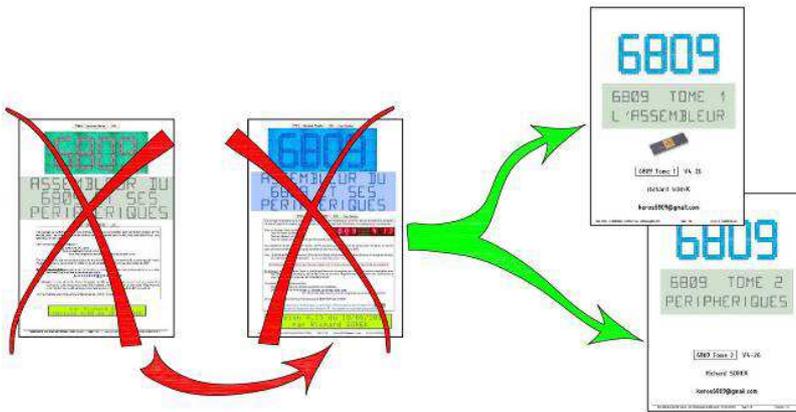


6809 TOME 2  
 PERIPHERIQUES



6809 Tome 2

T2.v4-26

Richard SOREK

keros6809@gmail.com

## PRÉFACE

Ces ouvrages de synthèse sur le microprocesseur 6809 (tome 1) et de ses périphériques (tome 2), sont le fruit de nombreuses années de travail, de modifications, de mise en page et de création durant mes soirées, mes nuits d'insomnies, mes week-ends et mes vacances.

Dans cet ouvrage (sans exception) : Tous **les textes** ont été saisis  
Tous **les tableaux** ont été créés  
Tous **les croquis** ont été dessinés par mes propres soins.

Après avoir recherché de la documentation sur le 6809 sur Internet, je me suis vite rendu compte que les documents que j'ai eu l'occasion de voir étaient trop succincts, incomplets et souvent avec quelques petites erreurs.

C'est pour cela que j'ai créé ces documents, ils sont destinés à la compréhension pragmatique et didactique de l'assembleur du 6809 (tome 1) et de ses périphériques (tome 2). Ce travail fût guidé par l'idée d'avoir une documentation précise, pragmatique, détaillée et vivante et surtout de faire partager ces informations gratuitement pour un usage uniquement personnel.

Dans la version précédente de cet ouvrage, j'avais mis une multitude de liens hypertextes. Suite à un bug de Word, j'ai eu énormément de difficulté à récupérer le texte. J'ai donc été contraint de revoir complètement la mise en page et de supprimer à mon grand regret les liens hypertextes. J'ai également scinder l'ancien ouvrage en deux tomes (**Tome 1** l'assembleur, **Tome 2** les périphériques).

Mes propres cours datant de mon passage à l'Ecoles des Mines de DOUAI 59500 ont été complétés et comparés aux informations trouvées dans les livres suivants :

- Ouvrage 01 : L'ASSEMBLEUR FACILE DU 6809 de François BERNARD de 1984
- Ouvrage 02 : LE MICROPROCESSEUR 6809 – SES PERIPHERIQUES ET LE PROCESSEUR GRAPHIQUE 9355-66 de Claude DARDANNE de 1991 neuvième édition
- Ouvrage 03 : PROGRAMMATION DU 6809 de Rodnay ZAKS de 1983 (édition SYBEX)
- Ouvrage 04 : MICROPROCESSEURS : DU 6800 AU 6809 MODES D'INTERFACE de Gérard REVILLIN
- Ouvrage 05 : ETUDES AUTOUR DU 6809 (CONSTRUCTIONS ET LOGICIELS) de Claude VICIDOMINI (2<sup>ème</sup> édition du hors série de la revue LED)
- Ouvrage 06 : PROGRAMMATION EN ASSEMBLEUR 6809 de BUI MINH DUC édition EYROLLES
- Ouvrage 07 : LE MICROPROCESSEUR 6809 DE MOTOROLA, partie 3 chapitre 4.2.1
- Ouvrage 08 : ASSEMBLEUR ET PÉRIPHÉRIQUE DES MO5 ET TO7/70 de Frédéric Blanc et de François Normand
- Ouvrage 09 : MANUEL DE L'ASSEMBLEUR 6809 DU TO7 de Michel Weissgerber
- Ouvrage 10 : MANUEL DE L'ASSEMBLEUR 6809 DU MO5 de Michel Weissgerber

**Je précise que ce document n'a aucun et n'aura jamais de caractère commercial.  
Ce document est à l'attention de quelques de personnes et uniquement pour un usage personnel.**

Si vous découvrez des erreurs, des fautes d'orthographe ou encore mieux si vous souhaitez m'apporter vos améliorations, merci de me les envoyer par mail. Je me ferai un devoir et un plaisir d'en apporter la correction ou d'en faire les ajouts.

Je vous demande d'avoir la grande gentillesse de me rapporter par mail vos critiques, vos appréciations, vos corrections éventuelles et/ou vos compléments d'informations, afin de faire vivre ce document.

En fonction des modifications, je vous transmettrai en retour de mail, mes dernières versions.

**Un petit message me ferai également un grand plaisir.**

Je souhaite remercier chaleureusement :

- **Jacques BRIGAUD** pour ces corrections sur l'assembleur.
- Les modérateurs du site <http://forum.system-cfg.com>

Un grand merci également aux sites suivants pour avoir mis mes documents en ligne.

[colorcomputerrchive.com](http://colorcomputerrchive.com)  
[studocu.com](http://studocu.com)  
[pdfcoffee.com](http://pdfcoffee.com)

[asm.developpez.com](http://asm.developpez.com)  
[kswichit.net](http://kswichit.net)

N'hésitez pas à me contacter par mail pour savoir si il n'y a pas une version plus récente de mes documents.  
Je souhaite une bonne lecture aux amoureux du 6809 et de ses périphériques. **Richard SOREK**

Pour votre information, j'ai développé un Assembleur-Désassembleur pour le 6809 c'est le **P30RS09**  
Il fonctionne pour **l'instant** uniquement sous Windows XP. Il est également disponible gratuitement.

# SOMMAIRE

<b>LES : LES ENTREES / SORTIES - GENERALITES</b>	006
--	-----

## 6821

<b>6821 : LES ENTREES / SORTIES- LE 6821 PIA</b>	007
6821 : Port A	007
6821 : Port B	007
6821 : Organisation Interne	007
6821 : Registres CRA et CRB (Control Register A et B)	008
6821 : Vue complète du registre CRA ou CRB	008
6821 : Détail du registre CRA ou CRB	009
6821 : bit <b>CRx0</b>	009
6821 : bit <b>CRx1</b>	009
6821 : bit <b>CRx2</b>	009
6821 : bits <b>CRx5 CRx4 CRx3</b>	009
6821 : <b>CRx5 = 0</b> la broche Cx2 est en ENTREE d'interruption	010
6821 : <b>CRx5 = 1</b> la broche Cx2 est en SORTIE de commande	010
6821 : 1er Mode HANDSHAKE ou mode Dialogue	
<b>CRx5 CRx4 CRx3 = %100</b>	011
Pour le port A	011
Pour le port B	011
6821 : 2ième Mode SET-RESET mode Programmé	
<b>CRx5 CRx4 CRx3 = %110 ou %111</b>	011
6821 : 3ième Mode PULSE-STROBE mode Impulsion	
<b>CRx5 CRx4 CRx3 = %101</b>	012
Pour le port A	012
Pour le port B	012
6821 : Les bits <b>CRx7 et CRx6</b>	013
6821 : bit <b>CRx6</b>	013
6821 : bit <b>CRx7</b>	013
6821 : Programmation des broches CA1, CA2, CB1 et CB2 en ENTREE	014
6821 : Broches CAx CBx : Modes Automatiques	014
6821 : Broches CAx CBx : Modes Manuels	014
6821 : CA1 et CB1	014
6821 : CA2 et CB2	015
6821 : Broche CA2	015
6821 : Broche CB2	015
6821 : Programmation des broches CA2 et CB2 en SORTIE	015
6821 : CRA4 ou CRB4 = 0 Mode Dialogue	015
6821 : CRA4 ou CRB4 = 1 Mode Programmé	015
6821 : Registres DDRA et DDRB (Data Direction Register A et B)	016
6821 : Registres ORA et ORB (Output Register A et B)	016
6821 : Organisation Externe	017
6821 : Brochage	017
6821 : Liaison avec le bus de données du 6809	017
6821 : Liaison avec le bus d'adresses du 6809	017
6821 : Broches CS0, CS1 et CS2  (Chip Select Line) Sélection de boîtier	017
6821 : Broches RS0 et RS1 (Register Select Line)	018
6821 : Liaison avec le bus de contrôle du 6809	018
6821 : Broche E : (Enable)	018
6821 : Broche RESET  broche en entrée :	018
6821 : Broche R/W  : (Read / Write )	018
6821 : Broches IRQA   et IRQB  (IR... = Interrupt Request) broches en sortie	019

6821 : Liaison avec la périphérie : lignes de transfert	019
6821 : Broches PA0 à PA7	019
6821 : Broches PB0 à PB7	019
6821 : Fonctionnement	020
6821 : Transfert d'une donnée Périphérie → 6809	020
6821 : Transfert d'une donnée 6809 → Périphérie	020
6821 : Sélection des registres internes	020
6821 : Méthode de programmation du PIA 6821	021
6821 : Exemple de programme 01, Port A en Entrée, Port B en Sortie	021
6821 : Exemple de programme 02, Utilisation des lignes de commande CA1 et CB2	021
6821 : Exemple de programme 03, Simulation d'un dialogue entre 2 microprocesseurs.	022
6821 : Exemple de programme 04, Génération d'un système d'impulsion corrélées	023
6821 : Exemple de programme 05, Transmission et réception de données en mode parallèle	026

## 6850

<b>6850 : LES ENTREES – SORTIES LE 6850 ACIA</b>	031
6850 : Organisation des données sérielles	031
6850 : Protocole Start-Stop	031
6850 : Protocoles DTR, XON-XOFF et ETX-ACK	031
6850 : Protocole DTR (Data Terminal Ready)	032
6850 : Protocole XON – XOFF	032
6850 : Protocole ETX-ACK	032
6850 : Brochage	032
6850 : Organisation Interne	033
6850 : Les échanges avec le 6809 se font par :	033
6850 : Les échanges avec les périphériques se font par :	034
6850 : Sélection des Registres Internes	034
6850 : Registre CR (control Register) registre de contrôle	036
6850 : Registre CR : Bits <b>CR1 CR0</b>	036
6850 : Registre CR : Bits <b>CR4 CR3 CR2</b>	036
6850 : Registre CR : Bits <b>CR6 CR5</b>	036
6850 : Registre CR : Bit <b>CR7</b>	037
6850 : Initialisation programmée (MASTER RESET)	037
6850 : Registre SR (Status Register) Registre d'états	038
6850 : Bit <b>SR0</b> : RDRF (Receiver Data Register Full)	039
6850 : Bit <b>SR1</b> : TDRE (Transmit Data Register Empty)	039
6850 : Bit <b>SR2</b> : DCD  (Data Carrier Detect)	039
6850 : Bit <b>SR3</b> : CTS  (Clear To Send)	040
6850 : Bit <b>SR4</b> : FE (Framing Error)	040
6850 : Bit <b>SR5</b> : OVRN (OVer RuN)	040
6850 : Bit <b>SR6</b> : PE (Parity Error)	041
6850 : Bit <b>SR7</b> : IRQ (Interrupt Request)	041
6850 : Transmission	041
6850 : Réception	042
6850 : Rapports 1/16 et 1/64	042
6850 : Rapport 1/1	043
6850 : Fonctionnement général du récepteur	043
6850 : Programmation Routine d'initialisation	043
6850 : Programme d'initialisation pour une émission	043
6850 : Programme d'initialisation pour une réception	044
6850 : Programmation Routine de transmission	044
6850 : 1er exemple de transmission	044
6850 : 2ième exemple de transmission	045
6850 : Exemples de programmation en Réception	046
6850 : 1er exemple de Réception	046
6850 : 2ième exemple de Réception	046

6850 : Exemple de Transmission des caractères Clavier vers Imprimante . . . . .	048
6850 : Exemple de Transmission des caractères vers Imprimante, Protocole DTR . . . . .	050
6850 : Exemple de Transmission des caractères vers Imprimante, Protocole ETX-ACK . . . . .	053
6850 : Exemple de Transmission des caractères vers Imprimante, Protocole XON-XOFF . . . . .	055
6850 : Exemple de Réception des données, avec diverses vérifications, Contrôle ligne RTS . . . . .	056

## 6840

<b>6840 : LES ENTREES – SORTIES LE 6840 TIMER</b> . . . . .	059
6840 : Généralités . . . . .	059
6840 : Brochage . . . . .	059
6840 : Organisation Externe . . . . .	060
6840 : Organisation Externe Schématique . . . . .	060
6840 : Organisation Externe Liaisons avec la périphérie . . . . .	061
6840 : Entrées horloges externes : <b>C1], C2], C3]</b> . . . . .	061
6840 : Entrées GATE : <b>G1], G2], G3]</b> . . . . .	061
6840 : Sorties des temporisateurs : <b>O1, O2, O3</b> . . . . .	061
6840 : Organisation Interne . . . . .	061
6840 : Fonctionnement . . . . .	062
6840 : Adressage, Sélection Du Boîtier . . . . .	062
6840 : Registres de commande CRx . . . . .	064
6840 : Registre d'Etat SR . . . . .	065
6840 : Rôle des registres Tampons (Initialisation) . . . . .	066
6840 : Rôle des compteurs (Initialisation) . . . . .	066
6840 : Différents modes de fonctionnement . . . . .	067
6840 : Mode Astable (Multivibrateur Astable ou Mode continu) <b>Mode 01, 02, 03 et 04</b> . . . . .	067
6840 : Mode Astable Fonctionnement en 2 x 8 bits . . . . .	067
6840 : Mode Astable : Exemple 01 . . . . .	069
6840 : Mode Monostable (Mode monocoup) <b>Mode 05, 06, 07 et 08</b> . . . . .	070
6840 : Mode Mesure d'Intervalle de Temps . . . . .	071
6840 : Mode Mesure d'Intervalle de Temps Comp. Fréquence <b>Mode 09 et 10</b> . . . . .	071
6840 : Mode Mesure d'Intervalle de Temps Comp. Largeur d'impulsion <b>Mode 11 et 12</b> . . . . .	072
6840 : Tableau regroupant tous les Modes de Fonctionnement . . . . .	073
6840 : Exemples de Programmation . . . . .	075
6840 : Exemple de Programmation Mode Astable . . . . .	075
6840 : Exemple de Programmation Mode Monostable . . . . .	076
6840 : Exemple de Programmation Mode Comparaison de Fréquence . . . . .	077

## 6829

<b>6829 : CIRCUIT DE GESTION MÉMOIRE LE 6829 MMU</b> . . . . .	078
6829 : Généralités . . . . .	078
6829 : Principe . . . . .	078
6829 : Utilisation . . . . .	078

## ANN : ANNEXES

ANN : Circuits d'Interfaces de la famille 6800 et 6809 . . . . .	079
ANN : Table ASCII (caractères 0 à 31) . . . . .	080
ANN : Table ASCII de 0 à 127 . . . . .	081
ANN : Table ASCII de 128 à 255 . . . . .	082

## PVM : POINT DE VUE MATERIEL

PVM : Interfaçage des Afficheurs . . . . .	084
PVM : AFF : Diode LED . . . . .	084

## NP : Notes Personnelles

NP : Notes Personnelles . . . . .	084
-----------------------------------	-----

## LR : LIENS RAPIDES

LR : LIENS RAPIDES . . . . .	086
------------------------------	-----

### E/S : GENERALITES

Le 6809 nécessite de communiquer avec l'extérieur.

Les périphériques d'entrées permettent de recevoir des données provenant par exemple de : clavier, disque ...

Les périphériques de sorties permettent d'envoyer des données vers par exemple : écran, imprimante, disque....

Le 6809 ne possède pas d'instruction spéciale pour les E/S.

Les entrées-sorties sont traitées comme de simples cases mémoires placées dans l'espace adressable du 6809.

On peut donc utiliser ces cases mémoires particulières avec toutes les instructions du 6809.

Le PIA 6821 (Peripheral Interface Adapter) permet la liaison parallèle entre le 6809 et le mode extérieur.

Il communique avec le 6809 par l'intermédiaire des bus de données (8 bits), de certains fils du bus d'adresse provenant du 6809 et du bus contrôle.

## 6821 : Port A :

La charge maximale d'une entrée représente 1,5 charge TTL standard.

Les broches du Port A peuvent être lues par le 6809 à la seule condition de respecter les niveaux de tensions.

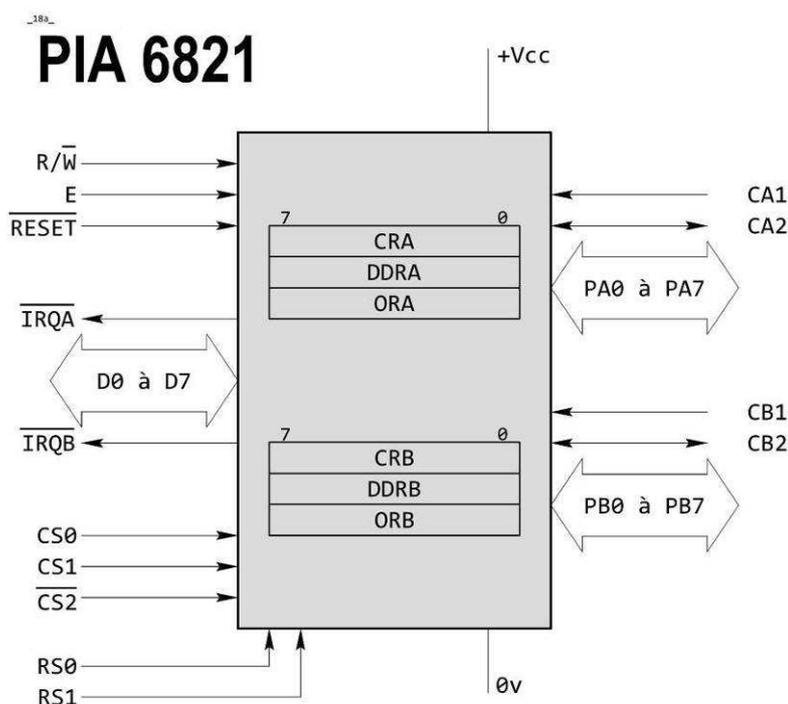
- U > 2 volts pour un 1 logique
- U < 0,8 volts pour un 0 logique

## 6821 : Port B :

Broches en logique trois états ce qui permet de les mettre en haute impédance lorsque le PIA n'est pas sélectionné.

Les sorties du port B (PB0 à PB7) sont compatibles TTL et peuvent fournir jusqu'à 1 mA sous 1,5 volts

## 6821 : Organisation Interne



Le PIA est divisé en 2 parties indépendantes A et B. Le 6821 possède :

- Un port de 8 bits bidirectionnel
- 2 lignes de contrôle par port (soit 4 lignes de contrôle au total)
- 3 registres internes par port (soit 6 registres pour l'ensemble du PIA)

Le PIA se comporte comme seulement 4 positions mémoire, bien qu'il comporte 6 registres internes.

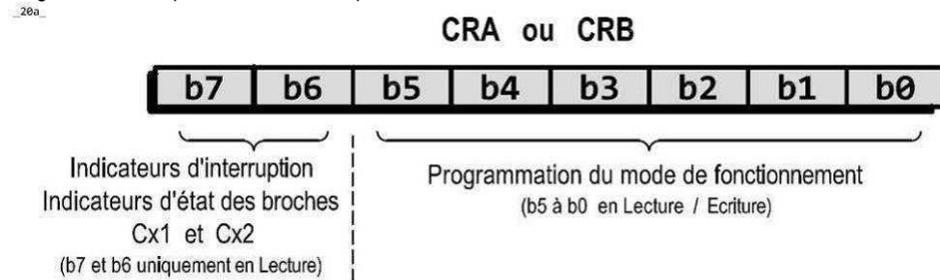
Les registres DDRx et ORx ont la même adresse, le bit 2 du registre de contrôle CRA ou CRB permettra la distinction entre ces deux registres.

Il en résulte qu'avant de programmer les registres (DDRA ou DDRB) et (ORA ou ORB) il faudra programmer le registre CRA ou CRB, quitte à les modifier par la suite.

Le 6821 peut gérer la génération automatique du signal STROBE (validation de données) dans une application mettant en œuvre un protocole d'échange de données de type CENTRONICS.

## 6821 : Registres CRA et CRB (Control Register A et B) Registre de contrôle des ports A et B

Registres 8 bits qui contiennent les paramètres de fonctionnement :



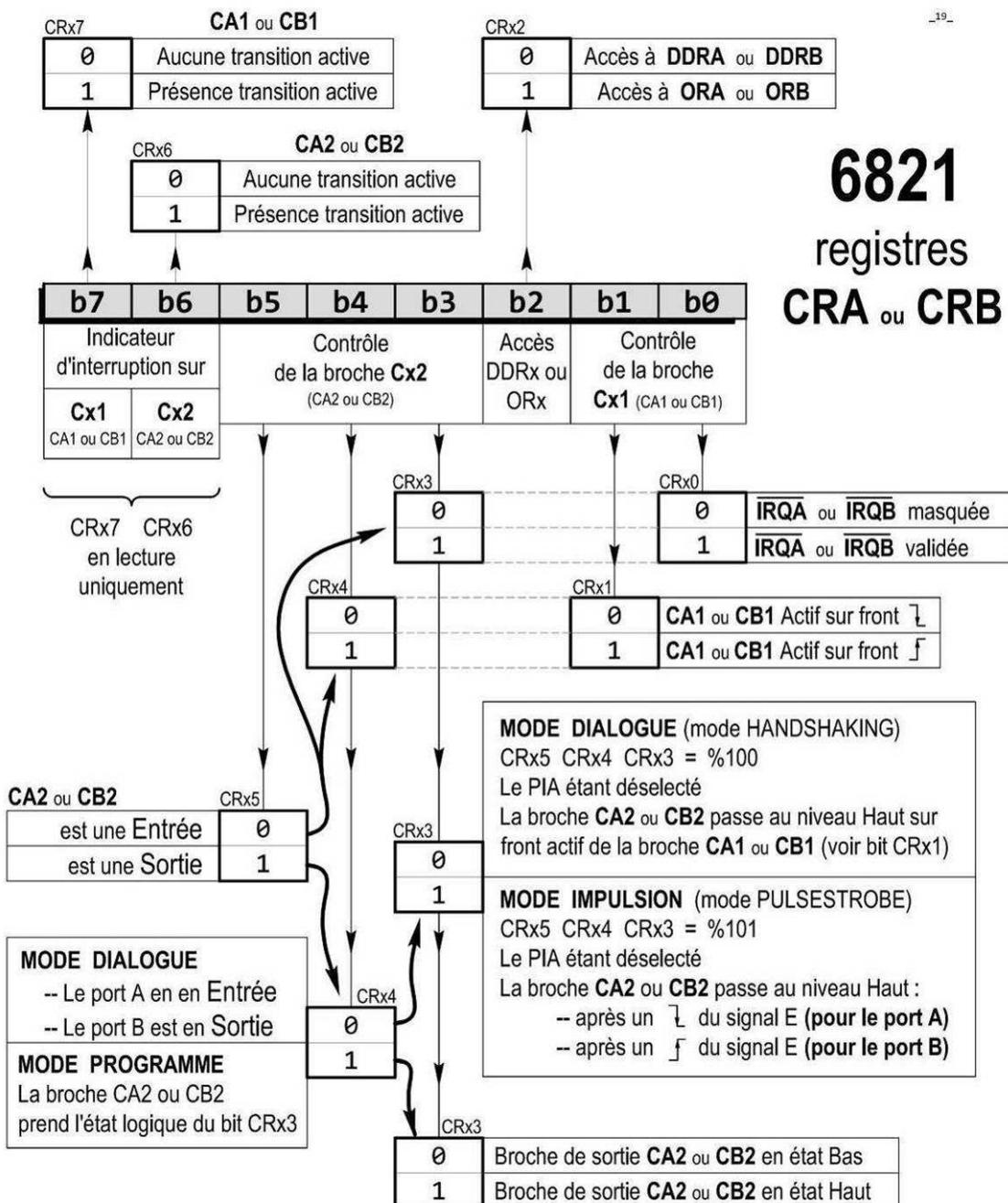
Ce registre concerne la programmation des broches spéciales CA1, CA2 pour le port A et CB1, CB2 pour le port B.

b7 et b6 : Les indicateurs d'état accessibles en lecture, c'est le PIA qui mettra à jour ces bits en fonction de l'activité sur les broches Cx1 et Cx2.

b5 à b0 : Les paramètres de fonctionnement accessible en lecture et en écriture.

### 6821 : Vue complète du registre CRA ou CRB

x sera remplacé par A ou B



## 6821 : Détail du registre CRA ou CRB

### 6821 : CRx0

x sera remplacé par A ou B

Autorise ou non l'envoi d'une interruption sur IRQx|, lorsqu'on a reçu la bonne transition active sur la broche Cx1.

Ce bit CRx0, valide la répercussion des interruptions vers le 6809 par l'intermédiaire des signaux IRQA| (broche 38) et IRQB| (broche 37).

- = 0 n'autorise pas IRQx|
- = 1 autorise l'envoi d'une IRQx|

Si une interruption arrive sur la broche Cx1 alors que la répercussion vers le 6809 est invalidée (cas où le bit CRx0 est à 0) et si le programme positionne le bit CRx0 à 1

Alors le signal de sortie IRQx| (broches 38 ou 37) basculera à l'état Bas pour signifier au 6809 qu'une interruption avait déjà été détectée.

### 6821 : CRx1

x sera remplacé par A ou B

Précise le sens de la transition active attendu sur la broche Cx1

Les bits CRx1 (CRA1 et CRB1) permettent ainsi de sélectionner le front actif des entrées d'interruption des broches CA1 ou CB1.

- = 0 on attend un front descendant sur la broche Cx1
- = 1 on attend un front montant sur la broche Cx1

### 6821 : CRx2 Adressage

x sera remplacé par A ou B

Utilisé pour l'adressage, permet la distinction entre les registres DDRx et ORx

- = 0 accès aux registres **DDRx** (DDRA ou DDRB) registres sens de transfert
- = 1 accès aux registres **ORx** (ORA ou ORB) registres de données

Il faut néanmoins respecter la configuration des broches RS0 et RS1

PIA 6821		A15 à A2 (voir la logique de décodage)			A1	A0	bit n°2 du registre CRx		Adresse
		CS0	CS1	CS2			RS1	RS0	
Partie A	ORA	1	1	0	0	0	1	0 ou 1	Adr
	DDRA	1	1	0	0	0	0	0 ou 1	Adr
	CRA	1	1	0	0	1	0 ou 1	0 ou 1	Adr + 1
Partie B	ORB	1	1	0	1	0	0 ou 1	1	Adr + 2
	DDRB	1	1	0	1	0	0 ou 1	0	Adr + 2
	CRB	1	1	0	1	1	0 ou 1	0 ou 1	Adr + 3

### 6821 : CRx5 CRx4 CRx3

x sera remplacé par A ou B

Définissent et régissent le fonctionnement des broches **Cx2**, broches CA2 et CB2 (broches programmables en Entrée ou en Sortie).

Si CRx5 = 0 le mode de fonctionnement des lignes Cx1 et Cx2 est identique. Ce sont des entrées dont les états électriques sont repérés par les bits CRx7 et CRx6.

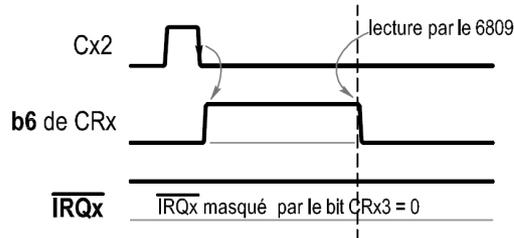
**6821 : CRx5 = 0 la broche Cx2 est en ENTREE** (en entrée d'interruption)

Les broches CA2 et CB2 sont respectivement analogues aux broches CA1 et CB1.

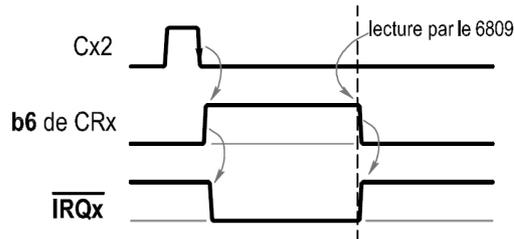
Les bits **CRx4 CRx3** : jouent un rôle identique aux bits CRx1 et CRx0 mais pour la broche Cx2 qui est programmé en entrée.

28b

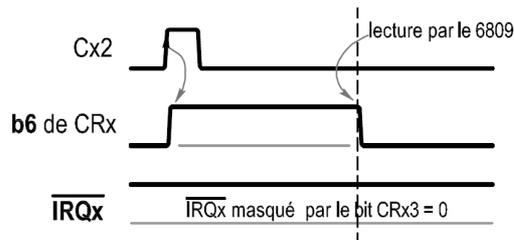
**CRx**  
**b5 b4 b3 = %000 CA2 OU CB2 en Entrée** (b5 = 0)



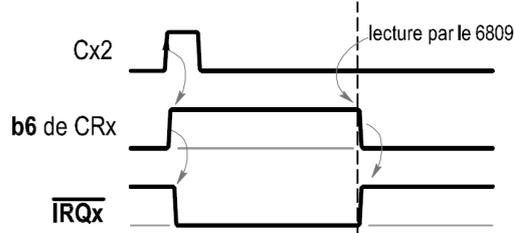
**CRx**  
**b5 b4 b3 = %001 CA2 OU CB2 en Entrée** (b5 = 0)



**CRx**  
**b5 b4 b3 = %010 CA2 OU CB2 en Entrée** (b5 = 0)



**CRx**  
**b5 b4 b3 = %011 CA2 OU CB2 en Entrée** (b5 = 0)



**6821 : CRx5 = 1 la broche Cx2 est en SORTIE** (en sortie de commande)

La programmation en sortie de commande des broches CA2 et CB2 peut par exemple trouver son utilité lors du contrôle de transmission des données parallèle vers la périphérie.

La sortie de commande étant gérée comme signal de validation de données (STROBE).

Le fonctionnement côté A et côté B est différent

Trois MODES de Fonctionnement :

- 1<sup>er</sup> **Mode HANDSHAKE** ou mode Dialogue
- 2<sup>ème</sup> **Mode SET-RESET** ou mode Programmé
- 3<sup>ème</sup> **Mode PULSE-STROBE** ou mode Impulsion

## 6821 : 1er Mode HANDSHAKE ou mode Dialogue CRx5, CRx4, CRx3 = %100

x sera remplacé par A ou B

Dans ce 1<sup>er</sup> mode les ports A et B ont un fonctionnement différent :

- Le port A travaille en ENTREE
- Le port B travaille en SORTIE

La broche Cx2 repasse à l'état Haut  Lorsque que la broche Cx1 recevra le prochain front actif.

La broche Cx2 passe à l'état Bas 

### Pour le port A

Sur le premier front Descendant du fil E (Enable) qui suit un ordre de lecture du registre ORA.

