

Chapitre 9

L'accès direct à la mémoire

Nous avons vu deux techniques pour contrôler le flot de données entre le microprocesseur et un périphérique : la technique de l'interrogation et le pilotage par interruption. Dans certains cas, la dernière technique n'est pas encore suffisamment rapide, par exemple pour le rafraîchissement de l'écran du moniteur. Dans ce cas on utilise une troisième technique, appelée **accès direct à la mémoire** ou **DMA** (pour l'anglais *Direct Memory Access*).

Le goulot d'étranglement dans le transfert des données est le microprocesseur lui-même. Lorsqu'un texte est sauvegardé sur un disque dur, on doit transférer les données depuis la mémoire sur le disque. Avec les techniques de l'interrogation et du pilotage par interruption, les données doivent transiter par le microprocesseur. Or passer par un registre du 8088 prend 39 tics d'horloge :

	Nombre de tics
BACK: MOV AL,[SI]	10
OUT PORT,AL	10
INC SI	2
LOOP BACK	17

ce qui ralentit le taux de transfert plus que nécessaire. L'idée de la DMA est de confier ce transfert à un circuit intégré spécialisé, de conception presque aussi délicate que le microprocesseur lui-même, mais adapté à ces transferts, qui ne durent alors que 4 tics d'horloge. Ce circuit intégré spécialisé est appelé **contrôleur DMA** (DMAC pour *DMA Controller*).

9.1 Partage des bus

Le DMA n'agit pas en parallèle avec le microprocesseur car il n'existe qu'un seul ensemble de bus (bus des données, bus des adresses et bus de contrôle). Les bus peuvent être utilisés soit par le microprocesseur, soit par le DMAC mais pas par les deux à la fois. Puisque le microprocesseur a le contrôle (primaire) sur les bus, il doit donner la permission au DMAC de les utiliser.

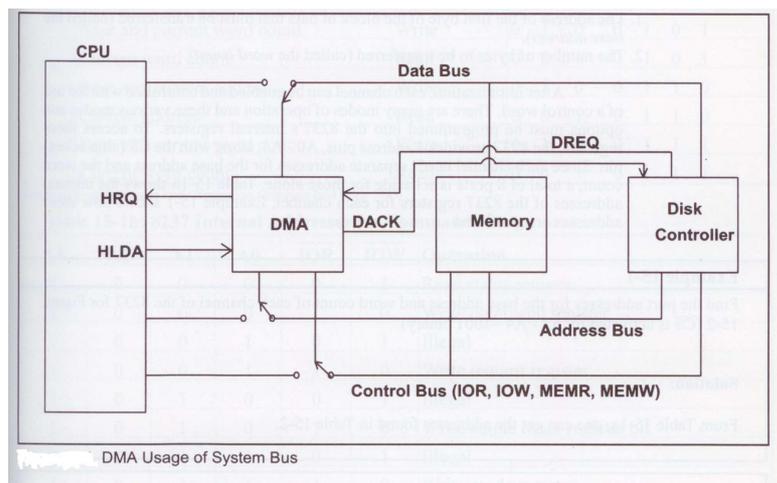


FIGURE 9.1 – DMA et bus

On utilise pour cela les deux broches HRQ et HLDA du microprocesseur (figure 9.1) :

- Le cycle commence par la requête d'un périphérique *via* l'entrée DREQ (pour *Dma REQuest*) du DMAC.
- Le DMAC positionne alors l'entrée HOLD du 8088 à niveau haut, demandant ainsi au microprocesseur d'entrer dans un état HOLD, laissant 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 (tristate). La broche HLDA (pour *HoLD Acknowledge*) est positionnée à niveau haut par le microprocesseur pour accuser réception de la requête.

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

- Lorsqu'il reçoit l'accusé de réception HDLA, le DMAC envoie le signal DACK (pour *Dma ACKnowledge*) au périphérique ayant demandé 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.

Le DMAC est programmé par le microprocesseur avant d'opérer pour un type donné de transfert. Il peut être programmé, par exemple, pour transférer 100 octets depuis l'emplacement mémoire commençant à l'adresse 1000:0000h vers le port d'entrées-sorties A3h.

9.2 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. C'est lui qui est utilisé sur l'IBM-PC d'origine.

9.2.1 Canaux

Le 8237 a quatre **canaux** de transfert de 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.

À 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.

9.2.2 Le brochage

Le 8237 est un circuit intégré à 40 broches se présentant de forme DIP. Les broches sont réparties de la façon suivante :

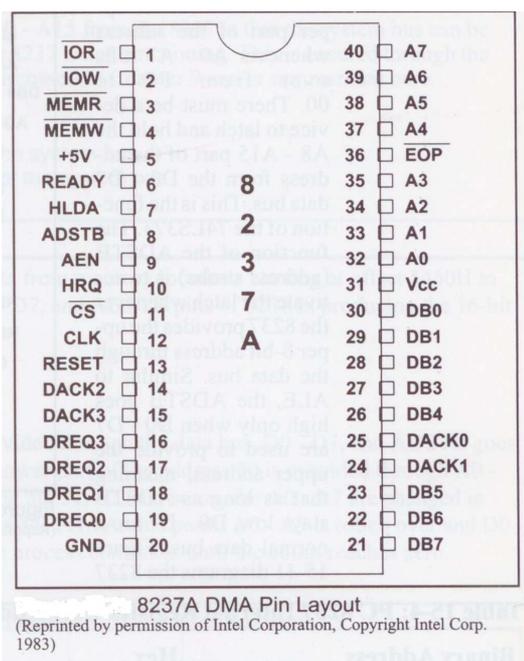


FIGURE 9.2 – Le brochage du DMAC 8237

— Bus des adresses

Il y a huit broches d'adresses, A0–A7, de numéros 32 à 35 et 37 à 40. Quatre de celles-ci, A0–A3, forment un bus d'adresses bidirectionnel, permettant d'envoyer des adresses au 8237 pour sélectionner un de ses 16 registres internes, la broche 11 (*chip select*) étant alors activée.

