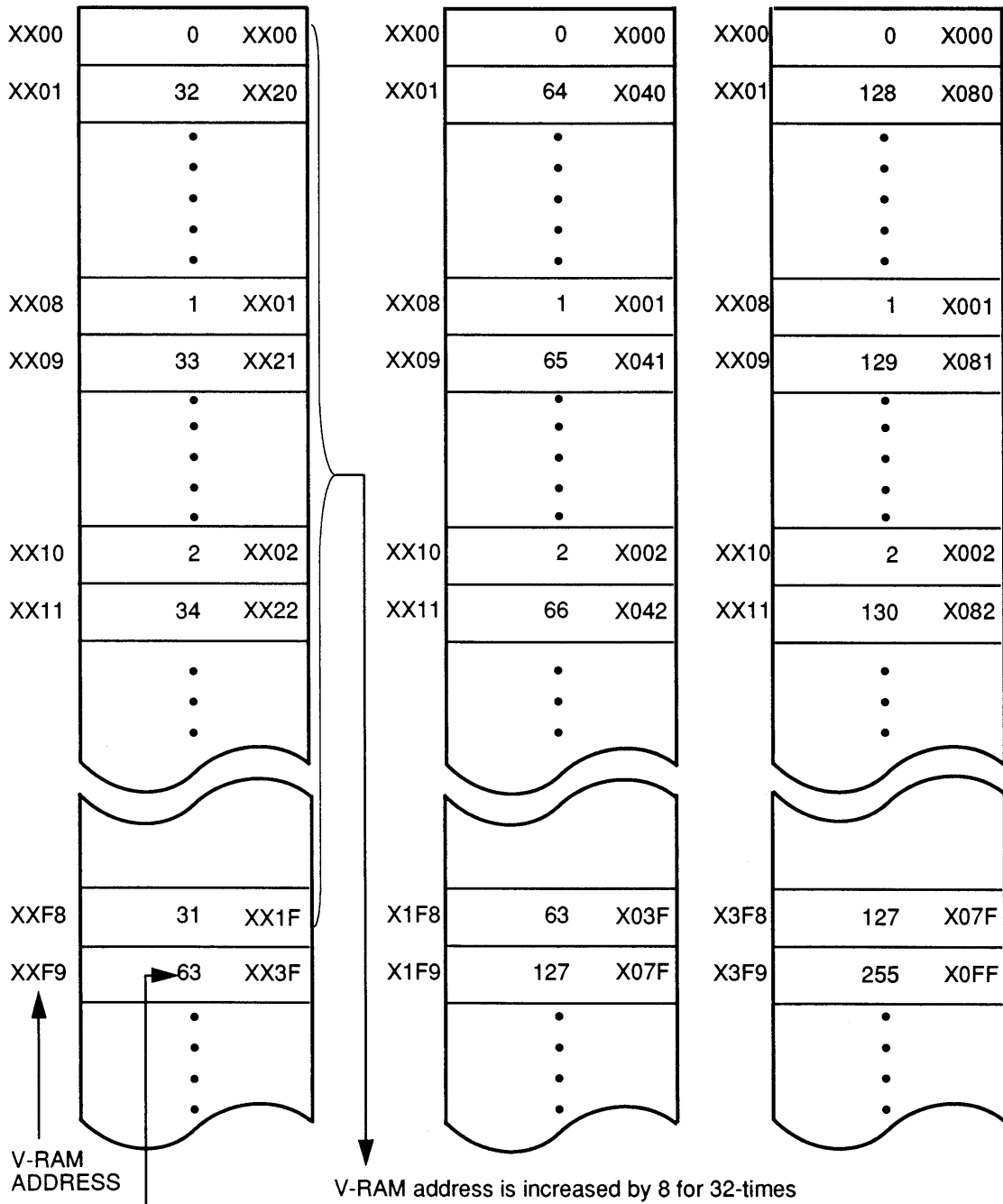


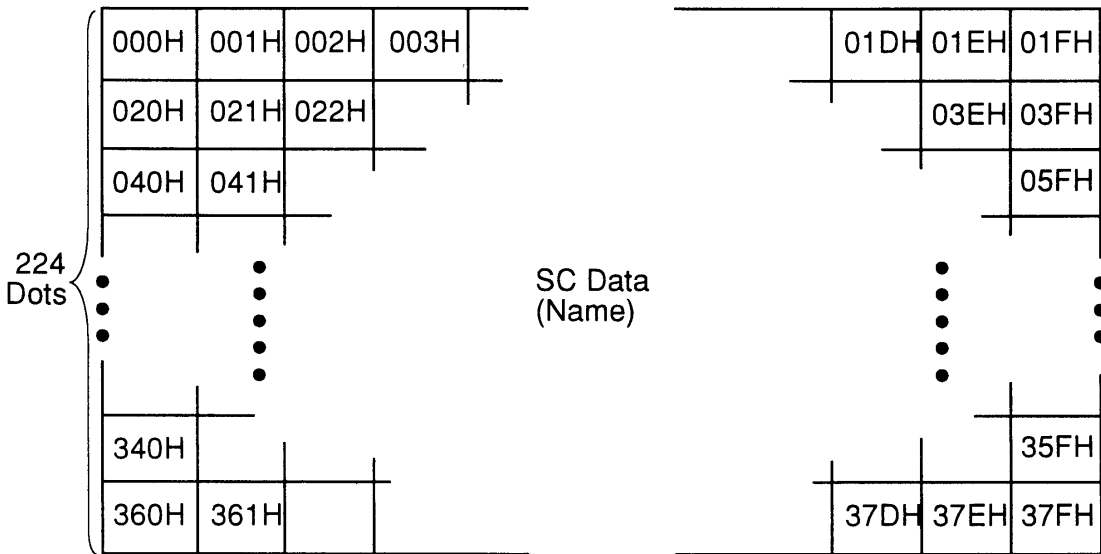
ADDRESS INCREMENT ORDER

V-RAM ADDRESS INCREMENT by 8, (32-TIMES) INCREMENT by 8, (64-TIMES) INCREMENT by 8, (128-TIMES)
 (G0 = 1, G1 = 0) (G0 = 0, G1 = 1) (G0 = 1, G1 = 1)



