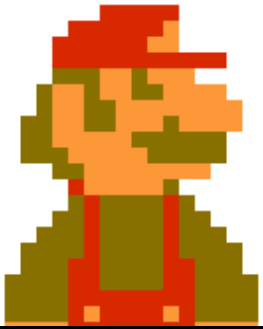
Le Nintendo Entertainment System et son architecture

Karabulut, Marchand de Kerchove, Achibet

Licence 3 Informatique
UFR Sciences et Techniques, Université du Havre
Année universitaire 2009–2010



Sortie du Famicom

Le Famicom (Family Computer), une console de jeux produite par Nintendo, sort au Japon en 1983.

Un succès immédiat

Elle est la console la plus puissante du marché japonais mais aussi la moins coûteuse (14.800¥ soit environ 100€). Deux mois après sa sortie, plus de 500.000 exemplaires ont été vendus.

Sortie dans les pays occidentaux

En 1985, une nouvelle version de la console sort en Europe et aux Etats-Unis. Son nom dans ces pays est changé en NES, pour Nintendo Entertainment System.

Différences NES/Famicom

Les deux versions de la Famicom/NES présentent des différences majeures :

- Design de la console modifié pour se débarasser de l'aspect "jouet"

Différences NES/Famicom

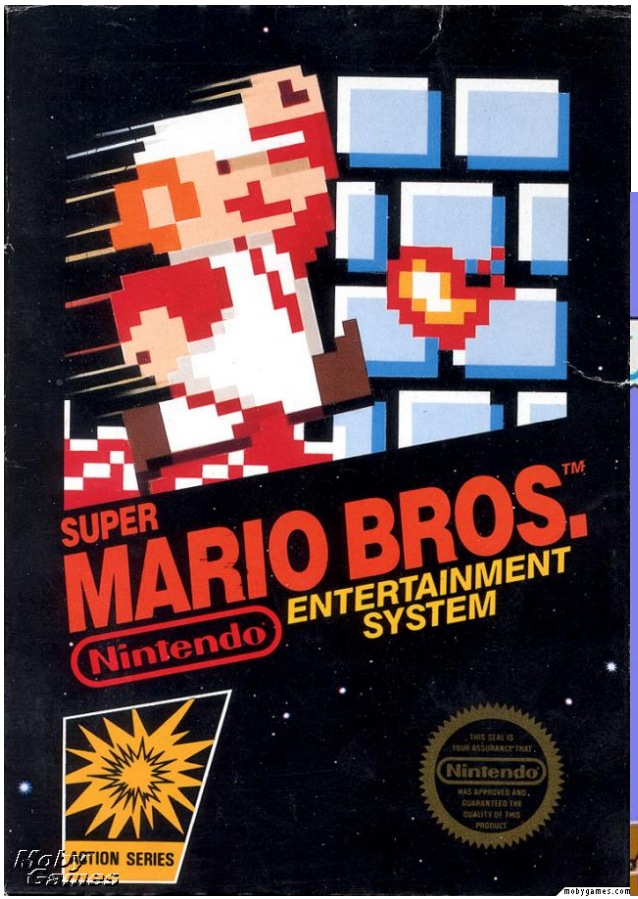
Les deux versions de la Famicom/NES présentent des différences majeures :

- Design de la console modifié pour se débarasser de l'aspect "jouet"
- Les cartouches n'ont pas le même format

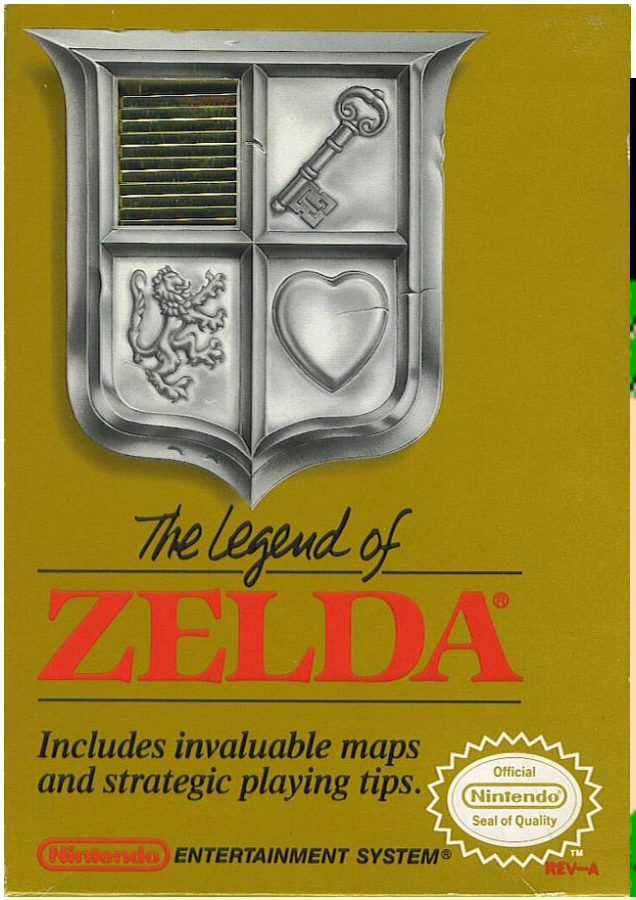
La NES relance l'industrie vidéoludique

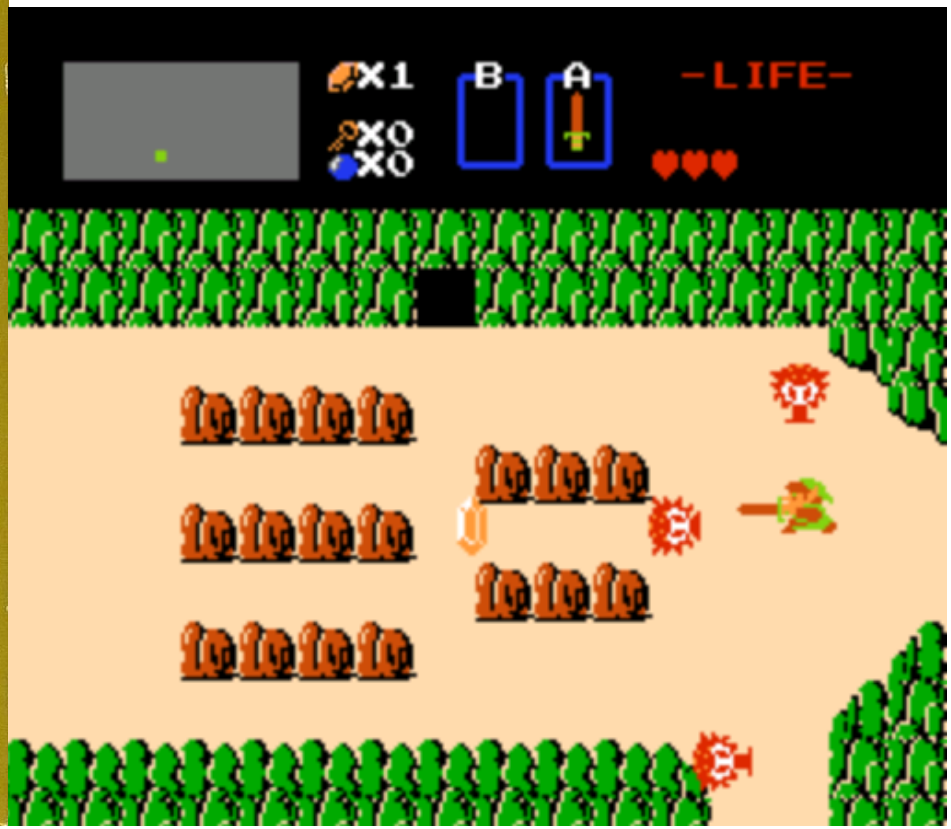
Au début des années 80, le secteur du jeu vidéo dans les pays occidentaux est en crise à cause d'un marché surchargé. Néanmoins, la NES parvient à attirer les acheteurs et plus de 42 millions d'exemplaires y sont vendus.

Super Mario Bros, 40 millions d'exemplaires vendus



The Legend of Zelda



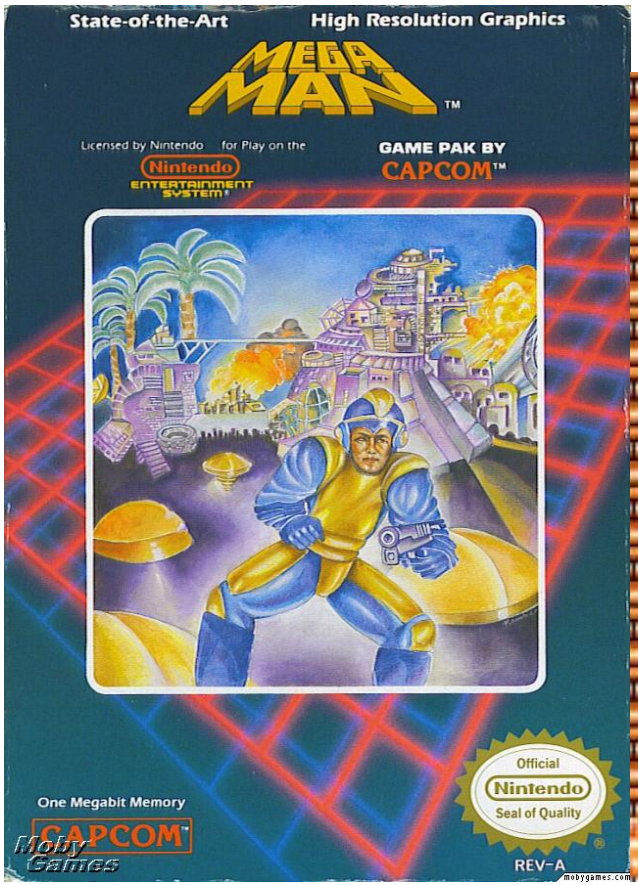


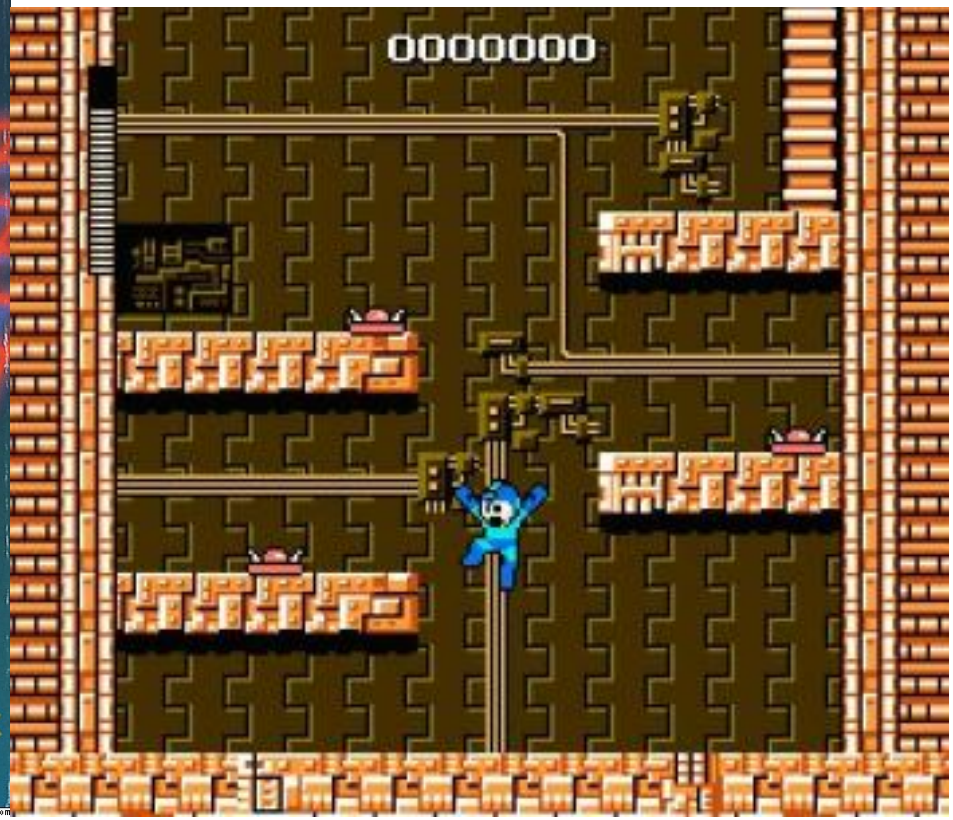
Metroid


METROID
Nintendo
ENTERTAINMENT
SYSTEM
EUROPEAN VERSION


EN..30

Mega Man


State-of-the-Art
High Resolution Graphics
MEGA MAN
Licensed by Nintendo for Play on the
Nintendo
ENTERTAINMENT SYSTEM
GAME PAK BY
CAPCOM
Official
Nintendo
Seal of Quality
One Megabit Memory
CAPCOM
REV-A
MobyGames


0000000

La ludothèque de la NES comprend plus de 800 jeux.

Les accessoires

Toujours dans l'optique de se démarquer de ses concurrents, Nintendo a sorti de nombreux accessoires originaux pour sa console.

Zapper

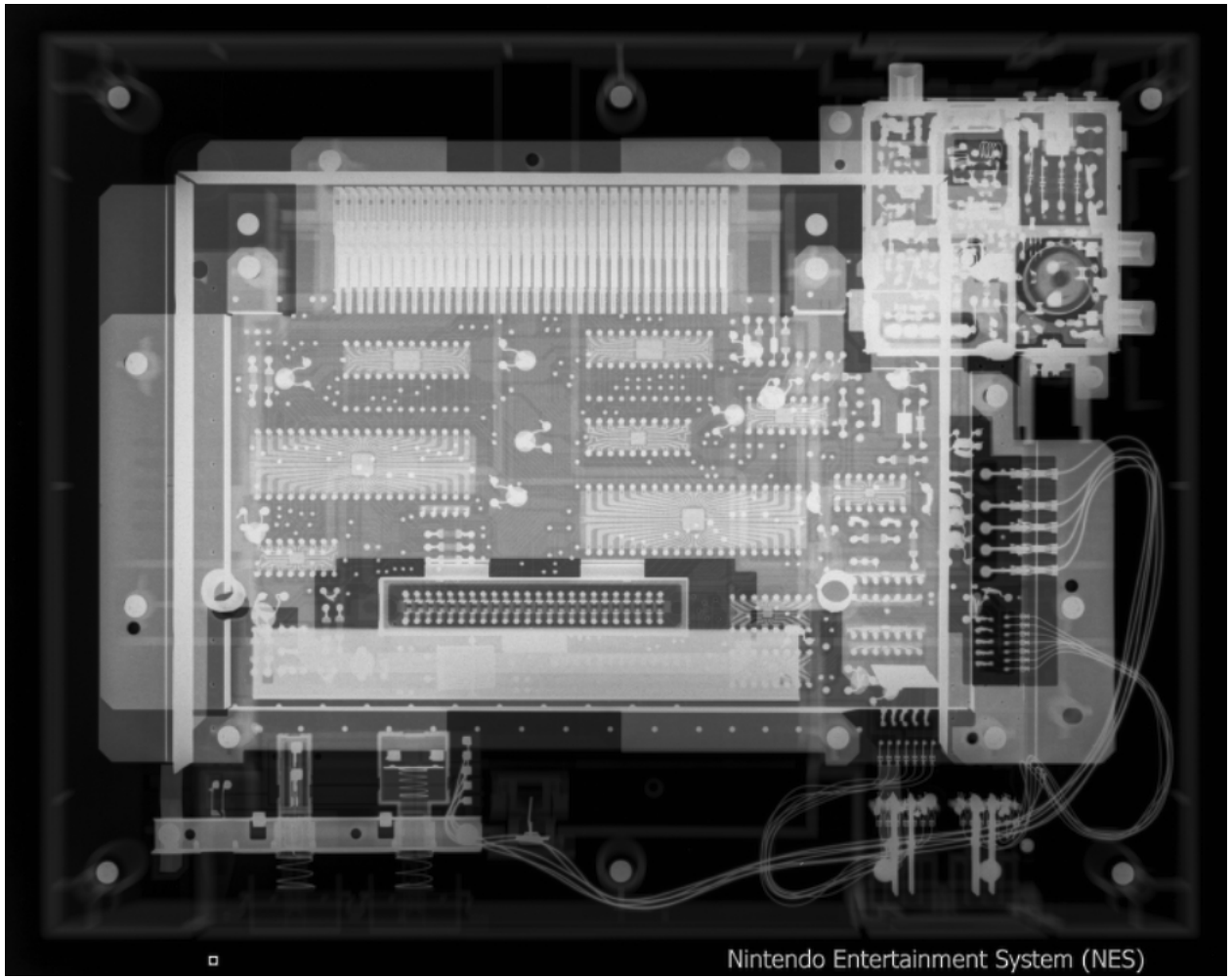
Le Zapper permet de tirer sur des éléments d'un jeu en visant le téléviseur. C'est le jeu de tir au canard Duck Hunt (vendu en bundle avec la console en Europe) qui l'a rendu populaire.

