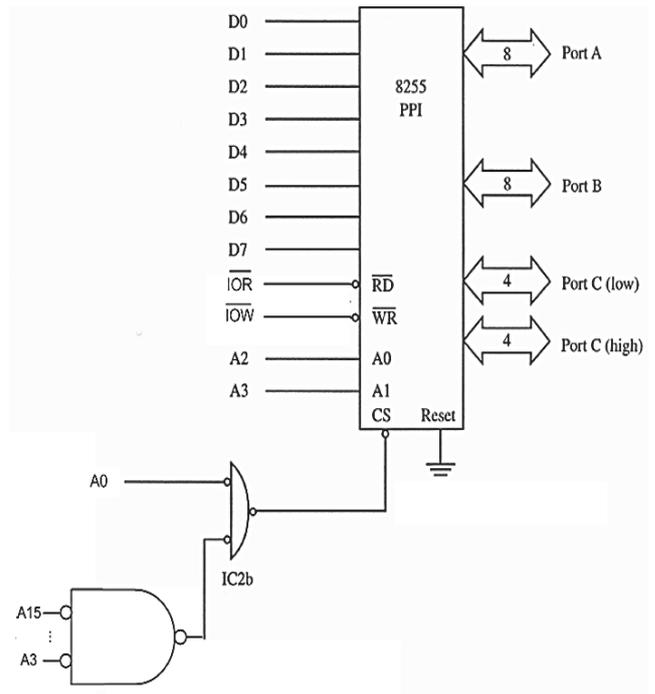
V.2.1 Interfaçage de 8086 avec le 8255

On peut connecter le bus de données du 8255 sur les lignes de données de poids faible du 8086 (D0 - D7) ou sur celles de poids fort (D8 - D15).

Donc l'un des deux signaux A0 ou BHE doit être utilisé pour sélectionner le 8255 alors les adresses des registres du 8255 se trouvent à des adresses paires (validation par A0) ou impaires (validation par BHE).

Exemple :

A15-A12	A11-A8	A7-A4	A2	A1	A0
0000	0000	0000	0	0	0
0000	0000	0000	0	1	0
0000	0000	0000	1	0	0
0000	0000	0000	1	1	0
Chip Select (CS)			Sélection de port		adresses paires



V.3 Le 8279

Intel 8279 est un circuit intégré d'interface programmable clavier / l'écran : Scans et encode jusqu'à un clavier de 64-clé.

Contrôle jusqu'à un affichage numérique de 16 chiffres.

La Section Clavier possède une mémoire tampon de 8 caractères FIFO intégrée.

L'affichage est commandé à partir d'une RAM interne 16x8 qui mémorise les informations d'affichage codées.

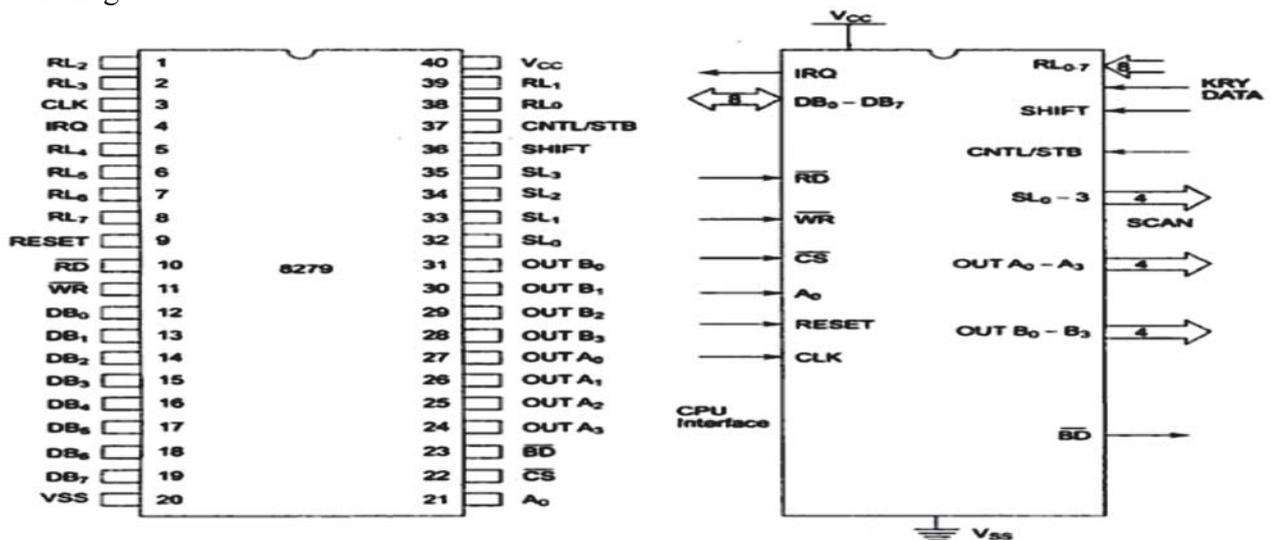


Figure V.5 : (a) Schéma fonctionnel et (b) brochage du 8279.