- **Bus des données**
Le bus des données est constitué par les broches DB0 à DB7, de numéros 30 à 26 et 23 à 21. Ce bus est bidirectionnel.
- **Bus de contrôle**
Le bus de contrôle est constitué par les broches 1 ($\overline{\text{IOR}}$), 2 ($\overline{\text{IOW}}$), 3 ($\overline{\text{MEMR}}$) et 4 ($\overline{\text{MEMW}}$), avec des significations parlantes.
- **AEN**
La broche 9 permet de savoir si le bus système est sous contrôle du microprocesseur (valeur 0) ou du DMAC (valeur 1).
- **RESET**
La broche 13 permet de remettre le DMAC à zéro.
L'entrée provient du 8284 dans le cas de l'IBM-PC.
- $\overline{\text{CS}}$ (*Chip Select*)
La broche 11 permet d'accéder aux registres internes du DMAC, comme nous l'avons déjà dit plus haut.
- **READY**
L'entrée de la broche 6 provient du signal RDYDMA du circuit d'engendrement de l'état d'attente. Ceci permet de prolonger le cycle de mémoire du DMA.
- **CLK** (*CLocK*)
La broche 12 est reliée à la broche CLK du 8284 dans le cas de l'IBM-PC, la fréquence étant la fréquence de travail (4,7 MHz dans le cas du PC/XT).
- **HRQ et HLDA**
Les broches 10 et 7 sont reliées aux broches HOLD et HLDA du microprocesseur.
- $\overline{\text{EOP}}$ (*End Of Process*)
Le signal de la broche 36 est activé lorsque le registre du compteur de l'un des quatre canaux atteint zéro. Ce signal peut être utilisé avec le signal DACK du canal concerné pour se prémunir de requêtes DMA multiples de ce canal au même moment ou peut être utilisé pour informer le périphérique ayant effectué la requête que le DMA a terminé son travail et qu'il devrait désactiver son signal DREQ.
- **DREQ0 et DACK0**
Les broches 19 et 25 véhiculent des signaux concernant le canal 0.
- **DREQ1–DREQ3 et DACK1–DACK3**
De même, les broches 18, 17, 16, 24, 14 et 15 véhiculent les signaux analogues concernant les canaux 1 à 3.
- **+5V, V_{CC} et GND**
Comme d'habitude, les broches 5, 31 et 20 permettent l'alimentation électrique du circuit.
- **ADSTB** (pour *ADdress STroBe*)
Nous avons vu que le 8237 a un bus d'adresses de 8 bits (broches A0 à A7) mais il est prévu pour adresser jusqu'à 64 KiO. On fait le pari que l'octet haut de l'adresse changera peu. Cette partie de l'adresse est donc d'abord envoyée via le bus des données D0 à D7. Il doit donc y avoir un élément de mémoire qui conserve la partie haute de l'adresse. C'est la fonction du 74LS373 qui doit être relié au 8237.
La fonction de la broche 8 est d'activer la bascule lorsque le 8237 fournit la partie haute d'une adresse.

9.2.3 Accès aux registres internes

Le 8237 possède seize registres internes. Les adresses des ports de ces registres internes sont données par le tableau suivant (en supposant que x soit égal à 0 pour l'adresse en hexadécimal) :

Adresse en binaire								Adresse en hexa	Fonction	Lecture/écriture
A7	A6	A5	A4	A3	A2	A1	A0			
0	0	0	x	0	0	0	0	00	CHAN0 memory address register	R/W
0	0	0	x	0	0	0	1	01	CHAN0 count register	R/W
0	0	0	x	0	0	1	0	02	CHAN1 memory address register	R/W
0	0	0	x	0	0	1	1	03	CHAN1 count register	R/W
0	0	0	x	0	1	0	0	04	CHAN2 memory address register	R/W
0	0	0	x	0	1	0	1	05	CHAN2 count register	R/W
0	0	0	x	0	1	1	0	06	CHAN3 memory address register	R/W
0	0	0	x	0	1	1	1	07	CHAN3 count register	R/W
0	0	0	x	1	0	0	0	08	Status/command register	R/W
0	0	0	x	1	0	0	1	09	Request register	W
0	0	0	x	1	0	1	0	0A	Single mask register bit	W
0	0	0	x	1	0	1	1	0B	Mode register	W
0	0	0	x	1	1	0	0	0C	Clear byte pointer	W
0	0	0	x	1	1	0	1	0D	Master clear/temporary register	R/W
0	0	0	x	1	1	1	0	0E	Clear mask register	W
0	0	0	x	1	1	1	1	0F	Mask register bits	W

Chaque canal du DMAC doit être initialisé séparément en ce qui concerne l'adresse du bloc de données à transférer ainsi que le nombre d'octets à transférer. Cette initialisation consiste à écrire pour chaque canal :

1. l'adresse du premier octet du bloc de données à transférer (appelée l'**adresse de base**) ;
2. le nombre d'octets à transférer (appelé le **compteur de mots**, en anglais *word count*).

Après initialisation, on doit activer chaque canal et le contrôler à l'aide d'un **mot de contrôle**. Il existe plusieurs modes opératoires ; le mode et les options doivent être programmés dans les registres internes du 8287. Pour accéder à ces registres, le DMAC dispose de quatre broches d'adresse, A0–A3, ainsi que de la broche CS. Puisque chaque canal a besoin d'une adresse pour l'adresse de base et d'une adresse pour le compteur de mots, on a besoin d'un total de 8 ports. La table suivante montre les adresses internes des registres de chaque canal :

Canal	Registre	R/W	CS	IOR	IOW	A3	A2	A1	A0
0	Adresse de base et en cours	W	0	1	0	0	0	0	0
	Adresse en cours	R	0	0	1	0	0	0	0
	Compteur de base et en cours	W	0	1	0	0	0	0	0
	Compteur en cours	R	0	0	0	1	0	0	1
1	Adresse de base et en cours	W	0	1	0	0	0	1	0
	Adresse en cours	R	0	0	1	0	0	1	0
	Compteur de base et en cours	W	0	1	0	0	0	1	0
	Compteur en cours	R	0	0	0	1	0	1	1
2	Adresse de base et en cours	W	0	1	0	0	1	0	0
	Adresse en cours	R	0	0	1	0	1	0	0
	Compteur de base et en cours	W	0	1	0	0	1	0	0
	Compteur en cours	R	0	0	0	1	1	0	1
3	Adresse de base et en cours	W	0	1	0	0	1	1	0
	Adresse en cours	R	0	0	1	0	1	1	0
	Compteur de base et en cours	W	0	1	0	0	1	1	0
	Compteur en cours	R	0	0	0	1	1	1	1