R.O.B.

R.O.B. (Robotic Operating Buddy) est un robot dont le joueur manipule les bras pour déplacer des objets à l'écran.

Power Glove

Le Power Glove est un gant censé représenter fidélement à l'écran les mouvements de la main du joueur. Son manque de précision a limité son succès mais cet accessoire a posé les fondations de la manette de la Wii de Nintendo, sortie en 2006.



Nintendo Entertainment System (NES)

Contenu de la puce

- Microprocesseur Ricoh 2A03 (NTSC) / 2A07 (PAL)

Contenu de la puce

- Microprocesseur Ricoh 2A03 (NTSC) / 2A07 (PAL)
- CPU : NMOS 6502 @ 1.79MHz (NTSC) / 1.66MHz (PAL)

Contenu de la puce

- Microprocesseur Ricoh 2A03 (NTSC) / 2A07 (PAL)
- CPU : NMOS 6502 @ 1.79MHz (NTSC) / 1.66MHz (PAL)
- APU : 5 canaux

Contenu de la puce

- Microprocesseur Ricoh 2A03 (NTSC) / 2A07 (PAL)
- CPU : NMOS 6502 @ 1.79MHz (NTSC) / 1.66MHz (PAL)
- APU : 5 canaux
- Timer programmable basse fréquence

Contenu de la puce

- Microprocesseur Ricoh 2A03 (NTSC) / 2A07 (PAL)
- CPU : NMOS 6502 @ 1.79MHz (NTSC) / 1.66MHz (PAL)
- APU : 5 canaux
- Timer programmable basse fréquence
- Unité de transfer DMA

Schéma du 2A03

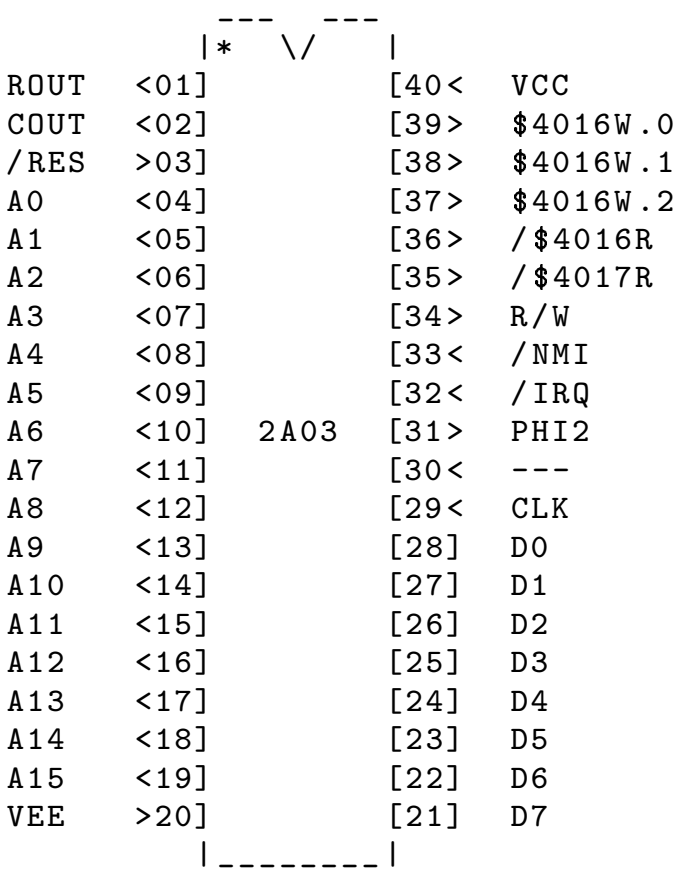
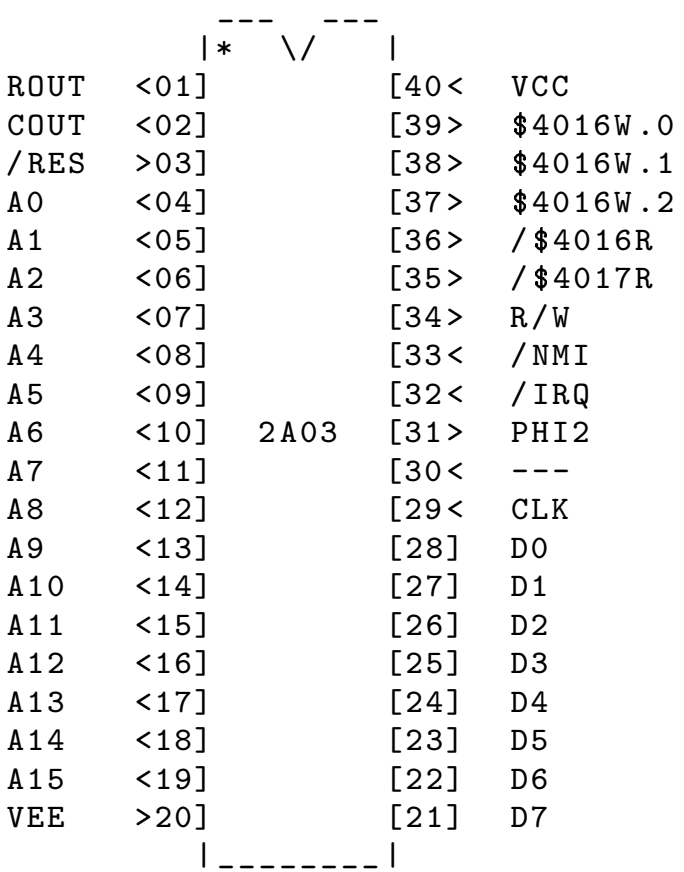


Schéma du 2A03



Notations

- VEE : masse
- VCC : +5VDC
- /RES : hard reset
- ROUT/COUT : sortie audio
- A0–A15 : bus d'adresse
- D0-D7 : bus de données
- CLK : entrée de l'horloge
- PHI2 : sortie de l'horloge après division
- /IRQ : interrompt le 6502
- R/W : direction du bus de données

Différences NTSC/PAL

- Puce 2A03 (NTSC) / 2A07 (PAL)

Différences NTSC/PAL

- Puce 2A03 (NTSC) / 2A07 (PAL)
- Diviseur d'horloge à 12 en NTSC contre 16 en PAL

Différences NTSC/PAL

- Puce 2A03 (NTSC) / 2A07 (PAL)
- Diviseur d'horloge à 12 en NTSC contre 16 en PAL
- CPU @ 1.79MHz (NTSC) / 1.66MHz (PAL)

Différences NTSC/PAL

- Puce 2A03 (NTSC) / 2A07 (PAL)
- Diviseur d'horloge à 12 en NTSC contre 16 en PAL
- CPU @ 1.79MHz (NTSC) / 1.66MHz (PAL)
- Playback APU plus lent en PAL

Un CPU populaire

- MOS 6502 modifié, sans mode décimal

Un CPU populaire

- MOS 6502 modifié, sans mode décimal
- Similaire au Motorola 6800, mais plus rapide

Un CPU populaire

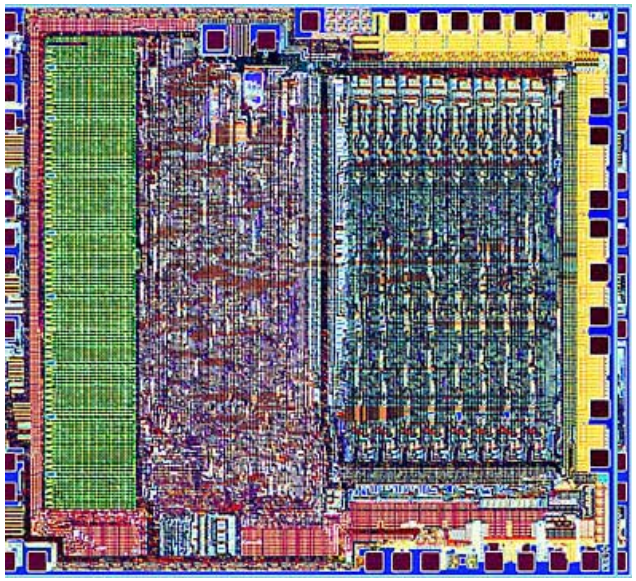
- MOS 6502 modifié, sans mode décimal
- Similaire au Motorola 6800, mais plus rapide
- 25$ à sa sortie (1975) contre 179$ pour les 6800 et 8080

Un CPU populaire

- MOS 6502 modifié, sans mode décimal
- Similaire au Motorola 6800, mais plus rapide
- 25$ à sa sortie (1975) contre 179$ pour les 6800 et 8080
- Utilisé dans les Apple I et II (1977), les Atari 400 et 800 (1979), les Commodore PET (1977) et VIC-20 (1981)

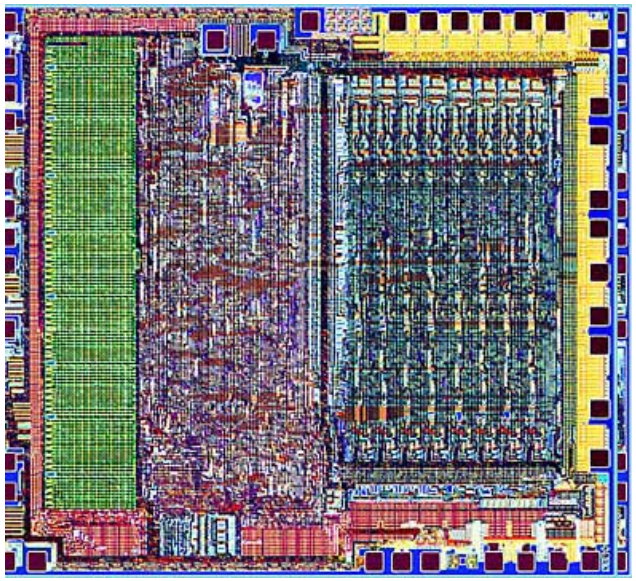
Un design simple

- Architecture 8 bits et bus d'adressage 16 bits



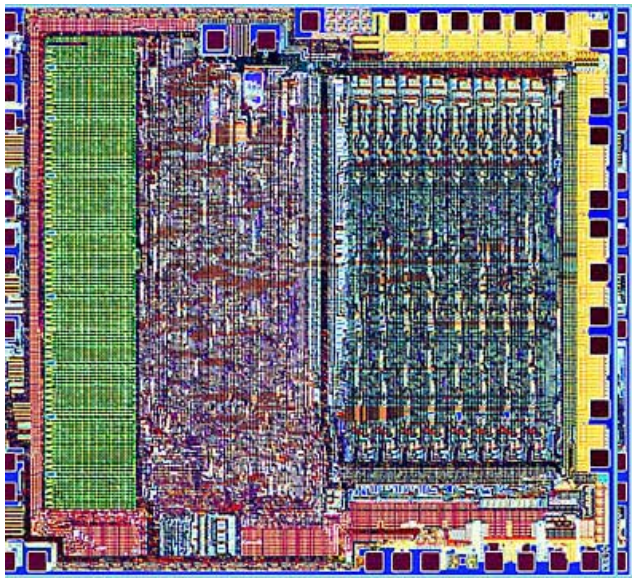
Un design simple

- Architecture 8 bits et bus d'adressage 16 bits
- Design simple à machines à états, rapide



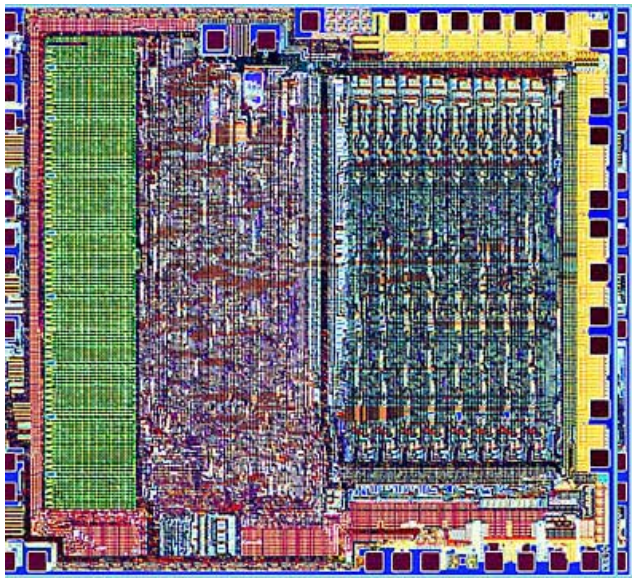
Un design simple

- Architecture 8 bits et bus d'adressage 16 bits
- Design simple à machines à états, rapide
- Pipeline à une seule étape : fetch pendant l'éxécution



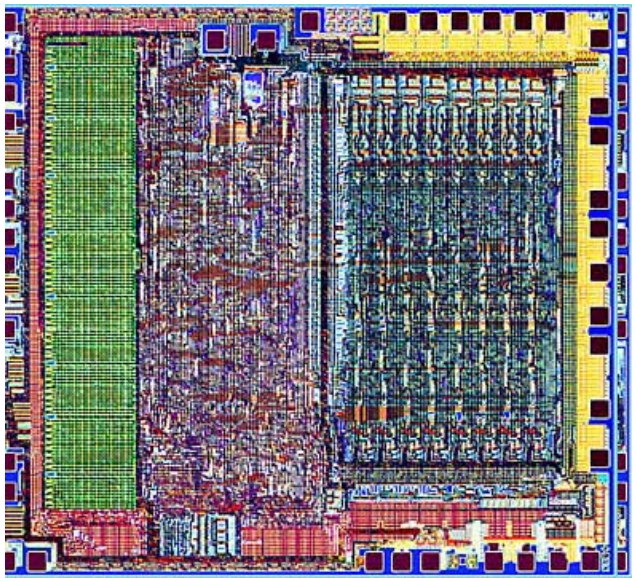
Un design simple

- Architecture 8 bits et bus d'adressage 16 bits
- Design simple à machines à états, rapide
- Pipeline à une seule étape : fetch pendant l'éxécution
- Pas de cache ni prédiction de branchement