- Passage de la broche CA2 à 0, sur le front négatif ( front descendant) de l'horloge broche E qui suit une lecture du registre ORA.
- Passage de la broche CA2 à 1, quand le bit CRA7 est positionné à 1 lors d'une détection d'un front actif sur la broche CA1.

### Pour le port B

Sur le premier front Montant de la broche E (Enable) qui suit un ordre d'écriture du registre ORB.

Le port B est adapté au dialogue en Sortie, sur le point de vue du matériel électronique, le port B est plus puissant que celui du port A.

- Passage de la broche CB2 à 0, sur le front positif ( front montant) de l'horloge broche E qui suit une écriture du registre ORB.
- Passage de la broche CB2 à 1, quand le bit CRB7 est positionné à 1 lors d'une détection d'un front actif sur la broche CB1.

## 6821 : 2ième Mode SET-RESET ou mode Programmé CRx5, CRx4, CRx3 = %110 ou %111

x sera remplacé par A ou B

La broche Cx2 suit l'état du bit CRx3

Il suffit donc de modifier le contenu du registre CRx (bit CRx3) pour changer l'état de la broche Cx2.

Si bit **CRx3** = 1 alors la broche Cx2 = 1

Si bit **CRx3** = 0 alors la broche Cx2 = 0

Quand CRx5, CRx4, CRx3 = %111, la broche Cx2 passe à 1 quand le bit CRx3 = 1 par écriture dans le registre CRx.

## 6821 : 3ième Mode PULSE-STROBE ou mode Impulsion CRx5, CRx4, CRx3 = %101

x sera remplacé par A ou B

### Pour le port A

La broche CA2 passe à l'état Bas sur le premier front négatif (front descendant) du fil E (Enable) qui suit un ordre de lecture du registre ORA.

La broche CA2 passe à l'état Haut sur le front négatif (front descendant) alors que le 6809 est désélectionné.

20c

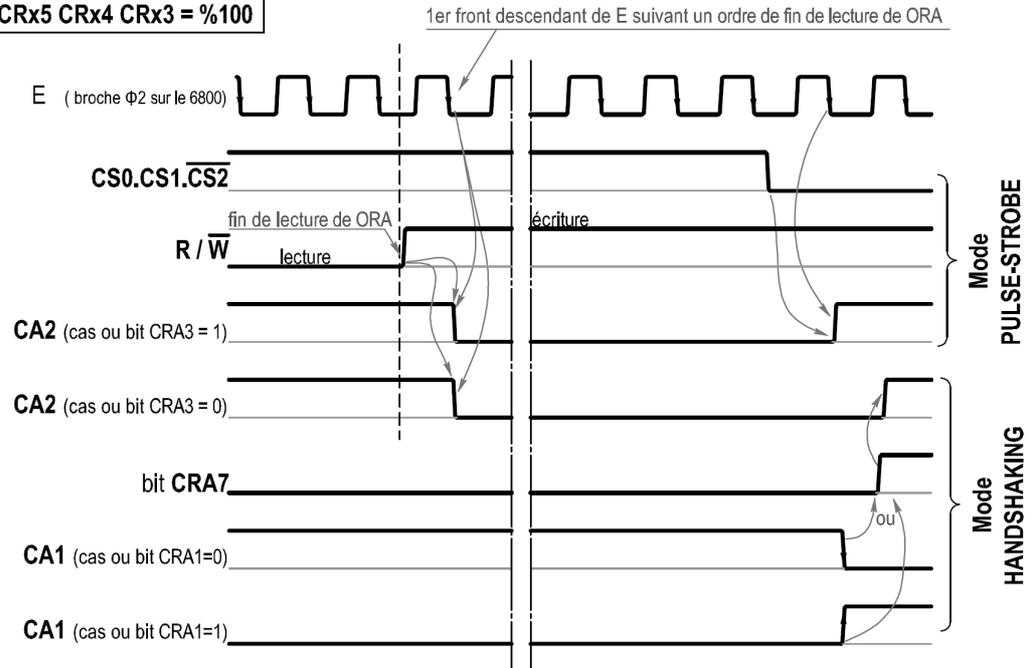
#### 6821 : le Port A (travaille en ENTREE)

#### broche CA2 en Sortie

(car bit CRA5 = 1 et bit CRA4 = 0)

CRx5 CRx4 CRx3 = %101

CRx5 CRx4 CRx3 = %100



### Pour le port B

La broche CB2 passe à l'état Bas sur le premier front Montant de la broche E (Enable) qui suit un ordre d'écriture dans le registre ORB.

La broche CB2 repasse à l'état Haut sur le premier front Montant de la broche E (Enable) alors que le 6821 est désélectionné (et donc lorsque que Cx1 recevra le prochain front actif).

20d

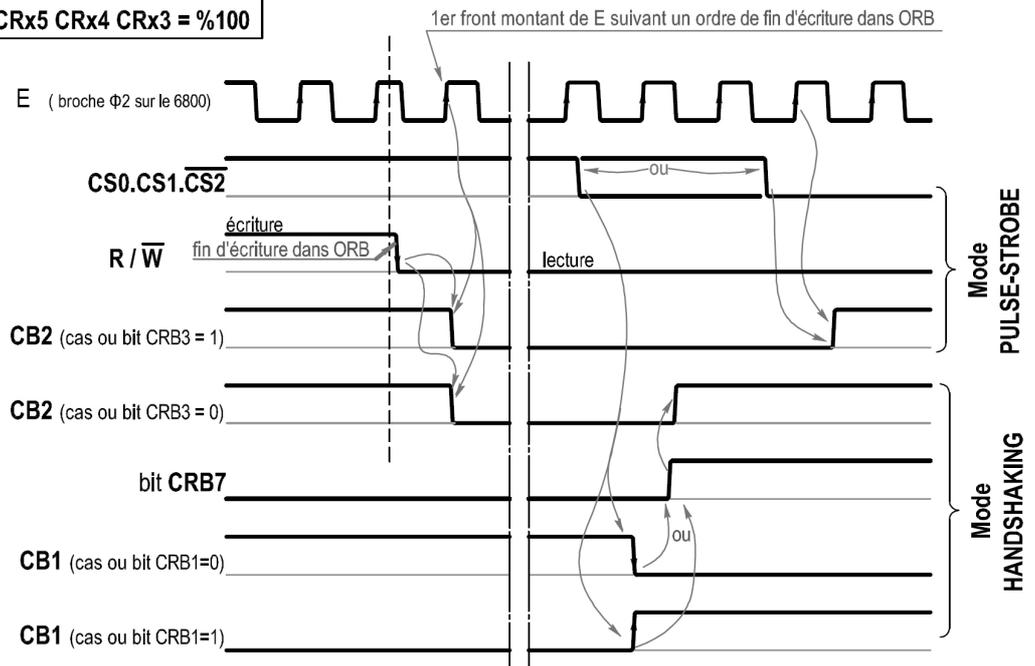
#### 6821 : le Port B (travaille en SORTIE)

#### broche CB2 en Sortie

(car bit CRB5 = 1 et bit CRB4 = 0)

CRx5 CRx4 CRx3 = %101

CRx5 CRx4 CRx3 = %100



## **6821 : Les bits CRx7 et CRx6**

x sera remplacé par A ou B

Ils sont des drapeaux internes d'interruption qui indique le passage à l'état Bas des broches [IRQA] et [IRQB] en fonction des 4 broches spéciales CA1, CA2 et CB1, CB2 et de leur programmation.

Ces indicateurs peuvent alors être mis en œuvre lors d'un masquage d'interruption.

Ils sont RAZ lors de chaque lecture du registre de données ORA ou ORB.

Après une telle réinitialisation, la prochaine interruption qui pourra être prise en compte devra intervenir au moins un cycle d'horloge E plus tard.

Après un RESET

- CRx6 et CRx7 sont à 0
- Les broches CA2 et CB2 sont en entrée
- L'on n'autorise pas d'interruption
- Les ports A et B sont en entrée

### **6821 : CRx6**

x sera remplacé par A ou B

Flag d'interruption lorsque les broches Cx2 sont en Entrée (en lecture uniquement) :

Si les broches Cx2 sont en Sortie alors ce bit CRx6 est forcé à 0 et non affecté par les transitions sur les broches Cx2.

= 0	par lecture du registre ORx
= 1	lorsqu'on reçoit la transition active attendue sur Cx2

### **6821 : CRx7**

x sera remplacé par A ou B

Flag d'interruption lorsque les broches Cx1 sont en Entrée (en lecture uniquement) :

Si les broches Cx1 sont en Sortie alors ce bit CRx7 est forcé à 0 et non affecté par les transitions sur les broches Cx1.

= 0	par lecture du registre ORx
= 1	lorsqu'on reçoit la transition active attendue sur Cx1

Attention ! : Ce bit CRx7 est mis à zéro après une lecture du registre ORx par le 6809 :

- Du registre de contrôle CRx
- Du registre de donnée

## **6821 : Programmation des broches CA1, CA2, CB1 et CB2 en ENTREE**

x sera remplacé par A ou B

Lorsque les broches **CA1 CA2 CB1** ou **CB2** sont programmées en entrée d'interruption, il est nécessaire qu'au moins une fois le signal E ait été à l'état haut pendant que le signal externe devant déclencher l'interruption était actif.

Les broches d'entrées **CA1** et **CB1** sont initialisées par le contenu des bits CRx1 et CRx0.  
Ces broches positionnent l'indicateur CRx7 quand elles sont actives.

Les broches d'entrées-sorties **CA2** et **CB2** (travaillent en entrées quand CRx5 est à 0), sont initialisées par le contenu des bits CRx4 et CRx3.

Ces broches positionnent l'indicateur CRx6 quand elles sont actives.

## **6821 : Broches CAx CBx : Modes Automatiques**

x sera remplacé par A ou B

Lorsque les lignes CB1 et CB2 sont programmées comme des sorties, les modes dits "automatiques" produisent des impulsions sur ces lignes sans intervention du 6809 lors d'une opération d'entrée-sortie (envoi ou réception d'une donnée).

En contre partie, la durée et polarité des impulsions échappent au contrôle du programmeur.

Les sorties Cx2 remplissent des fonctions différentes :  
-- La partie A sert à la réception des données.  
-- La partie B sert à la transmission des données.

Dans chaque cas juste au dessus, les 2 lignes de contrôle de la partie correspondante (CA1, CA2 pour la partie A et CB1, CB2 pour la partie B) sont mobilisées pour l'appel-réponse.

## **6821 : Broches CAx CBx : Modes Manuels**

x sera remplacé par A ou B

Dans les modes dits "Manuels", la polarité de l'impulsion peut-être programmée par le bit CRx3.  
La durée de l'impulsion est également réglable.

## **6821 : CA1 et CB1**

Ces broches sont toujours en entrées.  
Ces broches sont initialisées par les bits CRA1, CRA0 ou CRB1, CRB0.  
Ces broches positionnent les bits indicateurs CRA7 ou CRB7 quand elles sont actives.

17b

CRx1	CRx0	$\overline{IRQx}$	CRx7
0	0	inhibée $\overline{IRQA} = 1$ $\overline{IRQB} = 1$	Mis à 1 sur $\downarrow$ appliquée sur CA1 ou CB1
0	1	autorisée	Mis à 1 sur $\downarrow$ appliqué sur CA1 ou CB1 simultanément $\overline{IRQA}$ ou $\overline{IRQB}$ $\downarrow$
1	0	inhibée $\overline{IRQA} = 1$ $\overline{IRQB} = 1$	Mis à 1 sur $\uparrow$ appliquée sur CA1 ou CB1
1	1	autorisée	Mis à 1 sur $\uparrow$ appliqué sur CA1 ou CB1 simultanément $\overline{IRQA}$ ou $\overline{IRQB}$ $\downarrow$

CA1 pour le port A CB1 pour le port B.

Actifs sur un front montant ou descendant suivant la programmation du bit CRx1 et le positionnement du bit CRx7.

CA1 et CB1 ne peuvent qu'être en entrées  
Deux broches en entrée d'évènement plus spécialisées dans la détection de ce qui devrait être des interruptions.

Ces broches positionnent directement les indicateurs d'interruption des registres CRx, soit les bits CRx7.

## **6821 : CA2 et CB2**

Ces broches peuvent travailler en entrées, quand les bits CRA5 ou CRB5 sont à 0.  
Ces broches sont initialisées par les bits CRA4, CRA3 ou CRB4, CRB3.  
Ces broches positionnent les bits indicateurs CRA6 ou CRB6.

### **6821 : Broche CA2**

- Associée à la partie A
- Peut être programmée en entrée ou en sortie.
- Permet de contrôler l'échange à travers le PIA.
- Son rôle est déterminé en fonction des bits CRA3, CRA4 et CRA5
- Elle positionne en entrée, un flag d'interruption bit CRA6
- CA2 en entrée représente une charge TTL
- CA2 en sortie peut alimenter une charge TTL

### **6821 : Broche CB2**

- Associée à la partie B
- Peut être programmée en entrée ou en sortie
- Permet de contrôler l'échange à travers le PIA
- Son rôle est déterminé en fonction des bits CRB3, CRB4 et CRB5.
- Elle positionne en entrée, un flag d'interruption bit CRB6
- CB2 en entrée est compatible TTL
- CB2 en sortie est compatible TTL et peut fournir jusqu'à 1mA sous 1,5 volt au système périphérique connecté.
- Les broches CA2 et CB2 peuvent être programmées par logiciel, en entrée d'interruption à l'image des broches CA1 et CB1.
- Les broches CA2 et CB2 peuvent aussi être programmées en sorties et être alors destinées à commander tout ou partie d'un système périphérique au même titre que l'une des 16 broches des ports A et B.

## **6821 : Programmation des broches CA2 et CB2 en SORTIE**

x sera remplacé par A ou B

Ces broches peuvent travailler en sorties, quand les bits CRA5 ou CRB5 sont à 1.  
Le bit CRA4 ou CRB4 permet alors de choisir deux modes de fonctionnement :

### **6821 : CRA4 ou CRB4 = 0 Mode Dialogue**

Dans ce mode et contrairement aux autres modes les ports A et B ont un fonctionnement différent :

**Le port A travaille en ENTREE.** La lecture du registre ORA entraîne le passage à l'état Bas de la broche CA2 (voir le bit CRA4 = 0)

**Le port B travaille en SORTIE.** L'écriture dans le registre ORB entraîne le passage à l'état Bas de CB2

Dans ces deux cas ci-dessus, c'est le contenu du bit CRA3 ou CRB3 qui détermine le mode de retour à l'état initial des broches CA2 ou CB2.

**Si le bit CRA3 ou CRB3 = 0** Alors les broches CA2 ou CB2 repassent à l'état Haut quand un front actif est détecté sur la broche CA1 ou CB1.

**Si le bit CRA3 ou CRB3 = 1** Alors les broches CA2 ou CB2 reprennent leur état initial quand le circuit est désélectionné sur :  
-- Un front descendant pour CA2  
-- Un front montant pour CB2

### **6821 : CRA4 ou CRB4 = 1 Mode Programmé**

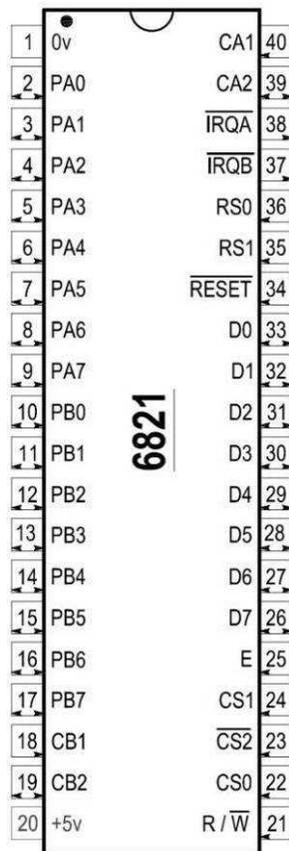
Les broches CA2 et CB2 prennent l'état du bit CRA3 ou CRB3.

Par exemple pour CRA3 = 0 la broche CA2 est à l'état Bas, l'écriture de CRA3 = 1 entraîne le passage de CA2 à l'état Haut. Le fonctionnement est identique pour la broche CB2 avec CRB3.



## 6821 : Organisation Externe

### 6821 : Brochage



### 6821 : Liaison avec le bus de données du 6809

De D0 à D7, ces 8 lignes sont bidirectionnelles directement reliées au bus de données du 6809.

Elles assurent l'échange des données. La circuiterie de sortie des broches D0 à D7 est à trois états.

Lorsque le composant n'est pas sélectionné  $CS0.CS1.CS2 = 0$  alors ces 8 broches ne sont pas utilisées, elles sont dans l'état haute impédance.

Le sens de transfert des informations est donné par l'état de la broche R/W

### 6821 : Liaison avec le bus d'adresses du 6809

#### 6821 : Broches CS0, CS1 et CS2 | (Chip Select Line) Sélection de boîtier :

Permettent l'adressage physique du boîtier, si ces trois lignes  $CS0.CS1.CS2 = \%110$  alors le 6821 est sélectionné.

On appliquera les broches adresses haute A15, A14, A13 sur les entrées CS0. CS1. CS2| lorsqu'il s'agit d'un petit système. On pourrait également mettre A12, A11, A10 pour un autre plan d'adressage.

L'état des signaux CS0, CS1 et CS2| doit être stabilisé pendant toute la durée du niveau Haut du signal E lorsqu'un transfert d'informations doit être effectué.

CS0, CS1, CS2| connectées directement au 6809 ou en passant par un circuit assurant un décodage d'adresse.

## **6821 : Broches RS0 et RS1** (Register Select Line)

Permettent de sélectionner (d'adresser) les registres internes (4 positions en mémoire).  
On leur appliquera nécessairement les broches d'adresse base A0 et A1.

Sélection de registre RS0, RS1 (broches en entrée). Ces 2 broches sont utilisées conjointement aux registres de contrôles CRA et CRB afin d'accéder à la totalité des 6 registres internes du 6821.

Ces deux broches RS0 et RS1 ne permettent théoriquement que l'adressage de quatre registres. Un bit de chacun des registres de contrôle CRx assure l'aiguillage vers les 2 registres manquant au décodage primaire assuré par les broches RS0 et RS1.

L'état des signaux RS0 et RS1 doit être stabilisé pendant toute la durée du niveau Haut de la broche E lorsqu'un transfert d'informations entre le 6821 et 6809 est à mettre en œuvre.

## **6821 : Liaison avec le bus de contrôle du 6809**

### **6821 : Broche E** (Enable)

Signal d'activation des échanges, assure la synchronisation des transferts d'information entre le 6821 et le 6809, la synchronisation de tous les autres signaux du composant est réalisée à partir des seuls fronts montant ou descendant de ce signal E.

Reçoit un signal d'horloge, généralement connecté à la broche E du 6809 (anciennement  $\Phi 2$  pour le 6800) pour assurer des échanges synchrones.

### **6821 : Broche RESET** broche en entrée

Sensible au niveau bas, reçoit le signal RESET général de la carte qui initialise le PIA.  
Cette broche doit être à l'état haut au moins une micro seconde avant la première sélection du PIA et donc avant d'adresser le PIA.

Remise à zéro des registres internes et de ce fait, programme en entrée toutes les broches des ports A et B ainsi que les 2 broches spéciales CA2 et CB2 (sachant que CA1 et CB1 ne peuvent qu'être en entrées).

De plus durant le RESET, les interruptions sont masquées, ainsi le 6809 ne risque pas d'être interrompu par inadvertance.

#### **Lors d'un RESET**

- Tous les bits du registre de contrôle sont à 0
- Les registres adressables immédiatement après le RESET sont les registres de contrôle --
- CRx et le registre de sens de transfert de données DDRx
- Les interruptions IRQA| et IRQB| sont inhibées.

#### **Après un RESET**

- Les bits b7 et b6 du registre CRx sont à 0
- Les broches Cx2 sont en Entrée
- On n'autorise pas d'interruption
- Les ports A et B sont en entrée

Le fait que toutes les broches en liaison avec la périphérie sont programmées en entrée, cela sécurise les organes périphériques, il ne risque pas d'être actionnés de manière intentionnelle.

### **6821 : Broche R/W** | : ( Read / Write | )

Lecture / Ecriture. Fixe le sens des transferts.

#### **R/W | = 0**

Pour un transfert du 6809 → 6821, écriture des informations dans le 6821, le bus de données du PIA est en entrée (si le boîtier est sélectionné et que la broche E est à l'état haut).

#### **R/W | = 1**

Pour un transfert du 6821 → 6809, lecture des informations du 6821, le bus de données du PIA est en sortie (si le boîtier est sélectionné et que la broche E est à l'état haut).

## **6821 : Broches IRQA | et IRQB |** (IR... = Interrupt Request) broches en sortie

2 lignes d'interruption supportant le OU câblé.

Reliées à IRQI, FIRQI ou NMII du 6809, ces lignes permettent d'interrompre l'exécution du programme en cours et d'appeler un sous-programme de traitement d'interruption.

L'interruption peut se faire soit directement, soit par l'intermédiaire d'une circuiterie de priorité d'interruption.

Ce type de signal peut être traditionnellement connecté en "OU câblé"; il peut donc être relié les uns aux autres et même à d'autres signaux de même type issus d'autres composants (ces sorties sont à drain ouvert).

Chaque ligne IRQ est associée à un port : IRQA| au port A et IRQB| au port B et aussi respectivement aux bits 6 et 7 des registres de contrôles CRA et CRB.

## **6821 : Liaison avec la périphérie : lignes de transfert**

### **6821 : Broches PA0 à PA7**

Permettent de transmettre ou de recevoir des informations sur 8 bits.  
Suivant la programmation du registre DDRA :

- = 0 la broche est en Entrée
- = 1 la broche est en Sortie

Exemple : Si le bit DDRA3=0 la broche PA3 sera une entrée

### **6821 : Broches PB0 à PB7**

Permettent de transmettre ou de recevoir des informations sur 8 bits.  
Suivant la programmation du registre DDRB :

- = 0 la broche est en Entrée
- = 1 la broche est en Sortie

Contrairement au port A, ces 8 lignes sont en logique 3 états

Quand le port B est mis en entrée, il est vu comme de la haute impédance. Il n'y a pas de résistances de Pull Up comme sur le port A.

Exemple : Si le bit DDRB5=1 la broche PB5 sera une sortie

## 6821 : Fonctionnement

### 6821 : Transfert d'une donnée Périphérie --> 6809

Exemple :

En mettant le registre DDRA = \$00, alors PA0-PA7 toutes les broches du port A sont en ENTREE.  
 La donnée disponible sur le port A est transmise à l'amplificateur de bus de données.  
 Cette donnée ne transite pas par ORA, il n'y a donc pas de mémorisation des données en entrée.  
 Ce transfert se fait sous le contrôle du registre CRA.  
 Un signal actif sur CA1 sera validé sur IRQA| si et seulement si le contenu du registre de contrôle le permet.  
 La fonction est identique pour la partie B

### 6821 : Transfert d'une donnée 6809 --> Périphérie

Exemple :

En mettant le registre DDRB = \$FF, alors PB0-PB7 toutes les broches du port B sont en SORTIE.  
 La donnée disponible sur le bus de donnée du 6809 est chargée dans le registre de sortie B.  
 La donnée est disponible tant qu'une nouvelle écriture n'est pas intervenue.

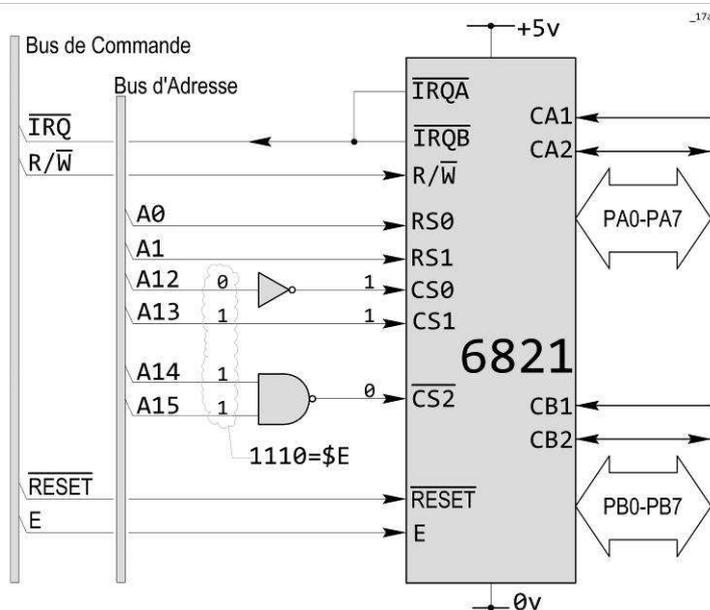
### 6821 : Sélection des registres internes

Le 6809 accède au registre interne du PIA par l'intermédiaire des lignes de sélection de boîtier CS0, CS1 et CS2| et par la sélection de registre RS0 et RS1.

6 registres internes pour 4 positions mémoires. C'est le rôle du bit CRx2 qui permettra de distinguer les registres DDRx et ORx.

Autrement dit, pour adresser les 6 registres avec 2 broches RS0 et RS2, on utilise en plus l'état du bit CRx2. Voir également le bit CRA2 et CRB2, sélection des registres et adressage du 6821.

Les broches A2 à A15 sont reliées à une logique de décodage qui détermine l'adresse de base du PIA.  
 Les broches A0 et A1 sont reliées à RS0 et RS1 pour accéder aux adresses Adr, Adr + 1, Adr + 2, Adr + 3



Exemple d'un 6821 implanté en \$E000

A15	A14	A13	A12							A1	A0	
1	1	1	0							0	0	\$E000 DDRA / ORA
1	1	1	0							0	1	\$E001 CRA
1	1	1	0							1	0	\$E002 DDRB / ORB
1	1	1	0							1	1	\$E003 CRB
\$E			\$0			\$0						

## 6821 : Méthode de programmation du PIA

Compte tenu du fait que les registres DDRx et ORx ont la même adresse, il faut programmer le registre CRx afin de pouvoir y accéder.

### 6821 : Exemple de programme 01, Port A en Entrée, Port B en Sortie

Dans cet exemple, on n'utilise pas les lignes de commandes, et il n'y a pas eu de RESET.  
On désire lire la donnée présente sur le port A et l'afficher sur le port B

```
                ; Prog 6821-01
                ;-----
PIACRA EQU $1000
PIACRB EQU $1010
PIADRA EQU $2000
PIADRB EQU $2010
PIAORA EQU $3000
PIAORB EQU $3010

                ;
                CLRA                ; raz de A (A=$00)
                STA PIACRA         ; $00->PIACRA
                STA PIACRB         ; $00->PIACRB
                STA PIADRA         ; port A en entrée
                COMA                ; A passe de $00 à $FF
                STA PIADRB         ; $FF->PIADRB port B en sortie
                STA PIACRA         ; $FF->PIACRA
                STA PIACRB         ; $FF->PIACRB b2 de CRB = 1
Br_Encore LDA PIAORA              ; lecture port A
                STA PIAORB         ; affichage port B
                BRA Br_Encore     ; branch.toujours vers Br_Encore
                END                ;
```

### 6821 : Exemple de programme 02, Utilisation des lignes de commande CA1 et CB2

Dans cet exemple, on effectue la lecture du port A après avoir eu un front descendant sur CA1.

Puis on écrit sur le port B après avoir eu un front montant sur CB2.  
Le port B recopie le port A.

```
                ; Prog 6821-02
                ;-----
PIACRA EQU $1000
PIACRB EQU $1010
PIADRA EQU $2000
PIADRB EQU $2010
PIAORA EQU $3000
PIAORB EQU $3010

                ;                CLRA                ; raz de A (A=$00)
                STA PIACRA         ; $00->PIACRA
                STA PIACRB         ; $00->PIACRB
                STA PIADRA         ; port A en entrée
                COMA                ; A passe de $00 à $FF
                STA PIADRB         ; $FF->PIADRB port B en sortie
                LDA #$04           ; $04 = %0000 0100
                STA PIACRA         ; $04->PIACRA
                LDA #$14           ; $14 = %0001 0100
                STA PIACRB         ; $14->PIACRB

                ;
                ATCA1 LDA PIACRA    ; | attente front descendant de CA1
                BPL ATCA1          ; | _test de b7 de CRA
                LDA PIAORA         ; lecture port A

                ;
                ATCB2 LDB PIACRB    ; | | attente front montant de CB2
                ROLB              ; | | décalage du b6 en position 7
                BPL ATCB2         ; | | _test de b6 de CRB, branch si > 0
                STA PIAORB         ; lecture port B
                LDA PIAORB         ; lecture fictive de ORB pour raz de b6
                BRA ATCA1         ; branche toujours vers ATCA1
                END                ;
```



## 6821 : Le programme d'initialisation

Doit également contenir l'initialisation de la pile et celle des vecteurs d'interruptions.

```

; Prog 6821-03a
;
;-----Définition des registres de programme-----
1000          ORG    $1000    ;
3FFF PILE    EQU    $3FFF    ;
2000 SPIRQ   EQU    $2000    ; Sous Prog d'interruption
F400 ADPIA   EQU    $F400    ; Adresse du PIA
F400 ORA     EQU    ADPIA    ; -----port A   $F400
F400 DDRA   EQU    ADPIA    ;                   $F400
F401 CRA    EQU    ADPIA+1  ;                   $F401
F402 ORB    EQU    ADPIA+2  ; -----port B   $F402
F402 DDRB   EQU    ADPIA+2  ;                   $F402
F403 CRB    EQU    ADPIA+3  ;                   $F403
;-----Initialisation du PIA-----
1000 10CE 3FFF      LDS    #PILE    ; init du pointeur de la pile
1004 8E   2000      LDX    #SPIRQ   ; init du vecteur d'interruption
1007 BF   FFF8      STX    $FFF8    ; vecteur pour IRQ | FFF8 et FFF9
100A 4F           CLRA           ; RAZ de A
100B B7   F401      STA    CRA      ; accès à DDRA bit CRA2=0
100E B7   F403      STA    CRB      ; accès à DDRB bit CRB2=0
1011 B7   F402      STA    DDRB     ; port B en Entrée car A=$00->DDRB
1014 43           COMA           ; pour A passage de $00 à $FF
1015 B7   F400      STA    DDRA     ; port A en Sortie car A=$FF->DDRA
1018 86   3C        LDA    #$3C     ; %0011 1100 voir ci-dessus
101A B7   F401      STA    CRA      ; init de CRA
101D 86   05        LDA    #$05     ; %0000 0101 voir ci-dessus
101F B7   F403      STA    CRB      ; init de CRB
```

**6821 : A la fin du programme d'initialisation** le PIA est prêt à transférer des octets du port A vers le port B  
Il faut écrire l'octet à transmettre dans le registre ORA, puis activer CA2 pour lancer la procédure de transfert.

```

; Prog 6821-03b
;
;-----Routine de transfert d'un octet dans le port A-----
;-----Programme principal
1022 B6   2017      LDA    OCTET    ; charg registre ORA avec l'octet à transférer
1025 B7   F400      STA    ORA      ;
1028 86   34        LDA    #$34     ; %00110100 CA2 passe à 0
; ça active en même temps CB1
; CA2 en sortie, mode SET-RESET
; CA2 passe de H à B ==> front descendant sur CB1
102A B7   F401      STA    CRA      ;
102D 12           NOP             ; }--attente de prise en compte du front
102E 12           NOP             ; }
102F 12           NOP             ; }
1030 20   FE        BRA    *        ; bouclage du prog sur lui-même
; ##### Bouclage Infini #####
; puis branch à $2000 (routine traitement
; d'interruption)
; si interruption IRQ voir SPIRQ
```

**6821 : Cette routine de transfert est interrompue** par l'interruption IRQ due au front actif sur CB1

Le microprocesseur interrompt le déroulement de la routine de transfert, puis exécute le programme d'interruption approprié.