Exemple.- Trouver l'adresses du port du registre d'adresse de base et du registre de compteur pour chacun des canaux du 8237 interfacé de la façon ci-dessous :

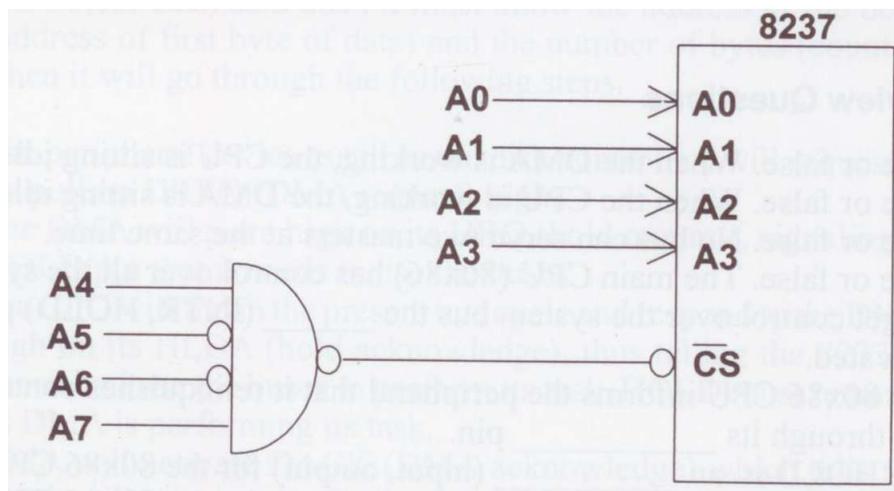


FIGURE 9.3 – Interfaçage du 8237

Le signal \overline{CS} est activé par $A_7A_6A_5A_4 = 1001$ donc, en tenant compte du tableau précédent pour $A_3A_2A_1A_0$, on obtient :

Adresse binaire								Hexa	Fonction
A7	A6	A5	A4	A3	A2	A1	A0		
1	0	0	1	0	0	0	0	90	CHAN0 memory address register
1	0	0	1	0	0	0	1	91	CHAN0 count register
1	0	0	1	0	0	1	0	92	CHAN1 memory address register
1	0	0	1	0	0	1	1	93	CHAN1 count register
1	0	0	1	0	1	0	0	94	CHAN2 memory address register
1	0	0	1	0	1	0	1	95	CHAN2 count register
1	0	0	1	0	1	1	0	96	CHAN3 memory address register
1	0	0	1	0	1	1	1	97	CHAN0 count register

9.2.4 Transfert

Principe.- Les informations nécessaires à la programmation du transfert de données d'un canal du DMAC sont (1) l'adresse du premier octet des données à transférer et (2) le nombre d'octets à transférer.

On écrit l'adresse dans le registre d'adresse mémoire. Puisque le registre d'adresse mémoire du 8237 est d'une taille de 16 octets et que le bus des données du 8237 est de 8 octets, on envoie deux octets, l'un après l'autre, à la même adresse de port : d'abord l'octet de poids faible puis celui de poids fort.

On écrit le nombre d'octets à transférer dans le registre de compteur du canal. On a une fois de plus le même problème, et la même solution, puisque le compteur peut atteindre FFFFh.

Les contenus des registres d'adresse et de compteur peuvent être lus de la même manière : d'abord l'octet inférieur, ensuite l'octet supérieur.

Exemple.- Supposons que le canal 2 du DMAC de l'exemple précédent doive transférer un bloc de 2 KiO (2 048 octets) depuis l'emplacement mémoire situé à l'adresse 53 400h. Écrire le programme nécessaire.

Les adresses des ports du canal 2 sont 94h pour l'adresse et 95h pour le compteur. L'initialisation sera donc :

```
MOV AX,3400h ; charge l'adresse de debut
OUT 94h,AL   ; envoie l'octet inferieur
MOV AL,AH
OUT 94h,AL   ; envoie l'octet superieur
MOV AX,2048h ; charge la taille du bloc
OUT 95h,AL   ; envoie l'octet inferieur
MOV AL,AH
OUT 95h,AL   ; envoie l'octet superieur
```

Problème de l'adresse.- On remarque un problème à propos de l'adresse : le bus des adresses système est de 20 bits alors que le bus des adresses du 8237 est de 16 bits. Dans notre cas, par exemple, seule la partie 3 400h de l'adresse 53 400h est prise en compte. N'oublions pas que le 8237 a été conçu pour les microprocesseurs 8080/8085 pour lesquels cela ne posait pas de problème.

Nous verrons comment le problème a été résolu dans le cas de l'IBM-PC, en utilisant un registre externe de 4 bits.

9.2.5 Description des registres internes

9.2.5.1 Registre de contrôle

Introduction.- Bien que le 8237 possède quatre canaux et que chacun de ces canaux puisse être programmé séparément en ce qui concerne l'adresse de base et le compteur, il n'existe qu'un seul ensemble de registres de commande et de contrôle, utilisé par tous les canaux.

Opérations sur les registres de contrôle.- Les opérations sur les registres de commande et de contrôle sont effectuées de la façon indiquée dans le tableau ci-dessous :

A3	A2	A1	A0	IOR	IOW	Opération
1	0	0	0	0	1	Lit le registre de statut
1	0	0	0	1	0	Écrit sur le registre de commande
1	0	0	1	0	1	Illégal
1	0	0	1	1	0	Écrit sur le registre de requête
1	0	1	0	0	1	Illégal
1	0	1	0	1	0	Écrit un bit du registre de masque
1	0	1	1	0	1	Illégal
1	0	1	1	1	0	Écrit sur le registre de mode
1	1	0	0	0	1	Illégal
1	1	0	0	1	0	Met à zéro le flip-flop pointeur d'octet
1	1	0	1	0	1	Lit le registre temporaire
1	1	0	1	1	0	Met à zéro le maître
1	1	1	0	0	1	Illégal
1	1	1	0	1	0	Met à zéro le registre de masque
1	1	1	1	0	1	Illégal
1	1	1	1	1	0	Écrit tous les bits du registre de masque

Exemple.- Utiliser le circuit de l'exemple ci-dessus pour trouver l'adresse des registres de contrôle du DMAC.