Un design simple

- Architecture 8 bits et bus d'adressage 16 bits
- Design simple à machines à états, rapide
- Pipeline à une seule étape : fetch pendant l'éxécution
- Pas de cache ni prédiction de branchement
- 56 instructions avec 13 modes d'adressage (dont zero page)



Répartition mémoire du 6502

Adresse	Taille	Utilité
$0000	$0800	RAM interne de 2Ko

Répartition mémoire du 6502

Adresse	Taille	Utilité
$0000	$0800	RAM interne de 2Ko
$0800	$0800	Mirroirs de $0000 – $07FF
$1000	$0800	
$1800	$0800	

Répartition mémoire du 6502

Adresse	Taille	Utilité
$0000	$0800	RAM interne de 2Ko
$0800	$0800	Mirroirs de $0000 – $07FF
$1000	$0800	
$1800	$0800	
$2000	$0008	Registres PPU

Répartition mémoire du 6502

Adresse	Taille	Utilité
$0000	$0800	RAM interne de 2Ko
$0800	$0800	Mirroirs de $0000 – $07FF
$1000	$0800	
$1800	$0800	
$2000	$0008	Registres PPU
$2008	$1FF8	Mirroirs de $2000 tous les 8 octets

Répartition mémoire du 6502

Adresse	Taille	Utilité
$0000	$0800	RAM interne de 2Ko
$0800	$0800	Mirroirs de $0000 – $07FF
$1000	$0800	
$1800	$0800	
$2000	$0008	Registres PPU
$2008	$1FF8	Mirroirs de $2000 tous les 8 octets
$4000	$0018	Registres APU et E/S

Répartition mémoire du 6502

Adresse	Taille	Utilité
$0000	$0800	RAM interne de 2Ko
$0800	$0800	Mirroirs de $0000 – $07FF
$1000	$0800	
$1800	$0800	
$2000	$0008	Registres PPU
$2008	$1FF8	Mirroirs de $2000 tous les 8 octets
$4000	$0018	Registres APU et E/S
$4018	$FFFF	PRG ROM cartouche, PRG RAM cartouche, et registres de mappers

Registres

- RAM plus rapide que le CPU donc peu de registres

Registres

- RAM plus rapide que le CPU donc peu de registres
- 5 registres 8 bits

Registres

- RAM plus rapide que le CPU donc peu de registres
- 5 registres 8 bits
- L'accumulateur pour l'arithmétique et la logique

Registres

- RAM plus rapide que le CPU donc peu de registres
- 5 registres 8 bits
- L'accumulateur pour l'arithmétique et la logique
- Deux registres d'index pour contenir des adresses et compter dans les boucles

Registres

- RAM plus rapide que le CPU donc peu de registres
- 5 registres 8 bits
- L'accumulateur pour l'arithmétique et la logique
- Deux registres d'index pour contenir des adresses et compter dans les boucles
- Le pointeur de pile ($0100 – $01FF, 128 routines de profondeur)

Registres

- RAM plus rapide que le CPU donc peu de registres
- 5 registres 8 bits
- L'accumulateur pour l'arithmétique et la logique
- Deux registres d'index pour contenir des adresses et compter dans les boucles
- Le pointeur de pile ($0100 – $01FF, 128 routines de profondeur)
- Le registre de status pour les tests, effets de bords des instructions

Schéma du registre de status

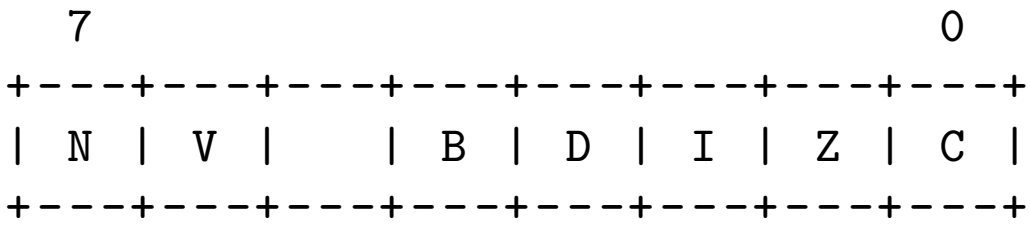
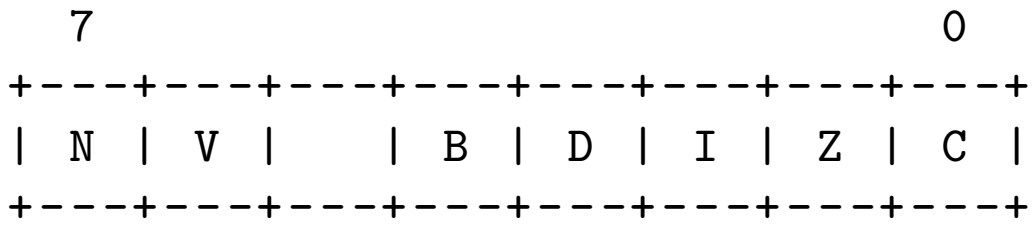


Schéma du registre de status



Flags

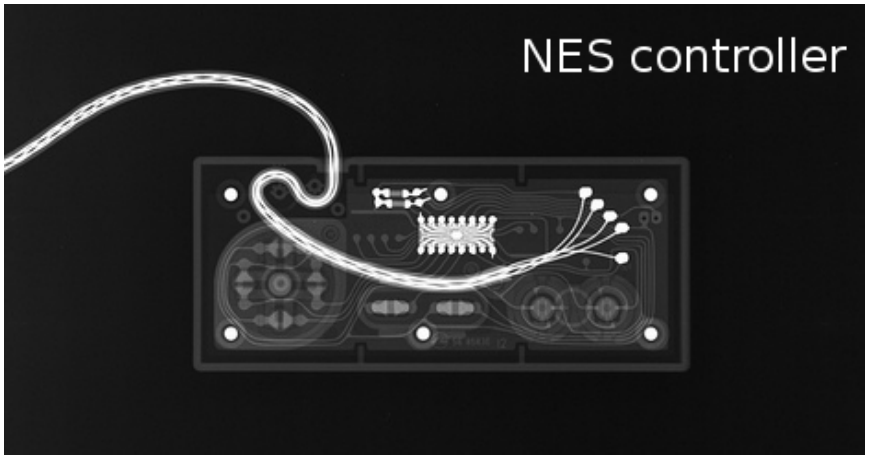
- Negative, si le bit 7 de l'accumulateur est 1
- Overflow, si dépassement lors d'un calcul
- BRK, si l'interruption a été causée par BRK
- Decimal, si le mode décimal est actif
- IRQ disable, si les interruptions sont désactivées
- Zero, si le résultat de la dernière opération est 0
- Carry, la retenue d'une opération

E/S projetées en mémoire

Les manettes, le PPU, le son se contrôlent par écriture et lecture dans des registres.

E/S projetées en mémoire

Les manettes, le PPU, le son se contrôlent par écriture et lecture dans des registres.



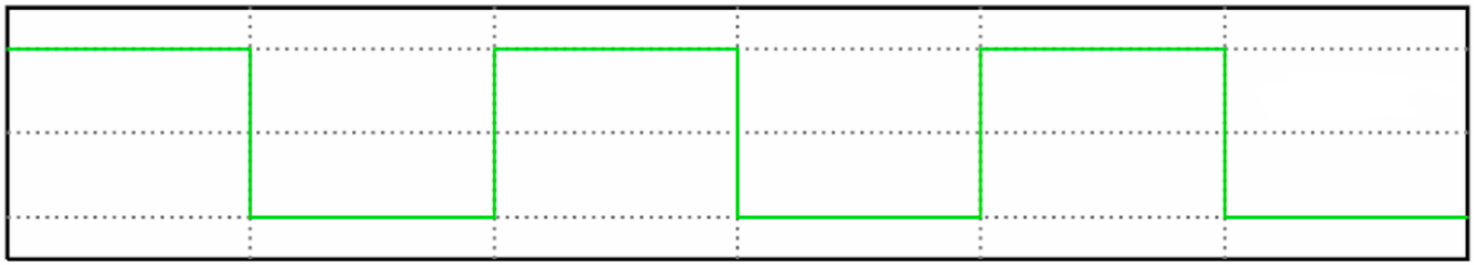
Lecture d'état de la première manette

```
LDA #$01
STA $4016
LDA #$00        ; latch both
STA $4016       ; controllers

LDA $4016       ; A
LDA $4016       ; B
LDA $4016       ; Select
LDA $4016       ; Start
LDA $4016       ; Up
LDA $4016       ; Down
LDA $4016       ; Left
LDA $4016       ; Right
```

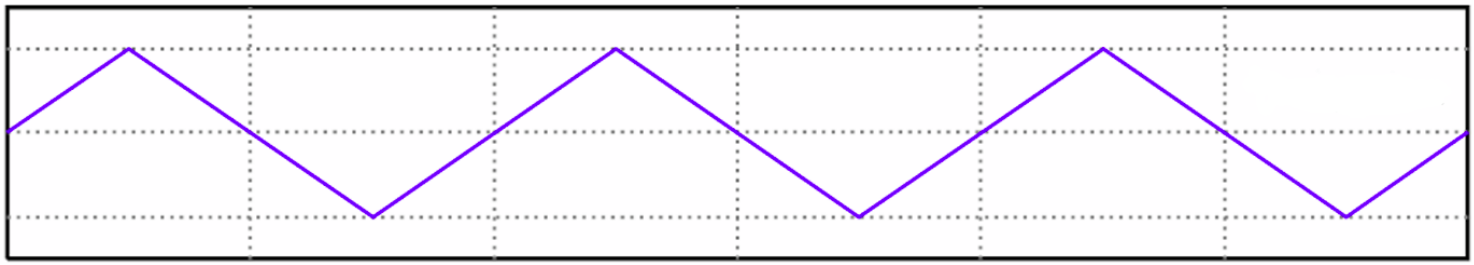
Cinq canaux

- 2 canaux à signaux rectangulaires



Cinq canaux

- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire



Cinq canaux

- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire
- 1 canal à longueurs d'ondes aléatoires pour le bruit

Cinq canaux

- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire
- 1 canal à longueurs d'ondes aléatoires pour le bruit
- 1 canal à modulation delta

Cinq canaux

- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire
- 1 canal à longueurs d'ondes aléatoires pour le bruit
- 1 canal à modulation delta

Cinq canaux

- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire
- 1 canal à longueurs d'ondes aléatoires pour le bruit
- 1 canal à modulation delta

Paramètres contrôlés

- Période (fréquence)

Cinq canaux

- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire
- 1 canal à longueurs d'ondes aléatoires pour le bruit
- 1 canal à modulation delta

Paramètres contrôlés

- Période (fréquence)
- Volume

Cinq canaux

- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire
- 1 canal à longueurs d'ondes aléatoires pour le bruit
- 1 canal à modulation delta

Paramètres contrôlés

- Période (fréquence)
- Volume
- Durée

Cinq canaux

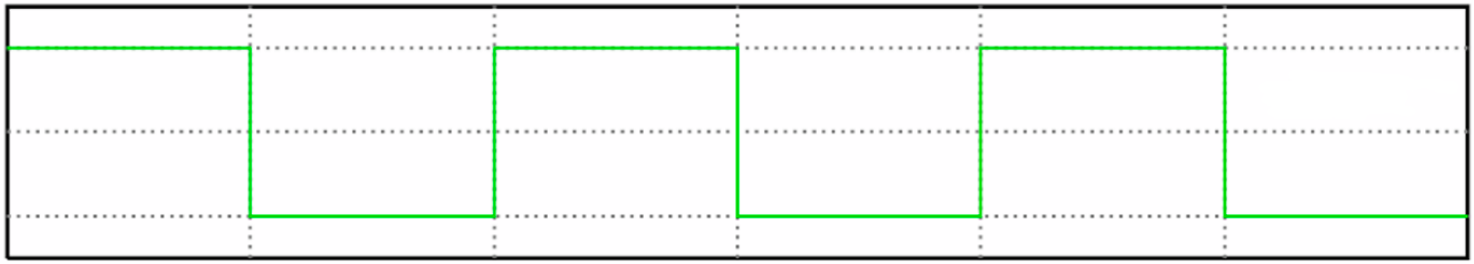
- 2 canaux à signaux rectangulaires
- 1 canal à signal triangulaire
- 1 canal à longueurs d'ondes aléatoires pour le bruit
- 1 canal à modulation delta

Paramètres contrôlés

- Période (fréquence)
- Volume
- Durée
- Ton

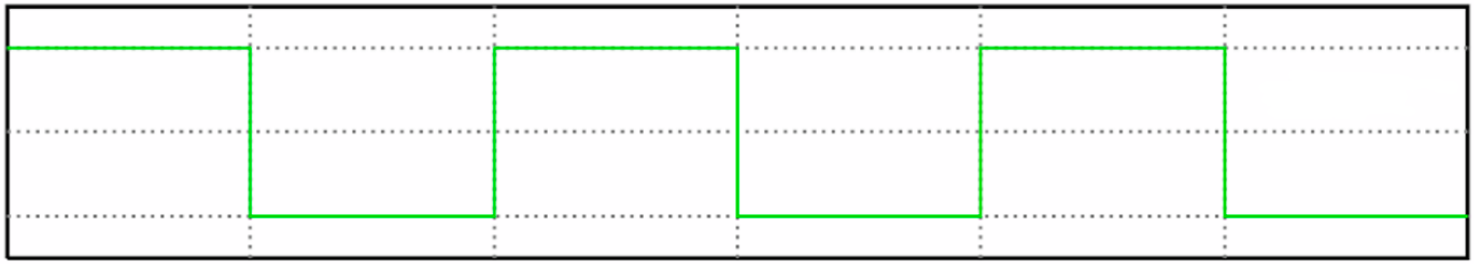
Signal rectangulaire

- Fréquence : de 54.6Hz à 12.4KHz



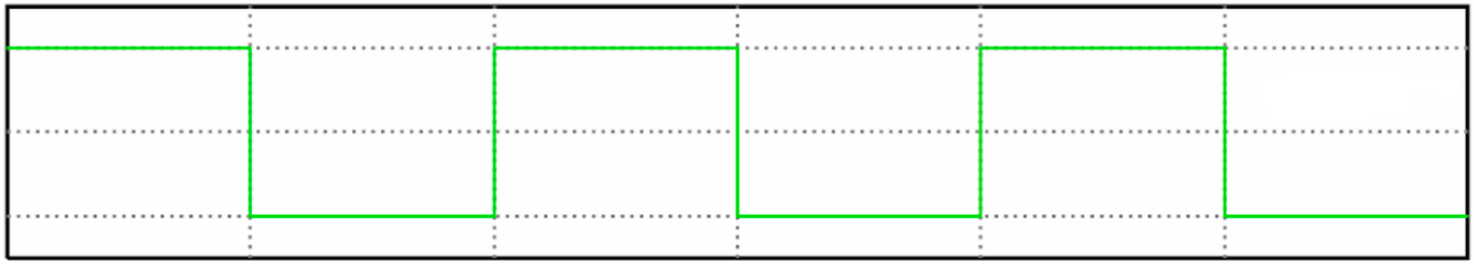
Signal rectangulaire

- Fréquence : de 54.6Hz à 12.4KHz
- Ajustement du ton sur 2 bits



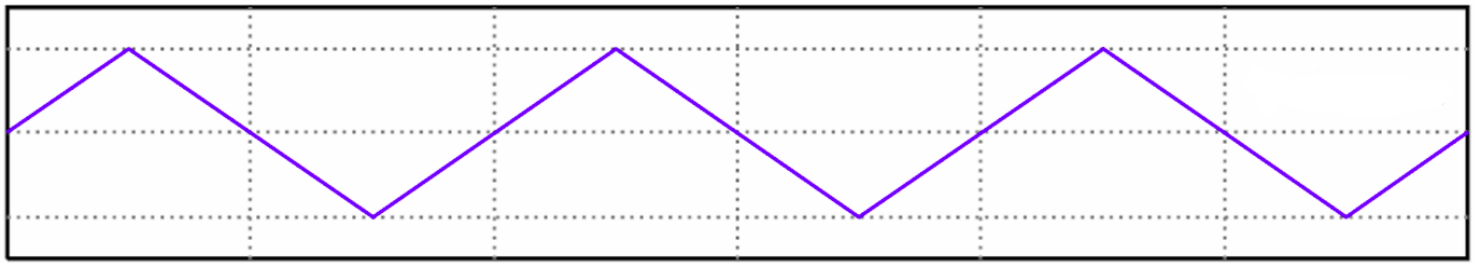
Signal rectangulaire

- Fréquence : de 54.6Hz à 12.4KHz
- Ajustement du ton sur 2 bits
- Ajustement de la longueur sur 4 bits