Dans le cas où plusieurs périphériques sont connectés sur la ligne IRQ du microprocesseur, il faut tester les bits d'état de ces périphériques, afin de déterminer l'origine de l'interruption.

```

; Prog 6821-03c
;
;-----Routine d'interruption-----
2000          ORG    $2000    ;
2000 B6   F403      LDA    CRB      ; test du bit CRB7
2003 85   80        BITA    #$80    ; %1000 0000
2005 26   04        BNE    GISSE    ;
2007 12           NOP             ; }--tester d'autre PIA
```

```

2008 12          NOP          ; }
2009 12          NOP          ; }
200A 12          NOP          ; }
                ; On teste ensuite, les différents registres
                ; d'état des autres périphériques, le niveau
                ; de priorité des uns par rapport aux autres
                ; est donné par le logiciel. La routine GISSE
                ; permet de lire le port B
                ;
200B B6  F402     GISSE  LDA   ORB      ; lecture du port B
200E B7  2018     STA   MEMOIR      ;
2011 86  3C       LDA   #$3C        ; %0011 1100
                ; CA2 en sortie, Accès à ORA
                ; CA2 passe de B à H
                ; CA2 repasse à l'état initial ainsi que CB1
2013 B7  F401     STA   CRA         ;
2016 3B          RTI              ; retour au programme principal
                ;
2017 12          OCTET  FCB   $12    ; origine ?
2018          MEMOIR  RMB   1       ; destination ?
                END

```

La lecture du port B, ramène l'indicateur d'état bit CRB7 à sa valeur initiale.

## 6821 : Exemple de programme 04, Génération d'un système d'impulsion corrélées

On désire générer 4 signaux du type de la figure ci-dessous.

Chaque signal est décalé d'un quart de cycle par rapport au signal adjacent.

On peut facilement les programmer symétriques ou non symétriques.

Une solution consiste à décaler un registre 8 bits comportant un groupement de 4 bit à 1, les autres étant à 0, comme dans la figure ci-dessous.

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	1
1	1	0	0	0	0	1	1
1	0	0	0	0	1	1	1
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	1	1	1	0	0

Il suffit alors de prélever les signaux sur les sorties paires ou impaire de la partie A ou B.

Pour régler la période de globale, on modifie le délai entre deux décalages. L'intérêt de cette solution réside en l'absence d'un test systématique à la fin de chaque période.

```

                ; Prog 6821-04a
                ;
                ;*****
                ;   Génération de 4 signaux décalés les uns par
                ;   rapport aux autres d'un quart de cycle
                ;*****
8000          ORG   $8000          ;
                ;-----Adresse des registres de la partie B
                ED0C  RCRB   EQU   $ED0C      ; Reg. de contrôle partie B
                ED08  RDDR  EQU   $ED08      ; Reg. sens transfert partie B
                ED08  ROR_B EQU   $ED08      ; Reg. Donnée partie B
8000 7F  ED0C          CLR   RCRB          ; adr reg. sens transf CRB2= 0
8003 86  FF           LDA   #$FF           ; Déf. Toutes lignes B comme sorties
8005 B7  ED08          STA   ROR_B          ;
8008 86  04           LDA   #%0000100     ; adrs reg.données CRB2 = 1
800A B7  ED0C          STA   RCRB          ;
                ;
                ;-----Charger mot définissant impulsions dans reg. données
800D 86  F0           LDA   #%11110000    ;
800F B7  ED08          STA   ROR_B          ;
                ;

```

```

;-----Boucle sans fin de génération d'impulsions
8012 8D 05      GNIMPU  BSR  DELAI          ; Délai réglable
8014 79 ED08    ROL  ROR_B          ;
8017 20 F9      BRA  GNIMPU          ;
;
;-----S/P DELAI permettant de régler la fréquence
8019 86 64      DELAI  LDA  #100          ; Valeur de réglage
801B 4A          DECA                    ;
801C 26 FD      BNE  DELAI+2          ; si A /= 0 décrémenter A
801E 39          RTS                    ; fin du S/P DELAI

```

## **6821 : Quelques explications sur le programme ci-dessus**

On peut envisager un deuxième système de signaux dont les différentes composantes ne sont pas liées entre elles. Dans ce cas, la solution consiste à ranger au préalable des mots dans une table qui seront ensuite transmis un par un dans le registre de données du 6821.

Les différents mots sont inscrits dans un tableau auquel le 6809 peut accéder rapidement en mode indexé.

Le phénomène est cyclique, la fin du tableau correspond à la fin d'un cycle.

Le 6809 doit détecter cette éventualité et se reporter de nouveau au début du tableau.

Quelques cycles machines nécessaires à l'actualisation de l'index doivent être comptabilisés dans le décompte des durées des impulsions.

```

; Prog 6821-04b
;
;*****
; Génération d'un système de signaux dont les formes
; sont définies par les valeurs inscrites dans un tableau
;*****
8000          ORG  $8000          ;
;-----Adresse des registres de la partie B
ED0C RCRB  EQU  $ED0C          ; Reg. de contrôle partie B
ED08 RDRB  EQU  $ED08          ; Reg. sens transfert partie B
ED08 ROR_B EQU  $ED08          ; Reg. Donnée partie B
8000 7F ED0C  CLR  RCRB          ; adr reg. sens transf CRB2= 0
8003 86 FF    LDA  #$FF          ; Déf. Toutes lignes B comme sorties
8005 B7 ED08  STA  ROR_B          ;
8008 86 2C    LDA  #%00101100    ; CRB2 = 1 adressage reg. Données.
; Brève impulsion sur la sortie CB2
; à chaque écriture
800A B7 ED0C  STA  RCRB          ;
;-----Initialisation à chaque début de cycle
800D 108E 8100 GNIMPU LDY  #$8100    ; adrs début tableau
8011 C6 41      LDB  #65          ; Nb de découpages élémentaire à
; l'intérieur d'un cycle. Lg tableau
8013 A6 A0      VALSVT LDA  ,Y+    ; charger une valeur tableau,
; VAleur SuiVanTe
8015 B7 ED08    STA  ROR_B          ;
8018 8D 05      BSR  DELAI          ; Délai réglable
801A 5A          DECB                    ; Fin Tableau ?
801B 26 F6      BNE  VALSVT          ; sinon, valeur suivante
801D 20 EE      BRA  GNIMPU          ; si oui, revenir début tableau
;-----S/P DELAI permettant de régler la durée de
;-----l'impulsion élémentaire
801F 8E 07D0    DELAI  LDX  #2000    ; valeur de réglage
8022 30 1F      LEAX -1,X          ;
8024 26 FC      BNE  DELAI+3          ; si (X) /= 0 décrémenter (X)
8026 39          RTS                    ; Fin S/P DELAI

```

## **6821 : Quelques explications sur le programme ci-dessus**

Le deuxième exemple est plus adapté ou séquenceur électronique, les durées des impulsions sont beaucoup plus longues. Pour obtenir des durées doubles, triples, il suffit d'appeler plusieurs fois le sous-programme DELAI.

Les séquences présentées s'effectuent indéfiniment. Pour les arrêter, il faut utiliser les réserves propres du système utilisateur, c'est-à-dire les interruptions manuelles.

Pour s'adresser au registre sens de transfert RDDRB nous employons CLR RCRB qui agit en fait sur 6 bits CRB0 à CRB5.

Comme les lignes CB1 et CB2 ne sont pas affectées à ce niveau, cette instruction est tolérable.

Dans le premier exemple, seules les sorties impaires sont utilisées.

Si l'on écrit `LDA #%01010101` au lieu de `LDA #$FF`, les sorties paires peuvent éventuellement jouer le rôle d'entrées.

Dans le second exemple, à chaque écriture dans le registre de données de la partie B, une impulsion est générée automatiquement sur la sortie CB2.

Dans le premier exemple, l'instruction qui permet de changer d'état est ROL ROR\_B. Elle implique une lecture de l'ancienne valeur puis une écriture de la nouvelle valeur.

Son emploi dans ce cas particulier évite l'écriture des séquences de branchement à la fin de chaque cycle.

Ces séquences rendent le système d'impulsion non symétrique, surtout aux fréquences élevées. La fréquence la plus élevée sera celle obtenue en enlevant le branchement vers DELAI.

Le sous-programme DELAI doit laisser à l'organe électromécanique suffisamment de temps pour commuter.

## **6821 : Exemple de programme 05, Transmission et réception de données en mode parallèle**

Un système processeur-interface PIA peut-être programmé en transmetteur ou récepteur de données en mode parallèle.

Dans une transmission ou réception de données par interface série ACIA 6850, les bits SR0 et SR1 informe le 6809 sur la disponibilité du récepteur ou du transmetteur.

Dans le cas d'une transmission parallèle les bits CRx7 jouent à peu près le même rôle. Ils sont liés aux états électriques des lignes Cx1.

Dans le PIA 6821, il existe quelques possibilités supplémentaires concernant ce mode de programmation des signaux de dialogue.

### **6821 : Ex-prog 05 : En réception**

Supposons que les lignes PA0 à PA7 soient programmées comme des entrées.

Lorsque le transmetteur externe a déposé une donnée sur les entrées du port A, il avertit le récepteur que la donnée est prête en actionnant l'entrée de dialogue CA1.

La transition active peut-être positive (front montant) ou négative (front descendant) selon la valeur du bit CRA1. Ce signal sur CA1 positionne le bit d'état CRA7 à 1.

Le 6809 reconnaît CRA7=1, lit la donnée disponible dans le registre de données RDDRA.

Si le mode de fonctionnement de CA2 est "Automatique", une brève impulsion sera générée sur la sortie CA2 par la lecture même du registre de données.

Si le mode de fonctionnement de CB2 est "Manuel" le 6809 doit modifier le bit CRA5 pour produire une impulsion sur la ligne CB2.

Cette impulsion avertit en retour le transmetteur que la donnée a été reçue avec succès, et qu'un autre cycle de transmission peut-être entamé.

Il est possible de se servir de la partie B pour effectuer une réception de données.

Cependant, après chaque lecture du registre de données RDDRB, le 6809 doit effectuer une écriture factice dans ce même registre pour produire l'impulsion de dialogue sur la ligne CB2, cette servitude ralentit la vitesse d'acquisition de l'ensemble.

## 6821 : Ex-prog 05 : En transmission

La partie B est conçue pour ce fonctionnement. Elle possède également un mode "automatique" et un mode "manuel".

Au début d'un cycle de transfert, le 6809 lit l'état électrique de la ligne CB1 à travers le bit CRB7 pour savoir si le récepteur est prêt à recevoir.

Si le bit CRB7=1, la donnée peut-être déposée dans le registre données RDDRb.

Cette écriture produit automatiquement une impulsion sur la sortie CB2 si un tel mode est sollicité. On a la possibilité de produire cette impulsion manuellement en jouant sur le bit CRB5.

Cette impulsion avertit le récepteur externe qu'une donnée est déposée sur les lignes PB0 à PB7. Après lecture, le récepteur actionne de nouveau l'entrée CB1 et autre cycle recommence.

On peut éventuellement se servir de la partie A du PIA 6821 pour effectuer une transmission.

Cependant, après chaque écriture dans le registre de données, le 6809 doit effectuer une lecture factice du même registre pour produire une brève impulsion sur la sortie CA2, ce qui ralentit la vitesse d'acquisition.

Dans cette application, nous nous servons du même PIA 6821 pour tester simultanément une transmission et une réception avec une poignée de main complète.

Il faut effectuer au préalable les liaisons électriques suivantes :

- Relier PB0...PB7 à PA0...PA7
- Relier CB2 à CA1
- Relier CB1 à CA2

L'application suivante comporte en réalité 3 programmes différents :

- Les 2 premiers se servent de la partie B comme transmetteur et de la partie A comme récepteur.
- Dans le 3ième programme, B joue le rôle de récepteur et A de transmetteur.

La programmation n'est pas identique si l'on considère le mode génération des impulsions de dialogue sur les lignes CA2 et CB2.

En étudiant ces programmes, il faut imaginer la présence de 2 PIA physiquement séparés pour pouvoir saisir l'utilité réelle des signaux de dialogue.

A l'exécution, chaque programme transfère une zone mémoire vers une autre pour simuler la transmission et la réception de données.

```

; Prog 6821-05
;
;*****
; Transmission et Réception de donnée par PIA page VI.44
;*****
;-----Adresses registres Port A
ED04 RCRA EQU $ED04 ; Reg. Contrôle A
ED00 RDDRA EQU $ED00 ; Reg. Sens transfert A
ED00 ROR_A EQU $ED00 ; Reg. Donnée A
;-----Adresses registres Port B
ED0C RCRB EQU $ED0C ; Reg. Contrôle B
ED08 RDDRb EQU $ED08 ; Reg. Sens transfert B
ED08 ROR_B EQU $ED08 ; Reg. Donnée B
;
;*****
;-----1ier PROGRAMME
;-----Port B = transmetteur Port A = Récepteur
;-----Brève impulsion en mode automatique
8000 ORG $8000 ; adrs du 1er programme
8000 7F ED04 CLR RCRA ; CRA2=0 adressage RDDRA
8003 7F ED0C CLR RCRB ; CRB2=0 adressage RDDRb
8006 4F CLRA ;
8007 B7 ED00 STA RDDRA ; port.A 8 entrées
800A 4A DECA ;
800B B7 ED08 STA RDDRb ; port.B 8 sorties
```

```

800E 86 2C          LDA  #00101100      ; impulsion automatique CB2 +
                                     ; adressage reg. donnée
                                     ; CRB7=1 sur transmission négative
8010 B7 ED0C       STA  RCRB          ;
8013 B7 ED04       STA  RCRA          ; impulsion automatique CB2 +
                                     ; adressage reg. donnée
                                     ; CRA7=1 sur transmission négative
8016 108E 9000     LDY  #9000         ; adrs zone à transférer
801A 8E 1000       LDX  #1000         ; bloc de 4096 octets
801D CE A000       LDU  #A000         ; Adrs de rangement
8020 B6 ED00       LDA  ROR_A         ; initialiser le transfert en
                                     ; mettant CRB7=1, récepteur prêt
                                     ;
                                     ;-----Boucle de transfert. ATTENTION
                                     ;-----CRB7 est remis à 0 après chaque écriture dans ROR_B
                                     ;-----CRA7 est remis à 0 après chaque lecture de ROR_A
8023 7D ED0C       TRRP1 TST  RCRB          ; récepteur prêt ?
8026 2A FB 8023    BPL  TRRP1         ; sinon attendre
8028 A6 A0         LDA  ,Y+          ; donnée à transmettre
802A B7 ED08       STA  ROR_B         ; Envoyer et produire impulsion
                                     ; sur CB2 vers CA1
802D 7D ED04       ATTEN1 TST  RCRA        ; donnée reçue sur port A ?
8030 2A FB 802D    BPL  ATTEN1        ; si CRA7=0 attendre
8032 B6 ED00       LDA  ROR_A         ; lire donnée puis production
                                     ; impulsion sur CA2 vers CB1
8035 A7 C0         STA  ,U+          ; rangement donnée
8037 30 1F         LEAX -1,X         ; bloc épuisé ?
8039 26 E8 8023    BNE  TRRP1        ; sinon continuer
803B 3F           SWI                ;
                                     ;
                                     ;*****
                                     ;-----2ième PROGRAMME
                                     ;-----Port B = transmetteur Port A = Récepteur
                                     ;-----production manuelle des impulsions
                                     ;-----sur les sortie CA2 et CB2
803C 7F ED04       CLR  RCRA          ; CRA2=0 adressage RDDRA
803F 7F ED0C       CLR  RCRB          ; CRB2=0 adressage RDDRB
8042 4F           CLRA                ;
8043 B7 ED00       STA  RDDRA         ; port.A 8 entrées
8046 4A           DECA                ;
8047 B7 ED08       STA  RDDRB        ; port.B 8 sorties
804A 86 36         LDA  #00110110     ; CB1 et CB2 initialement bas
                                     ; adressage reg. données +
                                     ; CRA7=CRB7=1 sur transmission
positive
804C B7 ED0C       STA  RCRB          ;
804F B7 ED04       STA  RCRA          ;
8052 108E 9000     LDY  #9000         ; adrs zone à transférer
8056 8E 1000       LDX  #1000         ; bloc de 4096 octets
8059 CE A000       LDU  #A000         ; Adrs de rangement
805C 8A 08         ORA  #00001000     ; porter CA2 Haut, donc CRB7=1
                                     ; pour initialiser le transfert
805E B7 ED04       STA  RCRA          ;
8061 84 F7         ANDA #11110111     ; remettre CA2 bas
8063 B7 ED04       STA  RCRA          ;
                                     ;
                                     ;-----Boucle de transmission
8066 7D ED0C       TRRP2 TST  RCRB          ; récepteur prêt ?
8069 2A FB 8066    BPL  TRRP2         ; sinon attendre
806B A6 A0         LDA  ,Y+          ; Donnée à transmettre
806D B7 ED08       STA  ROR_B         ; Envoi
8070 B6 ED0C       LDA  RCRB          ; Mot de configuration
8073 8A 08         ORA  #00001000     ; porter CB2 haut, donc CRA7=1
                                     ; pour avertir récepteur
8075 B7 ED0C       STA  RCRB          ;
8078 84 F7         ANDA #11110111     ; remettre CB2 Bas
807A B7 ED0C       STA  RCRB          ;
                                     ;
                                     ;-----Boucle de réception
807D 7D ED04       RECEP2 TST  RCRA        ; donnée reçue sur port A ?
8080 2A FB 807D    BPL  RECEP2        ; sinon attendre
8082 B6 ED00       LDA  ROR_A         ; lire donnée
8085 A7 C0         STA  ,U+          ; rangement donnée

```

```

8087 B6 ED04          LDA RCRA          ; Mot configuration port A
808A 8A 08           ORA  #00001000      ; porter CA2 haut, donc CRB7=1
                                     ; pour avertir transmetteur

808C B7 ED04          STA RCRA          ;
808F 84 F7           ANDA #11110111     ; remettre CA2 Bas
8091 B7 ED04          STA RCRA          ;
                                     ;
                                     ;
;-----Test fin de la séquence transmission-réception
8094 30 1F           LEAX -1,X          ; bloc épuisé ?
8096 26 CE 8066      BNE TRRP2         ; sinon continuer
8098 3F              SWI                ;
                                     ;
; *****
;-----3ième PROGRAMME
;-----Port B = récepteur  Port A = Transmetteur
;-----production des impulsions par des lectures et écritures
;-----factices des registres de données

8099 7F ED04          CLR RCRA          ; CRA2=0 adressage RDDRA
809C 7F ED0C          CLR RCRB          ; CRB2=0 adressage RDDRB
809F 4F              CLRA              ;
80A0 B7 ED08          STA RDDRb         ; port.B 8 entrées
80A3 4A              DECA              ; (A)=$FF
80A4 B7 ED00          STA RDDRA         ; port.A 8 sorties
80A7 86 24           LDA  #00100100     ; impulsion sur sortie CA2
                                     ; et CB2 + adressage reg. donnée
                                     ; + CR7 = 1 sur transition négative
                                     ;
80A9 B7 ED04          STA RCRA          ;
80AC B7 ED0C          STA RCRB          ;
80AF 108E 9000        LDY  #$9000       ; adrs zone à transférer
80B3 8E 1000          LDX  #$1000       ; bloc de 4096 octets
80B6 CE A000          LDU  #$A000       ; Adrs de rangement
                                     ;
;-----Produire une impulsion vers le transmetteur
;-----port A pour indiquer que le récepteur est
;-----prêt et pour initialiser le transfert
80B9 B7 ED08          STA ROR_B         ; écriture factice dans le registre
                                     ; donnée récepteur pour générer
                                     ; un état bas sur CB2 (CA1). Le
                                     ; bit CRA7 est mis à 1
                                     ;
;-----Boucle de transmission avec port A
80BC 7D ED04          TRRP3 TST RCRA      ; réception prêt ?
80BF 2A FB 80BC      BPL TRRP3         ; sinon attendre
80C1 A6 A0           LDA  ,Y+          ; donnée à transmetteur
80C3 B7 ED00          STA ROR_A         ; envoi de donnée
80C6 B6 ED00          LDA  ROR_A         ; lecture factice pour produire
                                     ; impulsion avertissant récepteur
                                     ;
;-----Boucle de réception avec port B
80C9 7D ED0C          RECEP3 TST RCRB     ; donnée présente sur port B ?
80CC 2A FB 80C9      BPL RECEP3        ; si CRB7=0, attendre
80CE B6 ED08          LDA  ROR_B         ; lire donnée
80D1 B7 ED08          STA  ROR_B         ; écriture factice pour mettre
                                     ; CB2 état bas avertissant le
                                     ; transmetteur (CRA7=1)
80D4 A7 C0           STA  ,U+          ; rangement donnée
80D6 30 1F           LEAX -1,X          ; bloc épuisé ?
80D8 26 E2 80BC      BNE TRRP3         ; sinon continuer
80DA 3F              SWI                ;

```

## **6821 : Quelques explications sur le programme ci-dessus**

Le premier type de transfert est le plus rapide, ce type de transfert émettant de brèves impulsions automatiquement après chaque lecture ou chaque écriture sur les lignes CA2 et CB2.

Cependant, si l'un des deux systèmes est plus lent que l'autre, il faut prolonger "manuellement" les impulsions. C'est l'esprit du second programme.

Dans le 3ième programme, à cause de la conception du PIA 6821, on est obligé d'écrire des instructions factices de lecture et d'écriture dans les registres de données pour produire des impulsions sur les sorties CA2 et CB2.

On a souvent tendance à les supprimer car elles ne jouent apparemment aucun rôle, d'où l'importance d'une documentation très explicite.

On peut confier le rôle de "maître" soit au transmetteur, soit au récepteur.

Dans les deux cas, l'organe assurant le contrôle doit détecter le nombre maximum de données requises et mettre en veilleuse l'autre pour transférer éventuellement d'autres données ultérieurement.

On remarque dans les trois programmes, la présence d'une séquence d'initialisation indispensable au démarrage du transfert.



## **6850 : Protocol DTR (Data Terminal Ready)**

On emploie une ligne de contrôle DTR issue d'une sortie du récepteur, qui sera connectée à l'entrée DCD de l'émetteur.

Cette ligne DCD est ramenée sur l'entrée DCD du 6850 au travers d'un convertisseur de ligne qui inverse également le signal.

Quand le buffer de réception est presque vide, le récepteur met la ligne DTR à +12V et le transfert est autorisé.

Lorsque le buffer de réception est presque plein, le récepteur force la ligne DTR à - 12 volts pour avertir l'émetteur qu'il faut arrêter momentanément l'envoi des données.

Dans ce protocole, la poignée de main est assurée au **niveau matériel** par l'implantation d'une ligne supplémentaire.

## **6850 : Protocole XON – XOFF**

La poignée de main est exécutée au **niveau logiciel**.

Quand le buffer de réception est presque plein, le récepteur renvoie vers l'émetteur, via la ligne de transmission le code DC3 (Device Control 3) caractère ASCII \$13.

Dans ce cas l'émetteur suspend l'envoi des données.

Quand le buffer de réception est presque vide le caractère renvoyé vers l'émetteur est le code DC1 caractère ASCII \$11. L'émetteur reprendra le transfert de données.

Ce protocole suppose que l'émetteur doit surveiller constamment le retour possible d'un caractère DC1 ou DC3.

On remarque que l'interface fonctionne simultanément en transmission et en réception, on dit qu'il fonctionne en FULL-DUPLEX.

Par opposition le mode HALF-DUPLEX implique soit une transmission, soit une réception, mais jamais une simultanéité des deux.

## **6850 : Protocole ETX-ACK**

Le buffer d'émission doit être plus petit que le buffer de réception (taille calculée en octets).

L'émetteur envoie un bloc entier de données, avec un caractère spécial marquant la fin du bloc. C'est le code ASCII \$03 (ETX - End Of Text).

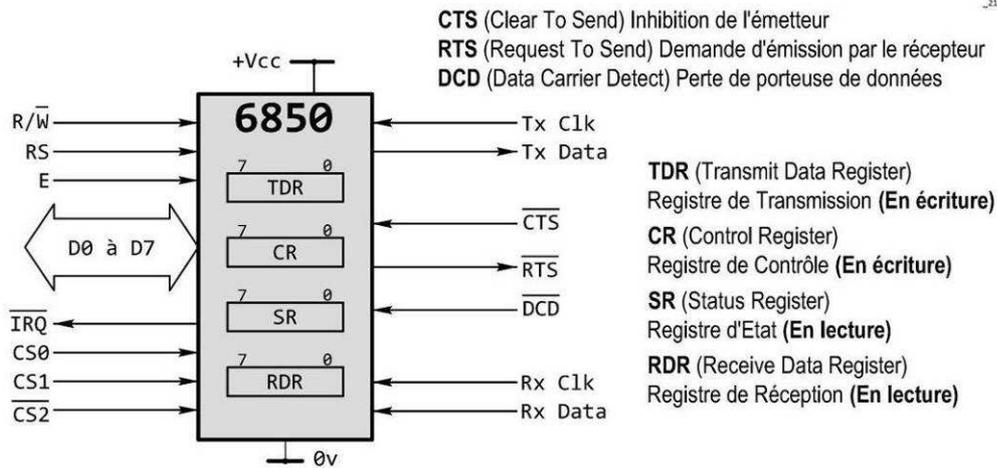
L'émetteur arrête ensuite l'envoi des caractères.

De son côté, le récepteur exploite les données à sa propre vitesse. Arrivant au caractère ETX le récepteur retourne à l'émetteur le caractère ASCII \$06 (ACK – acknowledge) indiquant à l'émetteur que le récepteur est prêt à accepter un autre bloc de données.

## **6850 : Brochage**



## 6850 : Organisation Interne



## 6850 : Les échanges avec le 6809 se font par :

-- Bus de données d0 à d7, ces 8 broches bidirectionnelles sont reliées au 6809, elles assurent l'échange des données entre le 6809 et le 6850. Si elles ne sont pas utilisées ces broches sont en haute impédance.

Ce bus assure en outre :

- Programmer le registre de contrôle
- Ecrire dans le registre transmission
- Lire dans le registre réception
- Lire le registre d'état

-- Bus d'adresses 4 lignes allant sur les broches :

- 3 Lignes de validation de boîtier **CS0, CS1, CS2** | (Chip Select Line), qui permettent l'adressage physique du boîtier. On appliquera des bits d'adresse haute sur ces 3 broches lorsque l'on a affaire à un petit système.  
**CS0, CS1** Sont validées au niveau Haut, **CS2** | est validé au niveau bas, d'où la combinaison de sélection pour **CS0, CS1, CS2** | = %110
- Une entrée de sélection de registre **RS** (Register Select Ligne) qui permet de sélectionner les registres internes.

-- Bus de contrôle

- Une entrée **E** (Enable Line) signal d'activation des échanges, reçoit un signal de l'horloge généralement E du 6809 (anciennement  $\Phi 2$  sur le 6800)
- **R/W** | (Read/Write) : lecture écriture
- **IRQ** | (Interrupt ReQuest line) : reliée à **IRQ**, **FIRQ** ou à **NMI** du 6809, cette sortie permet d'interrompre le 6809.

## 6850 : Les échanges avec les périphériques se font par :

- Une broche de transmission de données **TxD** (Transmit Data Line). Cette broche de sortie assure la transmission des données en série.
- Une broche de réception de données **RxD** (Receive Data Line). Cette broche d'entrée réceptionne les données série en provenance de la périphérie.

-- 3 broches de contrôle assurant la synchronisation des transferts :

**CTS** : (Clear To Send) Permet l'inhibition de l'émetteur.  
Broche d'entrée, permet le contrôle automatique de la fin de transmission par un MODEM.  
Non utilisée, elle doit être au niveau Bas.

**RTS** : (Request To Send) Permet une demande d'émission.  
Broche de sortie, permet de piloter un périphérique ou un modem.

**DCD** : (Data Carrier Detect) Perte de la porteuse de données.  
Broche d'entrée, permet le contrôle de la réception.  
Non utilisée, elle doit être au niveau Bas.

-- 2 entrées d'horloge permettant de fixer les cadences de transmission et de réception.

**TxCk** (Transmit Clock)  
Entrée horloge de transmission, elle sert de référence pour la transmission des données.

**RxCk** (Receive Clock)  
Entrée horloge de réception, elle est utilisée pour la synchronisation des informations reçues.

Les échanges avec le périphérique se font suivant le mode START-STOP, adapté pour les transmissions de 50 bps à 500 kbps (bps = bit par seconde).

Chaque caractère en informant de :

- 7 ou 8 bits.
- Suivi d'un bit de parité paire ou impaire.
- Suivi d'un ou deux bits de Stop (suivant la cadence désirée).

Les bits de Start et Stop ont pour but de synchroniser deux caractères consécutifs.

## 6850 : Sélection des Registres Internes

Bien que le 6850 ait 4 registres internes de 8 bits (2 à lecture seule, 2 à écriture seule), le 6809 voit cette interface comme s'ils occuperaient seulement 2 positions mémoires.

- CR** (Control Register) registre de contrôle.  
En Ecriture seule, contiennent les paramètres de fonctionnement.
- SR** (Statut Register) registre d'état.  
En Lecture seule, contiennent les informations sur les opérations en cours.
- TDR** (Transmit Data Register) registre de transmission de données.  
En Ecriture seule, contient le mot de 8 bits à émettre.
- RDR** (Receive Data Register) registre de réception de données.  
En Lecture seule, reçoit le mot de 8 bits en provenance de la périphérie.

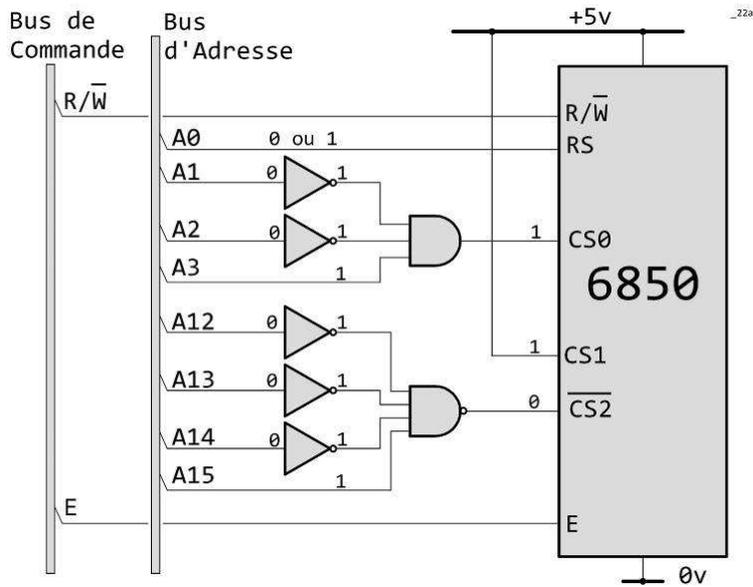
Dès la validation du boîtier par CS0, CS1 et CS2], l'adressage des 4 registres du 6850, se fera en conjonction des broches R/W] et RS.