Nous obtenons les adresses suivantes :

Adresse binaire								Hexa	Nom du registre	R/W
A7	A6	A5	A4	A3	A2	A1	A0			
1	0	0	1	1	0	0	0	98	Registre de statut/commande	R/W
1	0	0	1	1	0	0	1	99	Registre de requête	W
1	0	0	1	1	0	1	0	9A	Simple bit du registre de masque	W
1	0	0	1	1	0	1	1	9B	Registre de mode	W
1	0	0	1	1	1	0	0	9C	Mise à zéro du pointeur d'octet	W
1	0	0	1	1	1	0	1	9D	Master clear/temporary register	R/W
1	0	0	1	1	1	1	0	9E	Registre de mise à zéro du masque	W
1	0	0	1	1	1	1	1	97	Bits du registre de masque	W

De ces huit registres, seuls les plus essentiels d'entre eux seront expliqués en détail ci-dessous.

9.2.5.2 Registre de commande

Rôle.- Il s'agit d'un registre de 8 bits permettant de contrôler les opérations du 8237. Ce registre doit être programmé (c'est-à-dire écrit) *via* le microprocesseur.

Format.- Son format est le suivant :

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

dont la description rapide est la suivante :

- D0 = 1 pour activer le transfert de mémoire à mémoire ;
- D1 = 1 pour activer le maintien de l'adresse du canal 0 (n'a de sens que si D0 = 1) ;
- D2 = 1 pour activer le contrôleur ;
- D3 = 1 pour une temporisation compressée (n'a de sens que si D0 = 0) ;
- D4 = 1 pour une priorité en rotation, D4 = 0 pour une priorité fixée ;
- D5 = 1 pour une sélection de l'écriture étendue (n'a de sens que si D3 = 0) ;
- D6 = 1 pour que DREQ réponde à un niveau bas ;
- D7 = 1 pour que DACK réponde à un niveau haut.

Fonction détaillée de chaque bits.- Décrivons en détail la signification de chacun des bits du registre de commande.

- **D0** précise qu'on doit utiliser les canaux 0 et 1 pour transférer un bloc de données de mémoire à mémoire. En effet, le 8237 est capable de transférer des données (1) d'un périphérique à la mémoire, (2) de la mémoire à un périphérique ou (3) de la mémoire à la mémoire.
 Pourquoi a-t-on besoin de deux canaux dans le cas d'un transfert de mémoire à mémoire ? Le canal 0 est utilisé comme source et le canal 1 comme destination. Le canal 0 lit un octet et le place dans un registre temporaire du 8237, le canal 1 l'envoie ensuite à destination. C'est différent dans le cas d'un transfert de mémoire à périphérique, ou *vice-versa*, car alors les données sont lues sur le bus des données et transférées à destination sans sauvegarde temporaire.
- **D1** est seulement utilisé dans le cas d'un transfert de mémoire à mémoire. Il est utilisé pour activer l'incrémement/décrémement du canal 0 afin d'écrire une valeur fixe dans tout un bloc de mémoire.
- **D2** est utilisé pour activer la DMA.
- **D3** permet de choisir entre le cycle normal de mémoire à quatre impulsions d'horloge et le cycle *compressé* à deux impulsions d'horloge.
- **D4** permet d'utiliser les quatre canaux suivant un ordre de priorité fixe ou par rotation. Si l'ordre est fixe, DREQ0 possède la priorité la plus élevée et DREQ3 la priorité la moins élevée. Dans le cas d'une priorité par rotation chaque requête est servie à tour de rôle en bouclant dans l'ordre DRQ0, DRQ1, DRQ2 et DRQ3 ; ceci permet au DMA de ne pas être monopolisée par un seul canal.
- **D5** permet d'augmenter la durée du signal d'écriture pour accéder aux périphériques lents.
- **D6** permet de programmer le niveau d'activation de DREQ : niveau haut ou niveau bas.
- De même on peut programmer, grâce à **D7**, le niveau d'activation de DACK.

Exemple 1.- Programmons le registre de commande du 8237 de notre exemple pour les options suivantes : transfert non de mémoire à mémoire, temporisation normale, ordre de priorité fixe, dernière écriture, DREQ et DACK tous les deux actifs à niveau haut.

Nous avons déjà vu que l'adresse du registre de commande est 98h. D'après ce que nous venons de dire, l'octet de commande doit être 1000 0000b = 80h. Le programme est donc :

```
MOV AL,80h ; charge l'octet de commande dans AL
OUT 98h,AL ; le place dans le registre de commande
```

Exemple 2.- Imaginons que le microprocesseur doit effectuer une opération critique et donc que le DMA doit être désactivé. Écrire le programme correspondant.

Pour désactiver le DMA, il suffit d'envoyer l'octet 0000 0100b = 04h au registre de commande :

```
MOV AL,04h
OUT 98h,AL
```

9.2.5.3 Registre de statut

Rôle.- Le registre de statut est un registre de 8 bits pouvant seulement être lu par le microprocesseur à travers le port de même adresse que le registre de commande. Il contient des informations sur l'état des quatre canaux.

Format.- Son format est le suivant :

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

avec :

- D0 = 1 lorsque le canal 0 a atteint son TC (*Terminal Count*);
- D1 = 1 lorsque le canal 1 a atteint son TC;
- D2 = 1 lorsque le canal 2 a atteint son TC;
- D3 = 1 lorsque le canal 3 a atteint son TC;
- D4 = 1 lorsqu'il y a requête du canal 0;
- D5 = 1 lorsqu'il y a requête du canal 1;
- D6 = 1 lorsqu'il y a requête du canal 2;
- D7 = 1 lorsqu'il y a requête du canal 3.

9.2.5.4 Registre de mode

Rôle.- Le registre de mode est un registre de 8 bits qui peut seulement être écrit par le microprocesseur.