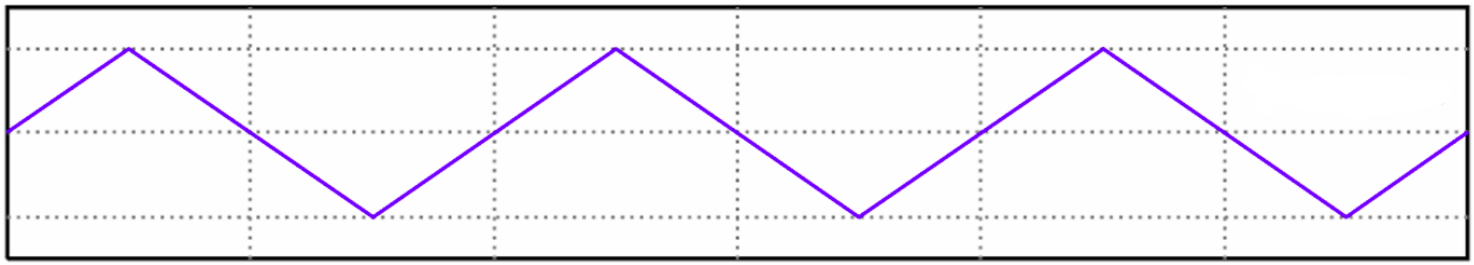
Signal triangle

- Résolution de 4 bits (16 étapes)



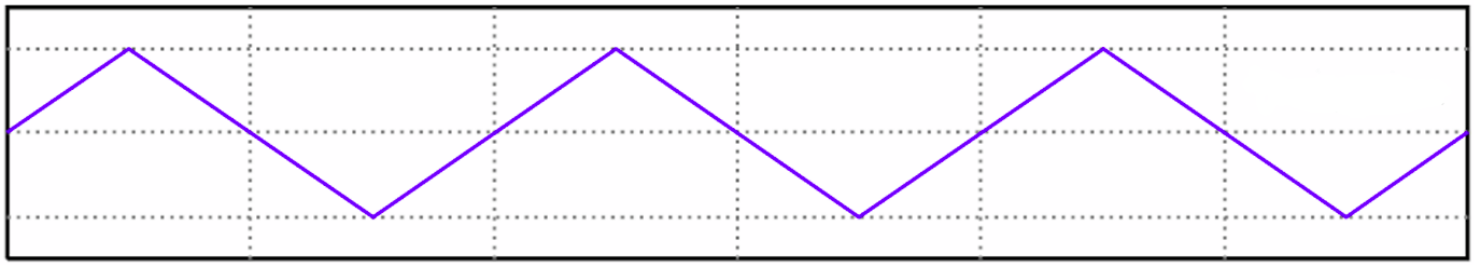
Signal triangle

- Résolution de 4 bits (16 étapes)
- Fréquence : de 27.3Hz à 55.9KHz



Signal triangle

- Résolution de 4 bits (16 étapes)
- Fréquence : de 27.3Hz à 55.9KHz
- Compteur linéaire pour améliorer la résolution



Signal de bruit

- Générateur d'entiers aléatoires sur 15 bits

Signal de bruit

- Générateur d'entiers aléatoires sur 15 bits
- Fréquence : de 29.3Hz à 447KHz

Delta Modulation Channel

- Série de compteurs digitaux et de registres

Delta Modulation Channel

- Série de compteurs digitaux et de registres
- Reproduit un son analogique

Delta Modulation Channel

- Série de compteurs digitaux et de registres
- Reproduit un son analogique
- Deux modes de reproduction sonore

Delta Modulation Channel

- Série de compteurs digitaux et de registres
- Reproduit un son analogique
- Deux modes de reproduction sonore

Delta Modulation Channel

- Série de compteurs digitaux et de registres
- Reproduit un son analogique
- Deux modes de reproduction sonore

DMA

Récupération des échantillons en une seule fois

Delta Modulation Channel

- Série de compteurs digitaux et de registres
- Reproduit un son analogique
- Deux modes de reproduction sonore

DMA

Récupération des échantillons en une seule fois

PCM

Mise à jour constante d'un registre mémoire

Le PPU

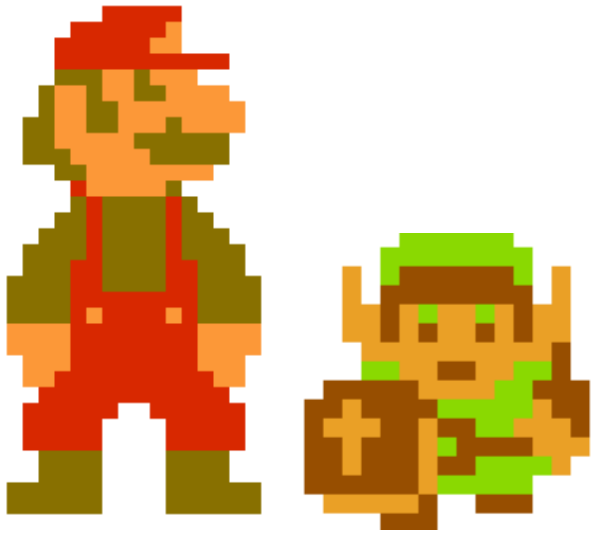
Picture Processing Unit

Le processeur graphique de la NES, aussi appelé PPU (Picture Processing Unit), existe en deux versions :

- RP2C02 pour les téléviseurs NTSC à 60 Hz (Amérique), 5.32 MHz
- RP2C07 pour les téléviseurs PAL à 50 Hz (Europe, Asie), 5.37 MHz

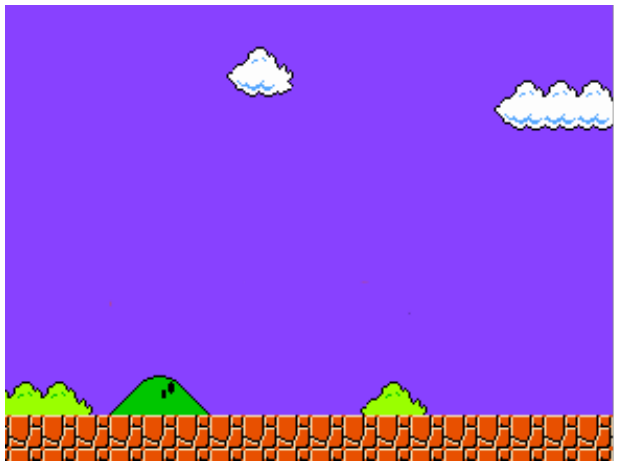
Son travail

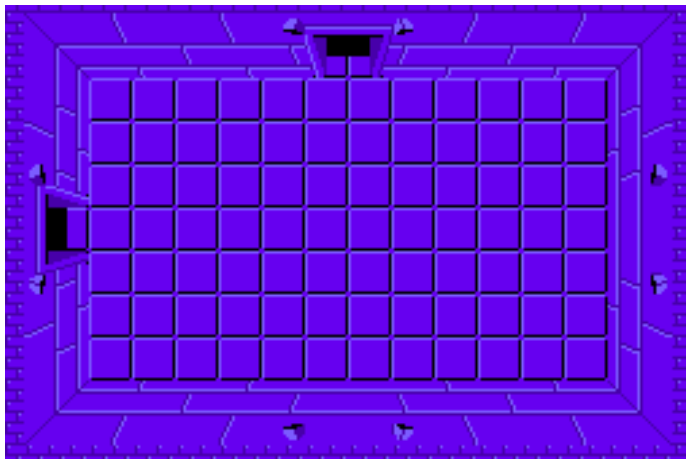
- afficher les personnages et les objets



Son travail

- afficher les personnages et les objets
- afficher les décors





Son travail

- afficher les personnages et les objets
- afficher les décors
- afficher les éléments de l'interface

Deux types d'élément graphiques

Le PPU manipule deux types d'éléments graphiques différents :

Deux types d'élément graphiques

Le PPU manipule deux types d'éléments graphiques différents :

Les sprites éléments mobiles et/ou animés (personnages, objets, interface)

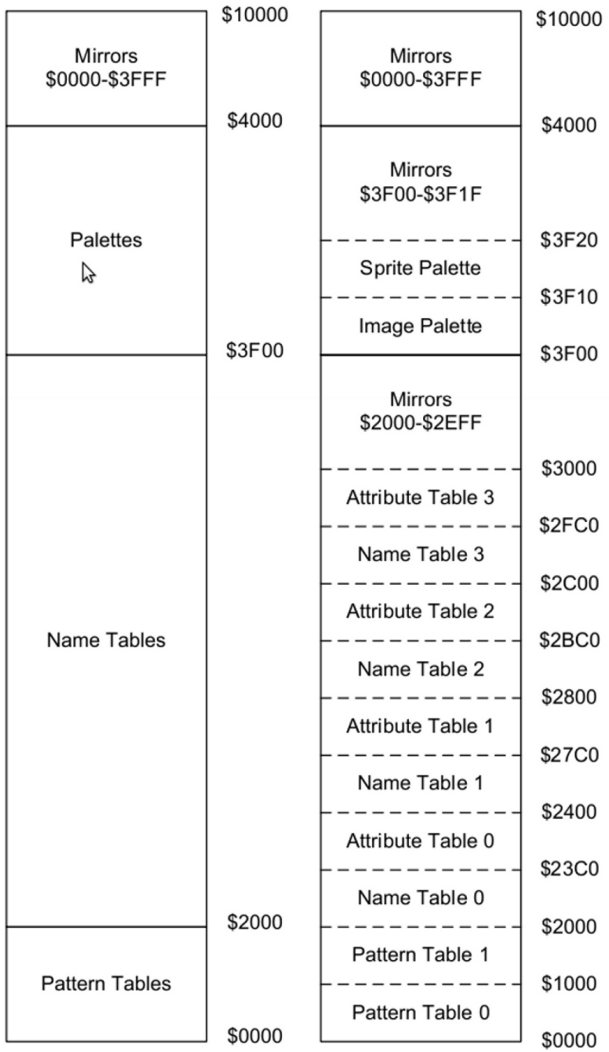
Deux types d'élément graphiques

Le PPU manipule deux types d'éléments graphiques différents :

Les sprites éléments mobiles et/ou animés (personnages, objets, interface)

L'arrière-plan le fond immobile (décors)

Répartition mémoire



Couleurs

Couleurs

Palette principale

Une palette principale prédéfinie contient l'ensemble des couleurs affichables par la console.

Couleurs

Palette principale

Une palette principale prédéfinie contient l'ensemble des couleurs affichables par la console.

Il y a 64 couleurs disponibles (52 différentes).

Couleurs

Palette principale

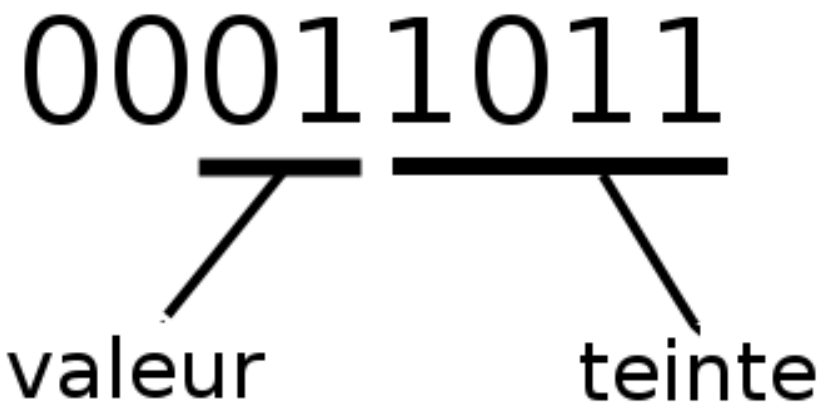
Une palette principale prédéfinie contient l'ensemble des couleurs affichables par la console.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F

Il y a 64 couleurs disponibles (52 différentes).

Codage des couleurs

Chaque couleur de la palette est codée en HSV (Hue, Saturation, Value ou Teinte, Saturation, Valeur) sur 8 bits.
Les deux bits de poids forts ne sont pas utilisés. La saturation est déterminée à partir des deux autres informations.



Deux autres palettes

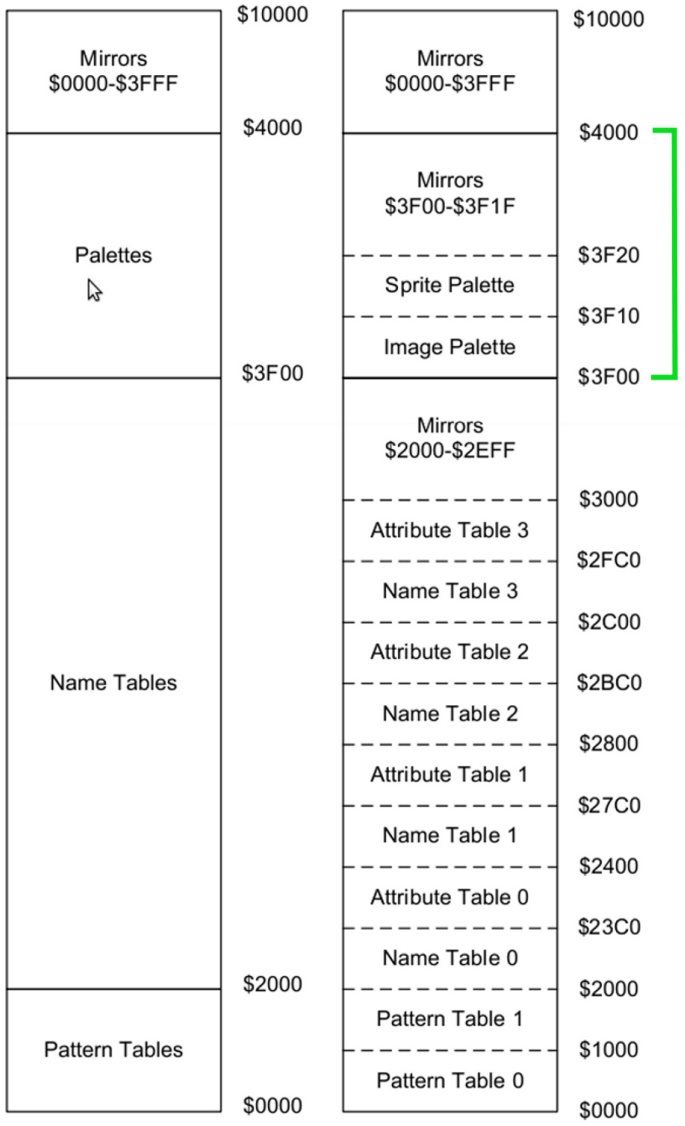
Deux plus petites palettes existent, une pour les sprites (sprite palette) et une pour l'arrière-plan (image palette).
Elles contiennent 16 couleurs chacunes (13 différentes car la première couleur, dite de transparence, est repétée toute les quatre entrées).