Le boîtier sera sélectionné si CS0.CS1.CS2] = %110

L'entrée RS recevra le bit A0 pour que les adresses soient consécutives.

ACIA 6850	Broches du 6809 →	Logique de décodage A15 à A1			A0	R/W	Adresse	
		Broches du 6850 →	CS0	CS1	CS2	RS		
<b>Ecriture</b> dans le registre de contrôle	<b>CR</b>	1	1	0	<b>0</b>	<b>0</b>	<b>Adr</b>	Reçoit les paramètres du fonctionnement
<b>Lecture</b> du registre d'état ou de statut	<b>SR</b>	1	1	0	<b>0</b>	<b>1</b>	<b>Adr</b>	Permet au 6809 de connaître l'état du registre d'émission, du registre de réception, du CTS 106, du DCD 109, de IRQ
<b>Ecriture</b> dans le registre Transmission de donnée	<b>TDR</b>	1	1	0	<b>1</b>	<b>0</b>	<b>Adr + 1</b>	L'octet à écrire va dans le registre TDR puis il est transmis par le 6850 au registre TSR pour être transmis sur la broche TxD 103 C'est le $\bar{L}$ de la broche E qui charge les 8 bits à émettre dans TDR, le bit 0 est émis en premier
<b>Lecture</b> du registre Réception de donnée	<b>RDR</b>	1	1	0	<b>1</b>	<b>1</b>	<b>Adr + 1</b>	Reçoit l'octet à lire, c'est le $\bar{L}$ de la broche E qui permet la lecture du registre RDR L'octet reçu par la broche RxD 104 est stocké bit par bit dans le registre RSR puis le 6850 transmet cet octet au registre RDR

**Exemple :** d'implantation d'un 6850 (implanté à l'adresse \$8008 et \$8009)

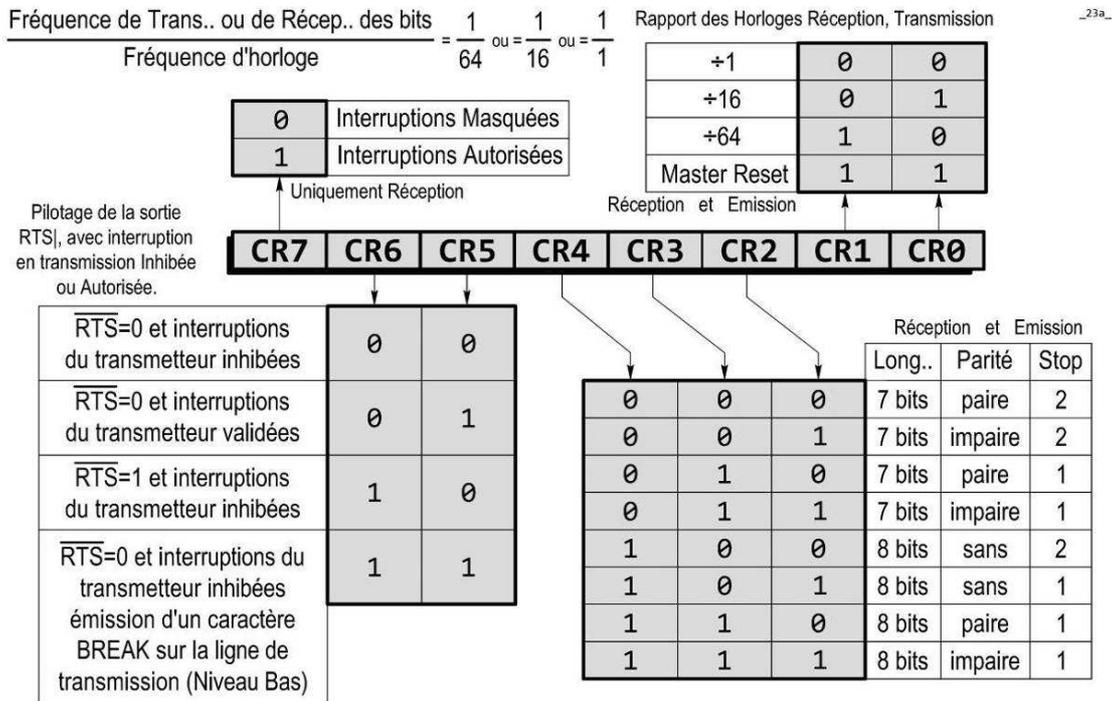


A15	A14	A13	A12					A3	A2	A1	A0	
1	0	0	0					1	0	0	0	\$8008
1	0	0	0					1	0	0	1	\$8009
			\$8				\$0		\$8		\$9	

## 6850 : Registre CR (control Register) registre de contrôle

Contient les paramètres de fonctionnement (format, vitesse, ...) de transmission de la réception.

Précise le mode fonctionnement : De la partie réception et de la partie émission  
Des interruptions  
De la broche RTS|



### 6850 : Bits CR1 CR0 :

Détermine les facteurs de division des horloges de transmission de réception.  
= %11 Est utilisé pour l'initialisation programmée du registre CR,  
Cette combinaison est appelée **MASTER RESET**.

### 6850 : Bits CR4 CR3 CR2 :

Détermine le format du mot transmis ou reçu : -- Longueur du mot  
-- Parité  
-- Nombre de bits d'arrêt

### 6850 : Bits CR6 CR5 :

Contrôle la partie transmission. Actifs en transmission seulement, ces bits déterminent la méthode de transfert. L'envoi d'un caractère vers la périphérie est toujours précédé d'un test sur le bit TDR du registre SR (lecture du bit SR1). Pour s'assurer que celui-ci est vide.

Les 4 combinaisons permettent :

- De fixer l'état de sortie de la demande d'envoi RTS
- De valider ou non une autorisation d'interruption TIE (Transmit Interrupt Enable).  
Si TIE est validé alors le bit SR7 suivra SR1 du même registre d'état.  
Donc si le bit SR1 égal à 1, alors le registre de transmission est vide, on générera une interruption de transmission (broche IRQ| au niveau bas) et le bit SR7 sera positionné à 1.
- De générer un BREAK (niveau bas sur la ligne)

## **6850 : Bit CR7 :**

Valide ou non une autorisation d'interruption de réception RIE (RIE = Receiver Interrupt Enable).

Si RIE est validé, alors le bit SR7 suivra le bit SR0 du même registre d'état.

Donc si le bit SR0=1 (indiquant que le registre réception est plein), alors la sortie IRQ passe au niveau Bas et le bit SR7 sera positionné à 1.

## **6850 : Initialisation programmée (MASTER RESET)**

A la différence du PIA 6821 qui possède une broche RESET, le 6850 a son propre circuit de mise sous tension qui le maintient dans son état inhibé jusqu'à son initialisation programmée, ceci afin d'éviter la transmission des d'informations erronées.

Pas de broche RESET sur le 6850 due à la limitation du nombre de broches 24 broches pour le 6850.

Le MASTER RESET est obtenu par programme en mettant b1 b0 du registre CR à %11, ce qui impose la broche RTS à l'état haut (donc sans action), les interruptions du transmetteur son inhibées.

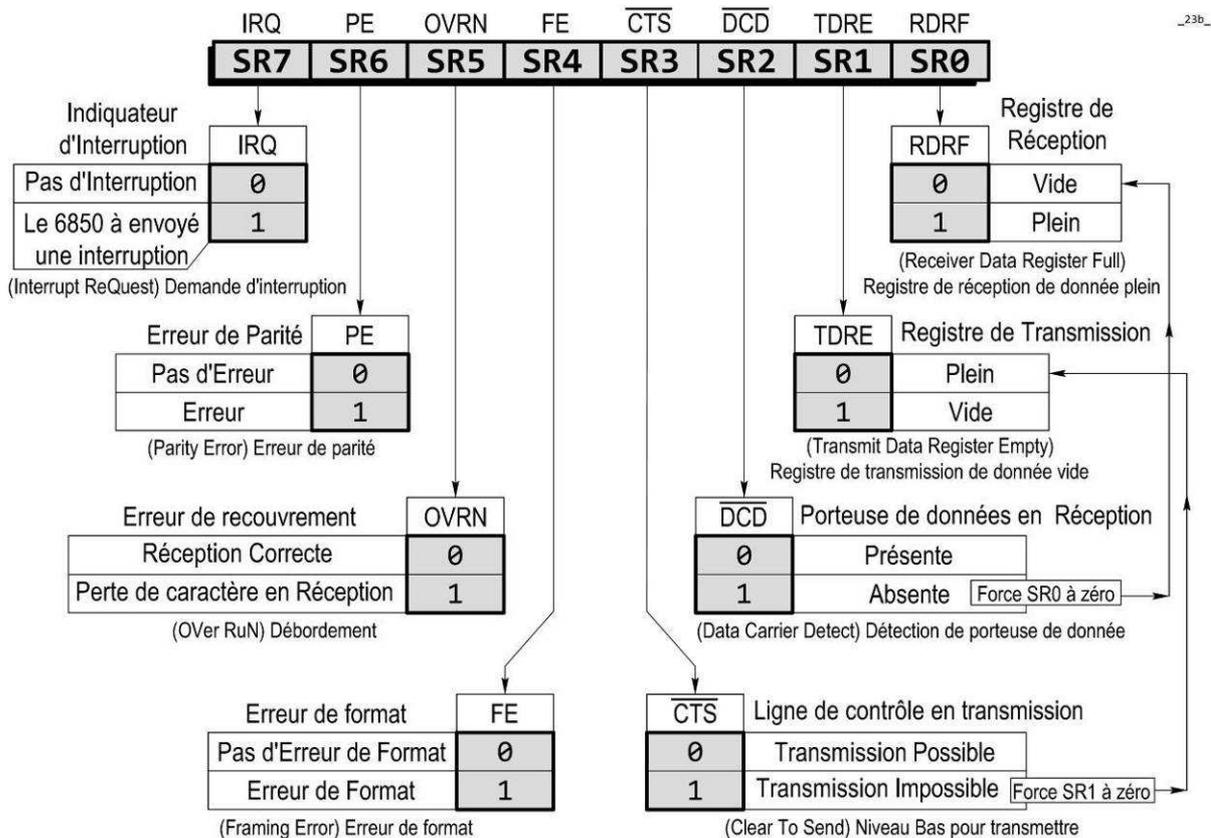
Après un MASTER RESET les bits b6 b5 du registre CR ne sont plus inhibés, le registre CR peut alors être programmé. D'autre part, tous les bits du registre SR sont mis à zéro sauf b3 et b2.

Avant de programmer un mot le 6850, la mise sous tension doit être suivie d'un MASTER RESET (initialisation programmée).

## 6850 : Registre SR (Status Register) Registre d'états

8 bits à lecture seule, le registre SR appelé registre d'état, contient le mot d'état qui renseigne le 6809 sur les opérations en cours.

Contient les informations d'état en provenance : Du registre de transmission, Du registre de réception, Des lignes de commande.



Pour la suite, il est intéressant de remarquer qu'en général les bits du registre d'état sont remis à 0 par :

- Une lecture du registre de réception.
- Une écriture dans le registre de transmission.

Ce procédé comporte un avantage car il accélère le transfert : les bits SR0 et SR1 sont automatiquement "préparés" pour le transfert du caractère suivant.

Pour les autres bits, la remise à 0 s'effectue de façon indirecte car on ne peut pas écrire dans le registre d'état. Dans un programme assembleur apparaissent parfois des instructions dites factices du type

**LDA RGRX** ou **STA RGTX**.

Les valeurs lues ou écrites ne sont pas utiles, on cherche tout simplement à agir sur les bits du registre d'état.

Pour une lecture factice, si l'on ne veut pas affecter la valeur dans les accumulateurs, on peut employer **TST RGRX** dans le mode étendu ou indexé pour obtenir le même effet.

Cette instruction affecte néanmoins le registre d'état CC du 6809.



### **6850 : SR3 : CTS (Clear To Send) Niveau Bas pour transmettre**

Ce bit indique l'état de la broche CTS| en provenance du modem.  
Inhibition de l'émetteur. Si CTS| est à 1, la transmission n'est pas possible, on force le bit SR1 (TDRE) à 0.

Le registre TDR (registre de transmission de données) est toujours considéré comme étant plein.

**SR3 = 0** Broche d'entrée CTS| à l'état Haut, le modem est prêt à émettre  
**SR3 = 1** Broche d'entrée CTS| à l'état Bas, le registre de transmission est inhibé, le bit SR1 (TDRE) mis à 0 et le bit SR7 (IRQ) mis à 0.

L'application d'un niveau haut sur la broche CTS| entraîne l'inhibition des bits SR1 et SR7 (SR1 = 0 et SR7 = 0).

Ce signal s'emploie dans le mode modem. Il indique que l'organe récepteur désire émettre et suspend l'activité du transmetteur.

La broche d'entrée CTS| n'a pas d'effet sur un caractère en cours de transmission ou placé dans le registre de transmission. Il y a seulement non initialisation de l'émission.

La broche d'entrée CTS| doit être mise à zéro en cas d'inutilisation.

Le bit CTS| n'est pas affecté par MASTER RESET.

### **6850 : SR4 : FE (Framing Error) Erreur de format**

Ce bit est positionné durant le transfert de données reçues et le restera positionné tant que le défaut subsiste. Ce bit b4 est remis à zéro par un MASTER RESET ou un niveau Haut sur la broche DCD|.

**SR4 = 0** Format correct, indique qui n'y a pas d'erreur de format.  
Les deux organes de dialogue (Emetteur et Récepteur) sont réglés sur un format identique.  
**SR4 = 1** Indique une erreur de format, qui peut provenir d'une erreur (absence de premier bit STOP) ou perte de synchronisation ou d'une réception défectueuse ou de la réception d'un BREAK.

### **6850 : SR5 : OVRN (OVer RuN) Débordement**

Indique que certains caractères reçus n'ont pas été lus par le 6809 ou un ou plusieurs caractères ont été perdus.

Ceci se produit lorsqu'un ou plusieurs caractères ont été reçus avant la lecture du caractère précédent dans le registre de réception.

**SR5 = 0** Réception correcte sans surcharge, aucun caractère n'a été perdu.  
**SR5 = 1** Indique une surcharge en réception. L'évacuation des données reçues est trop lente. Après la suppression de la surcharge, ce bit SR5 peut être remis à 0 par une seconde lecture factice du registre de réception.

Cependant la mise à 1 n'intervient que lorsque la lecture du caractère précédent la surcharge a eu lieu et elle se produit à partir du milieu du dernier bit du deuxième caractère reçu sans lecture du registre de réception.

Le bit SR0 (RDRF) reste à 1 jusqu'à ce que la condition de surcharge soit supprimée.  
Un niveau haut sur la broche DCD| ou un MASTER RESET entraîne le SR5 (OVRN) à zéro.

Le bit SR5 peut également être réinitialisé par une seconde lecture du registre RDR (registre de réception de données) Lors d'une nouvelle lecture le bit SR0 = 0 (RDRF) et le bit SR5 = 0 (OVRN)

### **6850 : SR6 : PE (Parity Error) Erreur de parité**

Ce bit sera positionné si l'on a programmé un contrôle parité par bits CR4 CR3 CR2, alors le bit SR6 sera activé lors du transfert interne.

**SR6 = 0** Parité correcte, indique qui n'y a pas erreur de parité.

**SR6 = 1** Indique une erreur de parité en réception.

Ce bit est actif seulement si le récepteur est configuré avec une détection de parité (voir les bits SR4, SR3, SR2).

Un niveau Haut sur la broche DCD| ou un MASTER RESET met ce bit SR6 à 0.

### **6850 : SR7 : IRQ (Interrupt Request) Demande d'interruption**

Ce bit passe à 1 entraînant le passage à 0 de la sortie IRQ| du 6850.

**SR7 = 0** Absence d'interruption, il n'a pas eu d'interruption.

**SR7 = 1** Demande d'interruption, une interruption est positionnée.

Ce bit SR7 Indique la présence d'une demande d'interruption qui provient (3 sources sont possibles) :

1°) Du récepteur (donc en réception), si le "mode par interruption" du récepteur est autorisé le bit CR7 = 1, le bit SR7 reproduit l'état du bit de réception SR0. (RDRF)

2°) Du transmetteur (donc en émission), si les bits CR6 CR5 = %01 alors les interruptions du transmetteur sont autorisées, le bit SR7 reproduit l'état du bit SR1. (TDRE)

3°) D'une perte de porteuse, lorsque le bit CR7 = 1, les interruptions du récepteur sont autorisées. Le bit SR7 passe également à 1, lorsque la broche DCD| est forcée à l'état Bas par une condition externe (broche d'entrée DCD| = 1).

Lorsque la broche DCD| (109) passe à un niveau haut  alors il y a génération d'une interruption (comme pour le bit SR2).

Le bit SR7 est remis à 0 par une lecture du registre de réception (interruption provenant du récepteur) ou par une écriture dans le registre de transmission (interruption provenant du transmetteur). Dans tous les cas, la broche DCD| doit être forcée à l'état Haut au préalable (entrée DCD| = 0).

### **6850 : Transmission**

Registre de transmission de donnée reçoit du 6809, par l'intermédiaire du bus de donnée, le mot à transmettre qui sera transféré dans le registre TSR (Registre à décalage de transmission) pour être ensuite sérialisé.

La transmission d'un caractère doit être précédée de la lecture du registre d'état SR, afin de connaître l'état du bit SR1 (TDRE). On pratique par interruption ou par boucle d'attente (polling).

Si le bit SR1 (TDRE) est à 1, le caractère à transmettre est chargé dans le registre TDR (registre de transmission de données) sur une commande d'écriture (front descendant de  $\Phi$ 2), le bit SR1 (TDRE) passe alors à 0 indiquant le que le registre TDR n'est pas libre.

Ensuite les données sont transférées du registre TDR dans le registre TSR (Registre à décalage de transmission) pendant une absence de transmission avec une durée correspondant à un bit série.

Le fond descendant du signal de transfert remet le bit SR1 (TDRE) à 1, ainsi un autre caractère peut être immédiatement chargé dans le registre TDR.

Le registre à décalage de sortie TSR, transmet sur la broche TxData le caractère en synchronisme avec une horloge interne dont la fréquence peut être 1/1, 1/16 ou 1/64 de la fréquence d'horloge appliquée sur la broche TxClk.

Le caractère est transmis bit par bit en commençant par D0 précédé d'un bit START, D7 étant suivi éventuellement par le bit de parité puis par un ou deux bits de STOP suivant la programmation du registre de contrôle.

Les bits sont transmis sur la transition négative de l'horloge interne.

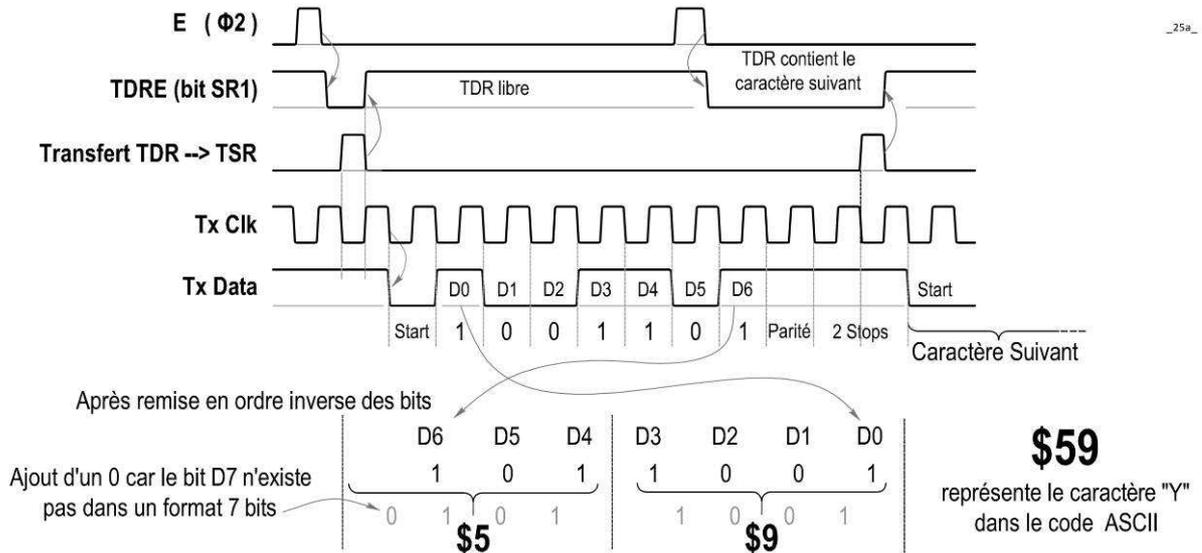
Pendant une opération de transmission la longueur d'un caractère et le nombre de bits d'arrêt peuvent être modifiés sans affecter le caractère en cours de transmission (sauf si la transition à lieu pendant le temps de transfert interne). La sélection de parité affecte immédiatement le caractère en cours de transmission.

Toute modification sur la partie transmission affecte également la partie réception. Si celle-ci est dommageable, on attendra une absence de réception.

### Exemple de chronogramme de transmission

On transmet la lettre "Y" codée en ASCII (59), donc sur 7 bits avec bit de parité paire et 2 bits de STOP.

On remarque que le deuxième caractère doit être inscrit dans le registre TDR avant le dernier bit de STOP du caractère précédent si mon il y a absence de transmission (IDLIND TIME).



### 6850 : Réception

Le registre de réception de donnée RDR reçoit le mot dé-sérialisé en provenance du registre à décalage de réception RSR.

Dans les systèmes de transmission asynchrone, les données sont transmises sans signal de synchronisation, ce sont les bits START et STOP qui vont permettre une synchronisation des bits du caractère aussi par rapport à la broche d'entrée RxClk (horloge de réception).

Cette broche RxClk synchronise une horloge interne dont le facteur de division 1/16 ou 1/64 est choisi par la programmation du registre de contrôle CR.

Pour le rapport 1/1 la synchronisation doit être externe.

### 6850 : Rapports 1/16 et 1/64

La synchronisation est assurée par la première transition négative (front descendant) suivant une période de repos.

Le bit de départ START est échantillonné durant les transitions positives (front montant) de l'horloge externe RxClk.

Si la broche d'entrée RxData est au niveau Bas pour 9 échantillons dans le mode 1/16 ou pour 33 échantillons dans le mode 1/64, ce qui représente plus de 50% de la durée d'un bit, le bit reçu et considéré comme bit START.

Ce bit START est rangé dans le registre à décalage RSR sur le front descendant de l'horloge interne.

Une fois que le bit de départ est détecté, la synchronisation de bit et de caractère est faite, les autres bits suivant le bit START sont décalés dans le registre RSR à peu près au milieu de leur durée.

Si la broche d'entrée RxData retourne à l'état Haut pendant la période d'échantillonnage du bit de départ, ce faux bit de départ est ignoré et le récepteur se place en attente d'un bit de départ correct.

## 6850 : Rapport 1/1

Dans ce mode, il n'y a pas de synchronisation de bit à l'intérieur du récepteur, donc l'horloge externe doit être synchronisée par la donnée.

Dès l'apparition de la première transition négative (front descendant), après une période de repos du signal reçu, l'échantillonnage se produit sur le front montant de l'horloge externe la broche RxClk et le bit START est chargé sur le front descendant suivant de l'horloge.

Pour améliorer la sécurité de détection, le front positif de l'horloge externe doit intervenir dans le milieu d'un bit.

## 6850 : Fonctionnement général du récepteur

La validité du caractère reçu est contrôlée pendant la réception et positionne les bits concernés du registre d'état. Il y a un test de la parité, du format et du débordement.

La réception complète du caractère provoque la mise en 1 du bit SR0 (RDRF).

Le caractère est transféré dans le registre RDR (registre de réception de données) après suppression des bits de départ, de parité, de STOP, c'est à ce moment que les indicateurs d'état sont positionnés.

Le premier bit série reçu correspondra sur le bus de données à D0.

Il est possible de lire un caractère dans le registre RDR (registre de réception de données) pendant qu'un autre caractère est mémorisé dans le registre RSR (registre à décalage en réception).

Si pendant la réception il y a absence du premier bit STOP, une erreur de format est signalée par le positionnement à 1 du bit SR4 (FE), sans qu'il y ait perte de synchronisation de caractère.

## 6850 : Programmation Routine d'initialisation

L'ACIA doit être initialisé avant de transmettre ou de recevoir des données.

Le premier état d'initialisation, qui se produit automatiquement à la mise sous tension doit être supprimé par un MASTER RESET. Ceci se produit en mettant CR1 CR0 à %11.

Puis l'on programme le registre de contrôle pour utilisation désirée.

On peut à tout moment, configurer à nouveau l'interface pour obtenir un autre protocole de dialogue :

```
                ; Prog 6850-01
                ;-----
EC00 RGCR EQU $EC00 ; Adrs reg.Contrôle
EC00 RGSR EQU $EC00 ; Adrs reg.Etat, on peut écrire RGSR EQU RGCR
EC01 RGTX EQU $EC01 ; Adrs reg.Transmission
EC01 RGRX EQU $EC01 ; Adrs reg.Réception

0000 86 03      Debut LDA #00000011 ; init programmée "Master Reset"
0002 B7 EC00      STA RGCR ;
0005 86 01      LDA #00000001 ; config pour un cas particulier
0007 B7 EC00      STA RGCR ;
```

Après avoir fixé le protocole de transfert, on peut effectuer une transmission ou une réception de donnée, ou les deux simultanément. En inhibant les "modes par interruptions" :

## 6850 : Programme d'initialisation pour une émission

La séquence débute par un test du bit SR1 :

Si SR1 = 1 le registre de transmission est vide, le 6809 peut alors placer la donnée à transmettre dans le registre de transmission RGTX

Si SR1 = 0 la donnée précédente est en cours de transmission, il faut attendre.

```

0000 34 02          PSHS  A          ; sauvarde de la donnée à transmettre
0002 96 10          TXDATA LDA  RGSR   ; lecture du reg.Etat
0004 85 02          BITA   #%0000010 ; test bit SR1
0006 27 FA          BEQ   TXDATA   ; Attendre que le reg.TX est vide
0008 35 02          PULS  A          ; récup donnée à transmettre
000A B7 2000        STA   RGTX     ; transmettre

```

Si le registre B est disponible le programme ci-dessus devient (en enlevant PSHS et PULS)

```

0000 D6 10          TXDATA LDB  RGSR   ; lecture reg.Etat
0002 54            LSRB           ; glisser le bit SR1 dans la
0003 54            LSRB           ; retenue (bit C) du reg CC du 6809
0004 24 FA          BCC   TXDATA   ; si bit SR1=0 alors attendre en bouclant
0006 B7 2000        STA   RGTX     ; transmettre donnée

```

## 6850 : Programme d'initialisation pour une réception

Il faut tester la validité d'une donnée reçue en veillant à l'absence d'erreur, en testant ces bits :

- SR4 pour les erreurs de format
- SR5 pour les erreurs de surcharge
- SR6 pour les erreurs de parité

Pour l'instant supposons que l'Emetteur et le Récepteur soient réglés sur un même protocole, dans ce cas la séquence de réception débute par un test de la présence d'une donnée dans le registre de réception RGRX à l'aide du bit SR0 du registre d'état.

```

0000 96 10          RXDATA LDA  RGSR   ; lecture reg.Etat
0002 44            LSRA           ; mettre le bit SR0 dans la retenu bit C
0003 24 FB          BCC   RXDATA   ; si SR0=0 attendre en bouclant
0005 B6 2000        LDA   RGRX     ; lecture du reg.Réception

```

## 6850 : Programmation Routine de transmission

### 6850 : 1er exemple de transmission

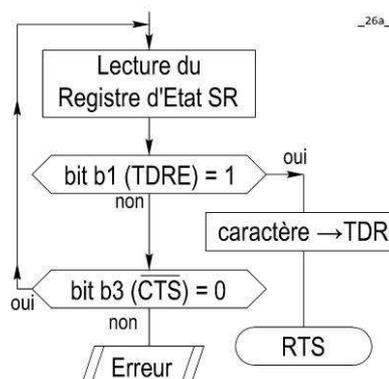
Dans un premier temps, on teste que le registre de transmission est vide, si non il faut vérifier, avant de tester de nouveau le bit b1 (TDRE), que le bit b3 (CTS) n'est pas à 1 ce qui inhiberait le bit b1 (TDRE).

Dans le cas d'une liaison avec un modem, cela signifierait une perte de porteuse.

```

0000 96 10          DEBUT LDA  ACIASR  ;registre d'état -->Acc.A
0002 47            ASRA           ;décal droite bits, mise dans flag C
0003 47            ASRA           ;décal droite bits, mise dans flag C
0004 25 07          BCS   TRANS     ;test du bit b1 TDRE
0006 47            ASRA           ;décal droite bits, mise dans flag C
0007 47            ASRA           ;décal droite bits, mise dans flag C
0008 24 F6          BCC   DEBUT     ;test du bit b3 CTS
000A 0E 11          JMP   ERROR     ;
000D F7 2000        TRANS STB  ACIATR  ;caractère --> registre TDR
0010 39            RTS            ;

```



## 6850 : 2ième exemple de transmission

On veut envoyer une donnée stockée à l'adresse \$1000 vers un télétype.

On ne travaille pas dans ce cas en interruption, il faut tester le contenu du registre de transmission afin de s'assurer qu'un caractère peut être envoyé.

### 1<sup>ère</sup> Solution Liv6809ModeInterface (test sur la transmission)

```
0000 86 03          LDA  #$03      ; %0000 0011 Master Reset b1 b0 = %11
0002 B7 1000       STA  ACIACR   ;
0005 86 45        LDA  #$45      ; %0100 0101 initialisation du CR
0007 B7 1000       STA  ACIACR   ;
000A 86 02        LDA  #%00000010 ;
000C B5 1020      ATENT BITA  ACIASR ;test sur bit b1 TDRE
000F 27 FB        BEQ  ATENT    ;TDR est plein, on recommence le test
0011 B7 1030      STA  ACIATD   ;TDR est vide on charge le contenu
                                   ; de la mémoire $1000 dans TDR
```

### 2<sup>ème</sup> Solution Liv6809et9365-66 (test sur la réception)

```
0000 86 03          LDA  #$03      ; %0000 0011 Master Reset b1 b0 = %11
0002 B7 1000       STA  ACIACR   ;
0005 86 45        LDA  #$45      ; %0100 0101 initialisation du CR
0007 B7 1000       STA  ACIACR   ;
000A B6 1020      GISSE LDA  ACIASR ; lecture du registre d'état
000D 44           LSRA          ; test sur SR0 RDR est plein
000E 24 FA        BCC  GISSE     ; RDR est vide SR0=0, on recommence le test
0010 B6 1030      LDA  ACIARD    ; le contenu du registre de réception est
0013 B7 2000      STA  $2000     ; transféré à l'adresse $1000
```

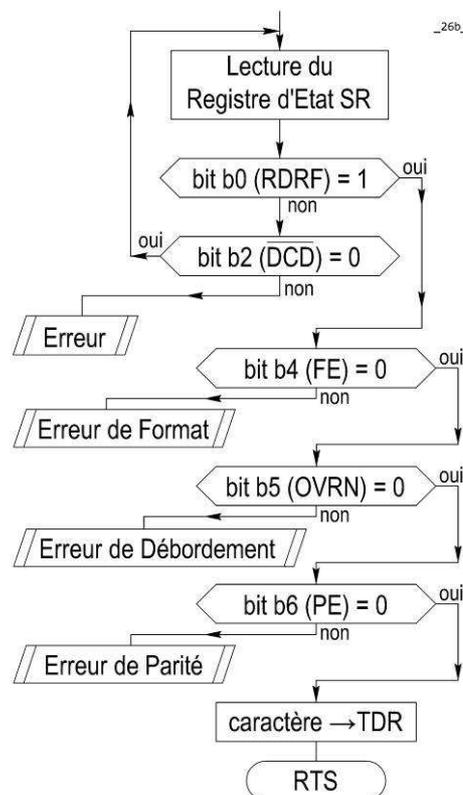
Dans le cas d'une transmission, il n'y a pas bien sûr de test sur les bits PE, FE et OVRN.  
La transmission se fait automatiquement à partir du chargement de TDR.

