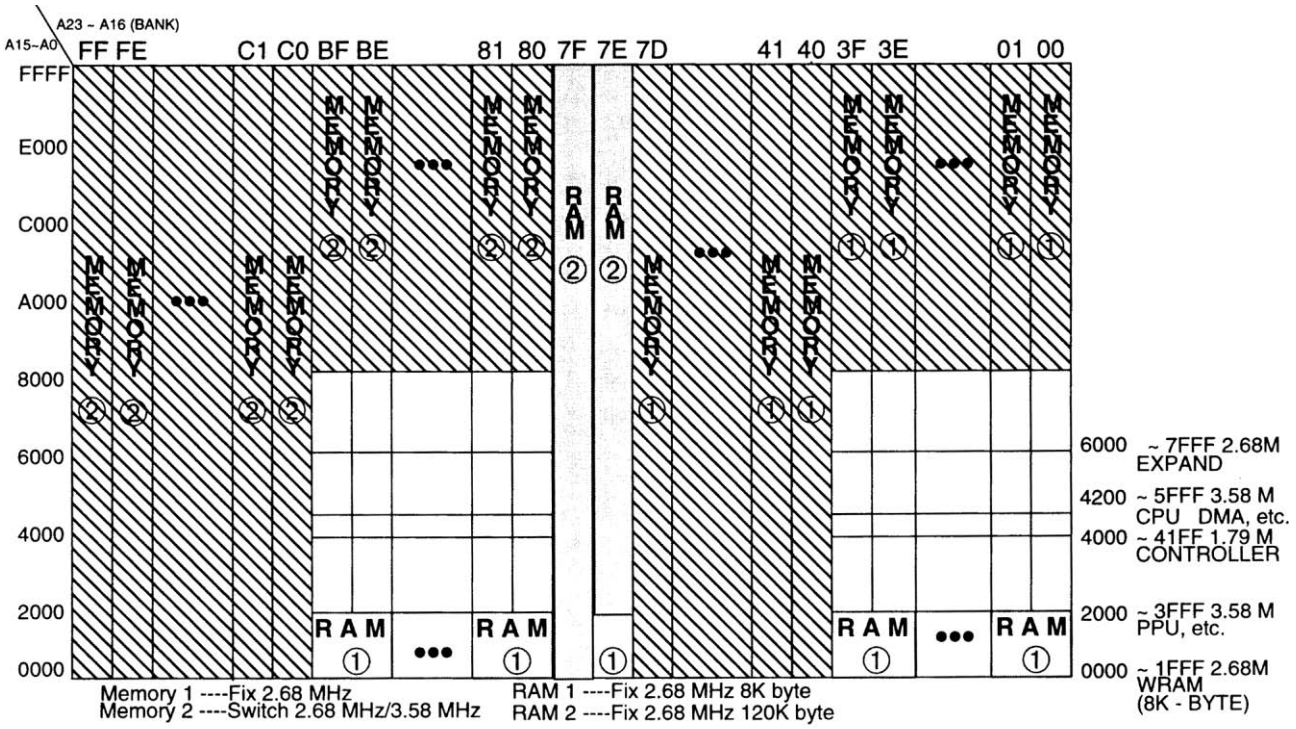
Quelques considérations sur la Vram

Pour comprendre ce qui suit, on gagnera à se reporter fréquemment aux deux schémas qui illustrent cet article : "Snes CPU Memory Map" et ″Super-Oric Memory Map".