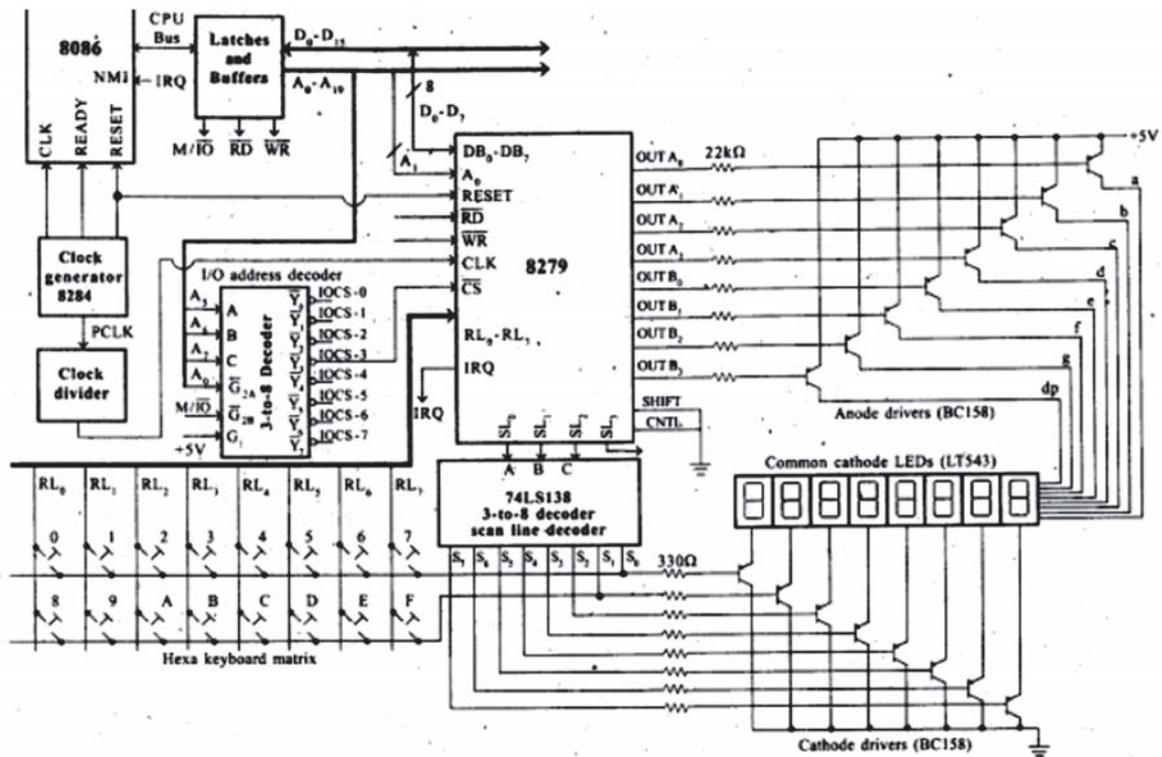


Figure V.6: Interface du 8279 (Un clavier Hexa et afficheur 7 segments) avec le 8086.

V.4 L'interface série 8250

L'interface entrées/sorties permet d'échanger des données entre le microprocesseur et un périphérique soit d'une manière parallèle (un mot de 8 bits) ou d'une manière série (bit par bit).

Mais sur des distances supérieures à quelques mètres, il est difficile de mettre en œuvre une transmission en parallèle. On utilise alors une liaison série.

Le circuit intégré 8250 est le composant chargé de la gestion des transmissions séries asynchrones il est appelé UART (Universal Asynchronous Receiver Transmitter).

Il existe deux types de transmissions séries:

- synchrone: les octets successifs sont transmis par blocs séparés par des octets de synchronisation.
- asynchrone : chaque octet peut être émis ou reçu sans durée déterminée entre un octet et le suivant ;

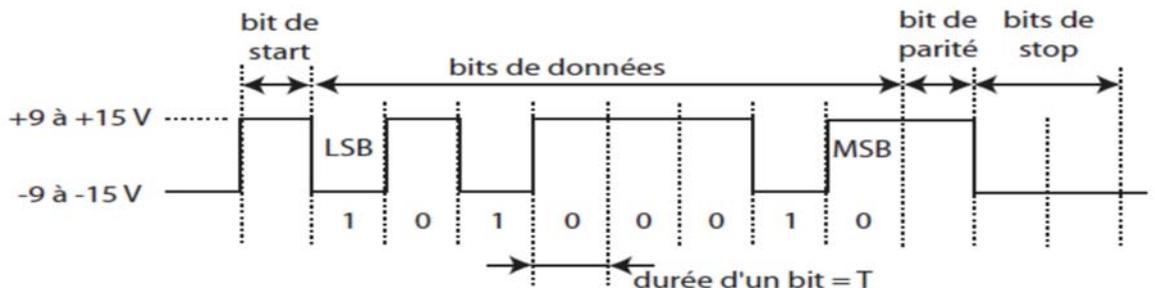


Figure V.7: Protocole de transmission de données asynchrone

- dans le protocole de transmission de données on a que deux états significatifs : 0 (ou low),

et 1 (ou high). Quand on ne transmet rien, la ligne est à l'état high (tension négative). Comme le montre la figure V.7,

- Le bit de Start permettra au récepteur de savoir que des données vont être transmises;
- Les bits de données sont transmis l'un après l'autre en commençant par le bit de poids faible. Ils peuvent être au nombre de 5, 6, 7 ou 8.
- Le bit de parité est un bit supplémentaire dont la valeur dépend du nombre de bits de données égaux à 1. Il est utilisé pour la détection d'erreurs de transmission ;
- Les bits de stop (1, 1.5 ou 2) marquent la fin de la transmission du caractère.