Format.- Son format est le suivant :

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

où :

- **D1D0** permettent de choisir le canal conformément au tableau suivant :

00 = sélection du canal 0 ;
 01 = sélection du canal 1 ;
 10 = sélection du canal 2 ;
 11 = sélection du canal 3 ;

- **D3D2** spécifient le mode de transfert conformément au tableau suivant :

00 = transfert de vérification ;
 01 = transfert en écriture ;
 10 = transfert en lecture ;
 11 = illégal ;

Dans le cas d'un **transfert en écriture**, le DMAC transfère des données d'un périphérique à la mémoire en activant IOR et MEMW. Dans le cas d'un **transfert en lecture**, le DMAC transfère des données de la mémoire à un périphérique en activant MEMR et IOW. Le **transfert de vérification** se fait comme dans le cas des deux premiers transferts sauf qu'il n'engendre pas de signaux de contrôle IOR, MEMW, MEMR et IOW.

- **D4** vaut 1 pour activer l'**auto-initialisation**, c'est-à-dire que le registre d'adresse de mémoire et le registre de compteur reprennent leurs valeurs initiales à la fin du transfert des données (lorsque le compteur prend la valeur zéro). De cette façon, ces registres sont seulement programmés une fois.
- **D5** permet soit d'incrémenter l'adresse ($D5 = 0$), soit de décrémenter l'adresse.
- **D7D6** permet de déterminer la façon dont le 8237 est utilisé, conformément au tableau suivant :

00 = sélection du mode à la demande,
 01 = sélection du mode simple,
 10 = sélection du mode bloc,
 11 = sélection du mode en cascade.

Pour un type DMA donné, il y a trois façons de mélanger les cycles DMA aux cycles de bus normaux du microprocesseur :

- Le **mode à la demande**, ou **mode éclaté** (en anglais *burst mode*), est celui dans lequel le transfert des données continue jusqu'à ce que DREQ soit désactivé ou que le compteur terminal ait été atteint. Ceci nous assure que la DMA puisse terminer le travail sans interruption, mais qui a évidemment pour conséquence la monopolisation des bus système par le DMA pendant toute la durée du transfert du bloc entier de données.

Ce mode est réservé aux périphériques ayant des tampons de données accessibles à haut débit. L'avantage de cette technique est que le tampon du périphérique est rempli très rapidement par le DMAC et vidé à loisir par le périphérique.

- Le **mode bloc**, ou **mode continu**, s'effectue de la même façon que le mode à la demande tout en permettant à DREQ d'être désactivé après le début du cycle de DMA. Le processus de transfert de données se termine alors lorsque le compteur terminal est atteint. En d'autres mots, il n'y a pas nécessité de garder DREQ à niveau haut pendant toute la durée du transfert.

Cette technique est intéressante pour les périphériques à très haut débit.

- Le **mode simple**, ou **mode octet**, est celui où, lorsque DREQ est rendu actif, il y a transfert d'un octet de données, le microprocesseur pouvant alors reprendre éventuellement le contrôle du bus système en désactivant HRQ pour un cycle de bus. Ce processus permet un accès alterné au bus système entre le microprocesseur et le DMAC jusqu'à ce que le compteur terminal soit été atteint.

C'est l'option choisie par l'IBM-PC, ce qui permet au microprocesseur et au DMAC d'effectuer leur travail sans monopolisation des bus par l'un ou l'autre.

- Le **mode cascade** est celui dans lequel plusieurs DMAC peuvent être montés en cascade pour étendre le nombre de DREQ.

Cette option est utilisée pour l'IBM-PC depuis l'AT.

Exemple.- Programmons le registre de mode du 8237 de notre exemple de façon à sélectionner le canal 2 pour un transfert allant de la mémoire vers un périphérique en utilisant l'auto-initialisation, l'incrémentation de l'adresse et le transfert simple.

Le registre de mode doit être 0101 1010b = 5Ah. L'adresse du port du registre de commande est 9Bh, ce qui donne :

```
MOV AL,5Ah
OUT 9Bh,AL
```

9.2.5.5 Registre de masquage simple

Rôle.- Ce registre peut seulement être écrit par le microprocesseur. C'est grâce à ce registre que l'entrée d'un DREQ d'un canal donné peut être masquée (désactivée) ou non masquée (activée). Un seul canal peut être masqué ou non masqué à la fois à travers ce registre. Pour masquer ou ne pas masquer plusieurs canaux en même temps, on utilise le registre suivant.

Format.- Son format est le suivant :

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

où seuls D2, D1 et D0 sont utilisés :

- **D1D0** permettent de choisir le canal conformément au tableau suivant :
 - 00 = sélection du bit de masquage du canal 0 ;
 - 01 = sélection du bit de masquage du canal 1 ;
 - 10 = sélection du bit de masquage du canal 2 ;
 - 11 = sélection du bit de masquage du canal 3.
- **D2** spécifie si on active le bit de masquage (D2 = 1) ou si on le désactive.

Exemple.- Si la valeur 00000101b est écrite dans ce registre, DREQ1 sera masqué et le DMAC ne répondra pas aux DREQ du canal 1.

9.2.5.6 Registre de masquage complet

Rôle.- Ce registre peut seulement être écrit par le microprocesseur.

Format.- Son format est le suivant :

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

où seuls D3, D2, D1 et D0 sont utilisés. Si D3 = 1, par exemple, alors le canal 3 sera masqué. De même D2, D1 et D0 correspondent aux canaux 2, 1 et 0.

Exemple.- Si on écrit 0000010b sur ce registre, l'entrée du canal 1 est masquée mais les entrées des trois autres canaux ne sont pas masquées.

Puisque 0000 0010b = 02h et que l'adresse du port de ce registre est 9Ah, on utilise le programme suivant :

```
MOV AL,02  
OUT 9Ah,AL
```

9.2.5.7 Registre Master clear/temporary

Ce registre peut seulement être écrit par le microprocesseur. L'octet envoyé à ce registre n'a pas d'importance puisqu'il permet seulement de remettre à zéro les registres de statut, de commande et de masquage et qu'il force le DMAC à entrer dans un cycle inactif (*idle*). Ceci revient à activer le RESET matériel du 8237.

Si on essaie de lire ce registre, le DMAC fournit le dernier octet de données transféré durant un transfert de mémoire à mémoire.