SNES CPU Memory Map

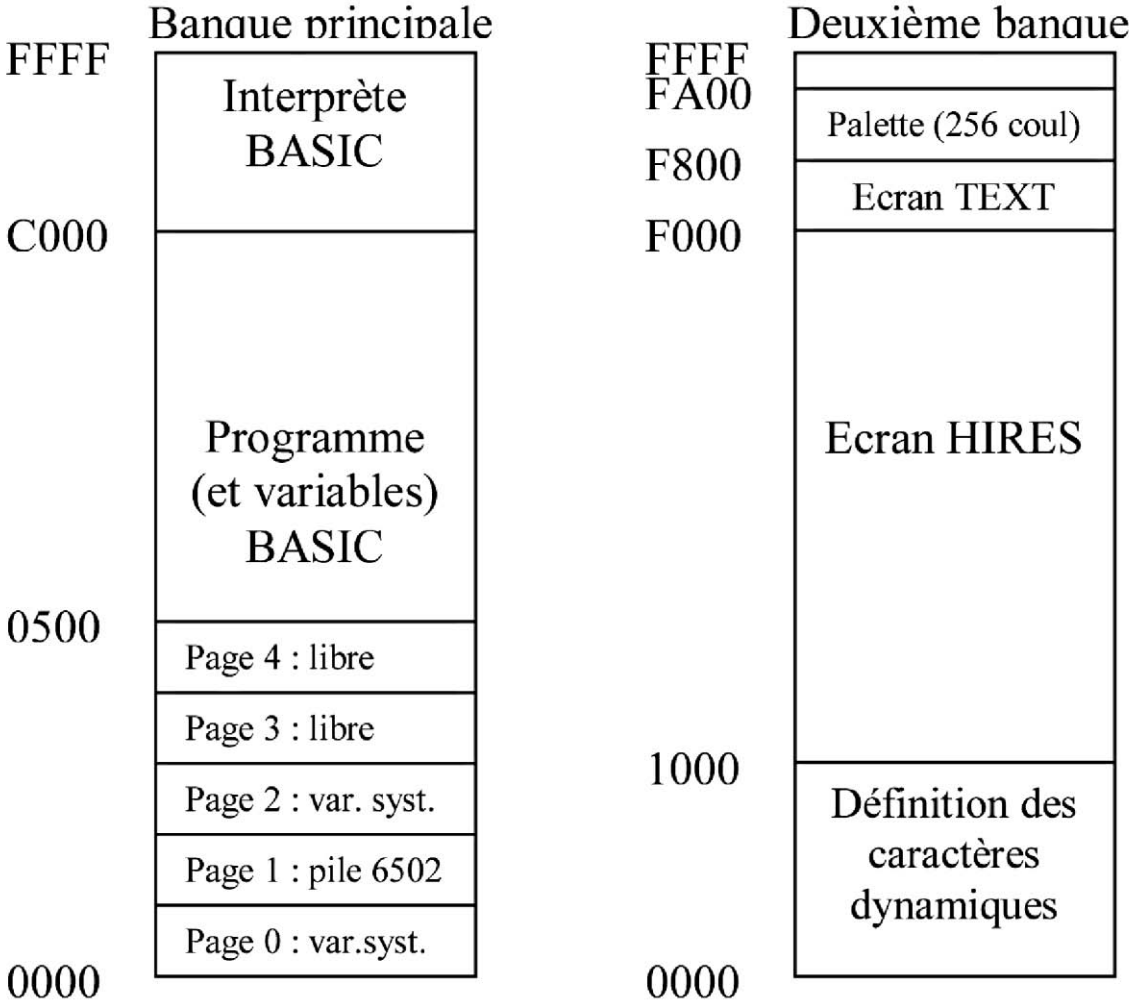
On peut distinguer 4 zones de Ram dans la Snes:

1) La totalité de la banque $7F (banque secondaire du Super-Oric), soit 64 ko (de $0000 à $FFFF).

2) La plus grande partie de la banque $7E (banque principale du Super-Oric), soit 56 ko (de $2000 à $FFFF).

3) Le petit reste de la banque $7E (banque principale du Super-Oric), soit 8 ko (de $0000 à $1FFF). Deux remarques: a) Cette petite zone est répliquée aux mêmes adresses dans les banques $00 à $3F et $80 à $BF. b) Dans le Super-Oric, nous l'utilisons pour y mettre les pages 0, 1, 2, 3 et 4, puis le début de la zone des programmes Basic. En passant, je me demande si cette "discontinuité" de la banque $7E entre la partie basse (de $0000 à $1FFF) et la partie haute (de $2000 à $FFFF) n'est pas responsable de "bug CSAVE" (qui marche bien avec les petits programmes, mais plante avec les programmes important). Même si ces deux morceaux sont probablement situés physiquement sur la même puce, ils font l'objet d'une différence de traitement. A revoir...

4) Une Ram système, utilisée par la Snes pour ses entrées/sorties, communications entre CPU, PPU, Vram etc. Cette zone occupe 24 ko (de $2000 à $7FFF) et est répliquée aux mêmes adresses dans les banques $00 à $3F et $80 à $BF, mais pas 7E, à la différence de la zone "3". On ne peut guère y récupérer de la place sans courir de risque, sauf peut-être dans la partie située juste sous la barre des $7FFF. En effet, je n'ai pas trouvé trace d'utilisation de cette partie de la Ram par le système, contrairement, par exemple à la zone $21XX (registres du PPU) ou à la zone $42XX (registres du CPU).