V.4.1 Structure de l'UART

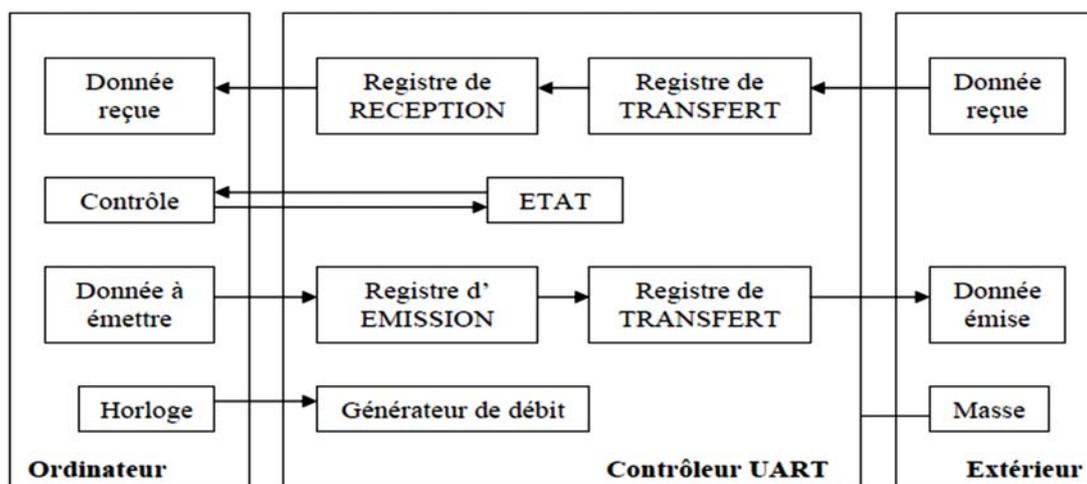


Figure V.8: Structure de l'UART

L'UART est conçue pour gérer les signaux en provenance d'un modem. Aussi est-il nécessaire de donner quelles broches sont utilisées par l'ETCD (Equipement Terminal de Communication de Données).

signal	Signification
TxD	Transmit Data
RxD	Receive Data
RTS	Request To Send
CTS	Clear To Send
DTR	Data Terminal Ready
DSR	Data Set Ready
DCD	Data Carrier Detect
RI	Ring Indicator

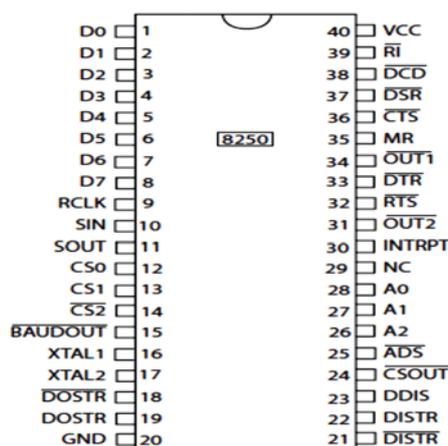


Figure V.9 : Brochage du 8250

Les 2 signaux TxD et RxD servent à transmettre les données. Les autres signaux sont des signaux de contrôle de l'échange de données.

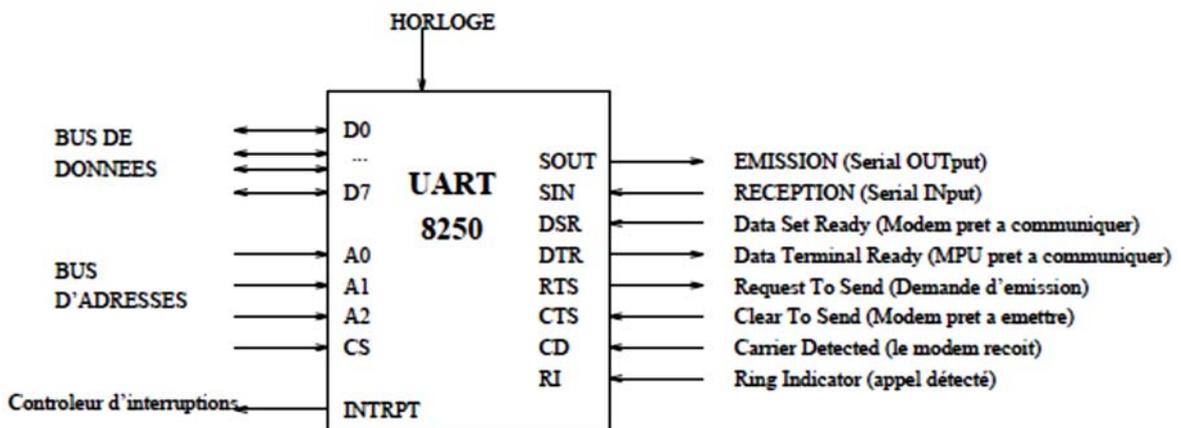


Figure V.10 : Bornes du circuit UART 8250.

V.4.2 Les registres du 8250

L'interface 8250 possède plus de 8 registres de 8 bits permettant de gérer la communication. Ces registres sont lus et modifiés par le processeur par le biais des instructions IN et OUT vues plus haut.

Comme on ne dispose que de 3 bits d'adresse :

(A0, A1 et A2), plusieurs registres doivent se partager la même adresse. La solution est d'utiliser un bit d'un registre spécial, **DLAB**.

En fonction de l'état de **DLAB** (Divisor Latch Access Bit), on a accès soit au registre d'émission / réception, soit au diviseur d'horloge, soit au masque d'interruptions.

Adresse				REGISTRES
DLAB	A2	A1	A0	
0	0	0	0	RBR : Receiver Buffer (registre de réception)
0	0	0	0	THR : Transmitter Holding Register (registre d'émission)
1	0	0	0	DLL : Divisor Latch LSB (poids faible diviseur horloge)
1	0	0	1	DLM : Divisor Latch MSB (poids fort diviseur horloge)
0	0	0	1	IER : Interrupt Enable Register
x	0	1	0	IIR : Interrupt Identification Register
x	0	1	1	LCR : Line Control Register
x	1	0	0	MCR : Modem Control Register
x	1	0	1	LSR : Line Status Register
x	1	1	0	MSR : Modem Status Register
x	1	1	1	non utilisé

Dans ce cours on ne va pas étudier le fonctionnement du 8250 ni sa programmation car d'habitude elle n'est pas utilisée d'une façon individuelle mais elle fait partie du système de communication.

Chapitre 6 : Les interruptions

VI.1 Définition d'une interruption

Dans un système à microprocesseur pour dialoguer avec ces périphériques le microprocesseur a deux façons de communiquer avec ces derniers :

- **polling (scrutation périodique)** : En questionnant de façon continue le périphérique pour vérifier que des données peuvent être lues ou écrites. Cela permettra de limiter les tâches qui pourraient être accomplies par le micro-ordinateur.
- **interruption** : En l'interrompant lorsqu' un périphérique est prêt à lire ou écrire des données, avec une demande d'interruption (**IRQ** : Interrupt Request).