Deux autres palettes

Deux plus petites palettes existent, une pour les sprites (sprite palette) et une pour l'arrière-plan (image palette).
Elles contiennent 16 couleurs chacunes (13 différentes car la première couleur, dite de transparence, est repétée toute les quatre entrées).

16	37	21	0A	16	29	0C	0D	16	2A	24	2D	16	12	02	20

Elles ne contiennent pas les valeurs HSV des couleurs mais leurs indices dans la palette principale.


$10000
Mirrors
$0000-$3FFF
$4000
Palettes
$3F00
Name Tables
$2000
Pattern Tables
$0000
$10000
Mirrors
$0000-$3FFF
$4000
Mirrors
$3F00-$3F1F
$3F20
Sprite Palette
$3F10
Image Palette
$3F00
Mirrors
$2000-$2EFF
$3000
Attribute Table 3
$2FC0
Name Table 3
$2C00
Attribute Table 2
$2BC0
Name Table 2
$2800
Attribute Table 1
$27C0
Name Table 1
$2400
Attribute Table 0
$23C0
Name Table 0
$2000
Pattern Table 1
$1000
Pattern Table 0
$0000

Limitation technique

Une image affichée par la console ne pourra pas contenir plus de 25 couleurs différentes :

Limitation technique

Une image affichée par la console ne pourra pas contenir plus de 25 couleurs différentes :

- 13 pour les sprites

Limitation technique

Une image affichée par la console ne pourra pas contenir plus de 25 couleurs différentes :

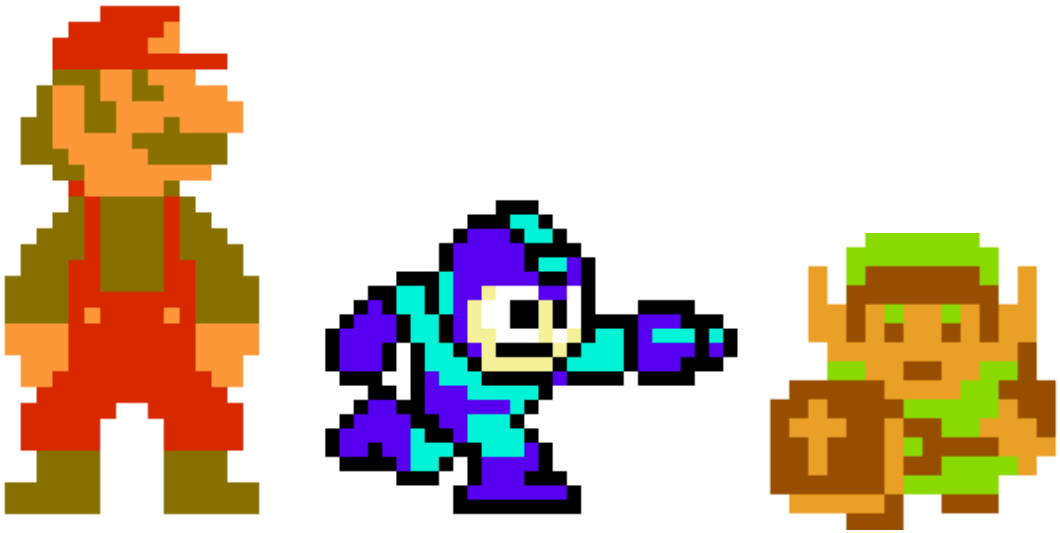
- 13 pour les sprites
- 13 pour l'arrière-plan

Limitation technique

Une image affichée par la console ne pourra pas contenir plus de 25 couleurs différentes :

- 13 pour les sprites
- 13 pour l'arrière-plan
- la couleur de transparence est commune aux deux

Sprites

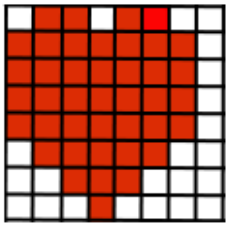


Sprites

Les sprites sont les éléments graphiques se déplaçant à l'écran et/ou étant animés.

Sprites

Les sprites sont les éléments graphiques se déplaçant à l'écran et/ou étant animés.



Ils sont stockés dans la mémoire sous forme de tiles de 8x8 pixels (ou plus rarement de 8x16 pixels).

Relation entre pixels et sprite palette

Chaque pixel de chaque tile est associé à une valeur représentant l'indice de sa couleur dans la palette des sprites.
La palette contient 16 couleurs, cet indice est codé sur 4 bits.

Les sprites en mémoire

Les informations concernant les sprites sont séparées dans deux zones de la mémoire :

Les sprites en mémoire

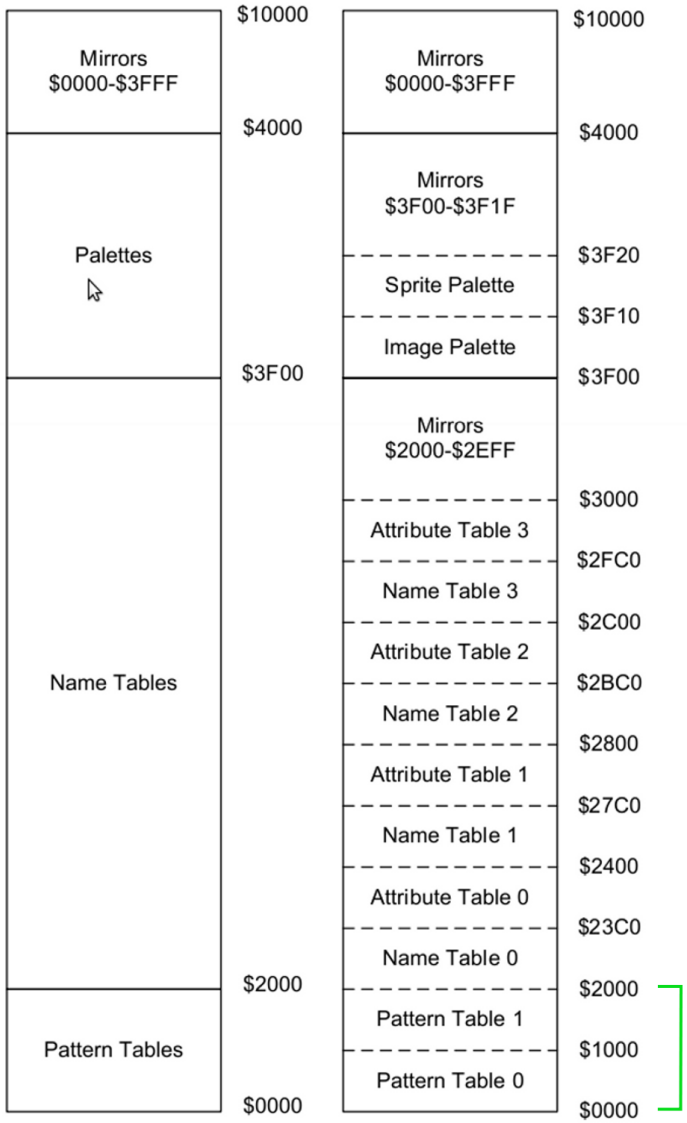
Les informations concernant les sprites sont séparées dans deux zones de la mémoire :

- une moitié des informations de couleur (2 bits de poids faible de l'indice de la couleur dans la palette) est définie dans une pattern table

Les sprites en mémoire

Les informations concernant les sprites sont séparées dans deux zones de la mémoire :

- une moitié des informations de couleur (2 bits de poids faible de l'indice de la couleur dans la palette) est définie dans une pattern table
- l'autre moitié des informations de couleur (2 bits de poids fort) ainsi que des informations de rendu (position sur l'écran, transformations à appliquer) se trouvent dans la Sprite RAM


$10000
Mirrors
$0000-$3FFF
$4000
Palettes
$3F00
Name Tables
$2000
Pattern Tables
$0000
$10000
Mirrors
$0000-$3FFF
$4000
Mirrors
$3F00-$3F1F
$3F20
Sprite Palette
$3F10
Image Palette
$3F00
Mirrors
$2000-$2EFF
$3000
Attribute Table 3
$2FC0
Name Table 3
$2C00
Attribute Table 2
$2BC0
Name Table 2
$2800
Attribute Table 1
$27C0
Name Table 1
$2400
Attribute Table 0
$23C0
Name Table 0
$2000
Pattern Table 1
$1000
Pattern Table 0
$0000

Pattern table

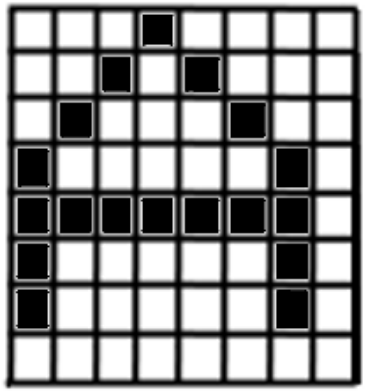
Une pattern table contient contient 4 Ko de données, pour 256 tiles, soit 16 octets par tile.

Address	Value
$0000	0 0 0 1 0 0 0 0
	0 0 1 0 1 0 0 0
	0 1 0 0 0 1 0 0
	1 0 0 0 0 0 1 0
	1 1 1 1 1 1 1 0
	1 0 0 0 0 0 1 0
	1 0 0 0 0 0 1 0
$0007	0 0 0 0 0 0 0 0

Pattern table

Une pattern table contient contient 4 Ko de données, pour 256 tiles, soit 16 octets par tile.

Address	Value
$0000	0 0 0 1 0 0 0 0
	0 0 1 0 1 0 0 0
	0 1 0 0 0 1 0 0
	1 0 0 0 0 0 1 0
	1 1 1 1 1 1 1 0
	1 0 0 0 0 0 1 0
	1 0 0 0 0 0 1 0
$0007	0 0 0 0 0 0 0 0



Si on utilisait seulement 8 octets par tile, les informations de couleurs seraient absentes.

Gestion des couleurs

On utilise en fait deux "couches" par tile afin d'avoir les deux bits de poids faible de la couleur de chaque pixel.

Address	Value
$0000	0 0 0 1 0 0 0 0
	0 0 0 0 0 0 0 0
	0 1 0 0 0 1 0 0
	0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
$0007	0 0 0 0 0 0 0 0

Gestion des couleurs

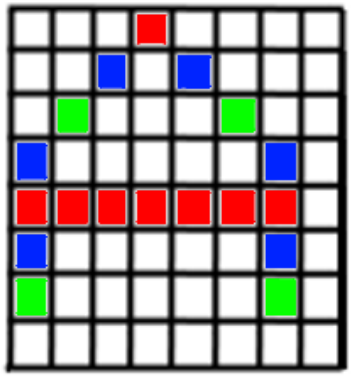
On utilise en fait deux "couches" par tile afin d'avoir les deux bits de poids faible de la couleur de chaque pixel.

Address	Value
$0000	0 0 0 1 0 0 0 0
	0 0 0 0 0 0 0 0
	0 1 0 0 0 1 0 0
	0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
$0007	0 0 0 0 0 0 0 0
$0008	0 0 0 0 0 0 0 0
	0 0 1 0 1 0 0 0
	0 1 0 0 0 1 0 0
	1 0 0 0 0 0 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
	1 0 0 0 0 0 1 0
$000F	0 0 0 0 0 0 0 0

Gestion des couleurs

On utilise en fait deux "couches" par tile afin d'avoir les deux bits de poids faible de la couleur de chaque pixel.

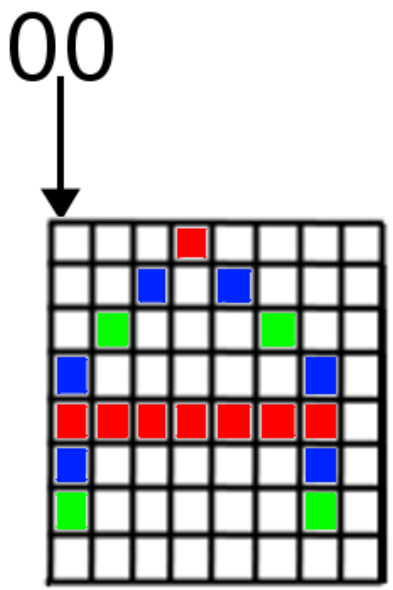
Address	Value
$0000	0 0 0 1 0 0 0 0
	0 0 0 0 0 0 0 0
	0 1 0 0 0 1 0 0
	0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
$0007	0 0 0 0 0 0 0 0
$0008	0 0 0 0 0 0 0 0
	0 0 1 0 1 0 0 0
	0 1 0 0 0 1 0 0
	1 0 0 0 0 0 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
	1 0 0 0 0 0 1 0
$000F	0 0 0 0 0 0 0 0



Gestion des couleurs

On utilise en fait deux "couches" par tile afin d'avoir les deux bits de poids faible de la couleur de chaque pixel.

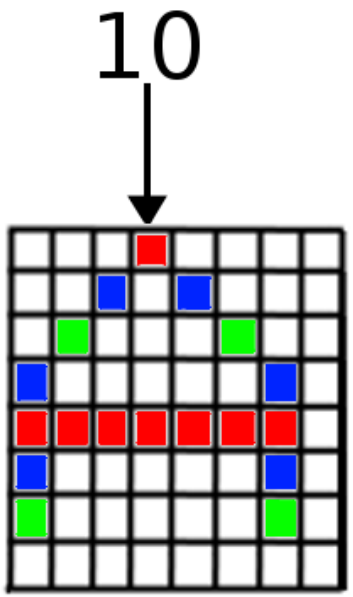
Address	Value
$0000	0 0 0 1 0 0 0 0
	0 0 0 0 0 0 0 0
	0 1 0 0 0 1 0 0
	0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
$0007	0 0 0 0 0 0 0 0
$0008	0 0 0 0 0 0 0 0
	0 0 1 0 1 0 0 0
	0 1 0 0 0 1 0 0
	1 0 0 0 0 0 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
	1 0 0 0 0 0 1 0
$000F	0 0 0 0 0 0 0 0



Gestion des couleurs

On utilise en fait deux "couches" par tile afin d'avoir les deux bits de poids faible de la couleur de chaque pixel.