Banque Principale = Banque Snes $7E ***Banque secondaire = Banque Snes $7F***

Super-Oric Memory Map

Le Super-Oric squatte la plus grande partie de la Ram de la Snes (voir la figure précédente). Il faut garder à l'esprit que les premiers 8 ko de la banque principale (de $0000 à $1FFF), qui accueillent les pages 0, 1, 2, 3, 4 et le début de la zone ″Programme et variables Basic″ sont répliqués dans les banques $00 à $3F et $80 à $BF. La seule partie de la Ram de la Snes qui ne soit pas directement utilisée par le Super-Oric est la Ram système (de $2000à $7FFF, zone répliquée dans les banques $00 à $3F et $80 à $BF). D'autre part, la vitesse d'accès à l'ensemble de la Ram (zone sans hachures dans la première figure) est de 2,68 MHz, cela inclus donc la zone $C000 à $FFFF de l'interprète Basic. Quant à la Rom (zone avec hachures dans la première figure), la vitesse d'accès à celle des banques $00 à $7D est aussi de 2,68 MHz, alors que la vitesse d'accès à celles des banques $80 à $FF est plus importante : 3,58 MHz. La mémoire flash de la cartouche Super-Oric est vue dans les banques $00 à $03, banques qui sont recopiées par le système dans les banques $80 à $83, d'accès plus rapide et qui sont utilisées par le Super-Oric (notamment pour le code 65816), plutôt que la zone d'origine.

Pour la Vram, le "Snes Manuel de Programmation", V4r01a, qui contient de nombreuses erreurs (il faut se méfier) indique (et c'est une connerie) qu'on peut utiliser deux zones:

1) La Vram 8ko, qui est répliquée en banque $00 à $3F, $7E et $80 à $BF. On peut donc la lire à partir de nombreuses banques, de $0000 et $1FFF. Quelle que soit celle de ces banques que l'on adresse, on sera toujours sur les mêmes 8ko de la Vram. Notons que ces 8 Ko ne pourraient pas

suffire pour le Super-Oric (les caractères dynamiques, l'écran Hires et l'écran Texte occupent 62 ko). Mais il y a un problème, car si une seule et unique Ram de 8 ko est répliquée dans les banques $00 à $3F, $7E et $80 à $BF, aux adresses de $0000 à $2000, alors il est impossible d'utiliser cette seule et unique Ram pour de la Vram, car dans la banque $7E, nous y logeons les pages 0, 1, 2, 3 et 4, puis le début de la zone des programmes Basic, données qui sont donc forcément répliquées aux banques $00 à $3F, $7E et $80 à $BF.

2) La Vram 120 ko, qui se trouve de $7E2000 à $7EFFFF, (#E000 octets = 56 ko) et de $7F0000 à $7FFFFF (#10000 octets = 64 ko). Mais dans le cas du Super-Oric, cela ne semble pas possible, car la banque $7E est notre banque principale et $7F notre banque secondaire. Sauf, si par un tour de passe-passe la Ram de la banque secondaire est la même que celle de la Vram (à l'exception des définitions de couleurs, qui doivent être stockées dans la CG-ram, dont j'ignore la localisation). Mais les routines de transfert DMA utiliseraient comme source la banque $7F et comme cible la même banque $7F. Cela ne me semble pas possible.

J'ai tourné ce problème dans tous les sens, en examinant les diverses routines de transfert DMA et j'en suis arrivé aux conclusions :

1) Qu'on ne peut pas savoir où se trouve la Vram.

2) Que cela n'a pas d'importance.

Par contre, tous les registres impliqués dans ces transferts ($21xx, $42xx et $43xx) sont en Ram dans la banque $00 et ses répliques ($00 à $3F et $80 à $BF). Leur adresse est par exemple de la forme $002115 ou $004305.

Tout se passe comme si la Vram n'appartenait pas au domaine de la "Snes CPU Memory Map". Il faut voir les routines de "rafraîchissement de la Vram", seulement comme l'envoi des données présentes en Ram vers le PPU (Picture Processing Unit) (élaboration de l'image), puis vers le CPU (Central Processing Unit) (envoi à l'écran physique durant une période de Blanking). La localisation de la Vram n'a pas d'importance, puisque tout est interfacé par les registres $2118-$2119 (pour écrire les données) $2139-$213A (pour lire les données).

Par contre les routines de transfert DMA requièrent :

1) Une adresse initiale en Vram.
2) Un mode d'incrémentation de cette adresse.
3) Le nombre d'octets à transférer.

Seul le dernier de ces trois paramètres (le nombre d'octets à transférer) est simple. L'explication que donne le "Snes, Manuel de Programmation" à propos du second paramètre (le mode d'incrémentation), est totalement obscur (inexploitable). Enfin, les adresses initiales en Vram, ainsi que les adresses finales (déductibles du nombre d'octets à transférer ?) sont pour le moins problématiques. C'est ce que nous allons voir maintenant.