En utilisant cette technique, le processeur peut passer la plupart de son temps à d'autres tâches, et n'exécute une lecture des ports d'E/S que lorsqu' une donnée est disponible.

Le processeur 8086 reçoit les interruptions de trois sources différentes:

- (1) **interruptions processeur** : c'est le processeur lui-même qui les génère, en raison d'un défaut interne (par exemple, une tentative de diviser par zéro)
- (2) **L'interruption logicielle** : à l'aide de l'instruction **INT n** (couramment utilisée dans le PC pour accéder au BIOS et fonctions DOS),
- (3) **interruptions matérielles** : produites par l'activation des lignes **INTR** et **NMI** du microprocesseur.

A la suite d'une demande d'interruption par un périphérique :

- le microprocesseur termine l'exécution de l'instruction en cours ;
- il range le contenu des principaux registres sur la pile de sauvegarde : CS : IP, flags, ...
- il émet un accusé de réception de demande d'interruption (Interrupt Acknowledge) indiquant au circuit d'E/S que la demande d'interruption est acceptée.
- il abandonne l'exécution du programme en cours et va exécuter un sous-programme de service de l'interruption (**ISR** : Interrupt Service Routine) ;
- il termine l'exécution de l'ISR avec l'instruction **IRET** (retour d'interruption).
- les registres sont récupérés à partir de la pile et le microprocesseur reprend l'exécution du programme qu'il avait abandonné.

Le processeur 8086 a seulement 2 broches interruption matérielle: INM ; INTR.

- **NMI (interruption non masquable)** : Elle ne peut pas être bloquée; le processeur doit répondre. Pour cette raison l'entrée NMI est habituellement réservée pour les fonctions critiques du système.
- **INTR (Interruption)** : Elle est masquable via le drapeau **IF**.

Pour (8086), l'adresse de la routine de service d'interruption est stockée dans 4 emplacements de mémoire consécutifs (un double-mot) dans une table de vecteur d'interruption commençant à l'adresse 00000h.

Remarque : **adresse vecteur d'interruption** = 4 × type de l'interruption

Exemple : interruption 10H, adresse du vecteur = 4 × 10H = 40H.

VI.2 Le contrôleur programmable d'interruptions 8259

Le microprocesseur 8086 ne dispose que de deux lignes de demandes d'interruptions matérielles (NMI et INTR). Pour pouvoir connecter plusieurs périphériques utilisant des interruptions, on peut utiliser le contrôleur programmable d'interruptions.

Un contrôleur d'interruption programmable (PIC) fonctionne comme un directeur général dans un environnement de système commandé par interruption.

Il accepte les demandes provenant de l'équipement périphérique, détermine laquelle des demandes entrantes est de la plus haute importance (priorité), vérifie si la demande entrante a une valeur de priorité plus élevée que le niveau en cours d'exécution et émet une interruption vers le processeur sur la base de cette détermination.

Exemple : le rôle de PIC 8259 :

- recevoir des demandes d'interruptions des périphériques ;
- résoudre les priorités des interruptions ;
- générer le signal INTR pour le 8086 ;
- émettre le numéro de l'interruption sur le bus de données.

Le 8259A peut gérer jusqu'à 8 demandes d'interruptions matérielles. Il est programmé par le logiciel du système comme un périphérique d'E / S.

Une sélection de modes de priorité est disponible pour le programmeur de sorte que la manière dont les demandes sont traitées par le 8259A peut être configurée pour correspondre à ses besoins du système.

Les modes de priorité peuvent être modifiés ou reconfigurés dynamiquement à tout moment pendant le programme principal. Cela signifie que la structure d'interruption complète peut être définie selon les besoins.

VI.3 Brochage du 8259 :

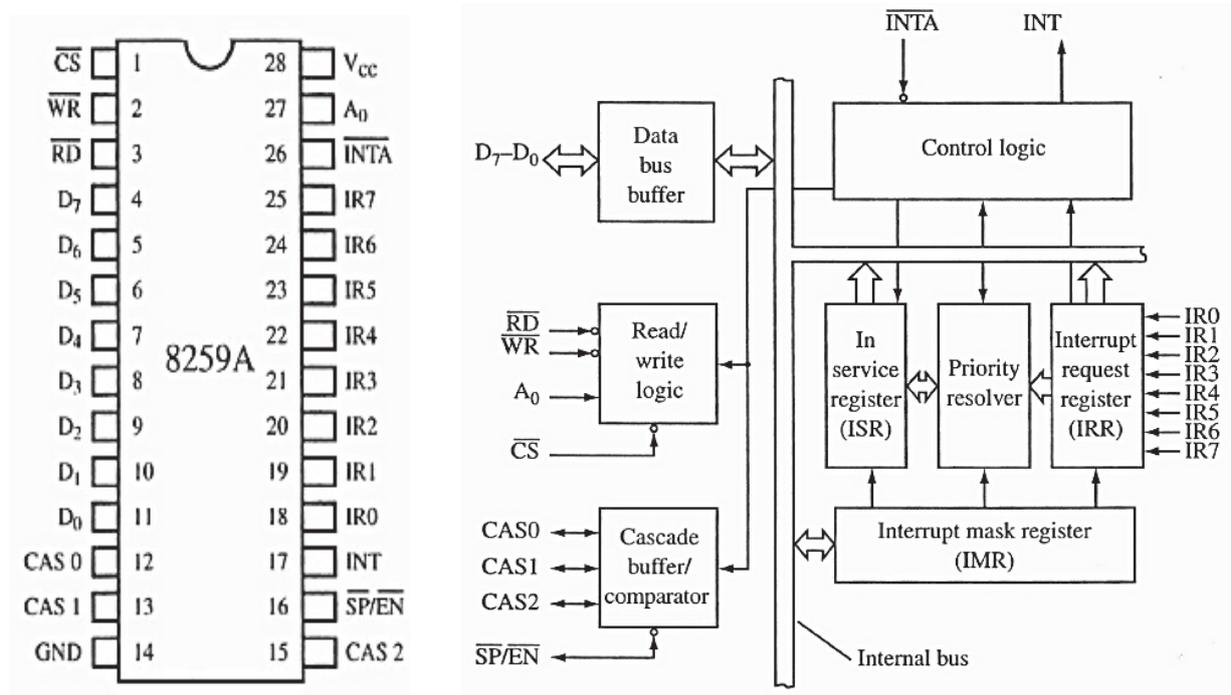


Figure VI.1 : Brochage et Schéma fonctionnel du 8259 .