Address	Value
$0000	0 0 0 1 0 0 0 0
	0 0 0 0 0 0 0 0
	0 1 0 0 0 1 0 0
	0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
$0007	0 0 0 0 0 0 0 0
$0008	0 0 0 0 0 0 0 0
	0 0 1 0 1 0 0 0
	0 1 0 0 0 1 0 0
	1 0 0 0 0 0 1 0
	0 0 0 0 0 0 0 0
	1 0 0 0 0 0 1 0
	1 0 0 0 0 0 1 0
$000F	0 0 0 0 0 0 0 0



Sprite RAM

La Sprite RAM, contient 256 octets pour 64 tiles, soit 4 octets par tile.

00110010

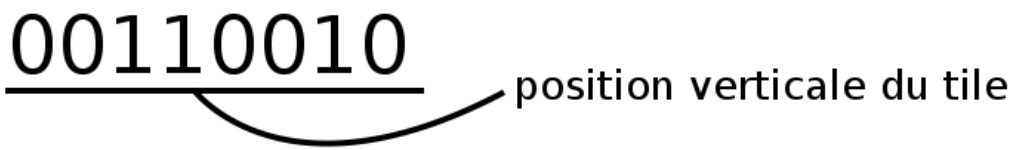
00001110

00000001

00000110

Sprite RAM

La Sprite RAM, contient 256 octets pour 64 tiles, soit 4 octets par tile.



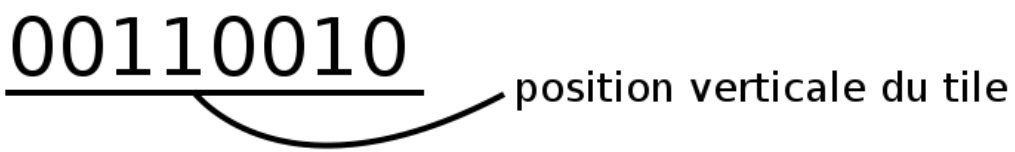
00001110

00000001

00000110

Sprite RAM

La Sprite RAM, contient 256 octets pour 64 tiles, soit 4 octets par tile.

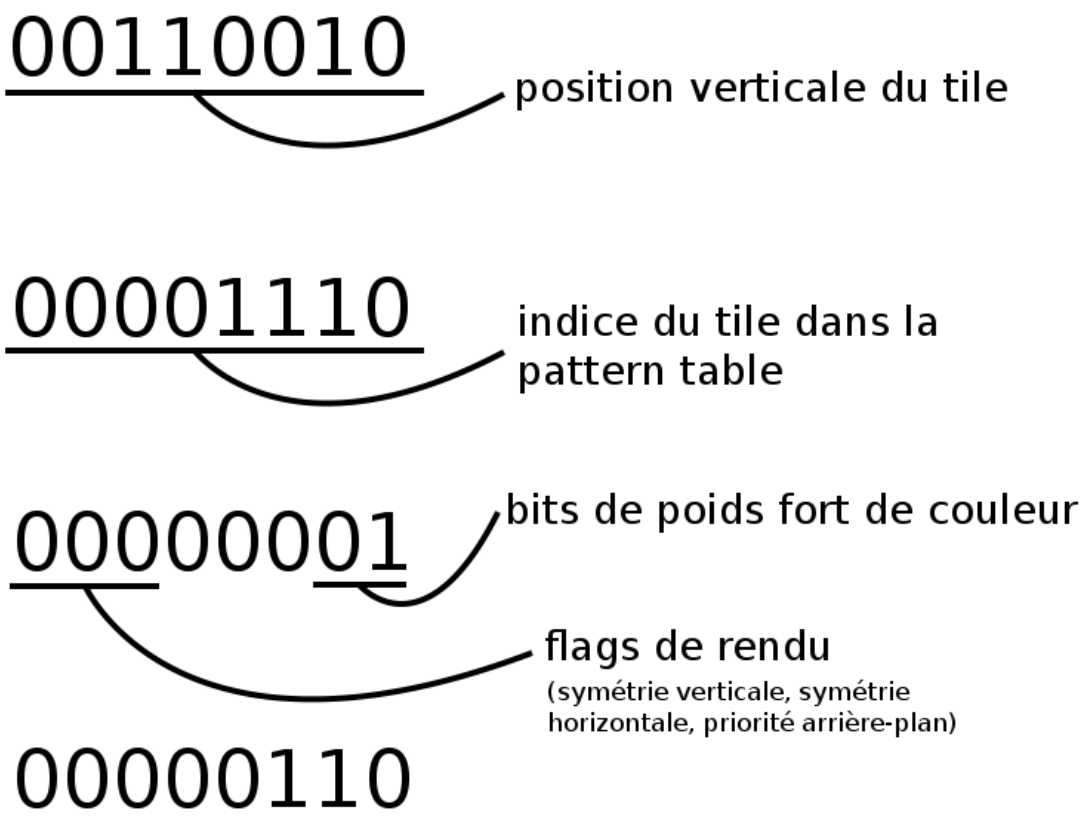


00000001

00000110

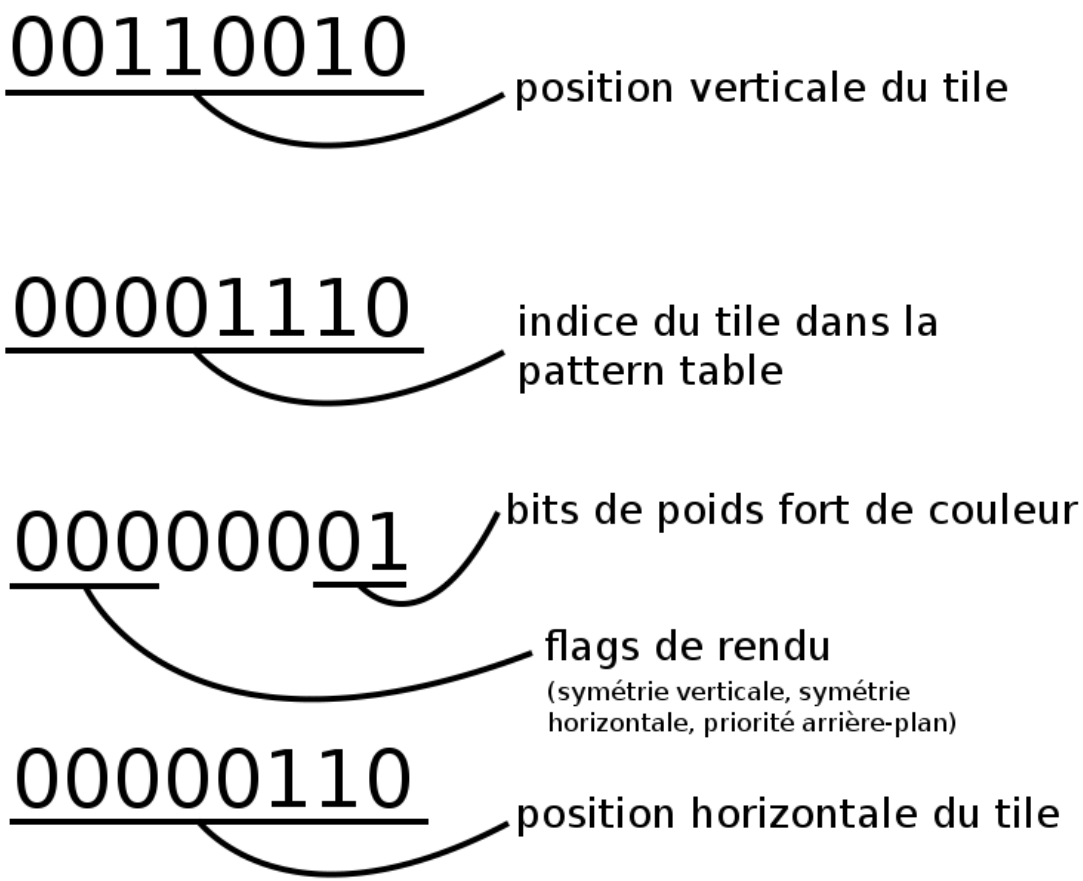
Sprite RAM

La Sprite RAM, contient 256 octets pour 64 tiles, soit 4 octets par tile.

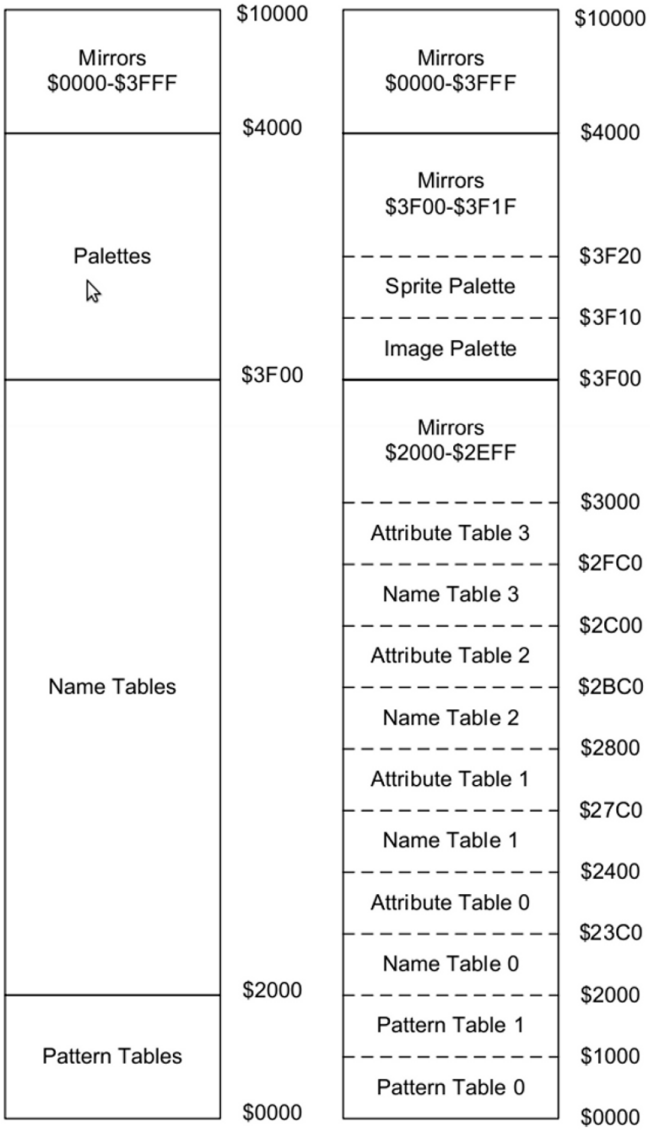


Sprite RAM

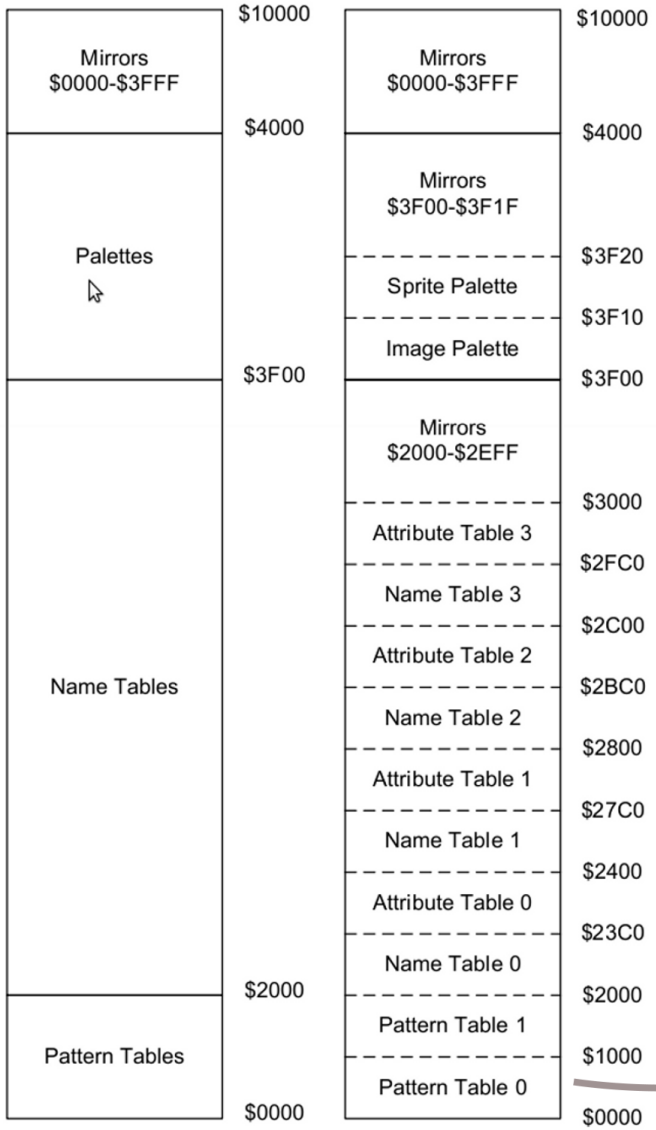
La Sprite RAM, contient 256 octets pour 64 tiles, soit 4 octets par tile.