Dans la Vram on a : Les caractères dynamiques, l'écran Hires et l'écran Texte.

1) Les caractères dynamiques (taille #$1000 octets, soit 4 Ko) sont écrits dans le registre $2118-$2119 et lus dans le registre $2139-$213A, selon l'adresse en $2116-$2117. Selon le code de la routine dma_charset (en $0080DA) la Vram concernée se trouve de $0000 à $0FFF.

2) L'écran Hires (taille #$E000 octets, soit 56 ko) est accédé en écriture dans le registre $2118-$2119 et en lecture dans le registre $2139-$213A, selon l'adresse en $2116-$2117. Selon le code de la routine hires_transfer (en $008168), pour l'écran Hires, la Vram est accédée en écriture de $0800 à $E7FF. Il doit y avoir un bug dans cette routine, car on a un chevauchement avec la zone des caractères dynamiques si ces deux zones sont dans la même Vram. L'adresse initiale devrait, par exemple, être $1000 et non $0800. L'adresse finale serait alors $EFFF et non $E7FF.

3) L'écran Texte (taille #$0800 octets, soit 2 ko) est accédé en écriture dans le registre $2118-$2119 et en lecture dans le registre $2139-$213A, selon l'adresse en $2116-$2117. Selon le code de la routine dma_screen (en $008136), pour l'écran Text, la Vram est accédée en écriture de $7800 à $7FFF. Comme ci-dessus, Il doit y avoir un bug dans cette routine, car on a un chevauchement avec la zone de l'écran Hires si ces deux zones sont dans la même Vram. L'adresse initiale devrait, par exemple, être $F000 et non $7800. L'adresse finale serait alors $F800 et non $7FFF.

Dans la CGram on a : Les palettes de couleurs.

Les définitions des 256 couleurs (2 octets par couleur, soit #$0200 octets) sont écrites et lues dans le registre $002122, selon l'adresse en $2121 (en fait, le n° de couleur). Selon le code de la routine dma_palette (en $810C), la CGram est accédée de $0000 à $0200 pour y écrire les définitions des 256 couleurs. Mais comme pour la Vram, on ne sait pas où se trouve la CG-ram. L'adresse réelle de la CG-ram est transparente. A partir du moment où on utilise le registre $2121 (Address for CG-RAM Write), le système sait où il doit transférer.

Deux notes importantes relatives aux caractères dynamiques:

1) En Lores0, il faut 8 words (16 octets) par caractère. 1024 caractères occupent 1024x16=16384 octets (#$4000 octets). Les 256 caractères dynamiques occupent 4096 octets (#$1000 octets). Le reste des caractères définis occupe l'espace situé juste au-dessus et normalement dédié à l'écran Hires. En mode Hires, les caractères outrepassant cette limite des 4096 octets (#$1000 octets) seront écrasés par l'écran Hires. On est donc limité à 256 caractères en tout.

2) En Lores1, il faut 16 words (32 octets) par caractère. 1024 caractères occupent 1024x32=32768 octets (#$8000 octets). On ne peut mettre que 128 caractères dynamiques dans cette même zone de #$1000 octets. En mode Hires, les définitions de caractères ne peuvent outrepasser cette limite des 4096 octets (#$1000 octets), sous peine d'être écrasées par l'écran Hires. On est donc limité à 128 caractères en tout.

Conclusion évidente:

Dans toutes les routines 65816, il faut impérativement essayer de comprendre, à quelle banque on s'adresse (valeur de DBR, Data Bank Register). Le PBR (Program Bank Register) est obligatoirement $00, puisque le code 65816 exécuté est dans la banque $00 (et sa réplique $80). Idem pour le registre DP (Direct Page), qui seul permet de comprendre les mnémoniques correspondants. Facile à dire, mais pas toujours facile à savoir, car DBR, PBR et DP peuvent avoir été initialisés dans la routine appelante ou être implicites (autrement dit : on est sensés le savoir !).
En ce qui concerne les routines de transfert Ram vers Vram, DBR pointe sur la banque $00 et cela permet d'accéder aux registres système ($21xx, $43xx, etc.), dont l'adresse réelle est de la forme $0021xx, $0043xx etc. Mais en aucun cas on a une idée claire de l'adresse cible en Vram, que l'on place en $2116-$2117. Cette adresse ne comporte que 2 octets, sans indication de banque ou autre. C'est le système qui complète pour cibler la Vram, là où elle se trouve. En fait, derrière cette ″adresse″ se cache des particularités de la Snes (les divers écrans BG, les segments, les tuiles etc.), qui ne sont pas aisées à comprendre.