9.2.5.8 Registre de remise à zéro des masques

Ce registre peut seulement être écrit par le microprocesseur. L'octet envoyé à ce registre n'a pas d'importance puisqu'il permet seulement de mettre à zéro les bits de masquage de chacun des quatre canaux, ce qui permet au DMAC d'accepter les requêtes de tous les canaux.

9.3 Cas de l'IBM PC/XT

Dans le cas de l'IBM-PC d'origine, un seul 8237 est utilisé.

9.3.1 Interfaçage

Description.- La figure 9.4 montre l'interfaçage du 8237 dans le cas de l'IBM-PC/XT. On remarquera la présence des trois circuits intégrés 74LS244, 74LS373 et 74LS670.

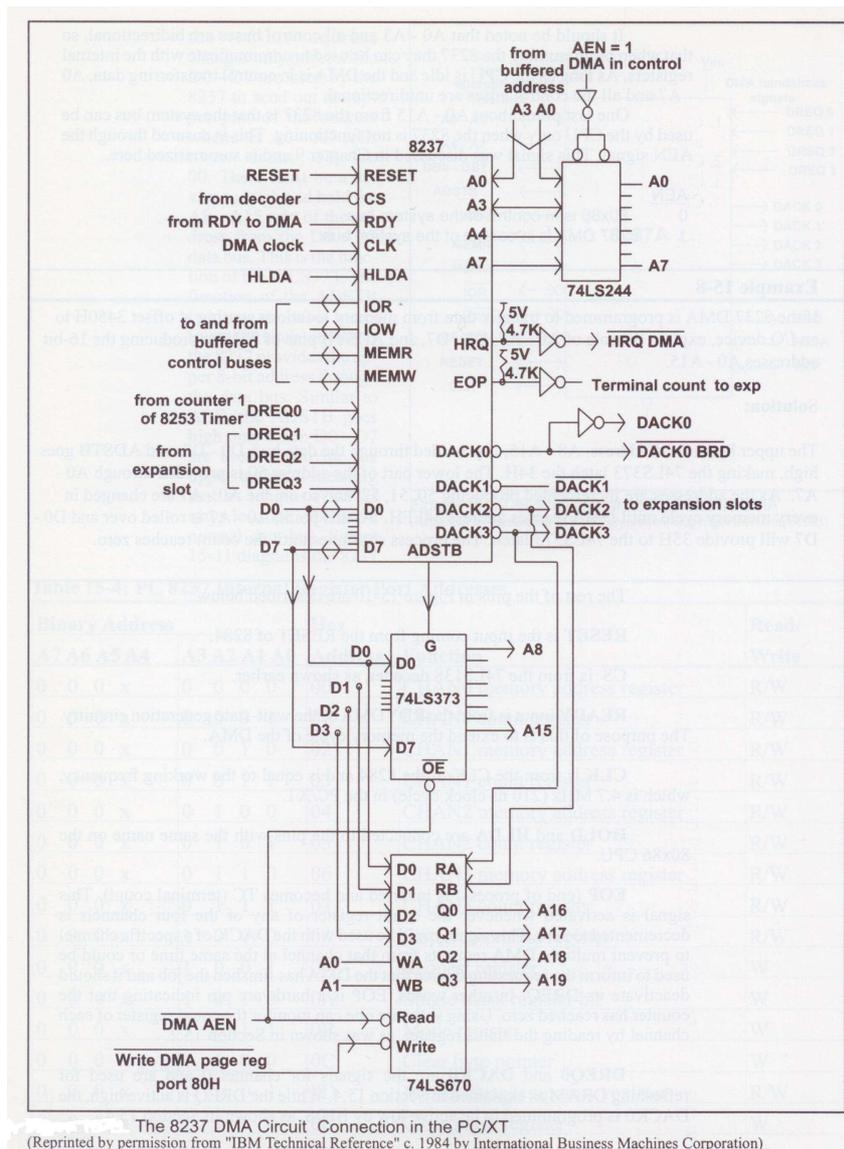


FIGURE 9.4 – Interfaçage du DMAC 8237 dans le cas de l'IBM-PC

Adresse des ports.- La broche \overline{CS} (*Chip Select*) du DMAC est reliée à la broche Y0 du décodeur 74LS138, comme le montre la figure 9.5, ce qui donne les adresses 00h à 0Fh pour les seize ports.

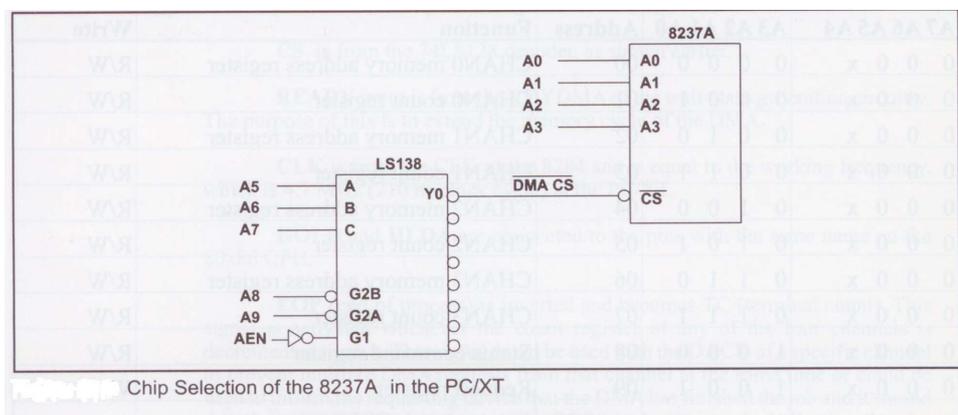


FIGURE 9.5 – Interfaçage du DMAC 8237 dans le cas de l'IBM-PC (2)

Utilisation des canaux.- Les quatre canaux sont utilisés de la façon suivante :

- le canal 0 sert à rafraîchir la mémoire DRAM (cette utilisation est abandonnée depuis le PC/AT) ;
- le canal 1 n'est pas réservé, mais il est souvent utilisé pour le réseau ;
- le canal 2 est utilisé pour le contrôleur de disquettes ;
- le canal 3 est utilisé pour le contrôleur de disque dur.

Le DMA était donc utilisé pour accélérer les opérations d'écriture et de lecture du lecteur de disquette, le microprocesseur étant relativement lent. Ceci n'est plus nécessaire de nos jours.