## 6850 : Exemples de programmation en réception

### 6850 : 1er exemple de Réception

```
0000 B6 1020      DEBUT LDA ACIASR      ; registre d'état --> Acc.A
0003 47          ASRA                ; décal droite bits, mise dans flag C
0004 25 06      BCS TESTFE          ; test du bit b0 RDRF
0006 47          ASRA                ; décal droite bits, mise dans flag C
0007 47          ASRA                ; décal droite bits, mise dans flag C
0008 24 F6      BCC DEBUT           ; test du bit b2 DCD
000A 20 14      BRA ERROR           ;

000C 47          TESTFE ASRA         ; décal droite bits, mise dans flag C
000D 47          ASRA                ; décal droite bits, mise dans flag C
000E 24 02      BCC TESTOV          ; test du bit b4 FE
0010 20 10      BRA ERFORM          ;
0012 47          TESTOV ASRA         ;
0013 24 02      BCC TESTPE          ; test du bit b5 OVRN
0015 20 09      BRA ERROR           ;
;
0017 47          TESTPE ASRA         ;
0018 24 02      BCC LECTUR          ; test du bit b6 PE
001A 20 08      BRA ERPAR           ; branchement vers SPgm Err Parité
001C B6 1030    LECTUR LDA ACIADR    ; lecture du caractère
001F 39          RTS                ;
```



### 6850 : 2ième exemple de Réception

On veut recevoir d'un télétape une donnée et la stocker en mémoire à l'adresse \$1000.

Il faut commencer par initialiser le 6850.

Après une initialisation logicielle (MASTER RESET), on détermine le contenu du registre de contrôle CR en fonction des caractéristiques de la ligne 7 ou 8 bits, la parité, le nombre de bits de stop, etc....

La fréquence RxClk est de 1760 Hz, il faut donc utiliser le diviseur par 16 pour obtenir la vitesse de 110 bauds (b1 b0 du registre CR = à %01).

Le télétape transmet des mots de 7bits, parité impaire et 2 bits de stop (b4 b3 b2 du registre CR = à %001).

La ligne de transmission est inactive, les interruptions du transmetteur sont inhibées (b6 b5 du registre CR = à %10).

Les interruptions du récepteur sont inhibées (b7 du registre CR = à %0).

Le programme d'initialisation du 6850 est donc le suivant :

```
0000 86 03          LDA    #%00000011 ; Master Reset
0002 B7 1000       STA    ACIACR    ;
0005 86 45        LDA    #%01000101 ; initialisation du CR
0007 B7 1000       STA    ACIACR    ;
```

Ce programme ne travaille pas en interruption, il faut tester le registre de réception pour savoir si une donnée a été envoyée.

Ce test se fait sur le bit b0 du registre d'état SR.

Le 6809 reste en attente tant que le registre de réception est vide.

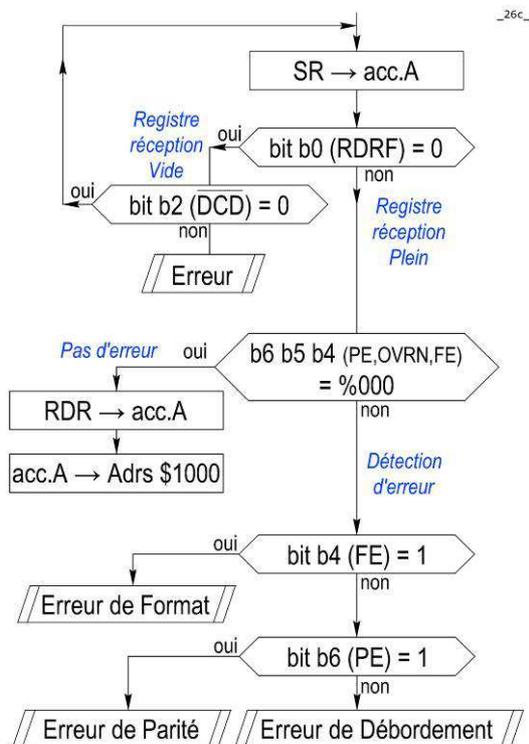
```
0000 B6 1020       GISSE LDA    ACIASR    ; test registre réception b0 SR = 1
                                ; alors RDR plein
0003 44           LSRA           ; décalage 1 bit vers la droite,
                                ; mise dans le flag C
0004 24 FA        BCC    GISSE    ; lag C = 0 alors RDR est vide,
                                ; on recommence le test
0006 B6 1030       LDA    ACIARD    ; contenu registre de réception
                                ; transféré à l'adresse $1000
0009 B7 1000       STA    $1000    ;
```

On suppose ici que le message reçu ne présente pas d'erreur, dans certains cas il est nécessaire de faire des tests sur le registre d'état.

L'organigramme d'un tel programme est le suivant :

#### Le programme est alors plus complexe.

```
0000 B6 1020       GISSE LDA    ACIASR    ;test registre réception
                                ; SR0=1 alors RDR plein
0003 44           LSRA           ;décal d'un bit --> droite
                                ; et mise dans flag C
0004 24 0A        BCC    LAINE    ;branch routine LAINE si
                                ; bit b0 = zéro (RDR vide)
0006 85 70        BITA    #$70    ;test bits b6 b5 b4 PE,OVRN,FE
0008 26 0C        BNE    SPERR    ;branch à SPER un
                                ; des 3 bits est à 1
000A B6 1030       LDA    ACIARD    ;le contenu du registre de
                                ; réception est transféré
                                ; à l'adresse $1000
000D B7 1000       STA    $1000    ;
0010 85 04        LAINE BITA    #4    ;Test sur le bit b2 DCD |
0012 27 EC        BEQ    GISSE    ;DCD|=1, recommence le test
0014 20 0A        BRA    SPDCD    ;DCD|=0, branche au SPgm SPDCD
                                ;
0016 85 10        SPERR BITA    #$10 ;test sur l'erreur de format
0018 26 09        BNE    SPFE    ;branche au SPgm SPFE
001A 85 40        BITA    #$40    ;test sur l'erreur de parité
001C 26 08        BNE    SPPE    ;branche au SPgm SPPE
001E 20 09        BRA    SPOVRN   ;branche au SPgm SPOVRN
```



## 6850 : Exemple de Transmission des caractères Clavier vers Imprimante

La plupart des imprimantes ayant une liaison série n'ont pas tous les signaux de la norme RS-232-C. L'imprimant peut être :

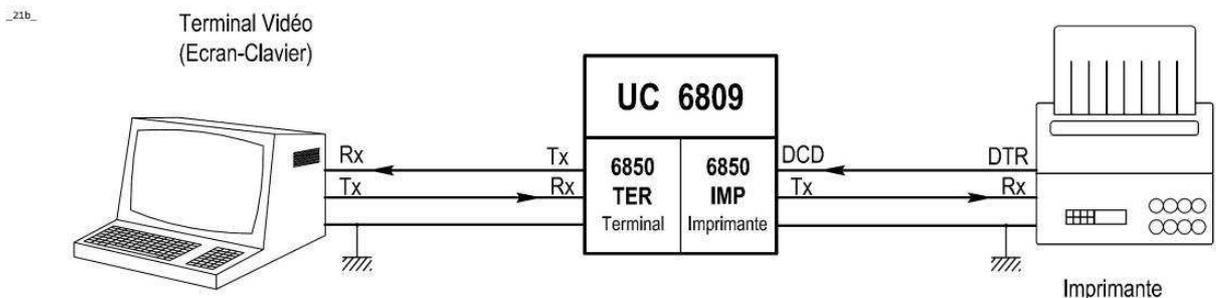
- Soit à réception seule.
- Soit d'un type plus complet possédant un clavier intégré. Dans ce cas les programmes en mode FULL-DUPLEX sont plus complexes.

Dans cet exemple nous utilisons l'imprimante comme une machine à écrire. L'opérateur envoie des données ASCII à partir d'un clavier vers une liaison série.

Après avoir analysé le caractère reçu du clavier, le 6809 renvoie le caractère, éventuellement modifié ou codé différemment, vers l'imprimante.

Deux interfaces série :  
 S0 fonctionnera en Transmission  
 S1 fonctionnera en Réception

Comme la vitesse de l'opérateur est très inférieure à celle réglée habituellement pour les lignes série, les protocoles de dialogues sont réduits à leur plus simple expression, c'est-à-dire au protocole START-STOP.



## Du point de vue électrique :

- Les lignes "masse signal" et "masse châssis" seront normalement connectées.
- Une seule ligne de réception est nécessaire pour S1 (Réception provenant du terminal Ecran-Clavier).
- Une seule ligne de transmission est nécessaire pour S0 (Emission vers l'imprimante).
- Les broches DCD] doivent être maintenues constamment à l'état haut +12 volts. C'est-à-dire les entrées DCD] =0 pour ne pas inhiber les récepteurs. Cette condition doit être vérifiée s'il y a un mauvais fonctionnement.

Le programme de gestion ci-dessous s'occupe également de l'envoi d'un caractère à l'écran.

```
8000                ORG    $8000    ; adrs de chargement
                   ;-----Adresses des registres ACIA IMP (imprimante)
EC00 RCRIMP EQU    $EC00    ; reg CR Contrôle IMP
EC00 RSRIMP EQU    $EC00    ; reg.SR Etat IMP
EC01 RTXIMP EQU    $EC01    ; reg.TX Transmission IMP
                   ;
                   ;-----Adresses des registres ACIA TER (Terminal)
EC80 RSRTER EQU    $EC80    ; reg. SR Etat TER
EC81 RRXTER EQU    $EC81    ; reg. RX Réception clavier TER
EC81 RTXTER EQU    $EC81    ; reg. TX Transmission TER
                   ;
                   ;-----Configuration de l'imprimante
8000 C6  03                LDB    #%00000011 ; Init Programmé
8002 F7  EC00              STB    RCRIMP    ; Registre contrôle imprimante
8005 C6  01                LDB    #%00000001 ; 7bits, Paire, 2 Stop, 1/16
8007 F7  EC00              STB    RCRIMP    ;
                   ;
                   ;-----Attendre un caractère du clavier
800A F6  EC80              ATCLAV LDB    RSRTER    ; registre Etat Terminal
800D 54                LSRB                ; examen bit réception
800E 24  FA    800A        BCC    ATCLAV    ; bit SR0=0 alors boucle d'attente
8010 B6  EC81              LDA    RRXTER    ; donnée
8013 81  03                CMPA   #$03     ; ETX ou CTRL C ?
8015 27  1C    8033        BEQ    FIN      ; si oui alors FIN
                   ;
                   ;-----Envoi caractère vers imprimante 6850 IMP
8017 F6  EC00              TXIMP  LDB    RSRIMP    ; registre Etat Transmission
801A C5  02                BITB   #%00000010 ; examen bit transmission
801C 27  F9    8017        BEQ    TXIMP    ; bit SR1 = 0 alors bouclage
                   ;                               d'attente
801E B7  EC01              STA    RTXIMP    ; caract à transm dans reg.trans
                   ;
                   ;-----Cette partie est optionnelle, utile pour clavier
                   ;-----autonome fonctionnant sans écho sur l'écran
8021 81  20                CMPA   #$20     ; caractère visualisable
8023 24  02    8027        BCC    *+4     ;
8025 86  2E                LDA    #$2E     ; Mettre un point
                   ;
                   ;-----Transmette caractère sur écran
8027 F6  EC80              TXTER  LDB    RSRTER    ; registre TER terminal
802A C5  02                BITB   #%00000010 ;
802C 27  F9    8027        BEQ    TXTER    ;
802E B7  EC81              STA    RTXTER    ; transmettre
8031 20  D7    800A        BRA    ATCLAV    ; Attendre un autre caractère
8033 3F                FIN    SWI                ;
```

Dans ce programme le branchement côté **TER**minial n'est pas initialisé, ni configuré car il doit l'être déjà la mise sous tension du système.

S'il n'en est pas ainsi, aucune information ne pourrait être reçue à partir du clavier.

Cependant, si le besoin s'en fait sentir, ce branchement peut être configuré à nouveau exactement comme pour le branchement côté IMPrimante (voir croquis ci-dessus).

La réception d'un caractère du clavier s'effectue sans vérification d'erreur à la réception.





```

803F 5A          DECB          ;
8040 26 FB      803D      BNE      *-3      ;
8042 A6 61          LDA      1,S      ; Code à transmettre
8044 8D 49      808F      BSR      BINHEX    ; conversion Binaire-Hexa
8046 ED 81          STD      0,X++     ; stockage 2 car. Convertis
8048 CC 2020     LDD      #$2020   ; 2 espaces entre les colonnes
804B ED 81          STD      0,X++     ;
804D C6 10      LDB      #16      ; Nbre Car. A visualiser
804F A6 61      LDA      1,S      ; code à visualiser
8051 A7 80      STA      0,X+    ;
8053 5A          DECB          ;
8054 26 FB      8051      BNE      *-3      ;
8056 C6 06      LDB      #6        ; 6 espaces entre les deux tableau
8058 86 20      LDA      #$20     ; code car. Espace
805A A7 80      STA      0,X+    ;
805C 5A          DECB          ;
805D 26 FB      805A      BNE      *-3      ;
;-----la deuxième colonne commence à partir
;-----du code $50 (code première colonne + $30)
805F A6 61      LDA      1,S      ; code deuxième colonne
8061 8B 30      ADDA     #$30     ;
8063 8D 2A      808F      BSR      BINHEX    ; conversion Binaire-Hexa
8065 ED 81          STD      0,X++     ;
8067 CC 2020     LDD      #$2020   ; 2 car espace
806A ED 81          STD      0,X++     ;
806C C6 10      LDB      #16      ;
806E A6 61      LDA      1,S      ; code lière colonne
8070 8B 30      ADDA     #$30     ; code à visualiser
8072 A7 80      STA      0,X+    ;
8074 5A          DECB          ;
8075 26 FB      8072      BNE      *-3      ;
;-----Mettre en fin de ligne "CR" return et "LF" line feed
8077 86 0D      LDA      #$0D     ; Car Return
8079 A7 80      STA      0,X+    ;
807B 86 0A      LDA      #$0A     ; Car Line Feed
807D A7 80      STA      0,X+    ;
;-----Test de fin de programme (48 lignes)
807F A6 61      LDA      1,S      ; Actualiser code pour lgn suivante
8081 4C          INCA          ;
8082 A7 61      STA      1,S      ;
8084 E6 E4      LDB      0,S      ; Actualiser Nbre de lgn restant
8086 5A          DECB          ;
8087 E7 E4      STB      0,S      ;
8089 26 AE      8039      BNE      LNSVT    ; écrire données ligne suivante
808B 32 62      LEAS     2,S      ;
808D 35 97      PULS     PC,X,B,A,CC ; fin du S/P GRAPH
;*****
; S/P conversion d'un octet binaire en 2 car. ASCII en Hexa
; Nom d'appel : BINHEX
; Entrée : A : Donnée à convertir
; Sortie : A : Code ASCII hexa 4 bits poids fort
;          B : Code ASCII hexa 4 bits poids faible
;          (ex : $3A sera converti en $33 $41
;*****
808F 1F 89      BINHEX   TFR      A,B      ;
8091 44          LSRA          ; ----obtenir 4 bits poids fort
8092 44          LSRA          ; -- |
8093 44          LSRA          ; -- |
8094 44          LSRA          ; -- |
8095 81 0A      CMPA     #$A      ; < 10 ?
8097 25 02      809B      BLO      *+4      ;
8099 8B 07      ADDA     #7        ;
809B 8B 30      ADDA     #$30     ;
809D CB 0F      ADDB     #$0F     ; 4 bits poids faible
809F C1 0A      CMPB     #$A      ; < 10 ?
80A1 25 02      80A5      BLO      *+4      ;
80A3 CB 07      ADDB     #7        ;
80A5 CB 30      ADDB     #$30     ;
80A7 39          RTS          ; fin du S/P BINHEX

```

Le programme comporte 2 parties :

- Gestion des transferts de données suivant le protocole DTR.
- Sous-programme de création de la base de données.

Dans le corps principal du programme de gestion, on observe un test systématique du bit SR2 du registre d'état pour détecter un éventuel changement d'état de la broche DCD] (transition de +12 volts → -12 volts).

Lorsque le bit SR1 = 1 la séquence aux adresse \$8018 et \$801B :

```
LDB   RSRIMP   ; lecture reg.Etat
LDB   RRXIMP   ; lecture reg.Réception
```

Effectue une remise à 0 "potentielle" du bit DCD. Ce bit sera effectivement remis à 0 lorsque la broche DCD] change d'état (transition de -12 volts → +12 volts) indiquant que le buffer de réception est à nouveau prêt pour recevoir d'autres données.

Durant la période d'attente, on peut orienter le 6809 vers le traitement d'une autre tâche, c'est ce qui est effectivement réalisé dans les systèmes multitâches.

#### Les quatre lignes du programme précédent

8031	C6	30	LDB	#\$30	; Nbre de ligne du texte ;
8033	E7	E4	STB	0,S	; ;
8035	86	20	LDA	#\$20	; code départ ;
8037	A7	61	STA	1,S	; ;

Peuvent être Optimisé par les deux lignes suivantes

```
LDD   #$3020   ; Nbre de ligne du texte
STD   0,S       ; + code départ
```

### 6850 : Exemple de Transmission des caractères vers Imprimante, Protocole ETX-ACK

Dans le protocole ETX-ACK (End of text, Acknowledge), les données sont envoyées par blocs.

La taille du bloc transmis doit-être en tout cas inférieure à celle du buffer de réception de l'imprimante.

Le dernier caractère d'un bloc doit-être le code ETX \$03.

De son côté, l'imprimante exploite les caractères reçus à sa propre vitesse.

A la détection du code ETX \$03, le récepteur renvoie au transmetteur le code à ACK \$06 lui indiquant sa disponibilité.

Pour tester cette application, seule les broches de transmission Tx et de réception Rx doivent être connectées électriquement.

L'entrée DCD] du système processeur-interface doit être maintenue à 0 (ligne DCD au niveau Haut +12 volts).

Pour la majorité des imprimantes, cette ligne peut être connectée à la broche DTR].

Ici, on surveille l'état du bit de réception SR0 dans le registre d'état, qui atteste la présence d'une donnée en retour.

Rappelons que dans le protocole DTR, la poignée de main est basée sur l'état du bit SR2 qui traduit l'état de l'entrée DCD].

```

;*****
;   Transmission de blocs de données vers une imprimante
;   suivant le protocole ETX/ACK
;*****
8000          ORG      $8000          ; adrs de chargement
;-----Adresses des registres ACIA IMP (imprimante)
EC00 RCRIMP EQU  $EC00          ; reg CR Contrôle IMP
EC00 RSRIMP EQU  $EC00          ; reg Etat IMP
EC01 RTXIMP EQU  $EC01          ; reg Transmission IMP
EC01 RRXIMP EQU  $EC01          ; reg Réception IMP

;-----Configuration de l'imprimante
8000 C6  03          LDB  #%00000011 ; init programmée Master Reset
8002 F7  EC00        STB  RCRIMP     ; reg.Contrôle IMP
8005 C6  01          LDB  #%00000001 ; 7bits, Paire, 2 Stop, 1/16
8007 F7  EC00        STB  RCRIMP     ;

```

```

;-----transmission. Les données à transmettre se
;-----trouve à partir de l'adresse $8100
800A 8E 8100          LDX  #$8100          ;
800D C6 06          LDB  #6            ; Nb. Blocs à transmettre
800F 8D 01 8012     BSR  TXIMP         ; Transmission
8011 3F             SWI                ;

;*****
; S/P de transmission de blocs de données
; suivant le protocole ETX/ACK  Nom d'appel : TXIMP
; Entrée : X: adresse basse de données
;         B: Nombre de blocs de 512 octets
; Sortie : aucun registre modifié
; Extention pile système: 12= $0C octets
;*****
8012 34 37          TXIMP  PSHS  Y,X,B,A,CC      ;
8014 32 7F          LEAS  -1,S          ; réserve pour compteur blocs
8016 E7 E4          STB   0,S          ; compteur blocs
8018 B6 EC01        RAZSR0 LDA  RRXIMP         ; remise à 0 du but SR0, tester la
; disponibilité de l'imprimante
801B 86 03          LDA  #$03          ; caractère ETX
801D 8D 25 8044     BSR  CARIMP         ; transmettre un caractère
801F B6 EC00        BLCSVT LDA  RSRIMP         ; reg.Etat, BLoCs SuiVanT
8022 44            LSRA                ; examen bit SR0
8023 24 FA 801F     BCC  BLCSVT         ; si SR0=0 bouclage BLoCs SuiVanT
8025 B6 EC01        LDA  RRXIMP         ; donnée reçue
8028 81 06          CMPA  #$06         ; caract ACK ?
802A 26 EC 8018     BNE  RAZSR0        ; sinon, nouvel essai,
; imprim prête, trnsmission
802C 108E 0200      LDY  #512          ; longueur d'un bloc en octets
8030 A6 80          CARSVT LDA  ,X+          ; code à transmettre
8032 8D 10 8044     BSR  CARIMP         ; transmettre un caractère
8034 31 3F          LEAY  -1,Y          ; Nb Car du bloc épuisé ?
8036 26 F8 8030     BNE  CARSVT         ; sinon, CARactère SuiVanT
8038 86 03          LDA  #$03          ; caractère ETX
803A 8D 08 8044     BSR  CARIMP         ; transmettre
803C 6A E4          DEC  0,S          ; compteur blocs épuisé ?
803E 26 DF 801F     BNE  BLCSVT         ; sinon, BLoCs SuiVanT
8040 32 61          LEAS  1,S          ;
8042 35 97          PULS  PC,X,B,A,CC    ; Fin du S/P TXIMP
;
;*****
; S/P transmission d'un caractère vers imprimante
; Nom d'appel : CARIMP
; Entrée : A: Code à transmettre
; Sortie : aucun registre modifié
; encombrement pile système : 3 octets
;*****
8044 34 02          CARIMP PSHS  A            ;
8046 B6 EC00        LDA  RSRIMP         ; reg.Etat
8049 85 02          BITA  #%00000010    ; examen bit transmission
804B 27 F9 8046     BEQ  *-5          ; si bit SR1=0, attendre
804D 35 02          PULS  A            ; code à transmettre
804F B7 EC01        STA  RRXIMP         ; registre transmission IMP
8052 39            RTS                ; Fin S/P CARIMP

```

Après les opérations habituelles de sauvegarde, le sous-programme TXIMP effectue une lecture factice du registre de réception, pour mettre à 0 le bit SR0, qui peut-être mis accidentellement à 1 par un autre programme.

Par précaution, le système teste d'abord la disponibilité du récepteur en envoyant le caractère ETX.

Le caractère renvoyé doit-être un à ACK. S'il n'en n'est pas ainsi, on peut s'orienter vers une séquence d'avertissement.

Ici le programme est simplifié, la boucle BNE RAZSR0 se referme indéfiniment si le caractère retourné n'est pas un à ACK.

Si le buffer de l'imprimante possède une taille supérieure à 512 octets, il faudra modifier la ligne  
802C 108E 0200 LDY #512 ;

La transmission se termine par l'envoi d'un caractère ETX à la fin de chaque bloc. Le test de fin de programme est basé sur le nombre total de blocs à transmettre.

## **6850 : Exemple de Transmission des caractères vers Imprimante, Protocole XON-XOFF**

Dans le protocole DTR, la disponibilité ou l'indisponibilité du buffer de réception sont traduites par l'activation d'une broche électrique DTR].

Dans le protocole XON-XOFF le transmetteur arrête le transfert à la réception du code XOFF ou DC3 (Device Control 3 : code ASCII \$13).

Lorsque le buffer de réception est prêt à recevoir, le code DC1 (code ASCII \$11) est retourné au transmetteur.

Comme dans le protocole ETX-ACK, le transfert met en œuvre une broche de transmission et une broche de réception pour le retour des codes de contrôle.

La différence est qu'ici la longueur du bloc de données est déterminée automatiquement par le récepteur et on n'a pas besoin de la fixer dans le programme assembleur comme pour le cas de ETX-ACK.

Le fonctionnement de XON-XOFF se rapproche plutôt du protocole DTR; seulement ici, le transmetteur doit reconnaître une donnée issue du récepteur au lieu de lire l'état électrique d'une broche.

```

;*****
;   Transmission de blocs de données vers une imprimante
;   suivant le protocole XON-XOFF ou DC1/DC3
;*****
8000          ORG      $8000          ; adrs de chargement
;-----Adresses des registres ACIA IMP (imprimante)
EC00 RCRIMP EQU  $EC00          ; reg CR Contrôle IMP
EC00 RSRIMP EQU  $EC00          ; reg Etat IMP
EC01 RTXIMP EQU  $EC01          ; reg Transmission IMP
EC01 RRXIMP EQU  $EC01          ; reg Réception IMP
;
;-----Configuration de l'imprimante
8000 C6  03          LDB  #00000011 ; init programmée Master Reset
8002 F7  EC00        STB  RCRIMP    ; reg.Contrôle IMP
8005 C6  01          LDB  #00000001 ; 7bits, Paire, 2 Stop, 1/16
8007 F7  EC00        STB  RCRIMP    ;
;
;-----Transmission. Les données à transmettre se trouvent
;-----à partir de l'adresse $8100
800A 8E  8100        LDX  #$8100    ;
800D 108E 0C00       LDY  #$C00     ; Nb Car à transmettre
8011 8D  01  8014    BSR  TXIMP     ; transmission
8013 3F              SWI             ;
;
;*****
;   S/P de transmission de donnée
;   Protocole XON / XOFF      Nom d'appel: TXIMP
;   Entrée : X: Adresse basse zone de données
;           Y: Nombre de car à transmettre
;   Sortie : Aucun registre modifié
;   Extension pile système : 9 octets
;*****
8014 34  37          TXIMP  PSHS  Y,X,B,A,CC ;
8016 B6  EC01        LDA  RRXIMP    ; mise à zéro bit réception
;
;-----tranqmission avec surveillance du retour de
;-----code XOFF (DC3 $13) avertissant buffer plein
8019 F6  EC00        CARSVT LDB  RSRIMP    ; reg.Etat CARactère SuiVanT
801C 56              RORB           ; examen bit réception
801D 25  0E  802D    BCS  STOPTD    ; si bit SR0=1, arrêter la trans.
801F 56              RORB           ; examen bit transmission
8020 24  F7  8019    BCC  CARSVT    ; si bit SR1=0, recommencer test
8022 A6  80          LDA  ,X+       ; code à transmettre
8024 B7  EC01        STA  RTXIMP    ; transmettre
8027 31  3F          LEAY -1,Y     ; Nb Caractères épuisé ?
8029 26  EE  8019    BNE  CARSVT    ; sinon, continuer
802B 35  B7          PULS  PC,Y,X,B,A,CC ; Fin S/P TXIMP
;

```

```

;-----détection d'un code XOFF
802D F6 EC01 STOPTD LDB RRXIMP ; Effacer bit réception
;
;-----Attente du code XON (Disponibilité)
8030 F6 EC00 ATXON LDB RSRIMP ; reg.Etat
8033 54 LSRB ; examen bit réception
8034 24 FA 8030 BCC ATXON ; si bit SR0=0, Attente XON
8036 F6 EC01 LDB RRXIMP ; lecture reg réception
8039 C1 11 CMPB #$11 ; XON ?
803B 27 DC 8019 BEQ CARSVT ; code correct, continer.
; Si le code reçu est incorrect
; -> possibilité d'écrire un S/P traitant
; cette erreur. Ceci n'est utile que pour
; une IMP munie d'un clavier Ici, tout
; code reçu est accepté
803D 20 DA 8019 BRA CARSVT ; instruct. à modifier éventuellement

```

Dans ce programme, lors de la rencontre d'un code en retour en cours de transmission, ce code est interprété automatiquement comme un XOFF et il n'y a pas de contrôle.

Ceci n'est nullement gênant si l'imprimante doit travailler en mode réception seule.

Pour les imprimantes dotées d'un clavier, on peut intervenir dans le système à partir du clavier.

Dans ce cas spécifique, chaque code retourné au transmetteur comporte une signification précise et une identification systématique s'impose.

Pour XON également, le contrôle s'avère inutile pour les mêmes raisons.

## **6850 : Exemple de Réception des données, avec diverses vérifications, Contrôle ligne RTS**

Dans les applications précédentes, la séquence de réception est présentée dans sa forme la plus simple, sans aucune vérification. Dans la majorité des systèmes pré-réglés, elle suffit amplement.

Pour les autres applications particulières, le 6850 possède de nombreuses options de contrôle réception :

### **Surcharge :**

La condition de surcharge se produit lorsque les données arrivent en flots et le programme assembleur se chargeant de la réception ne lit pas rapidement les données reçues pour la vitesse réglée.

### **Parité et Format :**

La parité est le format du mot reçu sont automatiquement échantillonnée par l'interface et toute erreur sera signalée par la mise à 1 des bits SR6 ou SR4.

Le bit de parité n'apparaît jamais dans le registre de réception, même avec un format 7 bits. Le contrôle est effectué au niveau "matériel" appelé aussi "mode câblé" pour simplifier la programmation au niveau utilisateur.

### **Etat de la ligne DCD :**

Elle traduit l'état de la broche DCD| dans un mode de transmission par modem (DCD Data Carrier Detect traduis par "Perte porteuse de données").

Il avertit le récepteur de l'absence de la porteuse modulée. Cette entrée est utile en transmission avec le protocole DTR. Elle sert de ligne de dialogue principale au mode "poignée de main câblée".

L'usage de cette broche d'entrée DCD et du bit associé SR2 est en réalité très générale.

Elle sert dans toutes circonstances où l'on a besoin d'avertir le 6809 d'un changement quelconque dans les conditions externes.

De son côté, le 6850 peut piloter un périphérique avec la broche RTS| en programmant les bit CR5 et CR6.

Dans le programme qui suit le 6850 joue le rôle de récepteur. Il supervise le transfert avec la broche de sortie RTS]. Lorsque RTS| = 0, le récepteur est prêt à recevoir.

