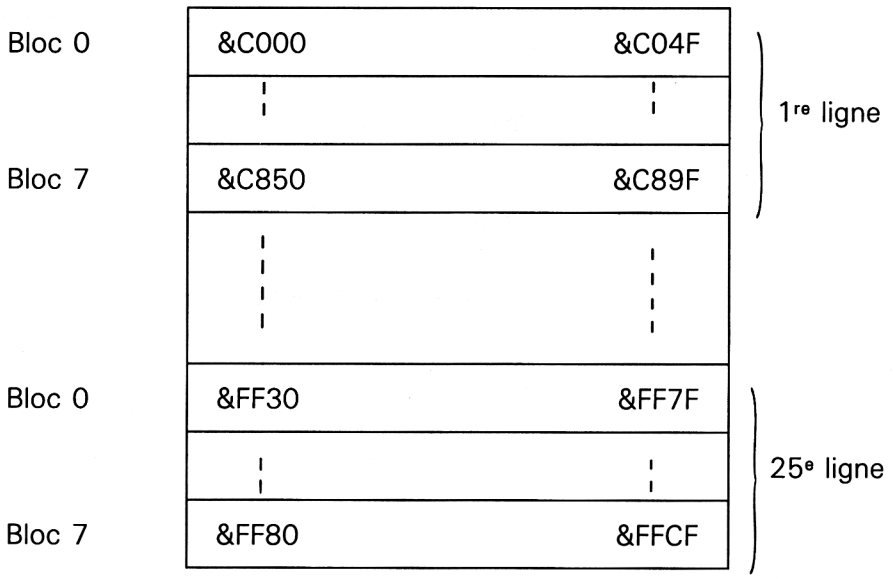
5/7

La mémoire d'écran

Une autre manière d'utiliser l'écran graphique est d'écrire dans la mémoire d'écran.

L'écran peut être considéré comme une mémoire RAM implantée entre les adresses &C000 et &FFFF, et donc, directement adressable par le micro-processeur. Quel que soit le mode d'affichage, la mémoire d'écran est divisée en huit blocs. Une bloc représente une ligne élémentaire (d'épaisseur un point élémentaire, et de largeur 80 octets).

Les blocs sont répartis comme suit :



Le bloc i (i compris entre 0 et 7) représente la ième ligne élémentaire de chacune des 25 lignes de l'écran, et fait donc 80 × 25 = 2000 octets. Pour des raisons de commodité de manipulation, un bloc a été défini sur 2^{12} octets (2048 octets), et les 48 octets supplémentaires de chaque ligne sont inutilisés.

Comme nous l'avons vu plus haut, la dimension en pixels d'un point élémentaire dépend du mode de résolution.

Dans le MODE 0, un octet comprend 2 points élémentaires de 4 pixels de large chacun.

Les octets sont codés comme suit :

Bit	0	1	2	3	4	5	6	7
Stylo	8	8	2	2	4	4	1	1
Point élémentaire	0	1	0	1	0	1	0	1

Ce tableau est à interpréter de la façon suivante :

— Si, par exemple, le bit 2 est à 1, le point élémentaire 0 sera allumé avec la couleur de stylo 2.

— De même, pour avoir le point 1 allumé avec la couleur 6, il faudra positionner à 1 les bits 3 et 5.

En MODE 1, un octet comprend 4 points élémentaires de 2 pixels de large chacun.

Les octets sont codés comme suit :

Bit	0	1	2	3	4	5	6	7
Stylo	2	2	2	2	1	1	1	1
Point élémentaire	0	1	2	3	0	1	2	3

Enfin, *en MODE 2*, un octet comprend 8 points élémentaires d'1 pixel de large chacun.

Les octets sont codés comme suit :

Bit	0	1	2	3	4	5	6	7
Stylo	1	1	1	1	1	1	1	1
Point élémentaire	0	1	2	3	4	5	6	7

En BASIC, on pourra afficher un ou plusieurs points sur l'écran par l'instruction **POKE <Adresse écran>, <Valeur sur 8 bits>.**

En assembleur, il faudra faire :

```
LD    A, <Valeur>
LD    (Adresse ecran),A
```