Exemple : interfaçage de 8259 avec le 8086 :

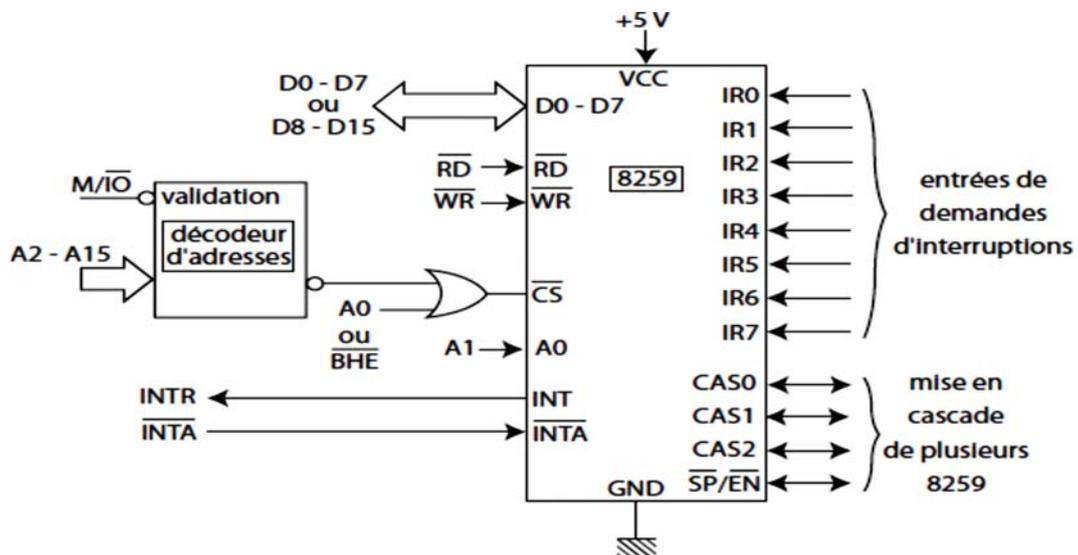


Figure VI.2 : interfaçage de 8259 avec le 8086 .

Remarque : si le nombre de demandes d'interruptions est supérieur à 8, on peut placer plusieurs 8259 en cascade.

VI.3 Le temporisateur programmable 8253/8254

On peut facilement exécuter une boucle de temporisation mais lorsque le microprocesseur exécute la temporisation il ne peut plus rien faire d'autre. Pour cela on fait appel à un circuit spécialisé c'est le PROGRAMMABLE TIMER.

Le 8253 programmable interval timer/counter est spécialement conçu pour les circuits d'Intel. Le programmeur configure le 8253 pour correspondre à ses besoins, initialise un des compteurs du 8253 avec la quantité désirée, puis avec le mot de commande le 8253 prendra en charge la temporisation et signalera la fin d'exécution avec une interruption du CPU.

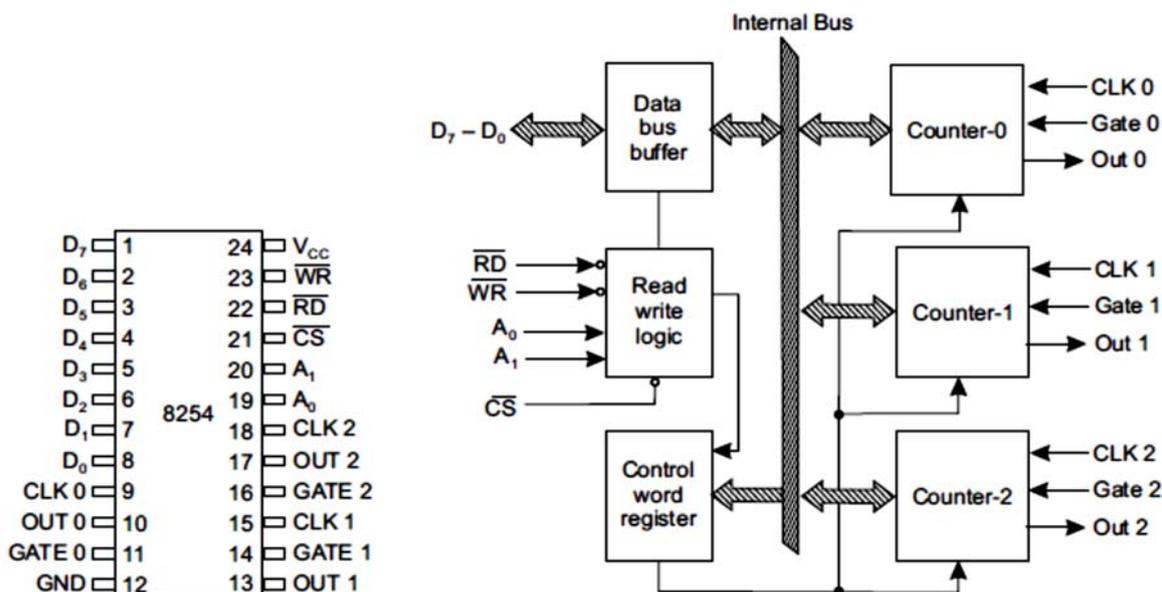


Figure VI.3 : Schéma fonctionnel et les descriptions des broches du 8254.