Il arrêtera le transfert en avertissant l'émetteur avec RTS| = 1.

```

;*****
; réception de donnée avec ligne série
; vérifications diverses (format, parité, surcharge)
; contrôle du transmetteur par ligne RTS
;*****
8000          ORG      $8000          ;
;-----adresses des registres du 6850
EC00 RCRIMP EQU $EC00 ; reg. contrôle
EC00 RSRIMP EQU $EC00 ; reg. Etat
EC01 RRXIMP EQU $EC01 ; reg. Réception
8000 C6 43          LDB      %%01000011 ; initialisation programmée avec
; RTS=1 (ligne RTS à -12v)
8002 F7 EC00          STB      RCRIMP ; registre contrôle
8005 8E 8100          LDX      #$8100 ; adrs rangement données
8008 108E 000A          LDY      #10 ; nb d'octet à recevoir
800C 31 3F          LEAY     -1,Y ; écarter dernière donnée
;
;-----Configurer le mode de réception et actionner ligne RTS
800E C6 01          LDB      %%00000001 ; 7 Bits,parité paire,2 stop, clock 1/16
; + RTS |=0 (ligne RTS= +12v)
8010 F7 EC00          STB      RCRIMP ;
;
;-----Réception données avec vérifications diverses
8013 F6 EC00          ATDON   LDB      RSRIMP ; reg. Etat
8016 C5 01          BITB     %%00000001 ; Examen bit réception
8018 26 06 8020          BNE      VERIF ; SR0=1, alors vérification
801A C5 04          BITB     %%00000100 ; examen bit DCD
801C 27 F5 8013          BEQ      ATDON ; SR2=0, alors ATtente DONnée
801E 20 2F 804F          BRA      TMDCD ; --> TraiteMent DCD
;
;-----Vérification format, parité, surcharge
8020 C5 70          VERIF   BITB     %%01110000 ; test global
8022 27 0E 8032          BEQ      RXDON ; sans erreur, --> Rx réception donnée
8024 C5 10          BITB     %%00010000 ; erreur format ?
8026 27 02 802A          BEQ      *+4 ; sinon test suivant adrs +4
8028 20 22 804C          BRA      ERFMT ; --> erreur format
802A C5 40          BITB     %%01000000 ; erreur parité ?
802C 27 02 8030          BEQ      *+4 ; sinon --> erreur surcharge
802E 20 1D 804D          BRA      ERPAR ; --> erreur PARité
8030 20 1C 804E          BRA      ERSCH ; --> erreur SurCharge
;
;-----Aucune erreur détectée, lecture donnée
8032 B6 EC01          RXDON   LDA      RRXIMP ; lecture reg. réception
8035 A7 80          STA      ,X+ ; rangement donnée épuisé ?
8037 31 3F          LEAY     -1,Y ; Nb données épuisé ?
8039 26 D8 8013          BNE      ATDON ;
;
;-----Inhibition de l'émetteur
803B C6 41          LDB      %%01000001 ; RTS |=1 (ligne RTS à -12v)
803D F7 EC00          STB      RCRIMP ; reg. contrôle
;
;-----Lecture dernière donnée en cours
8040 F6 EC00          LDB      RSRIMP ; reg. Etat
8043 56          RORB ;
8044 24 FA 8040          BCC      *-4 ;
8046 B6 EC01          LDA      RRXIMP ;
8049 A7 80          STA      ,X+ ; rangement
804B 3F          SWI ;
;
;-----Si une ou plusieurs erreurs sont présentes,
;-----elles sont indiquées dans le registre B
804C 3F          ERFMT   SWI ;
804D 3F          ERPAR   SWI ;
804E 3F          ERSCH   SWI ;
804F 3F          TMDCD   SWI ;

```

Après l'initialisation programmée de l'interface avec RTS| = 1 et après changement des paramètres d'acquisition, le récepteur autorise le transfert en mettant la broche de sortie RTS| = 0 (broche RTS| à +12volts).

Le récepteur teste ensuite le bit de réception et doit trouver normalement SR0=0 indiquant l'absence de donnée. La broche d'entrée DCD| doit être maintenue constamment à 0 (broche DCD| à +12v) pour ne pas inhiber le bit de réception.

Le récepteur entre normalement la phase d'attente de réception.

L'opérateur peut alors lancer le programme de transfert du côté de transmetteurs.

Ce dernier détecte l'autorisation de transfert en examinant l'état de la broche RTS|.

Si la donnée est disponible, le récepteur vérifie le format, la parité et la surcharge.  
Le test est d'abord global.

Si une ou plusieurs erreurs sont présentes, la discrimination s'effectue avec une priorité accordée au format, puis à la parité.

Dans ce programme la détection d'une erreur provoque une interruption logicielle SWI.

Il faut vérifier alors le mot de configuration de l'interface et la division de l'horloge à la fois du côté transmetteur et récepteur.

Si la donnée est correcte, le 6809 lit le registre de réception, range la donnée en mémoire puis vérifie si le nombre de caractères demandé est épuisé.

Lorsque le nombre de caractères, moins 1, est atteint, le récepteur actionne la broche RTS| pour avertir le transmetteur.

Or, tout ceci se produit pendant l'envoi de la dernière donnée dans le registre à décalage du récepteur, d'où l'utilité de la dernière séquence.

## 6840 : Généralités

Le principe du 6840 est simple : un compteur 16 bits initialement mis à une certaine valeur se décrémente au rythme d'une horloge externe ou provenant du 6809.

Le 6840 est un temporisateur programmable, qui peut être utilisé comme :

- Générateur d'interruption.
- Générateur de signaux périodiques : Multivibrateur Astable (train d'impulsions de durée et de période programmable).
- Générateur de signaux non périodique : Monostable.
- Chronomètre : mesure d'intervalle de temps (Fréquencemètre).
- Fréquencemètre : mesure de durée d'impulsion (largeur d'impulsion).
- Compteur d'événements.

Il comporte essentiellement 3 compteurs à 16 bits, dont le fonctionnement est commandé par 3 registres de commande.

Le 6840 est mono tension (0, +5v), sa consommation est de 500 mW environ.

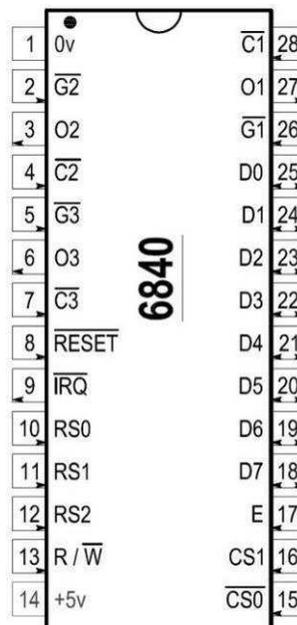
Les 3 temporisateurs peuvent fonctionner simultanément, ce qui donne beaucoup de souplesse circuit. Ces trois temporisateurs peuvent être bouclés les uns sur les autres.

On peut ainsi simultanément générer un signal carré, générer un signal unique et faire une mesure de durée.

Autres caractéristiques essentielles

- Fonctionnement à partir de l'horloge du 6809 ou d'une horloge externe.
- 3 entrées C| pour horloges externes.
- 3 entrées G| de déclenchement sont synchronisées à l'intérieur du timer.
- Fréquence maximum externe 4 Mhz (uniquement sur le timer 3).
- 3 sorties masquables.
- Les compteurs accessibles par lecture indiquent le temps qui sépare de la fin de la période programmée.

## 6840 : Brochage



## 6840 : Organisation Externe

Les échanges avec le 6809 se font par l'intermédiaire :

### 6840 : D0 à D7 :

Du bus de données, si ces broches ne sont utilisées elles se retrouvent dans l'état haute impédance.

Ce bus est utilisé pour programmer : -- Les 3 registres de contrôle CR1, CR2 et CR3  
-- Les registres tampon LSB et MSB de chaque timer  
-- Lire les registres tampon ou le registre d'état.

### 6840 : Les broches CS0 | et CS1 :

De 2 lignes de validation de boîtier qui permettent l'adressage physique du boîtier.

### 6840 : Les broches RS0, RS1, RS2 :

3 entrées de sélection de registre.

Les combinaisons de ces 3 broches permettent de sélectionner les registres internes (8 positions mémoire).

### 6840 : La broche E :

De l'entrée (Enable) qui reçoit l'horloge  $\Phi 2$  du 6809, afin de synchroniser et d'activer les échanges.

### 6840 : La broche R/W | :

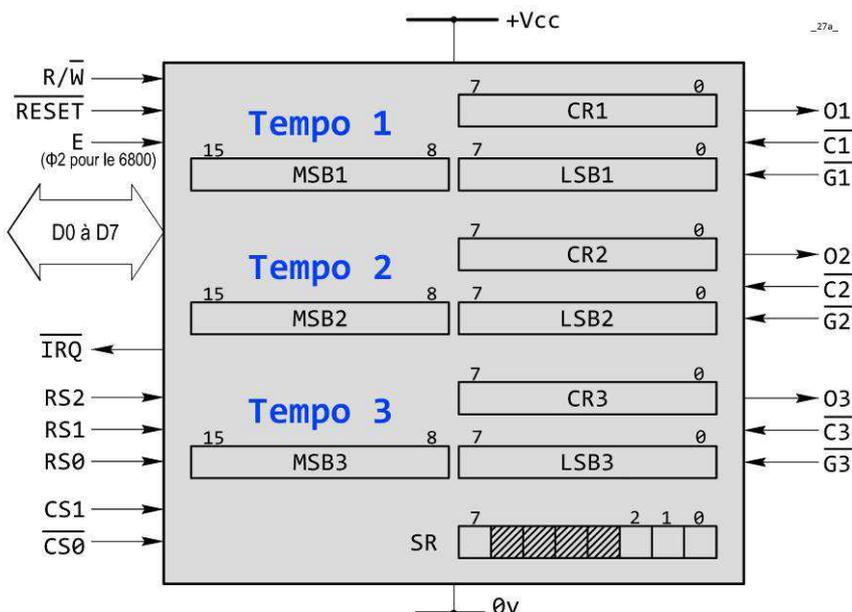
De l'entrée Lecture / écriture (Read / Write) qui fixe le sens des transferts, écriture du temporisateur ou lecture.

### 6840 : La broche IRQ | :

D'une ligne d'interruption à drain ouvert, donc supportant le OU câblé, et qui permet d'interrompre l'exécution d'un programme.

Reliée à la broche IRQ|, FIRQ| ou NMI| du 6809.

## 6840 : Organisation Externe Schématique



## **6840 : Organisation Externe Liaisons avec la périphérie**

### **6840 : Entrées horloges externes : C1 |, C2 |, C3 |**

Ces entrées sont compatibles avec les niveaux TTL et peuvent varier du continu à la valeur de l'horloge du 6809.

La fréquence appliquée peut aller du continu à la fréquence d'horloge appliquée sur la broche E (ENABLE).

Les compteurs internes du 6840 peuvent être activés par l'horloge du 6809 ou par des horloges externes.

Chaque temporisateur possède sa propre entrée d'horloge externe.

L'utilisation de ces entrées nécessite impérativement la prise en considération des paramètres dynamiques du 6840.

L'horloge du temporisateur numéro 3 peut être divisée par 8, mais elle est traitée de façon identique à C1| ou C2|

### **6840 : Entrées GATE | : G1 |, G2 |, G3 |**

Ces entrées sont compatibles avec des signaux asynchrones TTL, donc non synchrones de l'horloge du 6809.

Trois impulsions d'horloge sont nécessaires pour les prendre en compte.

Ces entrées sont directement liées aux compteurs 16 bits.

### **6840 : Sorties des temporisateurs : O1, O2, O3**

Chaque sortie peut commander 2 charges TTL et délivre un signal défini quand le temporisateur travaille en astable ou en monostable.

Une sortie non validée reste à 0

En mode intervalle de temps (chronomètre ou fréquencemètre), des signaux apparaissent en sortie si CRx7 = 1, mais leur forme est imprévisible.

## **6840 : Organisation Interne**

On doit adresser 10 registres de 8 bits différents :

-- 6 registres de 8 bits pour les timers **MSB1, LSB1, MSB2, LSB2, MSB3, LSB3**.

-- 3 registres de 8 bits **CR1 CR2 CR3** pour les contrôles

-- 1 registre d'état **SR**

### **6840 : MSB1, LSB1, MSB2, LSB2, MSB3, LSB3 :**

Elle comprend essentiellement 3 compteurs de 2 x 8 bits (permettant la génération de signaux de rapport cyclique 1/2).

Chaque Timers est divisé en 2 registres de 8 bits, correspondants aux poids forts MSB et aux poids faibles LSB.

Ces registres tampon, peuvent fonctionner en 16 bits, contiennent les paramètres de comptage.

Ces Timers permettant de générer des signaux de rapport cyclique variable.

Les données sont transférées des registres tampon dans les compteurs proprement dits lors d'un cycle d'initialisation des compteurs.

Les compteurs sont décrémentés à chaque impulsion d'horloge (interne ou externe).

Suivant le mode de fonctionnement spécifié, le compteur s'arrête ou recommence un nouveau cycle lorsqu'il arrive zéro.

## 6840 : CR1 CR2 CR3 :

Trois registres de contrôle en 8 bits définissent le mot de fonctionnement de chacun des compteurs : mode astable, mode monostable, comparateur de fréquence, comparateur de largeur d'impulsions.

Ces registres CR1 CR2 CR3 ne sont accessibles qu'en écriture.

## 6840 : SR registre :

Un registre d'état en 8 bits, à lecture seule, il nous fournit les indications d'interruption de chacun des timers (interruptions indépendantes) et de n'importe lequel d'entre deux bits SR7.

## 6840 : Fonctionnement

Pour chaque temporisateur, il faut programmer le registre de contrôle CR afin de définir lequel des trois modes (astable, monostable ou intervalle de temps) on va utiliser.

Le contenu du registre CR permet également de valider la sortie et les interruptions générées par le 6840, de choisir l'horloge d'activation et le mode de fonctionnement des compteurs.

Il faut ensuite charger le registre tampon associé au temporisateur choisi.

Dans le cas de fonctionnement en interruption, un test sur le registre d'état SR est nécessaire, afin de connaître le temporisateur qui est à l'origine de cette interruption.

## 6840 : Adressage, Sélection Du Boîtier

Le bit b0 du registre de contrôle 2 (registre CR20) permettra de différencier les registres de contrôle 1 et 3 qui ont donc même adresse.

Les broches RS0, RS1, RS2 combinées aux broches R/W|, CS0| et CS1 permettent de sélectionner les registres internes, mais ne suffisent pas pour adresser les 10 registres internes.

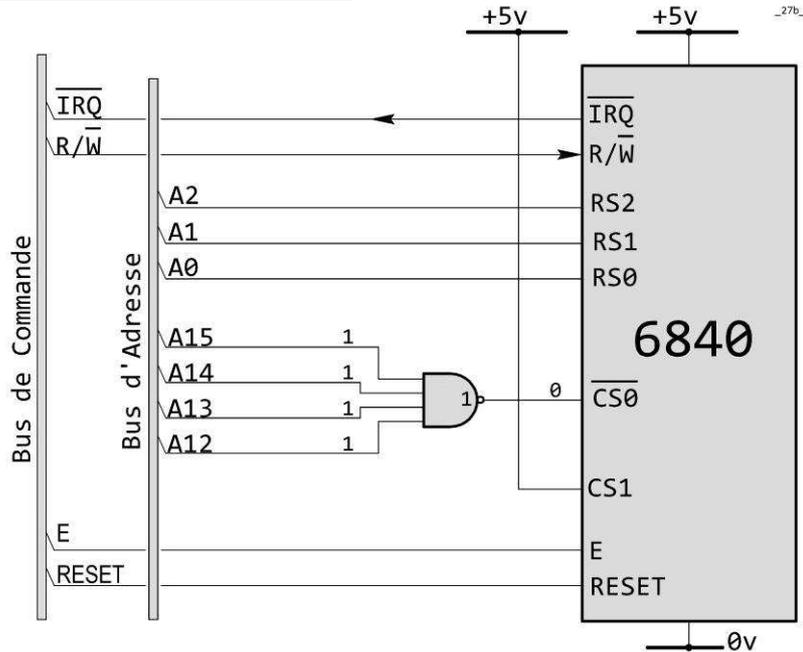
Le 6840 occupe 8 octets mémoire pour 10 registres interne.

Ces entrées RS0, RS1 et RS2 reçoivent nécessairement les bits A0, A1 et A2 du bus d'adresse pour que le 6809 voie le temporisateur comme 8 positions mémoires consécutives.

Pour adresser les registres CR1 ou CR3 il faut tenir compte de l'état du bit b0 du registre CR2.

PTM 6840	Broches du 6809 →	A15 ... A3	A2	A1	A0	En lecture R / W = 1	En écriture R / W = 0	CR20 c'est le bit 0 de CR2
	Broches du 6840 →	CS1	CS0	RS2	RS1			
Adresses	Adr	1	0	0	0	0	<b>Pas de lecture possible</b>	Ecriture registre <b>CR3</b> si CR20 = 0 Ecriture registre <b>CR1</b> si CR20 = 1
	Adr + 1	1	0	0	0	1	Lecture registre d'état <b>SR</b>	Ecriture registre <b>CR2</b>
	Adr + 2	1	0	0	1	0	Lecture compteur <b>MSB1</b>	Ecriture registre tampon <b>MSB1</b>
	Adr + 3	1	0	0	1	1	Lecture compteur <b>LSB1</b>	Ecriture registre tampon <b>LSB1</b>
	Adr + 4	1	0	1	0	0	Lecture compteur <b>MSB2</b>	Ecriture registre tampon <b>MSB2</b>
	Adr + 5	1	0	1	0	1	Lecture compteur <b>LSB2</b>	Ecriture registre tampon <b>LSB2</b>
	Adr + 6	1	0	1	1	0	Lecture compteur <b>MSB3</b>	Ecriture registre tampon <b>MSB3</b>
	Adr + 7	1	0	1	1	1	Lecture compteur <b>LSB3</b>	Ecriture registre tampon <b>LSB3</b>

**Exemple d'un 6840 implanté à l'adresse \$F000**



A15	A14	A13	A12		A8	A7		A2	A1	A0
1	1	1	1	0	0	0	0	0	X	X
\$F				\$0				\$0		\$0 à \$7

En affectant un 0 sur les bits d'adresse non connectés, le 6840 aura pour adresses :  
 LSB (Least Significant Bit) octet de poids faible.  
 MSB (Most Significant Bit) octet de poids fort.

**\$F000** : registres CR des tempos 1 et 3 suivant le bit b0 du registre CR de la tempo 2.

**\$F001** : registres CR de la tempo 2 et registre SR suivant la broche R/W].

**\$F002** : poids Fort MSB du registre tampon 1 et compteur num 1 suivant la broche R/W]

**\$F003** : poids Faible LSB du registre tampon 1 et compteur num 1 suivant la broche R/W]

**\$F004** : poids Fort MSB du registre tampon 2 et compteur num 2 suivant la broche R/W]

**\$F005** : poids Faible LSB registre tampon 2 et compteur num 2 suivant la broche R/W]

**\$F006** : poids Fort MSB du registre tampon 3 et compteur num 3 suivant la broche R/W]

**\$F007** : poids Faible LSB registre tampon 3 et compteur num 3 suivant la broche R/W]

**Exemple** : Autrement dit si un 6840 se trouve dans la zone partant de l'adresse \$7000,  
 On aura alors :

\$7000	écriture de CR1 ou CR3 suivant le bit 0 de CR2
\$7001	lecture de SR ou écriture de CR2 suivant la broche R/W]
\$7002	lecture ou écriture de MSB1 en fonction de la broche R/W]
\$7003	" " " de LSB1 " " " " " "
\$7004	" " " de MSB2 " " " " " "
\$7005	" " " de LSB2 " " " " " "
\$7006	" " " de MSB3 " " " " " "
\$7007	" " " de LSB3 " " " " " "

CR1, CR2 et CR3 précisent le mode de fonctionnement de chaque temporisateur.

Les différents bits de ces registres remplissent un rôle identique SAUF le bit 0 :

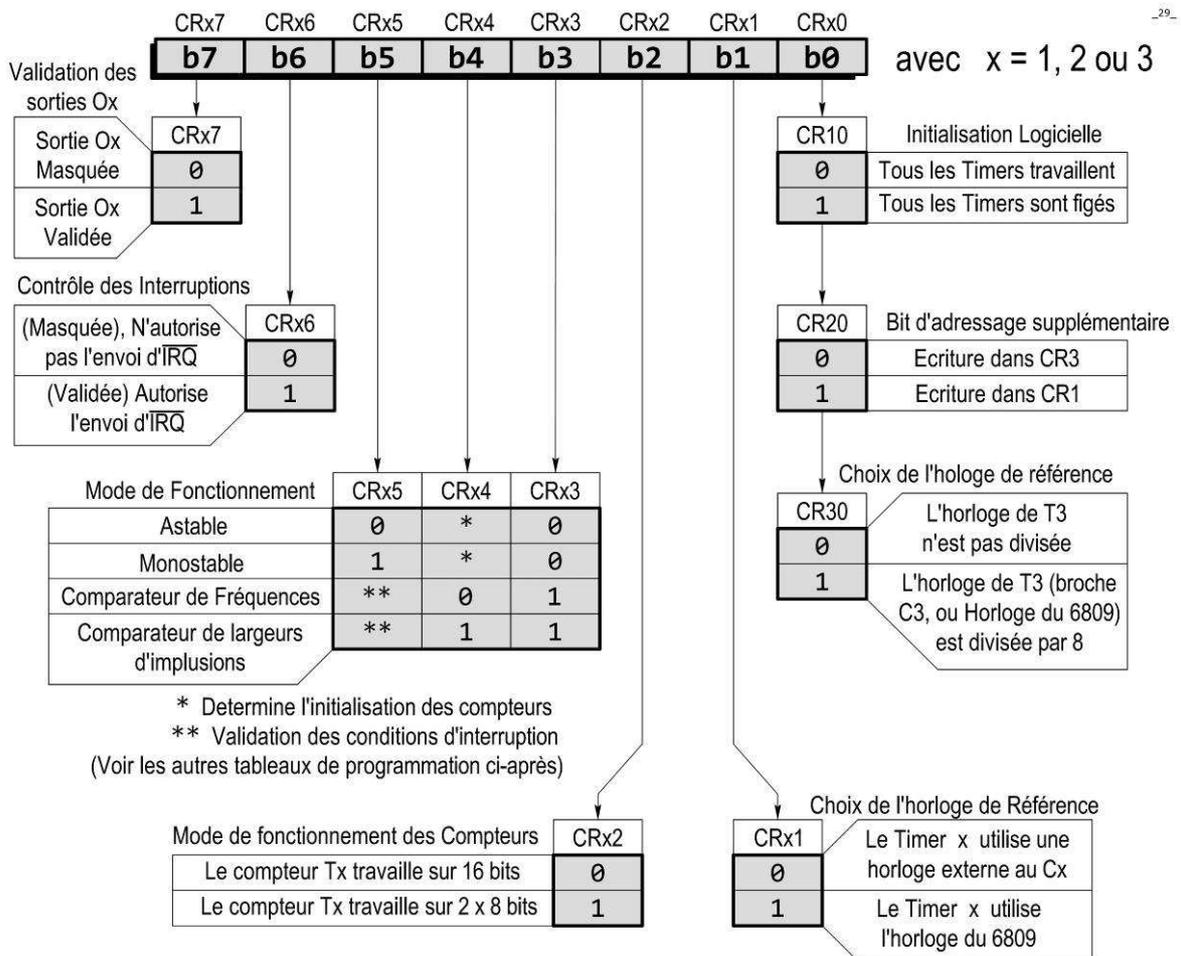
- Le bit 0 du CR1 sert de RESET logiciel.
- Le bit 0 du CR2 sert de bit d'adressage supplémentaire.
- Le bit 0 du CR3 permet de diviser ou non l'horloge du compteur 3 par 8.

Pour ces 3 registres, les autres bits permettent de choisir :

- Le bit 1 : l'horloge de référence.
- Le bit 2 : le mode de décrémentation des compteurs
- Les bits 3, 4 et 5 : le mode de travail des temporisateurs
- Le bit 6 : les interruptions.
- Le bit 7 : les sorties de la broche Ox

De plus, comme le montre la table des adresses, seul le registre CR2 (registre de commande numéro 2) est adressable directement.

Pour l'écriture dans les registres de commande numéro 1 et 3, il faut auparavant positionner le bit 0 du registre CR numéro 2.



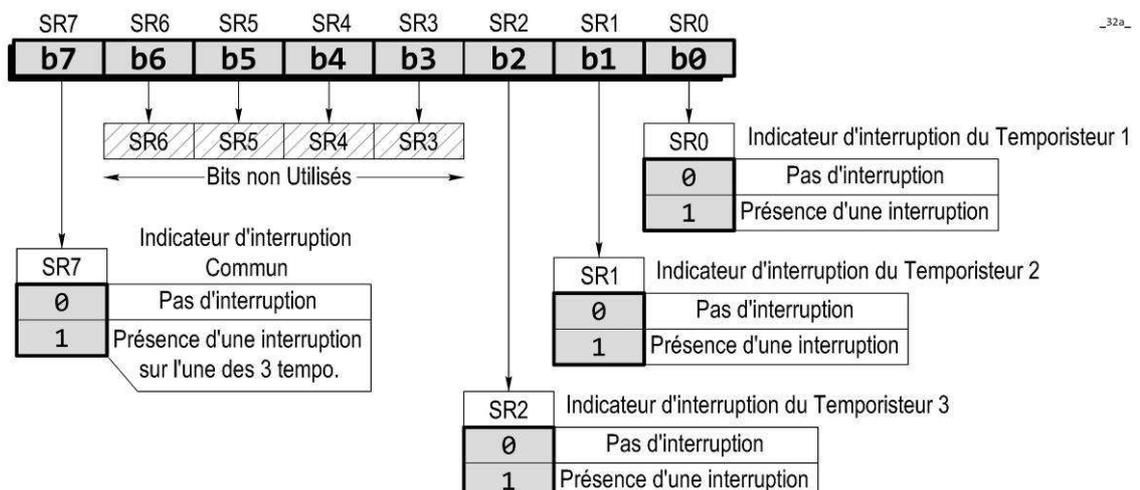
<b>6840 : CR10</b>	= 0	tous les temporisateurs sont autorisés à fonctionner	
	= 1	tous les temporisateurs sont figés dans leur état présent.	
<b>6840 : CR20</b>	= 0	accès au registre CR num 3	
	= 1	accès au registre CR num 1	
<b>6840 : CR30</b>		(Horloge externe 4 Mhz maxi)	
	= 0	facteur de division : 1	
	= 1	facteur de division : 8.	
<b>6840 : CRx1</b>	Définit si l'on utilise :		x sera remplacé par 1, 2 ou 3
	= 0	utilisation d'une horloge externe (appliquée sur la broche d'entrée Cx correspondante)	
	= 1	utilisation d'une horloge du microsysteme.	
<b>6840 : CRx2</b>	Définit la taille du compteur		x sera remplacé par 1, 2 ou 3
	= 0	compteur 16 bits	
	= 1	compteur 2 x 8 bits	
<b>6840 : CRx3 CRx4 CRx5</b>			x sera remplacé par 1, 2 ou 3
	0 x 0	Astable	
	0 x 1	Monostable	
	1 0 x	Comparaison de fréquences	
	1 1 x	Comparaison de largeurs d'impulsions	
	x	est utilisé pour modifier l'initialisation du compteur et sa validation, ou les conditions d'interruption.	
<b>6840 : CRx6</b>	Valide ou non les interruptions		x sera remplacé par 1, 2 ou 3
	= 0	n'autorise pas l'envoi d'IRQ  (niveau haut)	
	= 1	autorise l'envoi IRQ	
<b>6840 : CRx7</b>	Valide ou non la sortie correspondante		x sera remplacé par 1, 2 ou 3
	= 0	sortie masquée	
	= 1	sortie validée	

## 6840 : Registre d'Etat SR

C'est un registre à lecture seule. Seul 4 sur les 8 bits sont utilisés comme indicateurs d'interruption.

- Le bit **SR0** est associé au temporisateur 1
- Le bit **SR1** est associé au temporisateur 2
- Le bit **SR2** est associé au temporisateur 3

Le bit SR7 est le bit d'interruption commun aux 3 temporisateurs. Il est positionné à 1 lorsque n'importe quel bit indicateur indépendant sera à 1, à condition que le bit CRx6 soit égal à 1 (validation des interruptions).



Un indicateur d'interruption bit SR2, SR1 ou SR0 est remis à zéro :

- soit par un niveau actif sur la broche RESET|
- soit par CR10 = 1
- soit par une lecture du compteur du temporisateur à condition que le registre d'état ait été lu auparavant et l'indicateur d'interruption positionné.

Cette condition liant la lecture du registre d'état SR et la lecture du compteur a été prévue pour éviter l'oubli d'interruptions qui pourraient se produire après la lecture du registre d'état mais avant la lecture du compteur.

## **6840 : Rôle des registres Tampons (Initialisation)**

Chaque temporisateur indépendant comprend un registre tampon de 16 bits (en écriture) associé à un compteur 16 bits (en lecture).

La durée de comptage d'un compteur dépend du contenu du registre tampon.

Ce contenu est calculé en fonction de signaux désignés en sortie, voir les divers modes de fonctionnement du 6840.

Comme les compteurs sont en 16 bits et que le bus de données est en 8 bits, il est nécessaire de stocker dans un BUFFER les 8 bits de plus fort poids.

L'octet de poids fort est préalablement stocké dans le registre MSB Buffer, il sera automatiquement transféré dans le registre tampon MSB lors de l'écriture du registre LSB (poids faible).

**C'est pour cela que l'on doit d'abord écrire le MSB puis le LSB. Cet ordre est le même pour la lecture.**

Il serait bon de préconiser des opérations 16 bits avec le 6809 pour respecter l'ordre de lecture et d'écriture  
Pour l'écriture : STD, STX, ou STY  
Pour la lecture : LDD, LDX, ou LDY

L'initialisation d'un compteur peut se faire :

- En appliquant le niveau actif sur RESET| ou CR10=1
- Lors de la commande écriture des registres tampons
- Par application d'une transition descendante sur l'entrée Gx| (G=Gate)

## **6840 : Rôle des compteurs (Initialisation)**

L'initialisation d'un compteur est définie comme transfert d'une donnée du registre tampon dans le compteur. Avec pour conséquence l'effacement de l'indicateur d'interruption associé au compteur.

L'initialisation du compteur se produit dans le cas :

- D'un RESET externe (broche RESET| à 0)
- D'un RESET interne CR10 = 0 (dépendant du mode de fonctionnement)
- Lors d'une écriture dans le registre tampon associé d'une transition négative sur la broche gâchette Gx| (dépendant du mode de fonctionnement).

Une fois initialisé, le compteur est automatiquement décrémenté à la vitesse de l'horloge d'activation choisie.

A chaque fois que le compteur passe par une valeur nulle, il est automatiquement réinitialisé avec le contenu du registre tampon qui lui est associé.

Les compteurs sont accessibles en lecture, comme pour les tampons, il faut toujours lire l'octet de poids fort MSBx avant l'octet de poids faible LSBx.