9.3.2 Registres de page

Problème du bus des adresses.- Comme nous l'avons vu, il y a un problème à propos du bus des adresses puisque le 8237 a été conçu pour le microprocesseur 8080 ayant un bus des adresses de 16 bits et non de 20 bits.

Nous avons déjà expliqué comment on transmet les deux octets inférieurs (A0-A15) de l'adresse en utilisant le 74LS373.

On utilise le circuit intégré 74LS670 pour fournir les bits A16-A19 de l'adresse physique de chacun des quatre canaux :

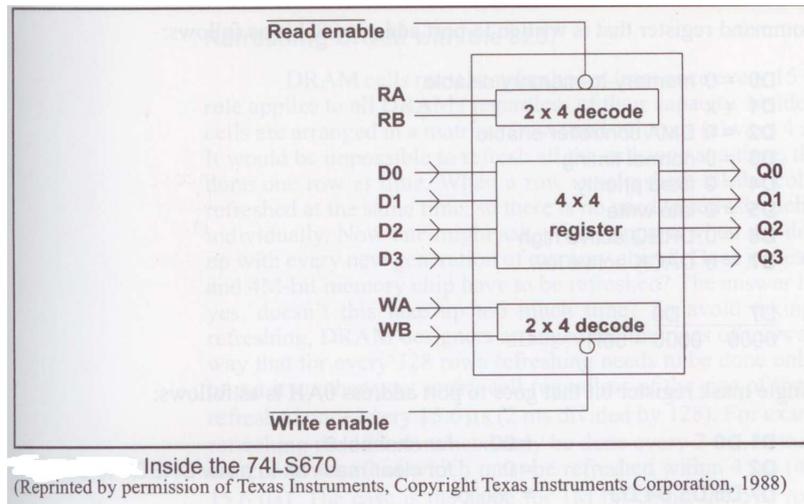


FIGURE 9.6 – Le 74LS670

On parle de **registres de page** du DMAC. Les quatre registres du 74LS670 peuvent seulement être écrits par le microprocesseur en utilisant D0-D3 du bus des données grâce à une instruction OUT. Puisque le 74LS670 est un registre 4x4, il y a quatre emplacements de port, un pour chacun des canaux. Le circuit intégré 74LS138 :

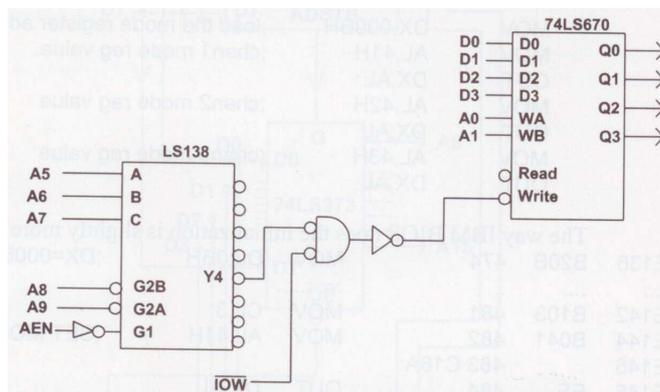


FIGURE 9.7 – Décodage des adresses pour le DMA

est utilisé pour décoder les adresses.

Le calcul de l'adresse de ces ports est (en supposant que tous les x sont nuls) :

AEN	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Hexa	Registre 4 bits
0	0	0	1	0	0	x	x	x	0	0	80	non utilisé pour le PC
0	0	0	1	0	0	x	x	x	0	1	81	pour le canal 2
0	0	0	1	0	0	x	x	x	1	0	82	pour le canal 3
0	0	0	1	0	0	x	x	x	1	1	83	pour le canal 1

Comme on le voit, seuls trois des registres de page sont utilisés puisque le canal 0 est utilisé pour rafraîchir la mémoire DRAM et qu'il n'a donc pas besoin des 4 bits supérieurs d'adresse (comme nous le verrons dans la section suivante).

Pour lire ces registres, on doit programmer le signal DACK du 8237 pour qu'il soit actif à niveau bas, en utilisant le registre de mode. Selon le DREQ actif, le 8237 peut lire le contenu du registre désigné de la façon suivante :

DACK2 RB	DACK3 RA	
1	1	canal 1
1	0	canal 3
0	1	canal 2
1	1	impossible

Les conditions de la dernière ligne ne peuvent jamais être réalisées puisque $\overline{\text{DACK2}}$ et $\overline{\text{DACK3}}$ ne peuvent pas être actifs en même temps.

Seul le microprocesseur peut écrire dans les registres de page puisque les broches d'adresse A0 et A1 du microprocesseur sont reliées aux broches WA et WB, respectivement, du 74LS670. La lecture des registres de page peut seulement intervenir lorsque le DMAC a le contrôle, puisque les broches RA et RB sont reliées aux broches DACK3 et DACK2 du DMAC et que les signaux correspondants ne sont pas activés tant que le DRQ correspondant n'est pas activé.

9.3.3 Taux de transfert

Cela nécessite beaucoup plus de tics d'horloge pour transférer un octet de données entre un périphérique et la mémoire en utilisant le microprocesseur 8088 qu'en utilisant le DMAC 8237. Quel est le taux de transfert des canaux 1, 2 et 3? Le circuit à état d'attente insère un tic d'horloge dans le cycle mémoire des canaux 1 à 3 en utilisant le signal $\overline{\text{RDYDMA}}$. Puisque le DMAC nécessite lui-même 5 impulsions d'horloge, ceci fait un total de 6 impulsions d'horloge pour le DMAC proprement dit. Mais entre chaque cycle de DMA il y a un cycle mémoire du microprocesseur, qui prend quatre tics d'horloge. Ceci donne donc un total de 10 impulsions d'horloge pour le transfert de chaque octet de données pour les canaux 1 à 3.