Il y a trois registres de comptage de 16 bits, dont chacun peut être programmé comme un timer ou un compteur d'événements. Plus un registre de commande qui peut ne peut-être qu'écrit.

L'accès aux registres du 8254 s'effectue de la façon suivante :

Tableau VI.1: Description des opérations de base

CS	RD	WR	A1	A0	fonction
0	1	0	0	0	Chargement compteur 0
0	1	0	0	1	Chargement compteur 1
0	1	0	1	0	Chargement compteur 2
0	1	0	1	1	Ecriture mot de mode
0	0	1	0	0	Lecture compteur 0
0	0	1	0	1	Lecture compteur 1
0	0	1	1	0	Lecture compteur 2
0	0	1	1	1	Etat de haute impédance
1	X	X	X	X	
0	1	1	X	X	

Table VI.2: Pin description of the 8254.

Pin Symbol	Name	Input/Output	Function
D ₇ - D ₀	Bidirectional Data Bus	Input/Output	Three-state 8-bit bidirectional data bus used when writing control words and count values, and reading count values upon reception of \overline{WR} and \overline{RD} signals from CPU.
\overline{CS}	Chip Select Input	Input	Data transfer with the CPU is enabled when this pin is at low level. When at high level, the data bus (D ₀ thru D ₇) is switched to high impedance state where neither writing nor reading can be executed. Internal registers, however, remain unchanged.
\overline{RD}	Read Input	Input	Data can be transferred from MSM82C53-2 to CPU when this pin is at low level.
\overline{WR}	Write Input	Input	Data can be transferred from CPU to MSM82C53-2 when this pin is at low level.
A ₀ - A ₁	Address Input	Input	One of the three internal counters or the control word register is selected by A ₀ /A ₁ combination. These two pins are normally connected to the two lower order bits of the address bus.
CLK ₀₋₂	Clock Input	Input	Supply of three clock signals to the three counters incorporated in MSM82C53-2.
GATE ₀₋₂	Gate Input	Input	Control of starting, interruption, and restarting of counting in the three respective counters in accordance with the set control word contents.
OUT ₀₋₂	Counter Output	Output	Output of counter output waveform in accordance with the set mode and count value.

La programmation du 8254 se fait par l'envoi d'un mot de commande dans le registre de commande CR. Les bits 6 et 7 de ce mot de commande permettent de l'affecter à l'un des trois compteurs. Le format du mot de commande est le suivant :

7 6 5 4 3 2 1 0							
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC - Selection du compteur		
SC1	SC0	Description
0	0	Compteur 0
0	1	Compteur 1
1	0	Compteur 2

RW - Lecture/Ecriture		
RW1	RW0	Description
0	0	Compteur Latch
0	1	D'abord LSB
1	0	D'abord MSB
1	1	D'abord LSB puis MSB

M - Mode			
M2	M1	M0	Description
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD - Format du compteur	
BCD	Description
0	Compteur binaire 16 bits
1	Compteur BCD 4 décimales

On remarque que chaque compteur peut être utilisé selon l'un des 6 modes possibles :

Mode 0 :

Il Permet au 8254 d'être utilisé comme un compteur ou temporisateur. Dans ce mode, la sortie est au niveau logique 0 Lorsque le décomptage atteint 0, la sortie passe au niveau 1. On notera que l'entrée **GATE** doit être un 1 logique pour permettre au compteur de compter. Si G devient 0 au milieu du comptage, le compteur s'arrête jusqu'à ce que G redevient 1.

Mode 1 :

Le 8254 à la fonction comme un monostable redéclenchable. Un front montant sur la gâchette GATE déclenche le décompte au top d'horloge qui suit. Alors la sortie passe à 0 et s'y maintient jusqu'à ce que le décompte arrive à 0. Chaque front montant sur GATE relance le processus à partir du compte initial (si celui-ci n'a pas été modifié).

Mode 2 :

C'est le mode diviseur par n il permet au 8254 de générer une série d'impulsions continues. La séparation entre les impulsions est déterminée par le comptage. Par exemple, pour un compte de 10, la sortie est un état logique 1 pendant neuf périodes d'horloge et 0 pour une période d'horloge. Ce cycle est répété jusqu'à ce que le compteur est programmé avec un nouveau comptage ou jusqu'à ce que l'entrée G est placée à un niveau logique 0. L'entrée G doit être un 1 logique pour ce mode pour générer une série continue d'impulsions.

Mode 3:

Ce mode est similaire au mode 2 sauf que OUT passe au niveau bas lorsque la moitié du compte initial est atteinte, soit $N/2$, et reste dans cet état jusqu'à ce que le compte arrive à 0 et le cycle recommence. Comme pour le mode 2, un niveau 1 sur GATE valide le décompte et un niveau 0 l'inhibe alors qu'un front montant le réinitialise. De ce fait, une valeur impaire amène $(N+1)/2$ avec sorties au niveau haut et $(N-1)/2$ au niveau bas.

Mode 4:

Ce mode est similaire au mode 0 sauf que OUT est au niveau haut pendant le décomptage et produit une seule impulsion négative lorsque le compte devient nul.

Mode 5:

Ce mode est similaire au mode 4. Cependant il est démarré par une impulsion de déclenchement sur la broche G au lieu par le programme. Ce mode est également similaire au mode 1, car il est redéclenchable.