Récapitulatif

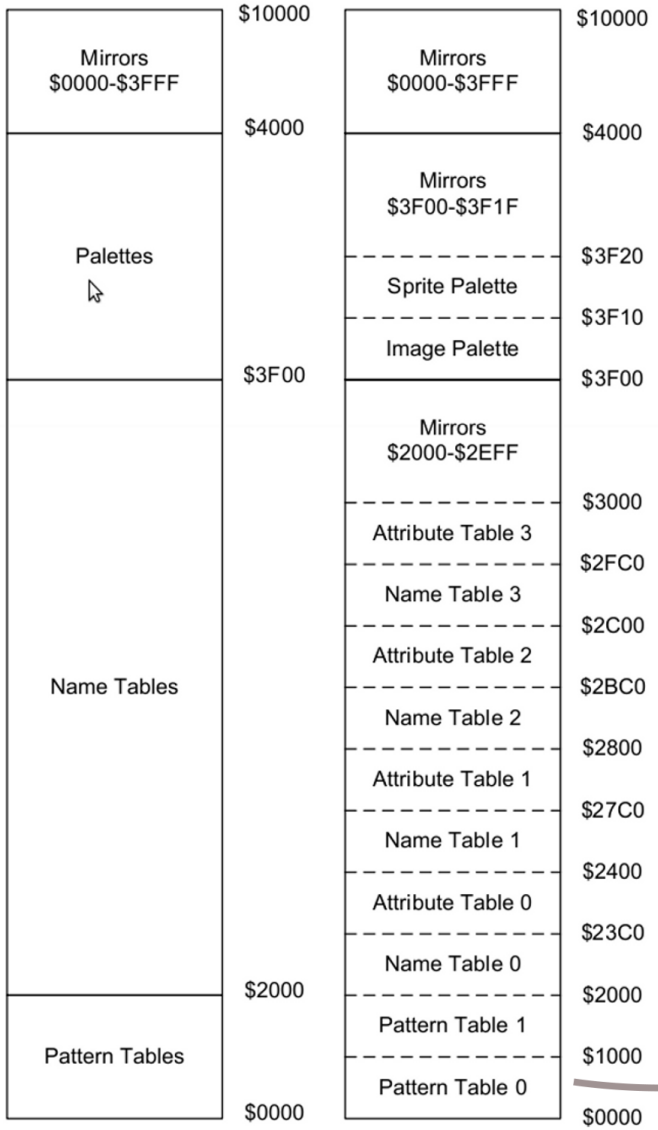


Récapitulatif

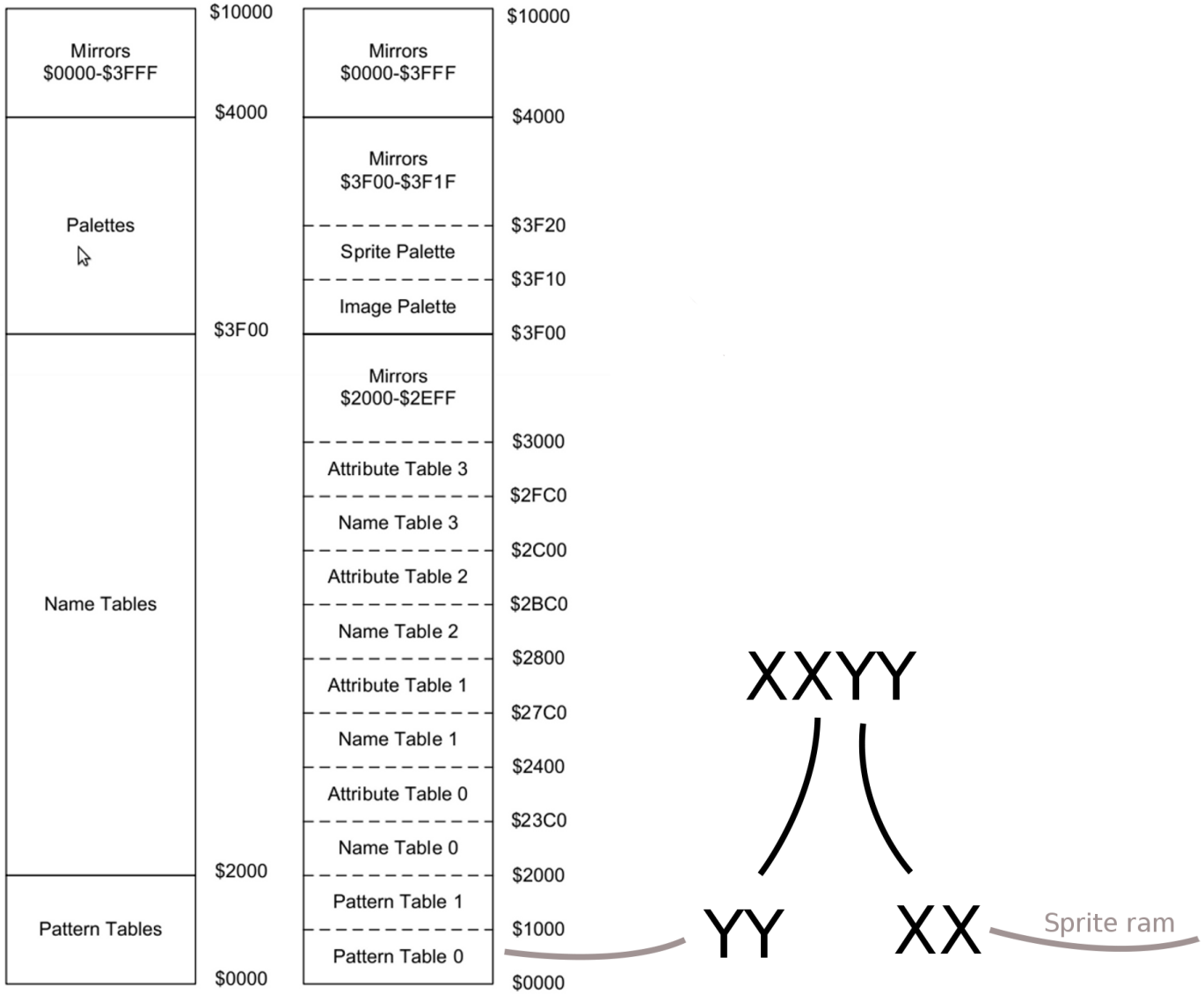


YY

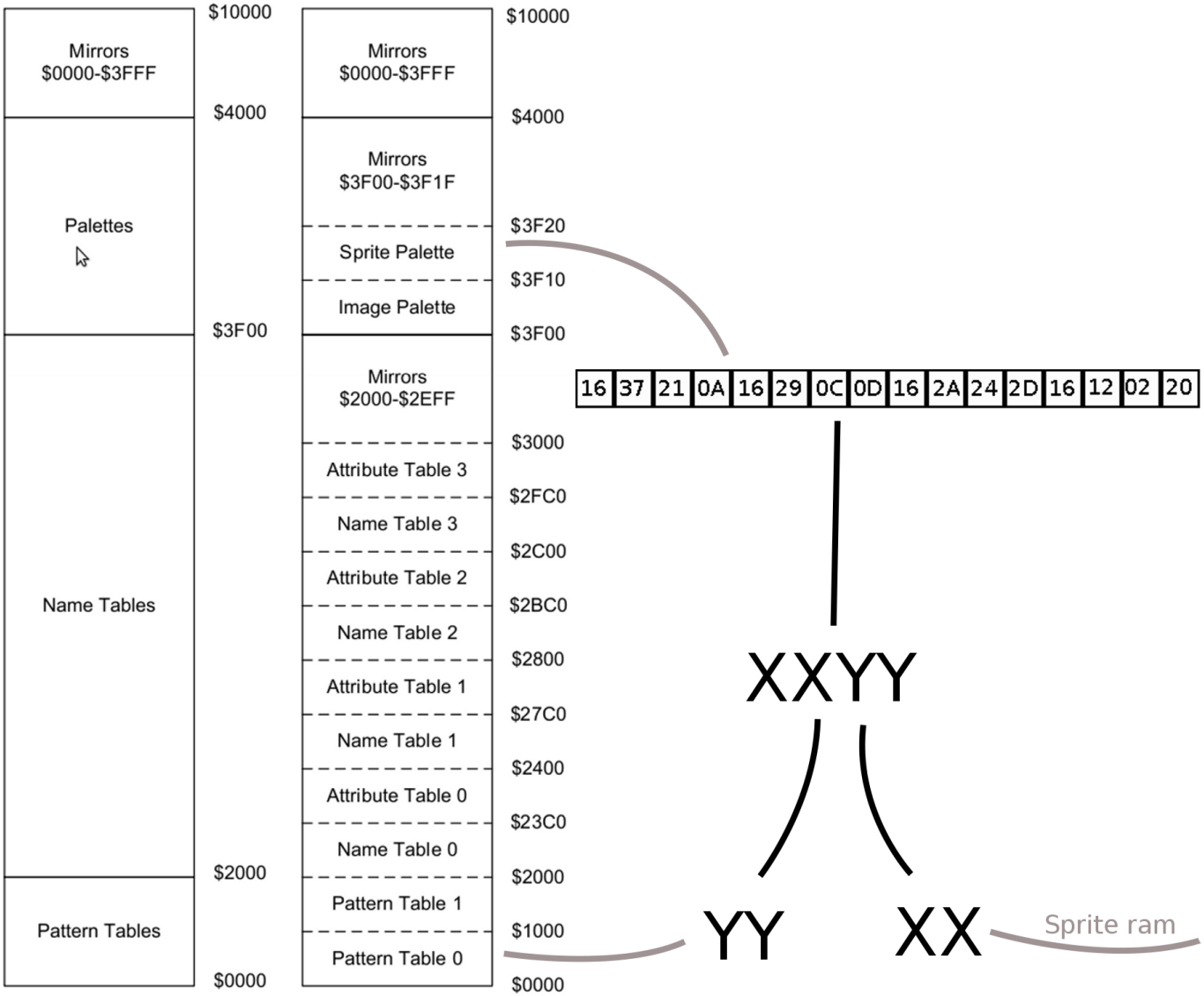
Récapitulatif



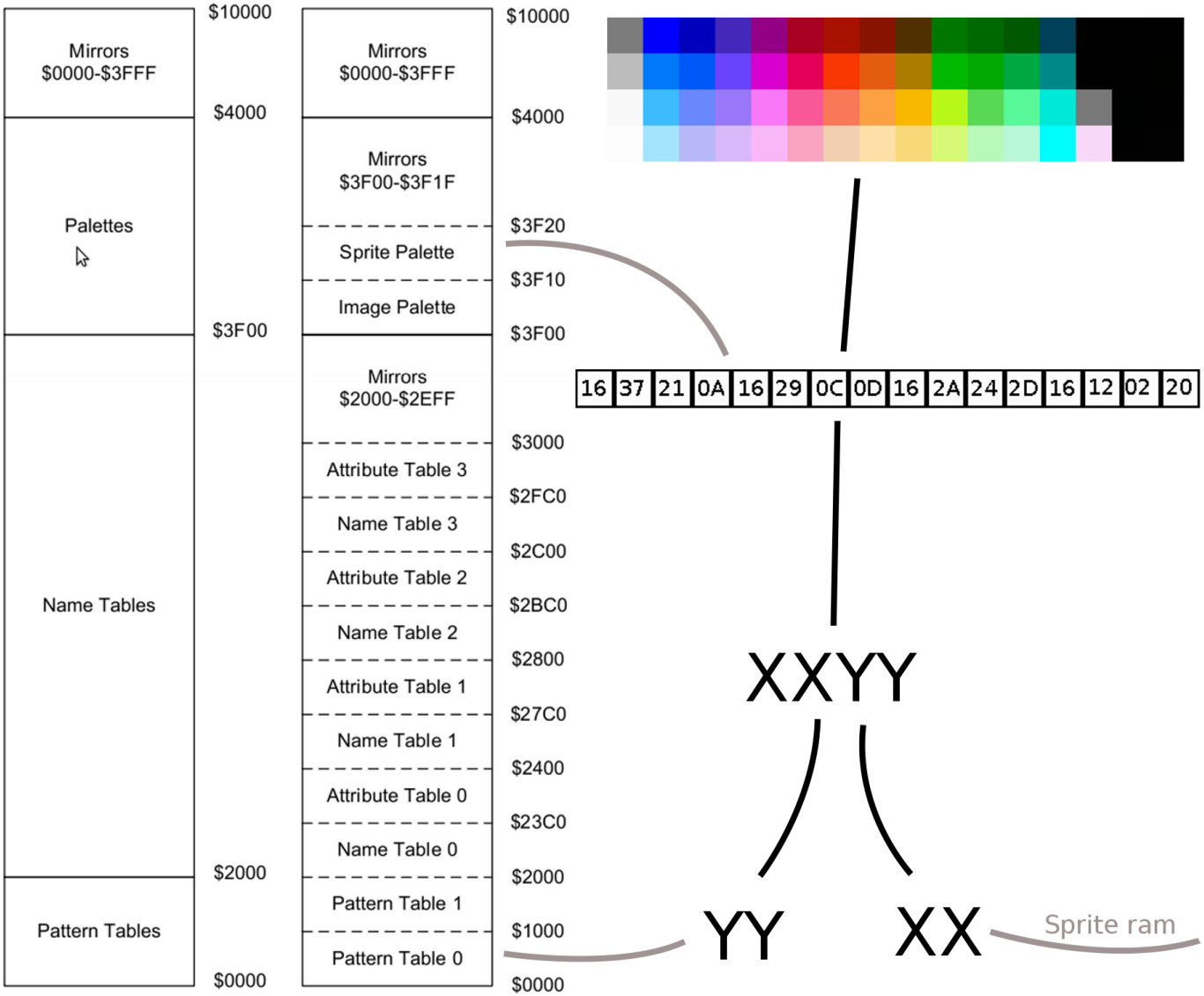
Récapitulatif



Récapitulatif



Récapitulatif



Limitation technique

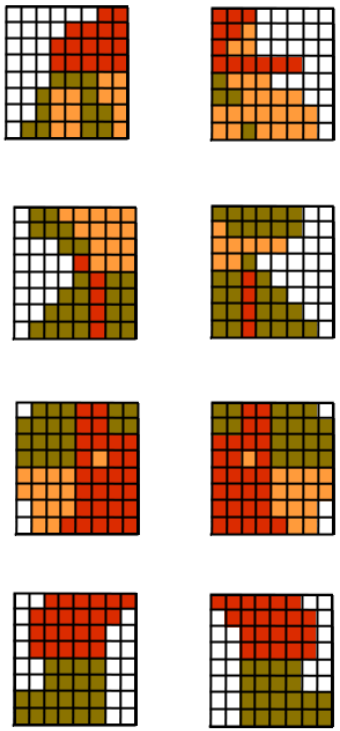
Chaque pixel de chaque tile est associé à 2 bits de poids faible de couleur mais les 2 bits de poids fort sont communs à tous les pixels d'un tile.
Un même tile ne peut donc contenir que quatre couleurs différentes.

Combinaison de plusieurs tiles

Un tile ne mesure que 8x8 pixels. On peut néanmoins en combiner plusieurs pour former un sprite plus grand.

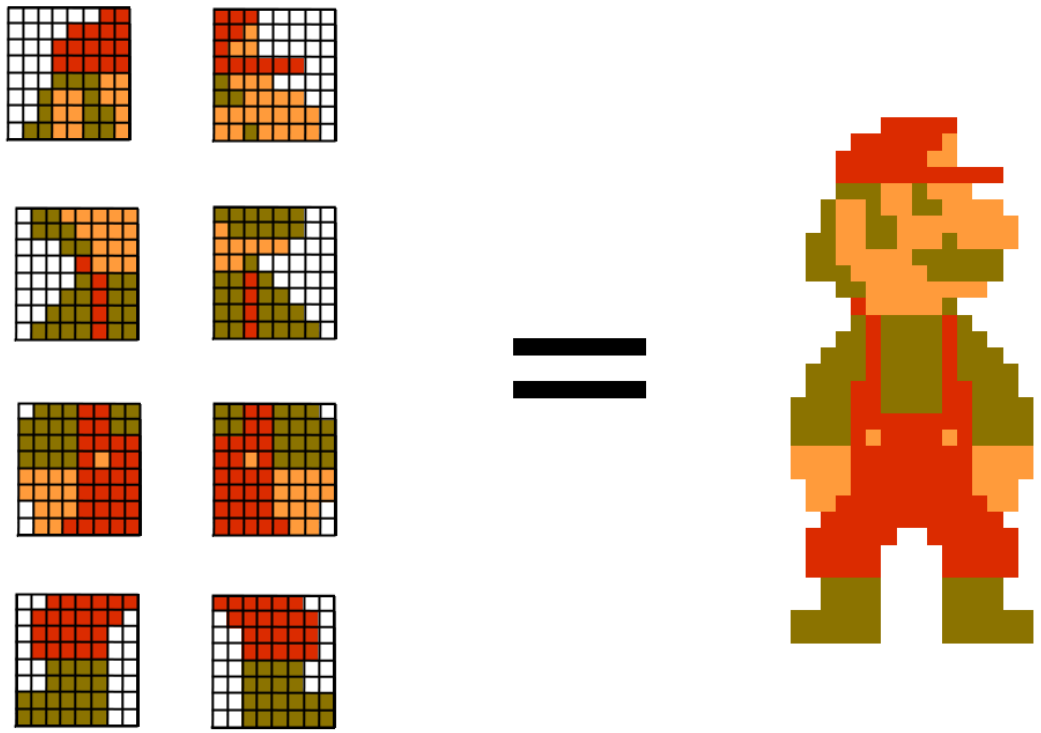
Combinaison de plusieurs tiles

Un tile ne mesure que 8x8 pixels. On peut néanmoins en combiner plusieurs pour former un sprite plus grand.