Puisque chaque impulsion d'horloge est de 210 ns (1/4,7 MHz) dans le cas de l'IBM PC/XT, cela fait $10 \times 210 \text{ ns} = 2100 \text{ ns}$ pour le transfert d'un octet de données par le DMAC. Ainsi le taux de transfert du 8237 dans le cas du PC/XT est-il de 476 190 octets par seconde ($1/2100 \text{ ns} = 476 190$). Ceci suppose qu'il n'y ait pas d'autre périphérique insérant un état d'attente dans le cycle de mémoire.


```

475
E135 BED8      476      MOV     DS,BX          ; SET UP ABSO INTO DS AND ES
E137 BEC3      477      MOV     ES,BX
478      ASSUME DS:ABSO,ES:ABSO
E139 B0FF      479      MOV     AL,OFFH        ; SET CNT OF 64K FOR REFRESH
E13B E601      480      OUT     DMA+1,AL
E13D 50        481      PUSH    AX
E13E E601      482      OUT     DMA+1,AL
E140 B058      483      MOV     AL,058H        ; SET DMA MODE, CH 0,RD.,AUOTIN
E142 E60B      484      OUT     DMA+0BH,AL    ; WRITE DMA MODE REG
E144 B0D0      485      MOV     AL,0           ; ENABLE DMA MODE REG
E146 8AE8      486      MOV     CH,AL          ; SET COUNT HIGH=00
E148 E608      487      OUT     DMA+8,AL      ; SETUP DMA COMMAND REG
E14A 50        488      PUSH    AX
E14B E60A      489      OUT     DMA+10,AL     ; ENABLE DMA CH 0
E14D B012      490      MOV     AL,18          ; START TIMER 1
E14F E641      491      OUT     TIMER+1,AL
E151 B041      492      MOV     AL,41H         ; SET MODE FOR CHANNEL 1
E153 E60B      493      OUT     DMA+0BH,AL
E155 50        494      PUSH    AX
E156 E4DB      495      IN      AL,DMA+08     ; GET DMA STATUS
E158 2410      496      AND     AL,00010000B  ; IS TIMER REQUEST THERE?
E15A 7401      497      JZ      C18C           ; (IT SHOULD'T BE)
E15C F4        498      HLT
E15D B042      499      C18C: MOV     AL,42H    ; SET MODE FOR CHANNEL 2
E15F E60B      500      OUT     DMA+0BH,AL
E161 8043      501      MOV     AL,43H         ; SET MODE FOR CHANNEL 3
E163 E60B      502      OUT     DMA+0BH,AL

```

- 3°) On initialise le temporisateur 1 pour rafraîchir la mémoire : on envoie 3h au port A du 8255 (lignes 446 et 447) pour indiquer que le coprocesseur arithmétique (en fait le DMA) est installé et on envoie la même valeur (celle-ci n'ayant pas d'importance) au registre *Master clear/temporary register* du 8257 (ligne 449) pour remettre à zéro ses registres de statut, de commande et de masquage pour qu'il entre dans un état inactif.

- 4°) On envoie le motif FFFFh aux huit registres de canaux 00h à 007h du 8257 (lignes 453 à 460), on change la valeur de AL (ligne 461) avant de lire les valeurs des ces registres (lignes 462 à 464) et on les compare aux valeurs envoyées (ligne 465) pour voir s'il s'agit bien des mêmes. On arrête le système (ligne 467) si ce n'est pas le cas pour au moins l'un des registres.

On recommence avec le motif 0000h (lignes 471 et 472).

- 5°) On initialise et on démarre le DMA pour rafraîchir la mémoire à partir du canal 0, celui de plus grande priorité.

Si tout s'est bien passé, le registre BX contient la valeur 0. On place cette valeur dans les registres DS et ES (lignes 476 et 477) pour parler d'adresses absolues.

On place le compteur FFFFh dans le registre de compteur du canal 0 du 8257 (lignes 479 à 482) pour rafraîchir un bloc de mémoire (c'est-à-dire 64 KiO), en attendant un petit peu entre l'envoi de l'octet de poids faible et celui de poids fort (ligne 481) à cause de problèmes de synchronisation avec le 8257.

L'octet du registre de mode envoyé à l'adresse 0Bh (lignes 483 et 484) est le suivant :

D1D0 = 00 pour le canal 0,
D3D2 = 10 pour le transfert en lecture,
D4 = 1 pour l'auto-initialisation,
D5 = 0 pour incrémenter l'adresse,
D7D6 = 01 pour le mode simple,

ce qui donne 0101 1000b = 58h (remarquer que la faute 'AUOTIN' au lieu de 'AUTO IN' se trouve bien dans le listing du BIOS).

L'octet du registre de commande, écrit au port d'adresse 08h (lignes 485 et 487), est le suivant :

D0 = 0 pour désactiver le transfert de mémoire à mémoire,

D1 = x,

D2 = 0 pour activer le contrôleur de DMA,

D3 = 0 pour une temporisation normale,

D4 = 0 pour une priorité fixe,

D5 = 0 pour une écriture tardive,

D6 = 0 pour un DREQ actif à niveau haut,

D7 = 0 pour un DACK actif à niveau bas,

ce qui donne 0000 0000b = 00h.

L'octet du registre de masquage simple, envoyé au port d'adresse 0Ah (lignes 488 et 489), est le suivant :

D1D0 = 00 pour le canal 0,

D2 = 0 pour mettre à zéro le bit, ce qui permet l'activation de DREQ0,

D7D6D5D4 = xxxx,

ce qui donne 0000 0000b = 00h.

On fait démarrer le temporisateur 1 (lignes 490 et 491).

- 6°) On initialise ensuite les canaux 1 à 3 du 8257.

L'octet du registre de mode, envoyé au port 0Bh, pour le canal 1 (lignes 492 et 493) est le suivant :

D1D0 = 01 pour canal 1

D3D2 = 00 pour transfert de vérification

D4 = 0 pour désactivation de l'auto-initialisation

D5 = 0 pour incrémentation de l'adresse

D7D6 = 01 pour mode simple

soit 0100 0001b = 41h.

Pour les canaux 2 et 3 (lignes 499 à 502), la valeur de l'octet du registre de mode est le même à part que D0D1 sont changés en 10 et 11 respectivement. Ainsi les canaux 2 et 3 ont-ils des valeurs du registre de mode égales à 42h et 43h.

- 7°) On vérifie enfin (en fait entre l'initialisation des canaux 1 et 3) que l'octet de statut du DMAC spécifie bien que le temporisateur 1 fonctionne, sinon on arrête le système (lignes 494 à 498).