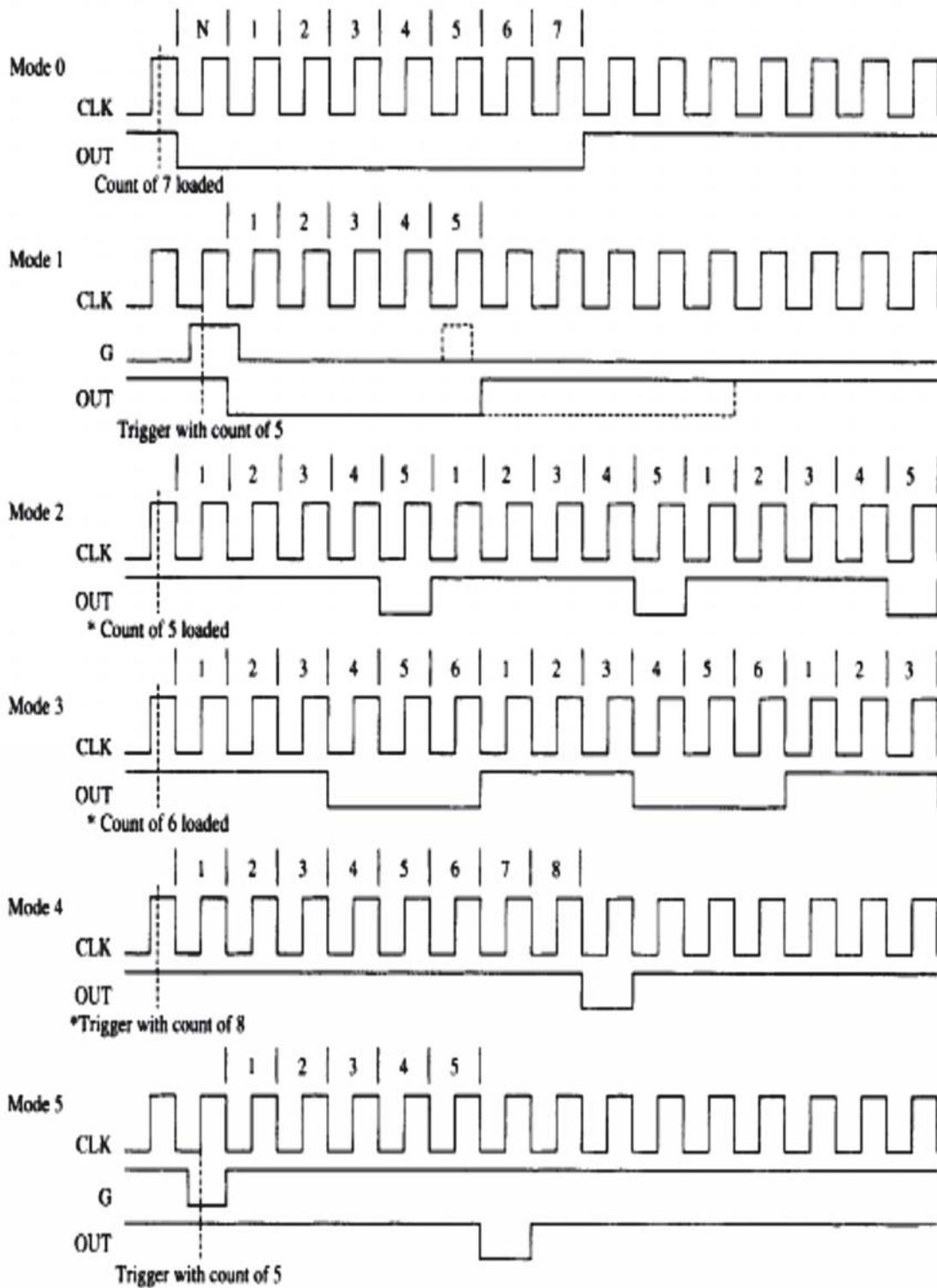


Figure VI.4 : Les six modes de fonctionnement pour le 8254. L'entrée de G = 0 arrête le compteur en mode 2, 3, et 4.

VI.4 DMA (Accès Direct à la Mémoire)

Précédemment, nous avons discuté d'E / S de base et de traitement d'interruption. Maintenant, nous nous tournons vers la forme finale d'E / S appelé accès direct à la mémoire (DMA). La technique DMA offre l'accès directe à la mémoire pendant que le microprocesseur est temporairement désactivé. Ceci permet aux données d'être transférées entre la mémoire et le dispositif d'E / S à une vitesse qui est limitée uniquement par la vitesse des composants de mémoire dans le système ou le dispositif de commande DMA. Les transferts DMA sont utilisés à de nombreuses fins, mais plus fréquents comme pour, transfert de données entre la mémoire et le disque dur, transfert de données vers la carte graphique ou la carte de son, etc.

On utilise pour cela le contrôleur de DMA un circuit intégré qui gère le transfert par DMA, lors de l'initialisation d'un transfert par DMA, le contrôleur de DMA négocie l'accès au bus de données avec le microprocesseur via les deux broches **HRQ** et **HLDA** du microprocesseur :

- Le cycle commence avec la requête du périphérique via une entrée **DREQ** (pour Dma REQuest) du DMAC.
- Le DMAC positionne alors l'entrée **HOLD** du 8086 à niveau haut, requérant ainsi que le microprocesseur entre dans un état HOLD qui laisse la main au DMAC.
- Le microprocesseur répond en terminant le cycle de bus en cours (s'il y en a un) et met ses adresses, données et la plupart des contrôles en position ouverte (haute impédance). La broche **HLDA** (pour **HOLD** Acknowledge) est positionnée au niveau haut par le microprocesseur pour accuser réception de la requête.

Dans un système avec des tampons de bus d'adresse, de données et de contrôle, **HDLA** est utilisé pour désactiver ces tampons de façon à ce que le microprocesseur soit complètement déconnecté de la mémoire et des entrées-sorties.

– Lorsqu'il reçoit **HDLA**, le DMAC applique **DACK** (pour Dma ACknowledge) au périphérique requérant le service. Le DMAC contrôle alors le système, fournissant les signaux de bus d'adresse et de contrôle comme si il était le microprocesseur (ce qu'il est réellement).

– Tant que le DMAC utilise les bus pour des transferts, le microprocesseur est inactif (et réciproquement, lorsque le microprocesseur est actif, le DMAC est inactif). Lorsque le DMAC a terminé son travail, il met **HOLD** à un niveau bas de façon à ce que le microprocesseur reprenne la main.