La lecture de MSBx entraîne le transfert automatique le LSBx dans LSB Buffer.  
La lecture de LSBx consiste à lire le contenu de LSB Buffer.

**6840 : Mode Astable (aussi appelé Multivibrateur Astable ou Mode continu)**

(Mode 01, 02, 03 et 04 voir tableau ci-dessous)

Chacun des 3 temporisateurs peut travailler en multivibrateur Astable CRx5 = 0 CRx3 = 0 avec la broche de sortie Ox est valide on obtient :

- Soit un signal carré CRx2 = 0 (on travaille en 16 bits).
- Soit un signal asymétrique CRx2 = 1 (on travaille en 2 x 8 bits).

Le compteur peut donc travailler :

- Soit en 16 bits.
- Soit en 2 x 8 bits.

L'initialisation d'un compteur peut se faire par remise à zéro du temporisateur de trois façons suivantes :

- Application d'un niveau Bas sur la broche RESET|
- Mise à 1 de b0 du registre CR1.
- Application d'un front descendant sur la broche d'entrée Gx| (G=Gate)

D'autre part, si le bit b4 des registres CRx (x =1, 2 ou 3) est à 0, alors on aura une initialisation du compteur à chaque commande d'écriture dans le registre tampon.

Autres Remarques :

- Il est indispensable pour le fonctionnement des compteurs que l'entrée Gx| (G=Gate) soit maintenue à l'état Bas.
- En fonctionnement 16 bits, la sortie du compteur, si elle est validée est à l'état Bas pendant toute la phase de mise à l'état initial et y restera ensuite pendant tout le temps de décrémentation de compteur TO (Time Out)
- Cette sortie passera ensuite à l'état Haut et s'y maintiendra pendant la même période de temps, et ainsi de suite.

**6840 : Mode Astable Fonctionnement en 2 x 8 bits**

L'utilisateur calcule la valeur de **M** (valeur contenue dans les 8 bits MSB) et la valeur de **L** (valeur contenue dans les 8 bits LSB) en fonction du signal désiré et l'horloge d'activation utilisée.

Une fois initialisé, le compteur décrémente **M** et **L**.

Suite au passage à zéro des poids faibles LSB, l'octet de poids fort MSB est décrémente.  
Suite au passage à zéro des poids forts, la sortie Ox passe à l'état haut.

La sortie passera à l'état Haut au début de la prochaine impulsion d'horloge, elle restera à l'état Haut jusqu'à ce que les compteurs LSB et MSB soient tous deux à zéro.

Autrement dit, le compteur LSB étant utilisé en décompteur, chaque fois que celui ci passe à 0, le compteur MSB est décrémente d'une unité.

Quand le LSB=0, le MSB est inchangé; sur le coup de l'horloge suivant, le LSB est remis à sa valeur initiale, chargée dans le registre tampon, puis le MSB est décrémente.

Chaque fois que le compteur passe à zéro, l'indicateur individuel d'interruption est positionné à 1, une interruption sera transmise au 6809 si CRx6 = 1.

La réinitialisation du compteur entraîne celle de l'indicateur d'interruption, et passage à l'état Bas de la sortie Ox

Le bit CRx4 de chaque temporisateur permet de choisir le type d'initialisation des compteurs.

x sera remplacé par 1, 2 ou 3

_30a_				
<b>Mode Astable</b> CRx7=1 CRx5=0 CRx3=0				
Condition nécessaire de fonctionnement du compteur ( $\overline{G} = 0$ ) la broche $\overline{GATE}$ soit maintenu à l'état Bas				
	CRx4	CRx2	Initialisation compteur	Signal en broche de sortie Ox
Fonctionnement sur 16 bits	0	0	01 $\overline{G}$ ou W ou R	
	1	0	02 $\overline{G}$ ou R	
Fonctionnement sur 2 x 8 bits	0	1	03 $\overline{G}$ ou W ou R	
	1	1	04 $\overline{G}$ ou R	

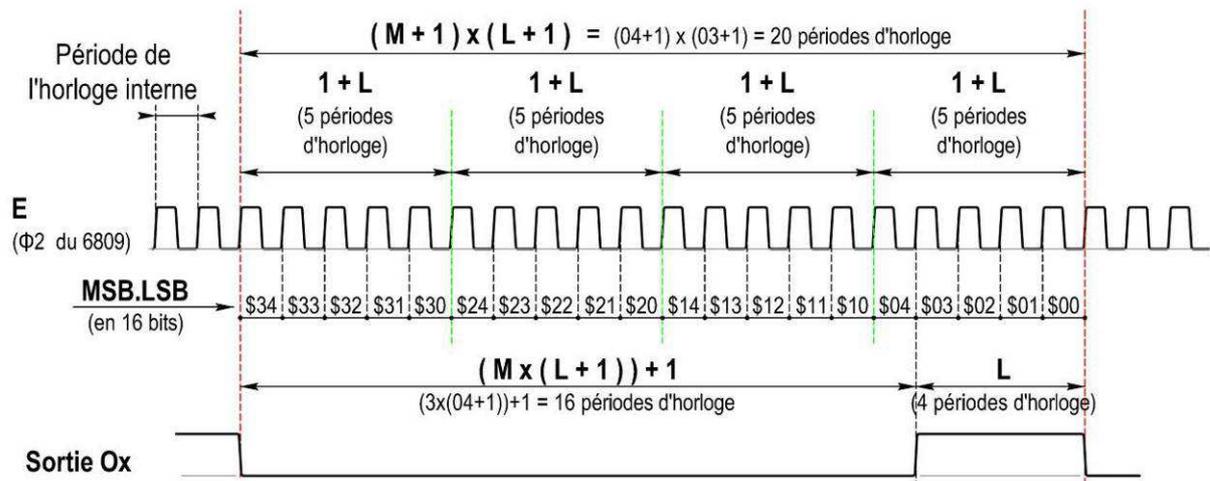
_31a_	
$\overline{G}$	Front descendant sur la broche d'entrée $\overline{Gx}$ ( $\overline{GATE}$ )
W	Commande d'écriture des registres tampon MSB puis LSB
R	Reset du Timer ( CR10=1 ou broche $\overline{RESET}$ niv. Bas )
N	Donnée sur 16 bits contenue dans les registres tampons du compteur
L	Donnée sur 8 bits contenue dans le registre tampon LSB du compteur
M	Donnée sur 8 bits contenue dans le registre tampon MSB du compteur
T	Front descendant de l'horloge appliquée au compteur
to	Initialisation du compteur
TO	Time Out : Temps de décrémentation des compteurs

### Exemple de signal en sortie du 6840 en mode Astable, 2 x 8 bits utilisant l'horloge interne

Contenu de M = MSB = \$ 03

Contenu de L = LSB = \$ 04

\_32b\_



## 6840 : Mode Astable : Exemple 01

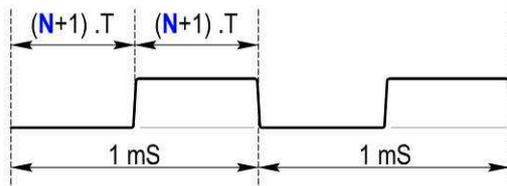
En reprenant la configuration du tableau ci-dessus (Mode Astable), **on désire obtenir simultanément** :

- En sortie du temporisateur 1 un signal carré de période 1 ms
- En sortie du temporisateur 2 un signal rectangulaire de période 0,5 ms et de rapport cyclique 1/4.

On utilisera l'horloge du 6809. On utilisera de ce fait :

- Le temporisateur 1 sur 16 bits
- Le temporisateur 2 sur 2 x 8 bits

### Temporisateur 1 (en 16 bits)



On souhaite obtenir une période de 1 mS

Donc  $2 \text{ fois } (N+1).T = 1 \text{ mS}$

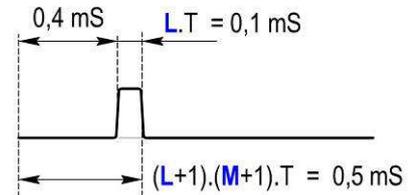
l'horloge du 6809 à 1 Mhz  $\rightarrow T = 1 \mu\text{S}$

$2 \times (N + 1) \times T = 1 \text{ ms}$

$2 \times (N + 1) \times 1 \mu\text{S} = 1 \text{ 000 } \mu\text{S}$

$$N = \frac{1 \text{ 000 } \mu\text{S}}{2 \times 1 \mu\text{S}} - 1 \quad N = 499_{(10)} \quad \text{soit } N = \text{\$01F3}$$

### Temporisateur 2 (en 2 x 8 bits)



On souhaite obtenir une période de 0,5 mS et un rapport cyclique de 1/4.

$L.T = 0,1 \text{ mS} = 100 \mu\text{S}$

l'horloge du 6809 à 1 Mhz  $\rightarrow T = 1 \mu\text{S}$

D'où  $L = 100_{(10)} = \text{LSB} = \text{\$64}$

$(L+1).(M+1).T = 0,5 \text{ mS} = 500 \mu\text{S}$

$M = \frac{500 \mu\text{S}}{(L+1).T} - 1 \approx 4 \quad \text{MSB} = \text{\$04}$

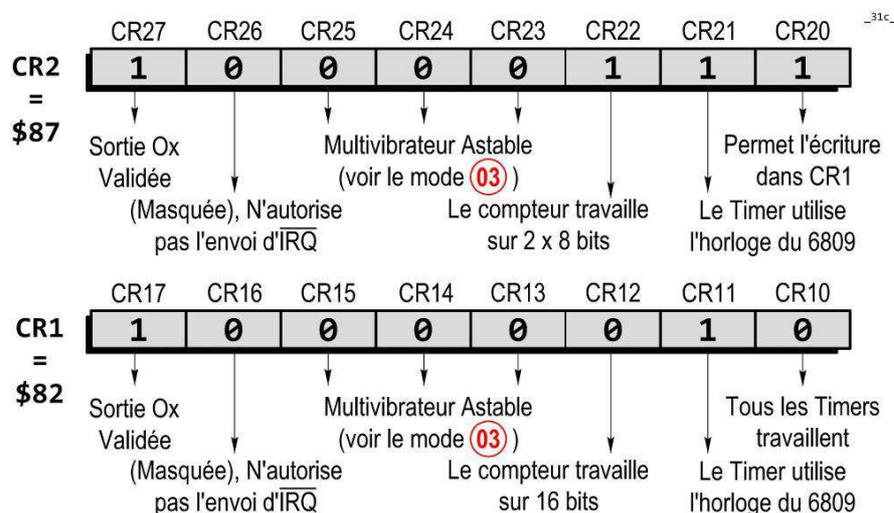
### Exemple 01 : le programme est le suivant

```

;-----Timer 2 (à faire avant le Timer 1-----
LDA  #\$87          ;%1000 0111
STA  \$F001         ;écriture dans CR2 avec CR20=1
                    ; pour autoriser l'écriture dans CR1
LDD  #\$0464       ;
STD  \$F004         ;écriture du registre tampon MSB 2 + LSB 2

;-----Timer 1-----
LDA  #\$82          ;%1000 0010
STA  \$F000         ;écriture dans CR1
LDD  #\$01F3       ;
STD  \$F002         ;écriture du registre tampon MSB 1 + LSB 1
    
```

### Exemple 01 : explication de la programmation des registres CR2 et CR1



### Exemple 01 : Remarques :

- En fonctionnement sur 2 x 8 bits, si L = M = 0 on obtient un signal de sortie de fréquence moitié de celle de l'horloge.
- En fonctionnement sur 2 x 8 bits, si L = 0, le compteur revient au fonctionnement 16 bits avec apparition du Time Out (TO) au bout de M + 1 périodes d'horloge.
- **Toujours charger les registres MSB avant les registres LSB.**

## 6840 : Mode Monostable (Mode Monocoup)

(Mode 05, 06, 07 et 08 voir tableau ci-dessous)

Le mode est identique au mode astable précédent à trois exceptions près :

**La première :** la sortie est validée pour une seule impulsion jusqu'à une réinitialisation.

- Après le premier Time Out la sortie reste à l'état Bas jusqu'au prochain cycle d'initialisation.
- Comme en fonctionnement astable, l'on peut travailler sur 16 bits ou en 2 x 8 bits.
- Le fonctionnement du compteur interne reste cyclique dans le fonctionnement en monostable.
- Chaque Time Out du compteur positionne à 1 l'indicateur d'interruption indépendant du registre d'Etat ainsi que la réinitialisation du compteur.

**La seconde :** la condition  $G| = 0$  n'est pas prise en considération.

- Il suffit d'appliquer une transition sur la broche G| (G=Gate) ou déclencher le Monostable.

**La troisième :** si L = M = 0 ou N = 0, la sortie est inhibée.

- Lorsque L = M = 0, en 2 x 8 bits ou N = 0 en 16 bits, la sortie tombe à l'état bas sur cette première impulsion d'horloge reçue pendant ou après l'initialisation du compteur.
- La sortie reste à l'état bas jusqu'à ce que l'on change le mode de fonctionnement ou que l'on rentre des données différentes de zéro dans les registres tampons.
- On a toujours un Time Out à la fin de chaque période d'horloge.
- En fonctionnement normal sur 16 bits, le compteur sera décrémenté jusqu'à zéro au bout de (N+1) périodes d'horloge, N étant la donnée de 16 bits contenue dans le registre tampon.

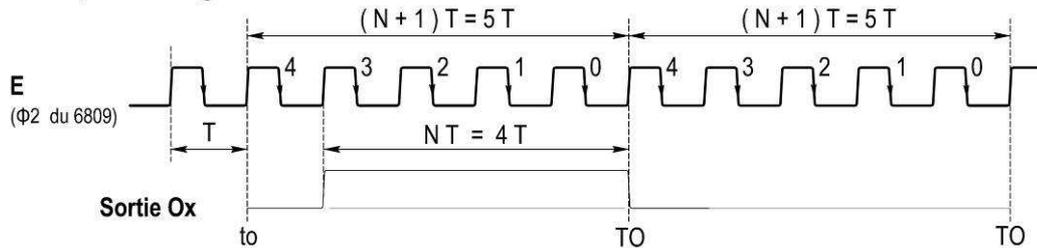
Mode Monostable CRx7=1 CRx5=1 CRx3=0				
	CRx4	CRx2	Initialisation compteur	Signal en broche de sortie Ox
Fonctionnement sur 16 bits	0	0	05 $\overline{G} $ ou W ou R	
	1	0	06 $\overline{G} $ ou R	
Fonctionnement sur 2 x 8 bits	0	1	07 $\overline{G} $ ou W ou R	
	1	1	08 $\overline{G} $ ou R	

$\overline{G} $	Front descendant sur la broche d'entrée $\overline{Gx}$ (GATE)
W	Commande d'écriture des registres tampon MSB puis LSB
R	Reset du Timer ( CR10=1 ou broche $\overline{RESET}$ niv. Bas )
N	Donnée sur 16 bits contenue dans les registres tampons du compteur
L	Donnée sur 8 bits contenue dans le registre tampon LSB du compteur
M	Donnée sur 8 bits contenue dans le registre tampon MSB du compteur
T	Front descendant de l'horloge appliquée au compteur
to	Initialisation du compteur
TO	Time Out : Temps de décrément des compteurs

### Exemple avec N=04

La sortie, si elle est validée, passe à l'état Haut après la première impulsion d'horloge qui se produit pendant ou après la durée de la mise à l'état initial et y reste pendant N impulsions d'horloge suivantes.

### Exemple de signal de Sortie en Mode Monostable



En fonctionnement sur 2 x 8 bits, le Time Out dure  $(L + 1) (M + 1) T$ .

**L** représente la donnée dans le registre tampon **LSB**  
**M** représente la donnée dans le registre tampon **MSB**

La sortie, si elle est validée est à l'état Bas pendant toute la phase d'initialisation et y reste jusqu'à ce que le MSB soit à zéro.

## 6840 : Mode Mesure d'Intervalle de Temps

x sera remplacé par 1, 2 ou 3

Ce mode est sélectionné quand le bit CRx3 = 1. Le bit CRx4 permet ensuite de choisir entre :

- Le mode comparaison de fréquence CRx4 = 0
- Le mode comparaison de largeur d'impulsion CRx4 = 1

Le mode intervalle de temps est utilisé dans des applications nécessitant plus de souplesse dans la génération des interruptions et l'initialisation des compteurs.

Dans ce mode les indicateurs individuels d'interruption sont fonctions à la fin du comptage d'un compteur (TO Time Out des compteurs) ou sur transition active de l'entrée Gx| (G=Gate).

Le principe de ce mode est de comparer une fréquence ou une impulsion externe, présente sur une des entrées Gx|, avec le contenu d'un compteur associé Cx|

Dans chacun de ce mode le signal de sortie Ox n'est pas défini, mais malgré cela le compteur peut travailler soit en 16 bits soit en 2 x 8 bits.

Un front descendant sur l'entrée G| active le compteur et commence un cycle d'initialisation.

Le compteur est alors décrémenté à chaque coup d'horloge pendant ou après l'initialisation du compteur et jusqu'à ce qu'une interruption soit engendrée.

## 6840 : Mode Mesure d'Intervalle de Temps Comparaison de Fréquence CRx4.CRx3 = %01

x sera remplacé par 1, 2 ou 3

(Mode 09 et 10 voir tableau ci-dessous) Ce mode présente deux aspects en fonction du contenu du bit CRx5

### CRx5 = 0 (mode 09)

Le compteur Cx est initialisé par un front descendant sur Gx|, une interruption est générée si l'entrée Gx| revient à l'état Bas avant la fin du comptage.

Si la période de comptage de compteur s'est écoulée avant l'arrivée de front descendant sur Gx|, le compteur recommence son cycle de décomptage et ainsi de suite jusqu'au moment où un front descendant est appliqué sur Gx|.

Dans ce cas, un bit est positionné à l'extérieur du temporisateur à la fin de la première période de comptage, afin d'empêcher la génération d'une interruption, tant que le compteur n'a pas été réinitialisé.

Cette réinitialisation est obtenue par l'application d'un front descendant sur Gx| (la condition front descendant sur Gx| et Bit indicateur d'interruption et Time Out est satisfaite puisque la fin d'une période est apparue sans positionnement indicateur d'interruption individuel).

### **CRx5 = 1** (mode 10)

Le signal à mesurer est présent sur une des entrées Cx|.  
Le compteur Cx associé est initialisé par un front descendant sur Gx|.

Si à la fin de comptage du compteur apparaît avant un nouveau front négatif sur Gx|, l'indicateur individuel d'interruption est positionné.

Le compteur est inhibé, tant que cet indicateur individuel reste à 1 et qu'un nouveau front descendant n'a pas été détecté.

Si au contraire le front négatif apparaît sur Gx| avant la fin du comptage, l'indicateur individuel d'interruption reste à 0 et ce front négatif sur Gx| ne fait que réinitialiser le compteur.

Ce fonctionnement continue jusqu'à la fin du comptage.

## **6840 : Mode Mesure d'Intervalle de Temps Comparaison de Largeur d'Impulsion CRx4.CRx3 = %11**

(Mode 11 et 12 voir tableau ci-dessous) Ce mode est similaire au mode de comparaison de fréquence, l'application d'un front descendant sur Gx| lance le compteur mais cette fois, c'est une transition positive de Gx| qui arrête le comptage.

Comme en comparaison de fréquence, ce mode présente deux aspects de fonctionnement, en fonction du contenu de CRx5

### **CRx5 = 0** (mode 11)

Gx| passe à l'état Bas, le compteur est initialisé.

Si Gx| revient à l'état Haut avant la fin de la période de comptage, l'indicateur d'interruption individuel est positionné et le compteur se bloque.

### **CRx5 = 1** (mode 12)

Si à la fin du comptage TO (Time Out) apparaît avant le retour à l'état Haut de Gx|, une interruption est générée.

	CRx5	CRx4	CRx3	Initialisation du Compteur	Activation du compteur	Désactivation du compteur	Positionnement de l'indicateur d'interruption SR (0,1 ou 2) à 1
Comparateur de Fréquences	0	0	1	$\overline{Gx}$ et bit SR 0,1,2 à 0 Compteur inactif	$\overline{Gx}$ et bit SR (0,1 ou 2) à 0 Pas de commande d'écriture des Registres Tampons	Commande d'écriture des Registres Tampons Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	$\overline{Gx}$ avant TO (Time Out)
				Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	Pas de Reset du Timer (CR10=0 et $\overline{RESET}$ niv. Haut)	bit SR (0,1 ou 2) à 1	
Comparateur de Fréquences	1	0	1	$\overline{Gx}$ et bit SR 0,1,2 à 0	$\overline{Gx}$ et bit SR (0,1 ou 2) à 0 Pas de commande d'écriture des Registres Tampons	Commande d'écriture des Registres Tampons Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	TO (Time Out) avant $\overline{Gx}$
				Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	Pas de Reset du Timer (CR10=0 et $\overline{RESET}$ niv. Haut)	bit SR (0,1 ou 2) à 1	
Comparateur de largeurs d'impulsions	0	1	1	$\overline{Gx}$ et bit SR 0,1,2 à 0	$\overline{Gx}$ et bit SR (0,1 ou 2) à 0 Pas de commande d'écriture des Registres Tampons	Commande d'écriture des Registres Tampons Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	$\overline{Gx}$ avant TO (Time Out)
				Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	Pas de Reset du Timer (CR10=0 et $\overline{RESET}$ niv. Haut)	bit SR (0,1 ou 2) à 0 $\overline{Gx}$ niv. Haut	
Comparateur de largeurs d'impulsions	1	1	1	$\overline{Gx}$ et bit SR 0,1,2 à 0	$\overline{Gx}$ et bit SR (0,1 ou 2) à 0 Pas de commande d'écriture des Registres Tampons	Commande d'écriture des Registres Tampons Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	TO (Time Out) avant $\overline{Gx}$
				Reset du Timer (CR10=1 ou $\overline{RESET}$ niv. Bas)	Pas de Reset du Timer (CR10=0 et $\overline{RESET}$ niv. Haut)	bit SR (0,1 ou 2) à 0 $\overline{Gx}$ niv. Haut	

$\overline{G}$	Front descendant sur la broche d'entrée $\overline{Gx}$ (GATE)
W	Commande d'écriture des registres tampon MSB puis LSB
R	Reset du Timer ( CR10=1 ou broche $\overline{RESET}$ niv. Bas )
N	Donnée sur 16 bits contenue dans les registres tampons du compteur
L	Donnée sur 8 bits contenue dans le registre tampon LSB du compteur
M	Donnée sur 8 bits contenue dans le registre tampon MSB du compteur
T	Front descendant de l'horloge appliquée au compteur
to	Initialisation du compteur
TO	Time Out : Temps de décrémentation des compteurs

# 6840 : Tableau regroupant tous les Modes de Fonctionnement

- Le mode Astable (ou mode continu) (mode 01, 02, 03 et 04)
- Le mode Monostable (ou mode monocoup) (mode 05, 06, 07 et 08)
- Le mode comparaison de fréquence (mode 09 et 10)
- Le mode comparaison de largeur d'impulsion (mode 11 et 12)

-33-

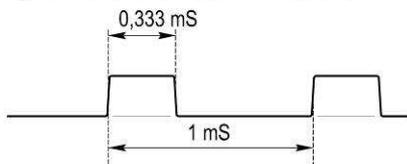
Mode de Fonctionnement	CRx5	CRx4	CRx3	CRx2	Format	Initialisation compteur	Signal en broche de sortie Ox		
<b>Astable</b> (mode continu)	0	0	0	0	16 bits	01 $\overline{Gx} \downarrow$ ou W ou R			
		1				02 $\overline{Gx} \downarrow$ ou R			
		0		1	2 x 8 bits	03 $\overline{Gx} \downarrow$ ou W ou R			
		1				04 $\overline{Gx} \downarrow$ ou R			
<b>Monostable</b> (mode monocoup)	1	0	0	0	16 bits	05 $\overline{Gx} \downarrow$ ou W ou R			
		1				06 $\overline{Gx} \downarrow$ ou R			
		0		1	2 x 8 bits	07 $\overline{Gx} \downarrow$ ou W ou R			
		1				08 $\overline{Gx} \downarrow$ ou R			
<b>Comparateur de Fréquences</b> (Fréquence-mètre)	0	0	1	0	16 bits	09		Une interruption est engendrée si la période de la broche $\overline{Gx}$ est < au TO du compteur	
				1	2 x 8 bits	10			Mesure de durée plus GRANDE que le Time Out
				1	0	16 bits	10		Une interruption est engendrée si la période de la broche $\overline{Gx}$ est > au TO du compteur
					1	2 x 8 bits	10		
<b>Comparateur de largeurs d'impulsions</b>	0	1	1	0	16 bits	11		Une interruption est générée si la durée de l'état Bas sur la broche $\overline{Gx}$ est < au TO (Time Out) du compteur	
				1	2 x 8 bits	11			
	1			0	16 bits	12		Une interruption est générée si la durée de l'état Bas sur la broche $\overline{Gx}$ est > au TO (Time Out) du compteur	
				1	2 x 8 bits	12			

$\overline{G}$	Font descendant sur la broche d'entrée $\overline{Gx}$ ( $\overline{GATE}$ )
W	Commande d'écriture des registres tampon MSB puis LSB
R	Reset du Timer ( CR10=1 ou broche $\overline{RESET}$ niv. Bas )
N	Donnée sur 16 bits contenue dans les registres tampons du compteur
L	Donnée sur 8 bits contenue dans le registre tampon LSB du compteur
M	Donnée sur 8 bits contenue dans le registre tampon MSB du compteur
T	Front descendant de l'horloge appliquée au compteur
to	Initialisation du compteur
TO	Time Out : Temps de décrémentation des compteurs

## 6840 : Exemples de Programmation

### 6840 : Exemple de Programmation Mode Astable

#### Exemple Multivibrateur Astable



On souhaite obtenir un signal dont la forme est la suivante (avec une période de 1 mS)

Le 6840 travaille à 1 Mhz

On prend le temporisateur n° 3 afin d'utiliser le diviseur d'horloge.

$$(L + 1) (M + 1) \times T = 1 \text{ mS}$$

$$L \times T = 333 \mu\text{S}$$

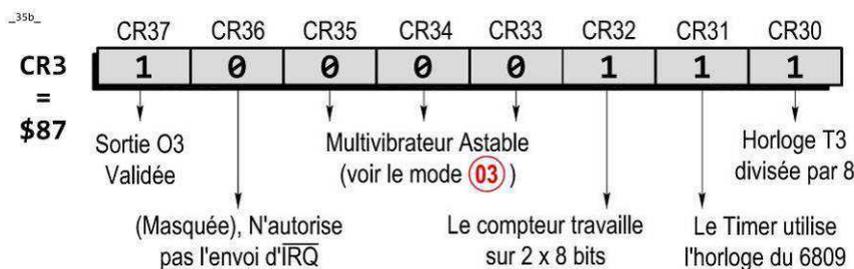
On utilise donc l'horloge interne divisée par 8 d'où  $T = 8 \mu\text{S}$

$$L \times T = 333 \mu\text{S} \quad \text{donc} \quad L = \frac{333 \mu\text{S}}{8 \mu\text{S}} = 41,625 \approx 42 \quad \mathbf{L = 42 = \$2A}$$

$$\text{d'où } M = \frac{1 \text{ mS}}{(L + 1) \times T} - 1 = \frac{1000 \mu\text{S}}{(42 + 1) \times 8} - 1 = 1,90698 \approx 2 \quad \mathbf{M = 2 = \$02}$$

La logique de décodage du 6840 donne comme première adresse \$CFF8

La valeur du registre CR3 est de \$87 = %1000 0111



On obtient le programme suivant :

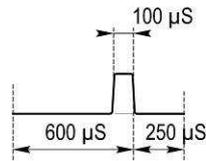
```

LDA  #$01    ;Accès à CR1
STA  ADRCR2  ;
CLRA                ;init logicielle de tous les Timers
STA  ADRCR1  ;
STA  ADRCR2  ;accès à CR3
LDA  #$87    ;init Timer 3
STA  ADRCR3  ;
LDX  #$022A ;M=$02 L=$2A init registre tampon 3
STX  TEMP3   ;

```

## 6840 : Exemple de programmation Mode Monostable

### Exemple En Monostable



Soit à programmer le temporisateur n°1 en mode Monostable afin d'obtenir une impulsion ci-dessous.

On utilise l'horloge interne du 6840 à 1 Mhz  
Le compteur n°1 fonctionnera sur 2 x 8 bits .

$$(L + 1) (M + 1) \times T = 600 \mu\text{s}$$

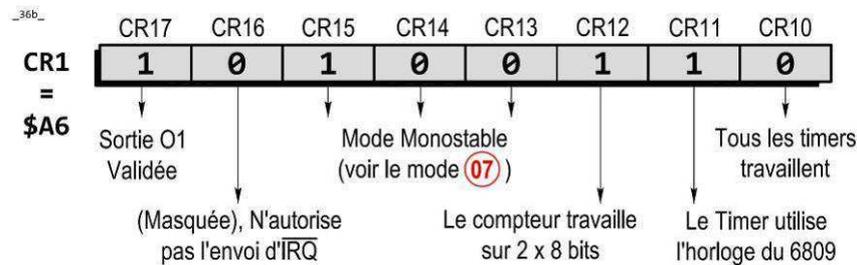
$$L \times T = 100 \mu\text{s}$$

On utilise donc l'horloge interne d'où  $T = 1 \mu\text{s}$

$$L \times T = 100 \mu\text{s} \quad \text{donc} \quad L = \frac{100 \mu\text{s}}{1 \mu\text{s}} = 100 \quad \mathbf{L = 100 = \$64}$$

$$\text{d'où } M = \frac{1 \text{ mS}}{(L + 1) \times T} - 1 = \frac{600 \mu\text{s}}{(100 + 1) \times 1} - 1 = 4,94059 \approx 5 \quad \mathbf{M = 5 = \$05}$$

La logique de décodage du 6840 donne comme première adresse \$8000  
La valeur du registre CR1 est de \$A6 = %1010 0110



On obtient le programme suivant :

```
LDA  #$01    ;Accès à CR1
STA  ADRCR2  ;
LDA  #$A6    ;init Timer 1
STA  ADRCR1  ;
LDX  #$0564  ;M=$05 L=$64 init registre tampon 1
STX  TEMP1   ;
```

## 6840 : Exemple de programmation Mode Comparaison de Fréquence

\_37a\_

### Exemple En Comparaison de Fréquence

On désire mesurer une fréquence 60 kHz < Fx < 200 kHz. On utilise le temporisateur n°1 activé avec l'horloge du 6840 à 1 MHz, une interruption sera envoyée au 6809.