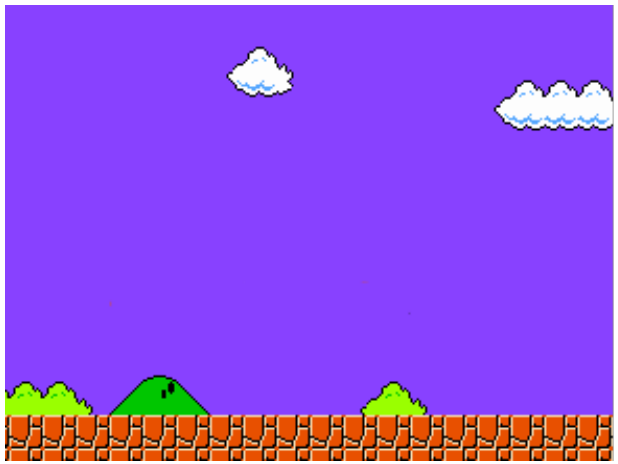


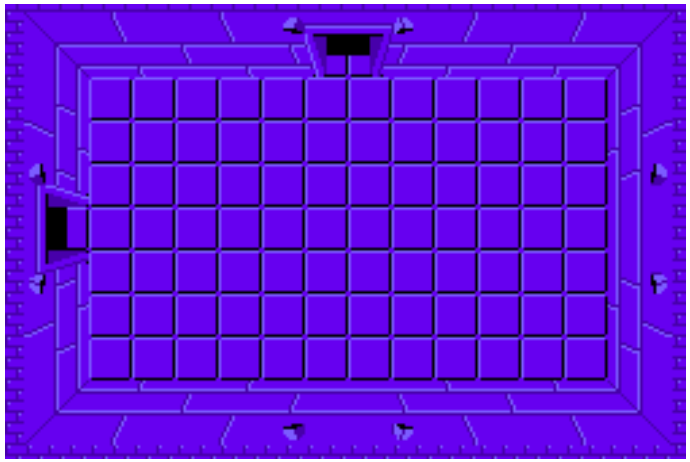
Combinaison de plusieurs tiles

Un tile ne mesure que 8x8 pixels. On peut néanmoins en combiner plusieurs pour former un sprite plus grand.



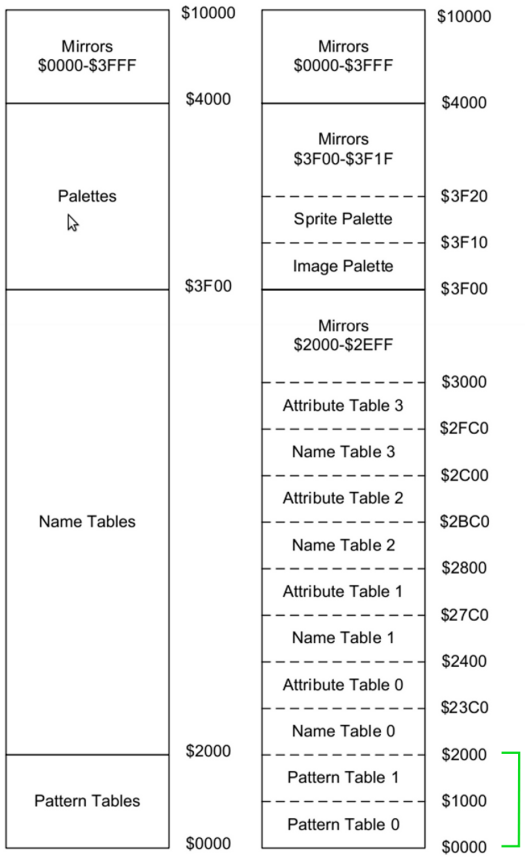
Arrière-plan





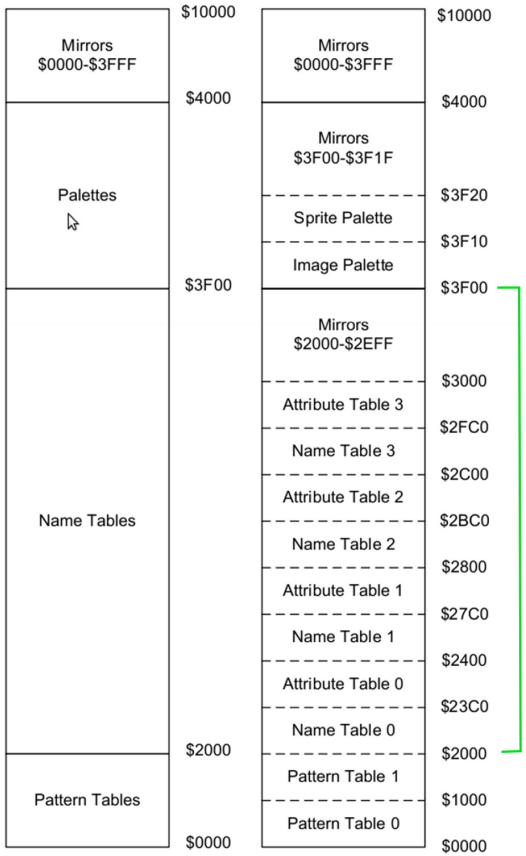
Arrière-plan en mémoire

De la même façon que pour les sprites, la moitié des informations de couleurs de l'arrière-plan est définie dans une pattern table.



Name table et attribute table

Mais le reste des informations se trouve dans les name tables et les attribute tables.

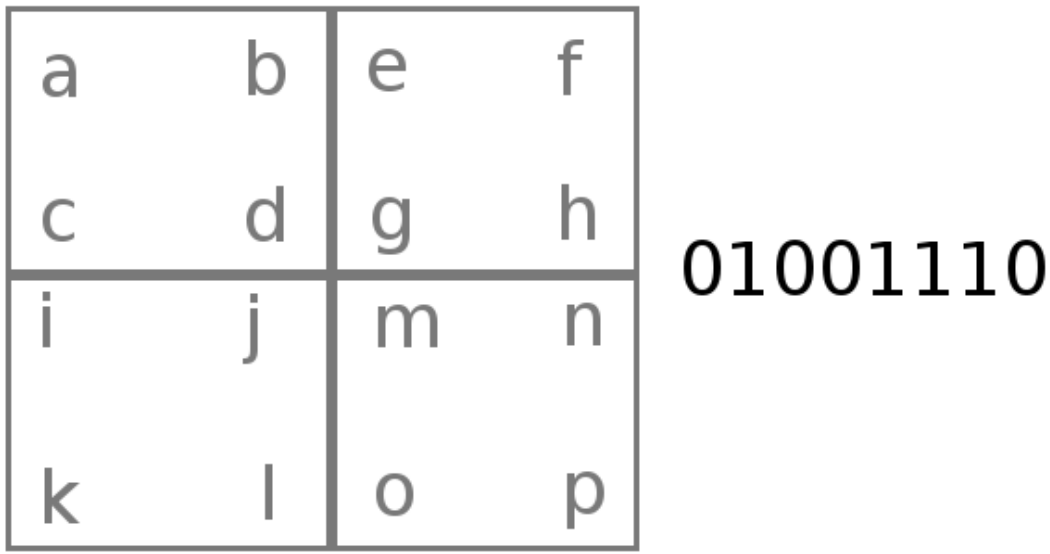


Name table

Chaque name table contient 960 octets, ce qui correspond aux 960 (30 x 32) tiles composant l'arrière-plan.
Chaque octet représente l'indice du tile dans la pattern table.
La position de chaque tile n'est pas spécifiée, ils sont affichés en fonction de leur ordre dans la name table.

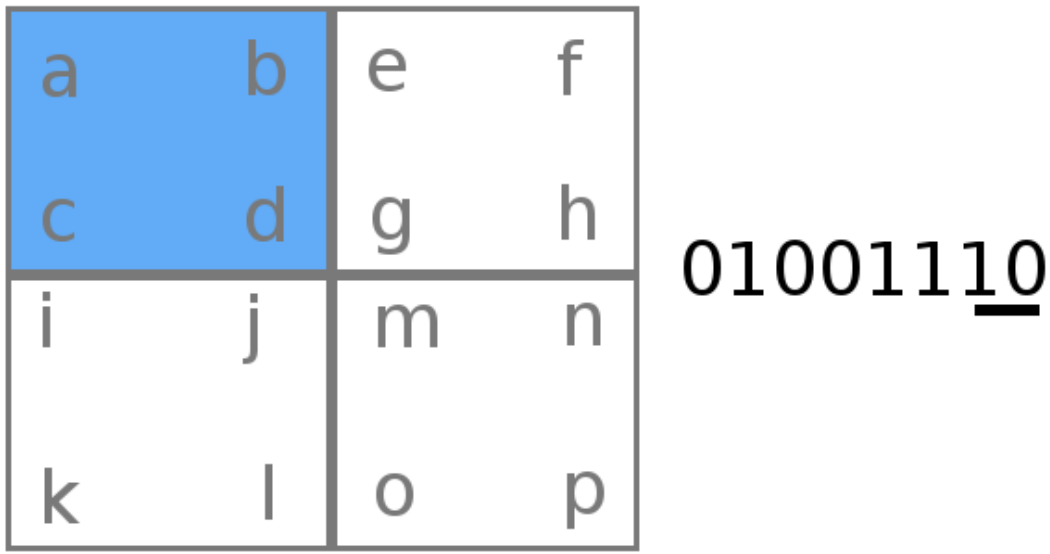
Attribute table

Chaque name table est liée à une attribute table qui contient les deux bits de poids forts indiquant la couleur.
Un attribute table ne contient que 64 octets, chaque entrée contient les bits de poids forts pour 16 tiles différents groupés par 4 (matrice de 4x4 tiles).



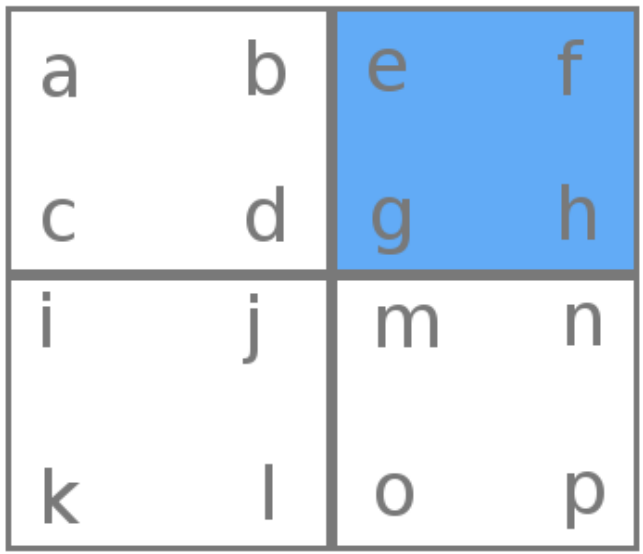
Attribute table

Chaque name table est liée à une attribute table qui contient les deux bits de poids forts indiquant la couleur.
Un attribute table ne contient que 64 octets, chaque entrée contient les bits de poids forts pour 16 tiles différents groupés par 4 (matrice de 4x4 tiles).



Attribute table

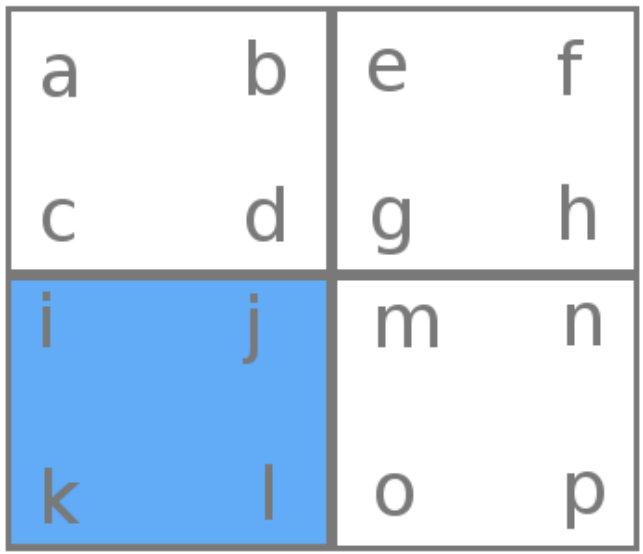
Chaque name table est liée à une attribute table qui contient les deux bits de poids forts indiquant la couleur.
Un attribute table ne contient que 64 octets, chaque entrée contient les bits de poids forts pour 16 tiles différents groupés par 4 (matrice de 4x4 tiles).



01001110

Attribute table

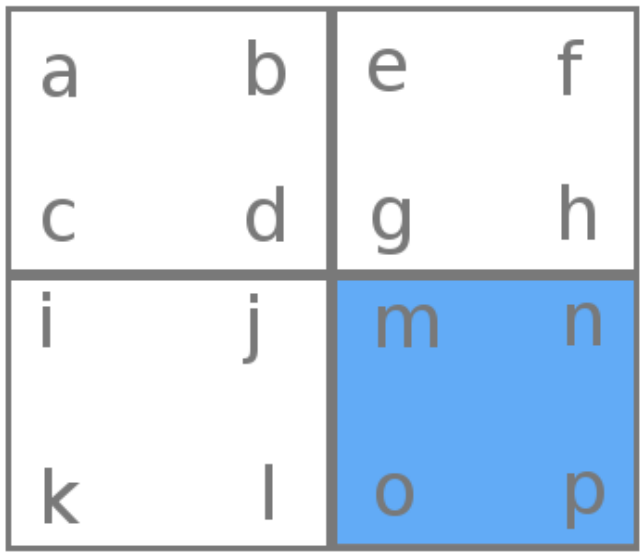
Chaque name table est liée à une attribute table qui contient les deux bits de poids forts indiquant la couleur.
Un attribute table ne contient que 64 octets, chaque entrée contient les bits de poids forts pour 16 tiles différents groupés par 4 (matrice de 4x4 tiles).



01001110

Attribute table

Chaque name table est liée à une attribute table qui contient les deux bits de poids forts indiquant la couleur.
Un attribute table ne contient que 64 octets, chaque entrée contient les bits de poids forts pour 16 tiles différents groupés par 4 (matrice de 4x4 tiles).



01001110

Limitation technique

Comme pour les sprites, un tile d'arrière-plan ne peut contenir que quatre couleurs différentes.

4 Appendices

- Sources
- Fin

Documentation

2A03 technical reference, Brad Taylor, http://nesdev.parodius.com/2A03%20technical%20reference.txt
NES ASM Tutorial, Brian 'bunnyboy' Parker, http://www.nespowerpak.com/nesasm/nesasm.pdf
6502 Guide, Inconnu, http://nesdev.parodius.com/6502guid.txt
Nintendo Entertainment System Documentation, Patrick Diskin, http://nesdev.parodius.com/NESDoc.pdf
How NES Graphics Work, Inconnu, http://nesdev.parodius.com/nesgfx.txt
Nintendo Entertainment System Documentation, JDC, http://tuxnes.sourceforge.net/nestech100.txt
Nintendo Entertainment System Architecture, Marat Fayzullin, http://fms.komkon.org/EMUL8/NES.html

Images

X-Ray Funnies, Reintji, http://www.flickr.com/photos/ravanderende/
Video Games, FatLittleNick, http://www.flickr.com/photos/fatlittlenick/sets/72157615644355863/
Molecular Expressions Chip Shots, http://micro.magnet.fsu.edu/chipshots/mos/
NES game covers, http://en.wikipedia.org/wiki/Category:NES_game_covers