VI.5 Le 8237

Intel a conçu le DMAC 8237 pour être associé au microprocesseur 8080. Il est également utilisé pour les microprocesseurs 8085 et 8086/88. Le 8237 est en fait un microprocesseur à usage spécial dont le travail est le transfert de données à grande vitesse entre la mémoire et les E / S. la Figure suivante montre le brochage et le diagramme du contrôleur de DMA programmable 8237.

Le 8237 a quatre canaux pour transférer les données, c'est-à-dire qu'il peut être relié à quatre périphériques. Bien entendu, à un instant donné, un seul périphérique peut utiliser le DMAC pour transférer des données.

A chaque canal est associé deux signaux : **DREQ** et **DACK**. Il y a un seul signal HOLD et un seul signal HLDA, ce qui signifie que les quatre canaux utilisent les mêmes bus système, mais le DMAC décide quel périphérique doit prendre le contrôle à partir d'un registre des priorités qui peut être programmé.

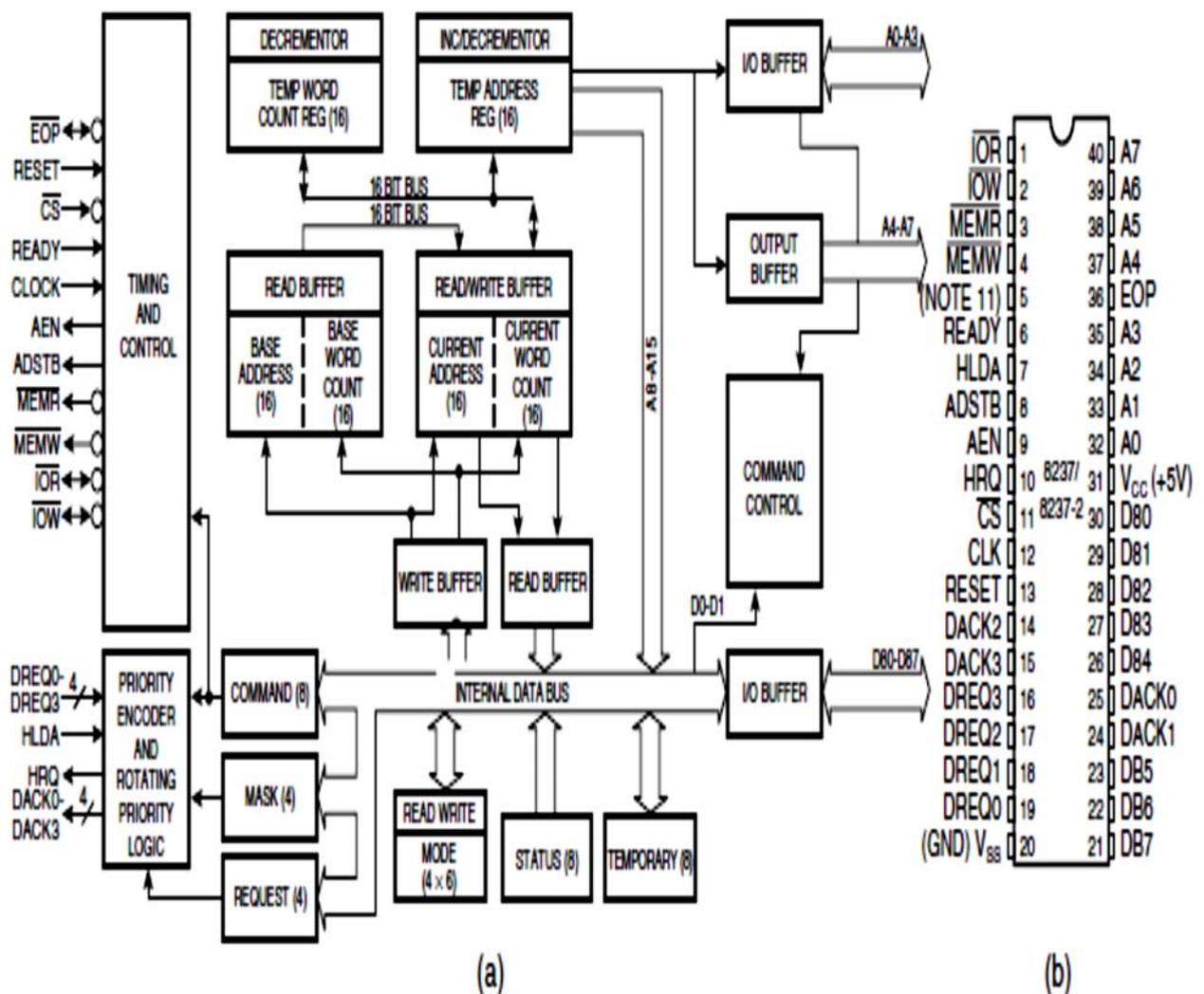


Figure VI.5 : (a) Schéma fonctionnel et (b) brochage du contrôleur DMA programmable 8237.

On ne va pas aller plus loin dans l'étude du contrôleur DMA programmable 8237 car il est rarement utilisé d'une façon individuelle.

Bibliographie

- [1] Brey, Barry B. **The Intel microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-bit extensions: architecture, programming, and interfacing** ;8th ed; Pearson Prentice Hall ; 2009.
- [2] D.a.godse A.p.godse. **Microprocessors and Interfacing**; first edition ; technical publications pune ; 2009 .
- [3] M.aumiaux . **Les systèmes à microprocesseurs** ; 2^{ème} edition ; masson ; 1982.
- [4] **Intel The 8086 Family**; Users Manual ; 1979.
- [5] Dr J.Y. Haggége. **MICROPROCESSEUR** ; (cours Institut Supérieur des études Technologiques de Radés) ; 2003
- [6] A. Oumnad. **MICROPROCESSEURS DE LA FAMILLE 8086**.
- [7] www.chipdocs.com
- [8] www.datasheetcatalog.com