On charge le registre tampon T1 de manière à ce que TO > T

Le compteur fonctionne sur 2 x 8 bits, on prend  $TO > \frac{1}{F_{x \text{ min}}}$  (TO = 20 µS)

On a (M + 1) (L + 1) x T = 20 µS

Comme l'horloge est à 1MHz ⇒ T = 1 µS

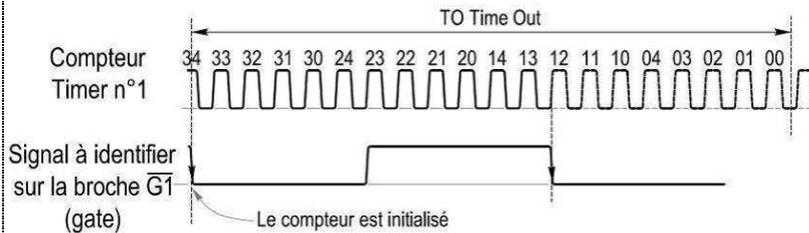
On se donne L x T = 4 µS donc  $L = \frac{4 \mu S}{T} = \frac{4 \mu S}{1 \mu S} = 4$  L = 4 = \$04

(M + 1) (L + 1) x T = 20 µS

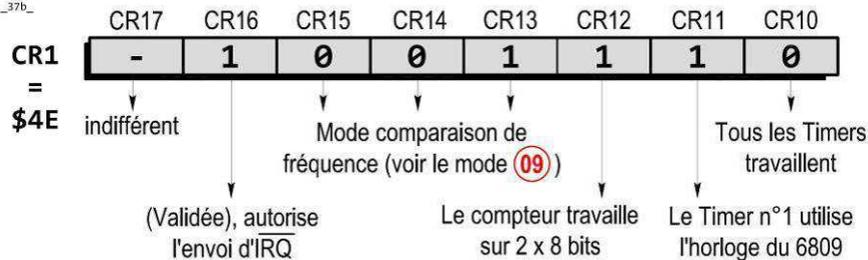
$M = \frac{20 \mu S}{(L + 1) x T} - 1 = \frac{20 \mu S}{(4 + 1) x 1} - 1 = 3$  M = 3 = \$03

Un front descendant sur  $\overline{G1}$  entraîne l'initialisation du compteur.

Le fonctionnement sera alors le suivant :



\_37b\_

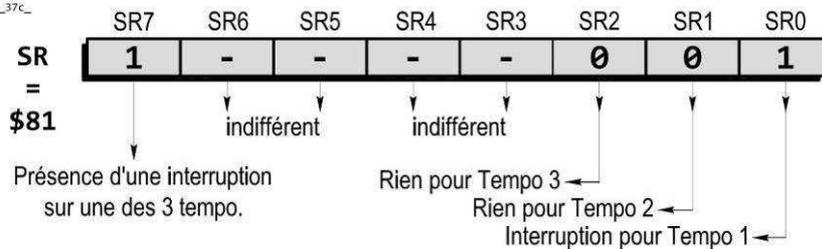


### On obtient le Programme suivant :

```
LDA #$01 ;Accès à CR1
STA ADRCR2 ;
LDA #$4E ;init CR1
STA ADRCR1 ;
LDX #$0304 ;M=$03 L=$04 init registre tampon 1
STX TEMP1 ;
```

Lorsque le compteur se bloque sur un second front descendant de G1| le registre d'état SR est positionné, son contenu est alors le suivant :

\_37c\_



Le 6809 est interrompu, le programme d'interruption qui suit consiste à lire le compteur et à calculer la fréquence Fx. Au départ on avait M = 3 et L = 4. Le compteur donne M = 1 et L = 3

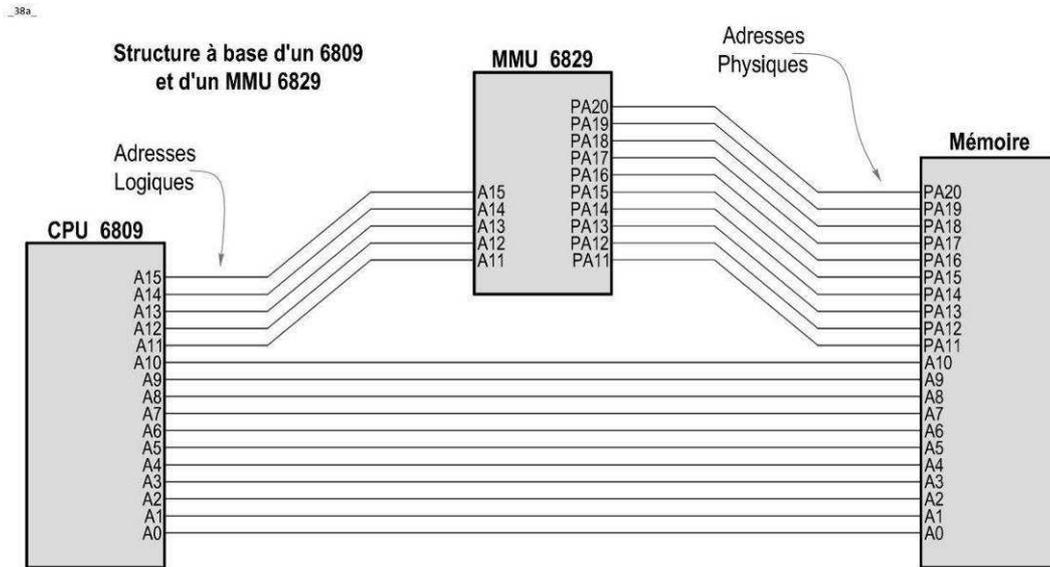
Ce qui fait  $T_x = (M + 1) (L + 1) x T = (1 + 1) (3 + 1) x 1 \mu S = 8 \mu S$

D'où la fréquence F (Hz) = 1 / Période (S) = 1 / 0,000 008 = 125 000 Hz Fx = 125 KHz

Le registre d'état est réinitialisé automatiquement par la lecture du compteur qui suit celle de ce même registre d'état.

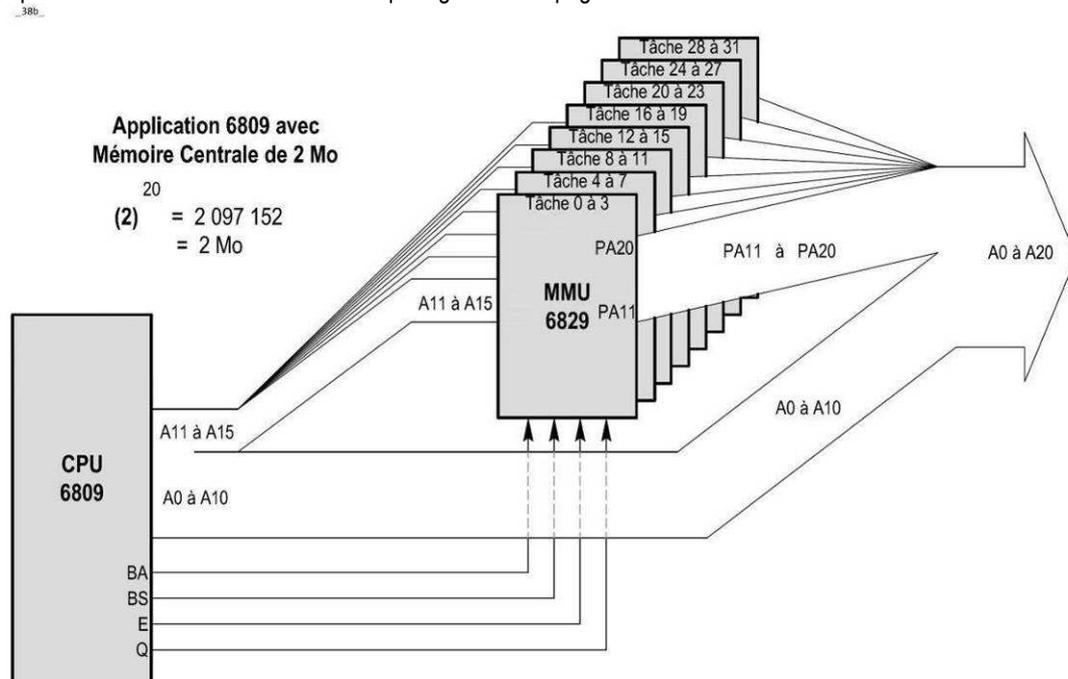
## 6829 : Généralités MMU (Management Memory Unit) Interface d'Extension Mémoire

Ce circuit permet d'étendre l'espace mémoire du 6809 de 64 Ko à 2 Mo.  
Le principe est de connecter celui-ci entre le bus d'adresse du 6809 et la mémoire.



## 6829 : Principe

Le principe de fonctionnement du 6829 est de traduire les adresses Logiques (issues du 6809) en adresses Physiques commandant la mémoire.  
Chaque MMU 6829 peut générer 4 tâches séparées de 64 Ko chacune.  
La capacité d'adressage maximum est de 2 Mo obtenue en connectant 8 circuits MMU 6829 en parallèle.  
Les broches A0 à A10 déterminent l'accès à des pages de 2 Ko.  
Les broches A11 à A15 permettent de sélectionner une des 32 pages d'une tâche sélectionnée au préalable.  
La combinaison de la tâche choisie en fonction du numéro de la page permet d'obtenir les poids forts A11 à A20 de l'adresse Physique.  
Chaque tâche peut être isolée et protégée en écriture.  
L'espace mémoire total du 6809 est donc partagé en 1024 pages de 2 Ko.



## 6829 : Utilisation

Dans le cas où l'utilisateur a besoin d'une mémoire centrale de 2 Mo, il est possible de connecter 8 circuits 6829 en parallèle.  
La tâche 0 du MMU 6829 n°1 est réservée au système d'exploitation.  
La tâche 1 du MMU 6829 n°1 est réservée aux accès directs à la mémoire.  
Les tâches 2 à 31 sont disponibles pour l'utilisateur.

### ANN : Circuits d'Interfaces de la famille 6800 et 6809

La plupart des circuits d'interface du 6800 sont compatibles avec le 6809

Origine	Référence	Désignations	
6800	6810	RAM	128 Ko 8 bits
"	6830	ROM	1024 Ko 8 bits
"	<a href="#">6821</a>	<a href="#">PIA</a>	Interface parallèle programmable (Peripheral Interface Adaptor)
"	6828	PIC	Contrôleur de priorité d'interruption (Priority Interrupt Controler)
"	<a href="#">6829</a>	MMU	Interface de gestion mémoire Permet d'aborder un espace mémoire de 2 Mo
"	6830	ROM	1 Ko par 8 bits
"	6839	ROM	Mathématique
"	<a href="#">6840</a>	<a href="#">PTM</a>	3 temporisateurs programmable
"	6843	FDC	Contrôleur de disque souple simple densité
"	6844	DMAC	Contrôleur d'accès mémoire
"	6845	CRTC	Contrôleur de visualisation
"	6846	ROM	Mémoire ROM 2 Ko, port parallèle 8 bits avec Temporisateur 16 bits
"	6847	CRTC	Color Video Display Generator
"	<a href="#">6850</a>	<a href="#">ACIA</a>	Interface série <b>asynchrone</b> (RS 232)
"	6852	SSDA	Interface série <b>synchrone</b>
"	6854	ADLC	Contrôleur de transmission avec protocole
"	6855	DMA	Contrôleur d'accès mémoire (pas introduit sur le marché)
"	68488	GPIA	Interface IEEE-488
"	9365	GPIA	Contrôleur d'écran graphique 512 x 512 (entrelacé) THOMSON
"	9366	GPIA	Contrôleur d'écran graphique 512 x 256 (non entrelacé)
6809	<a href="#">6829</a>	<a href="#">MMU</a>	Interface d'extension mémoire
"	6839	ROM	Mémoire ROM mathématique
	6883		Synchronous Adress Multiplexer ( idem que 74 LS 783 et 74 LS 785 )

## ANN : Table ASCII Description étendue de l'usage des caractères de contrôle (caractères 0 à 31)

\$00	000	<b>NUL</b> (NULL) : caractère nul Typiquement (et spécialement en PureBasic) utilisé pour indiquer la fin d'une chaîne. Originellement une NOP, un caractère à ignorer. Lui donner le code 0 permettait de prévoir des réserves sur les bandes perforées en laissant des zones sans perforation pour insérer de nouveaux caractères a posteriori. Avec le développement du langage C il a pris une importance particulière quand il a été utilisé comme indicateur de fin de chaîne de caractères.
\$01	001	<b>SOH</b> (Start Of Heading) : début de titre ou début d'en-tête Indique le début d'un bloc de données, ou la zone d'en-tête d'un bloc de données. Il est aujourd'hui souvent utilisé dans les communications séries pour permettre la synchronisation après erreur <sup>14</sup> .
\$02	002	<b>STX</b> (Start of TeXt) : début de texte Typiquement envoyé comme premier caractère dans un bloc de texte, pendant les communications.
\$03	003	<b>ETX</b> (End of TeXt) : fin de texte Typiquement envoyé comme dernier caractère dans un bloc de texte, pendant les communications.
\$04	004	<b>EOT</b> (End Of Transmission) : fin de transmission Utilisé pour indiquer la fin d'une transmission.
\$05	005	<b>ENQ</b> (ENQuiry) : requête - invitation à la transmission Envoyé à un récepteur afin d'obtenir une réponse.
\$06	006	<b>ACK</b> (ACKnowledge) : accusé de réception Envoyé par un récepteur pour indiquer qu'il a reçu et/ou compris la requête.
\$07	007	<b>BEL</b> (BELL) : cloche Produit un signal sonore (provoque l'émission d'un 'bip' par le haut-parleur du PC)
\$08	008	<b>BS</b> (BackSpace) : retour arrière Déplace le curseur d'une position vers la gauche (pourrait également effacer le caractère à gauche du curseur avant d'effectuer le mouvement)
\$09	009	<b>HT</b> (Horizontal Tab) : tabulation horizontale Typiquement utilisé pour la mise en forme de tableaux dans un texte.
\$0A	010	<b>LF</b> (LineFeed) : saut de ligne Le caractère utilisé pour représenter l'action de passer une ligne sur une machine à écrire ou une imprimante en mode texte. Typiquement utilisé comme, ou partie des, caractères de fin de ligne.
\$0B	011	<b>VT</b> (Vertical Tab) : tabulation verticale Même chose que la tabulation (horizontale), mais le déplacement s'effectue d'une rangée vers le bas au lieu d'une colonne vers la droite.
\$0C	012	<b>FF</b> (Form Feed) : saut de page Caractère typiquement utilisé pour indiquer à une imprimante (en mode texte) de passer à la page (feuille) suivante.
\$0D	013	<b>CR</b> (Carriage Return) : retour chariot Le caractère qui représente l'action de ramener la tête d'une machine à écrire ou d'une imprimante au début de la ligne. Typiquement utilisé comme, ou partie des, caractères de fin de ligne.
\$0E	014	<b>SO</b> (Shift Out) : mouvement sortant Début d'un bloc de caractères dont la signification dépend de l'implémentation.

\$0F	015	<b>SI</b> (Shift In) : mouvement entrant Ferme la transmission du type de bloc ci-dessus.
\$10	016	<b>DLE</b> (Data Link Escape) : échappement de lien de donnée Utilisé pour indiquer que le caractère de contrôle suivant devrait être interprété comme donnée et non comme caractère de contrôle.
\$11	017	<b>DC1</b> (Device Control 1) : contrôle de périphérique 1 Typiquement utilisé pour activer une partie d'un équipement. L'usage le plus courant aujourd'hui est en tant que caractère XON dans les communications série à contrôle de flux logiciel.
\$12	018	<b>DC2</b> (Device Control 2) : contrôle de périphérique 2 Un autre caractère de contrôle de périphérique. Son usage dépend du contexte.
\$13	019	<b>DC3</b> (Device Control 3) : contrôle de périphérique 3 Typiquement utilisé pour désactiver une partie d'un équipement. L'usage le plus courant aujourd'hui est en tant que caractère XOFF dans les communications série à contrôle de flux logiciel.
\$14	020	<b>DC4</b> (Device Control 4) : contrôle de périphérique 4 Un autre caractère de contrôle de périphérique.
\$15	021	<b>NAK</b> (Negative Acknowledge) : accusé de réception négatif Typiquement utilisé pour signaler des données non-reçues ou non-comprises (erronée).
\$16	022	<b>SYN</b> (SYNchronous idle) : attente synchronisée Comme son nom l'indique, il s'agit d'un signal envoyé à intervalle régulier pour indiquer que le canal de communication est en attente, mais toujours actif.
\$17	023	<b>ETB</b> (End of Transmission Block) : fin de transmission de bloc Utilisé pour contrôler la transmission de donnée en indiquant la fin de bloc. A ne pas confondre avec EOT.
\$18	024	<b>CAN</b> (CANcel) : annulation Signifie généralement que la donnée envoyée précédemment devrait être ignorée, bien que les détails dépendent de l'application.
\$19	025	<b>EM</b> (End of Medium) : fin de média Utilisé pour indiquer la fin d'un média, par exemple la fin d'un lecteur de bande
\$1A	026	<b>SUB</b> (SUBstitute) : substitution Un caractère utilisé pour indiquer qu'un caractère a été substitué.
\$1B	027	<b>ESC</b> (ESCape) : échappement Le caractère produit habituellement en appuyant sur la touche 'ECHAP' de votre clavier, utilisé dans les "séquences d'échappement" pour fournir des informations de formatage aux afficheurs de texte (consoles, imprimantes, etc..)
\$1C	028	<b>FS</b> (File Separator) : séparateur de fichier
\$1D	029	<b>GS</b> (Group Separator) : séparateur de groupe
\$1E	030	<b>RS</b> (Record separator) : séparateur d'enregistrement
\$1F	031	<b>US</b> (Unit separator) : séparateur d'unité
\$7F	127	<b>DEL</b> (Delete) : effacement. Lui donner le code 127 (1111111 en binaire) permettait de supprimer a posteriori un caractère sur les bandes perforées qui codaient les informations sur 7 bits. N'importe quel caractère pouvait être transformé en DEL en complétant la perforation des 7 bits qui le composaient.

**Table ASCII ( 0 - 127 )**

Déci	Hexa	Octal		Déci	Hexa	Octal		Déci	Hexa	Octal		Déci	Hexa	Octal	
000	\$00	000	<b>NUL</b>	032	\$20	040	space	064	\$40	100	@	096	\$60	140	`
001	\$01	001	<b>SOH</b>	033	\$21	041	!	065	\$41	101	<b>A</b>	097	\$61	141	<b>a</b>
002	\$02	002	<b>STX</b>	034	\$22	042	"	066	\$42	102	<b>B</b>	098	\$62	142	<b>b</b>
003	\$03	003	<b>ETX</b>	035	\$23	043	#	067	\$43	103	<b>C</b>	099	\$63	143	<b>c</b>
004	\$04	004	<b>EOT</b>	036	\$24	044	\$	068	\$44	104	<b>D</b>	100	\$64	144	<b>d</b>
005	\$05	005	<b>ENQ</b>	037	\$25	045	%	069	\$45	105	<b>E</b>	101	\$65	145	<b>e</b>
006	\$06	006	<b>ACK</b>	038	\$26	046	&	070	\$46	106	<b>F</b>	102	\$66	146	<b>f</b>
007	\$07	007	<b>BEL</b>	039	\$27	047	'	071	\$47	107	<b>G</b>	103	\$67	147	<b>g</b>
008	\$08	010	<b>BS</b>	040	\$28	050	(	072	\$48	110	<b>H</b>	104	\$68	150	<b>h</b>
009	\$09	011	<b>HT</b>	041	\$29	051	)	073	\$49	111	<b>I</b>	105	\$69	151	<b>i</b>
010	\$0A	012	<b>LF</b>	042	\$2A	052	*	074	\$4A	112	<b>J</b>	106	\$6A	152	<b>j</b>
011	\$0B	013	<b>VT</b>	043	\$2B	053	+	075	\$4B	113	<b>K</b>	107	\$6B	153	<b>k</b>
012	\$0C	014	<b>FF</b>	044	\$2C	054	,	076	\$4C	114	<b>L</b>	108	\$6C	154	<b>l</b>
013	\$0D	015	<b>CR</b>	045	\$2D	055	-	077	\$4D	115	<b>M</b>	109	\$6D	155	<b>m</b>
014	\$0E	016	<b>SO</b>	046	\$2E	056	.	078	\$4E	116	<b>N</b>	110	\$6E	156	<b>n</b>
015	\$0F	017	<b>SI</b>	047	\$2F	057	/	079	\$4F	117	<b>O</b>	111	\$6F	157	<b>o</b>
Déci	Hexa	Octal		Déci	Hexa	Octal		Déci	Hexa	Octal		Déci	Hexa	Octal	
016	\$10	020	<b>DLE</b>	048	\$30	060	<b>0</b>	080	\$50	120	<b>P</b>	112	\$70	160	<b>p</b>
017	\$11	021	<b>DC1</b>	049	\$31	061	<b>1</b>	081	\$51	121	<b>Q</b>	113	\$71	161	<b>q</b>
018	\$12	022	<b>DC2</b>	050	\$32	062	<b>2</b>	082	\$52	122	<b>R</b>	114	\$72	162	<b>r</b>
019	\$13	023	<b>DC3</b>	051	\$33	063	<b>3</b>	083	\$53	123	<b>S</b>	115	\$73	163	<b>s</b>
020	\$14	024	<b>DC4</b>	052	\$34	064	<b>4</b>	084	\$54	124	<b>T</b>	116	\$74	164	<b>t</b>
021	\$15	025	<b>NAK</b>	053	\$35	065	<b>5</b>	085	\$55	125	<b>U</b>	117	\$75	165	<b>u</b>
022	\$16	026	<b>SYN</b>	054	\$36	066	<b>6</b>	086	\$56	126	<b>V</b>	118	\$76	166	<b>v</b>
023	\$17	027	<b>ETB</b>	055	\$37	067	<b>7</b>	087	\$57	127	<b>W</b>	119	\$77	167	<b>w</b>
024	\$18	030	<b>CAN</b>	056	\$38	070	<b>8</b>	088	\$58	130	<b>X</b>	120	\$78	170	<b>x</b>
025	\$19	031	<b>EM</b>	057	\$39	071	<b>9</b>	089	\$59	131	<b>Y</b>	121	\$79	171	<b>y</b>
026	\$1A	032	<b>SUB</b>	058	\$3A	072	:	090	\$5A	132	<b>Z</b>	122	\$7A	172	<b>z</b>
027	\$1B	033	<b>ESC</b>	059	\$3B	073	;	091	\$5B	133	[	123	\$7B	173	{
028	\$1C	034	<b>FS</b>	060	\$3C	074	<	092	\$5C	134	\	124	\$7C	174	
029	\$1D	035	<b>GS</b>	061	\$3D	075	=	093	\$5D	135	]	125	\$7D	175	}
030	\$1E	036	<b>RS</b>	062	\$3E	076	>	094	\$5E	136	^	126	\$7E	176	~
031	\$1F	037	<b>US</b>	063	\$3F	077	?	095	\$5F	137	_	127	\$7F	177	<b>DEL</b>

## Table ASCII ( 128 - 255 )

Déci	Hexa	Octal	
128	\$80	200	Ç
129	\$81	201	ü
130	\$82	202	é
131	\$83	203	â
132	\$84	204	ä
133	\$85	205	à
134	\$86	206	á
135	\$87	207	ç
136	\$88	210	ê
137	\$89	211	ë
138	\$8A	212	è
139	\$8B	213	ï
140	\$8C	214	í
141	\$8D	215	ì
142	\$8E	216	Ä
143	\$8F	217	Å

Déci	Hexa	Octal	
160	\$A0	240	á
161	\$A1	241	í
162	\$A2	242	ó
163	\$A3	243	ú
164	\$A4	244	ñ
165	\$A5	245	Ñ
166	\$A6	246	ª
167	\$A7	247	º
168	\$A8	250	¿
169	\$A9	251	¬
170	\$AA	252	¬
171	\$AB	253	½
172	\$AC	254	¼
173	\$AD	255	¡
174	\$AE	256	«
175	\$AF	257	»

Déci	Hexa	Octal	
192	\$C0	300	Ł
193	\$C1	301	ł
194	\$C2	302	ł
195	\$C3	303	ł
196	\$C4	304	ł
197	\$C5	305	ł
198	\$C6	306	ł
199	\$C7	307	ł
200	\$C8	310	ł
201	\$C9	311	ł
202	\$CA	312	ł
203	\$CB	313	ł
204	\$CC	314	ł
205	\$CD	315	=
206	\$CE	316	ł
207	\$CF	317	ł

Déci	Hexa	Octal	
224	\$E0	340	α
225	\$E1	341	β
226	\$E2	342	Γ
227	\$E3	343	π
228	\$E4	344	Σ
229	\$E5	345	σ
230	\$E6	346	μ
231	\$E7	347	τ
232	\$E8	350	Φ
233	\$E9	351	Θ
234	\$EA	352	Ω
235	\$EB	353	δ
236	\$EC	354	∞
237	\$ED	355	φ
238	\$EE	356	ε
239	\$EF	357	∩

Déci	Hexa	Octal	
144	\$90	220	É
145	\$91	221	æ
146	\$92	222	Æ
147	\$93	223	ô
148	\$94	224	ö
149	\$95	225	ò
150	\$96	226	ù
151	\$97	227	û
152	\$98	230	ÿ
153	\$99	231	Ö
154	\$9A	232	Ü
155	\$9B	233	ç
156	\$9C	234	£
157	\$9D	235	¥
158	\$9E	236	Pts
159	\$9F	237	f

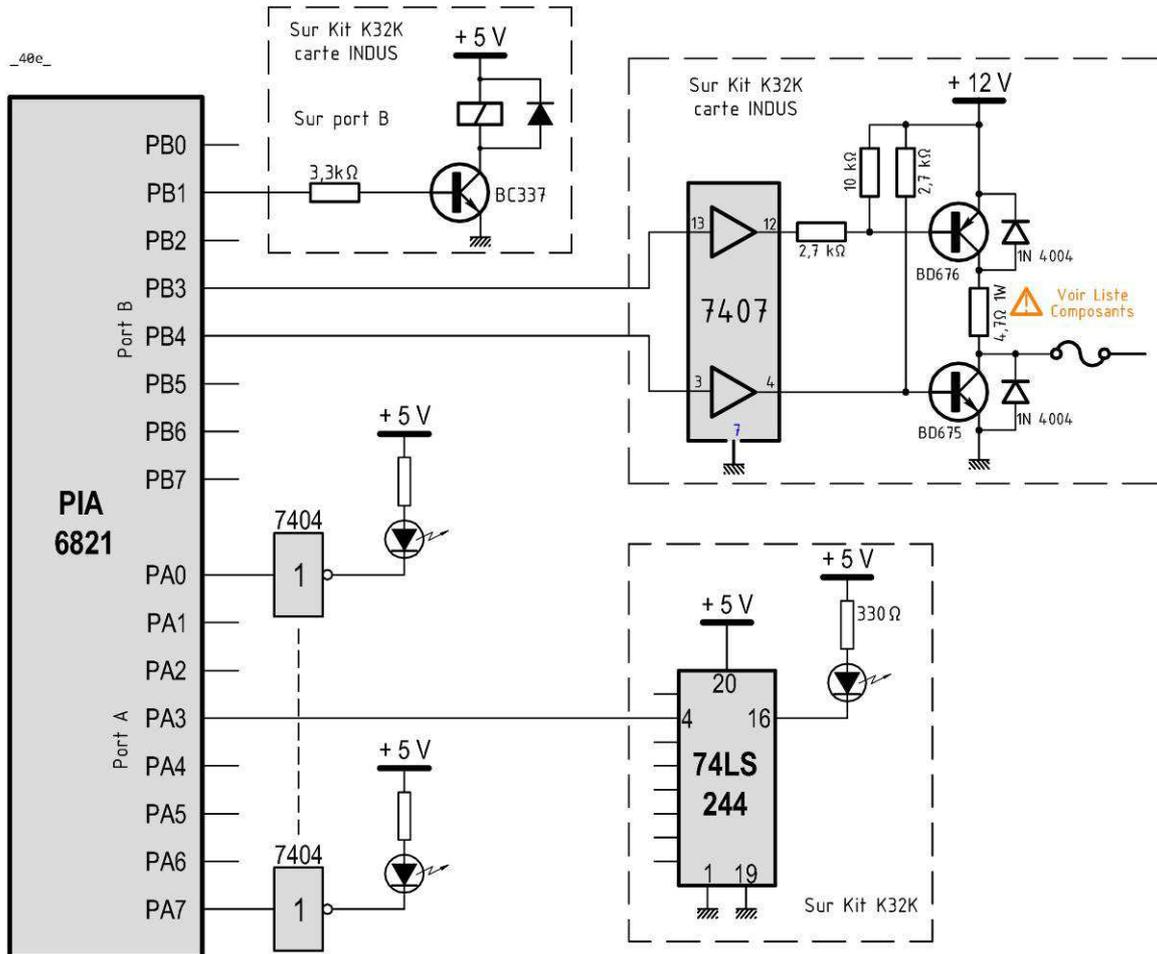
Déci	Hexa	Octal	
176	\$B0	260	⋮
177	\$B1	261	⋮
178	\$B2	262	⋮
179	\$B3	263	
180	\$B4	264	└
181	\$B5	265	┘
182	\$B6	266	┌
183	\$B7	267	┐
184	\$B8	270	┘
185	\$B9	271	┘
186	\$BA	272	
187	\$BB	273	┘
188	\$BC	274	┘
189	\$BD	275	┘
190	\$BE	276	┘
191	\$BF	277	┘

Déci	Hexa	Octal	
208	\$D0	320	ł
209	\$D1	321	ł
210	\$D2	322	ł
211	\$D3	323	ł
212	\$D4	324	Ō
213	\$D5	325	ł
214	\$D6	326	ł
215	\$D7	327	ł
216	\$D8	330	ł
217	\$D9	331	ł
218	\$DA	332	ł
219	\$DB	333	■
220	\$DC	334	■
221	\$DD	335	■
222	\$DE	336	■
223	\$DF	337	■

Déci	Hexa	Octal	
240	\$F0	360	≡
241	\$F1	361	±
242	\$F2	362	≥
243	\$F3	363	≤
244	\$F4	364	
245	\$F5	365	
246	\$F6	366	÷
247	\$F7	367	≈
248	\$F8	370	≈
249	\$F9	371	·
250	\$FA	372	·
251	\$FB	373	√
252	\$FC	374	ⁿ
253	\$FD	375	²
254	\$FE	376	■
255	\$FF	377	

## PVM : Interfaçage des Afficheurs

### PVM : AFF : Diode LED



La commande d'une diode LED se fait par un port en sortie d'un PIA.

Un inverseur du type 7404 est nécessaire pour absorber le courant de la LED, limité par une résistance.

La LED est allumée si  $Pax = 1$



## LR : LIENS RAPIDES

<b>6821 : LE 6821 PIA</b> .....	007
6821 : Vue complète du registre CRA ou CRB .....	008
6821 : Registres DDRA et DDRB .....	016
6821 : Registres ORA et ORB .....	016
6821 : Sélection des registres internes .....	020
<b>6850 : LE 6850 ACIA</b> .....	031
6850 : Sélection des Registres Internes .....	034
6850 : Registre CR .....	036
6850 : Registre SR .....	037
<b>6840 : LE 6840 TIMER</b> .....	059
6840 : Adressage, sélection du boîtier .....	062
6840 : Registres CR .....	064
6840 : Registre SR .....	065
6840 : Tableau regroupant tous les modes de fonctionnement .....	074
<b>ANNEXES</b> .....	079
ANN : Table ASCII caractères de contrôle (caractères 0 à 31) .....	080
ANN : Table ASCII ( 0 - 127 ) .....	081
ANN : Table ASCII ( 128 - 255 ) .....	082