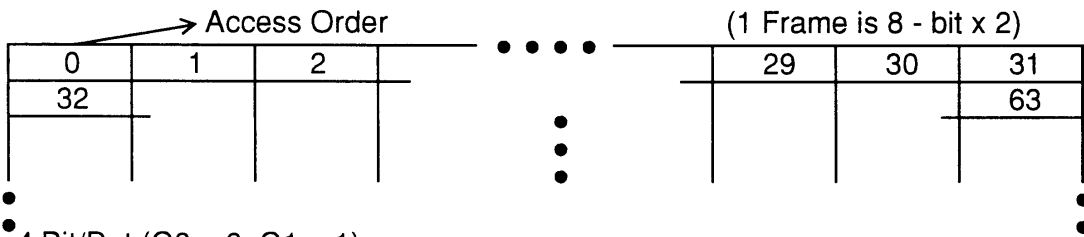
ORDER OF ACCESS (value designated by register <2116H> <2117H>)

(NCL PG 74)

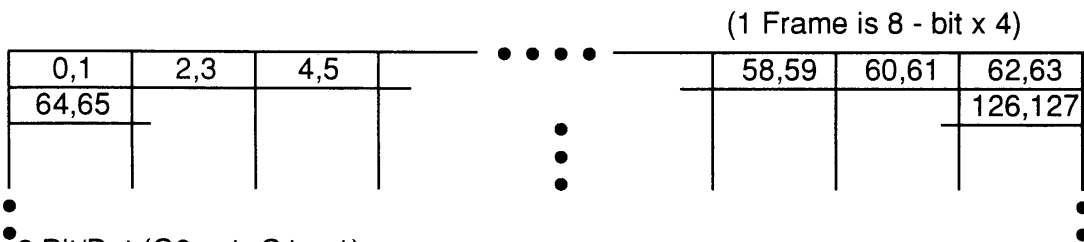


When SC data (Name) is set during BG Mode 0 ~ 6 as demonstrated in the table above, Character data accesses horizontally by 8 dots in Full Graphic (G0, G1) of register <2115H>.

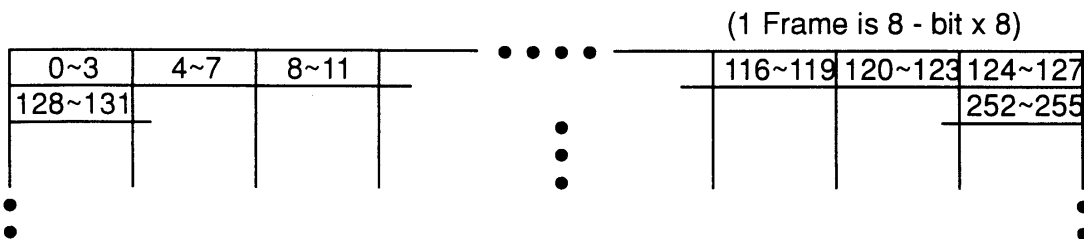
2 Bit/Dot (G0 = 1, G1 = 0)



4 Bit/Dot (G0 = 0, G1 = 1)

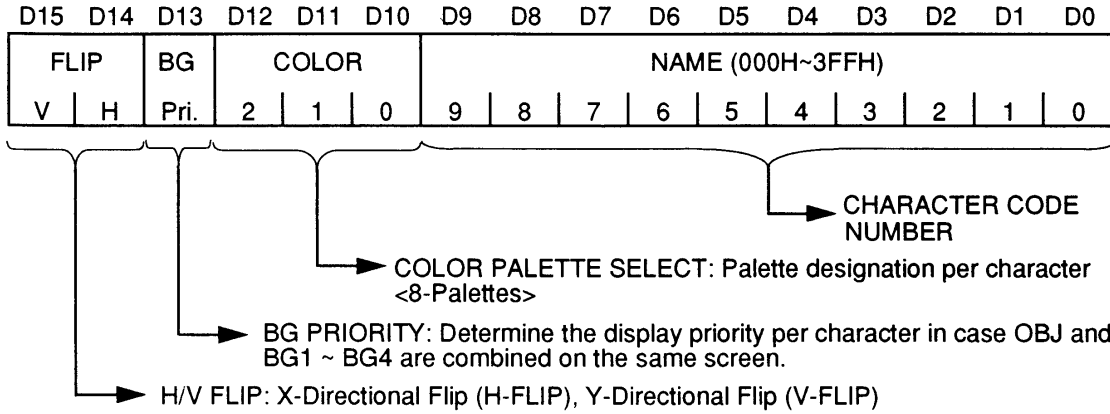


8 Bit/Dot (G0 = 1, G1 = 1)



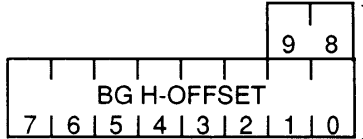
(NCL PG 74a)

BG SC DATA (MODE 0 ~ 6)

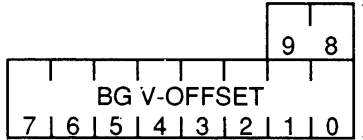


BG SCREEN H/V SCROLL

REGISTER <210DH> <210FH>
<2111H> <2113H>

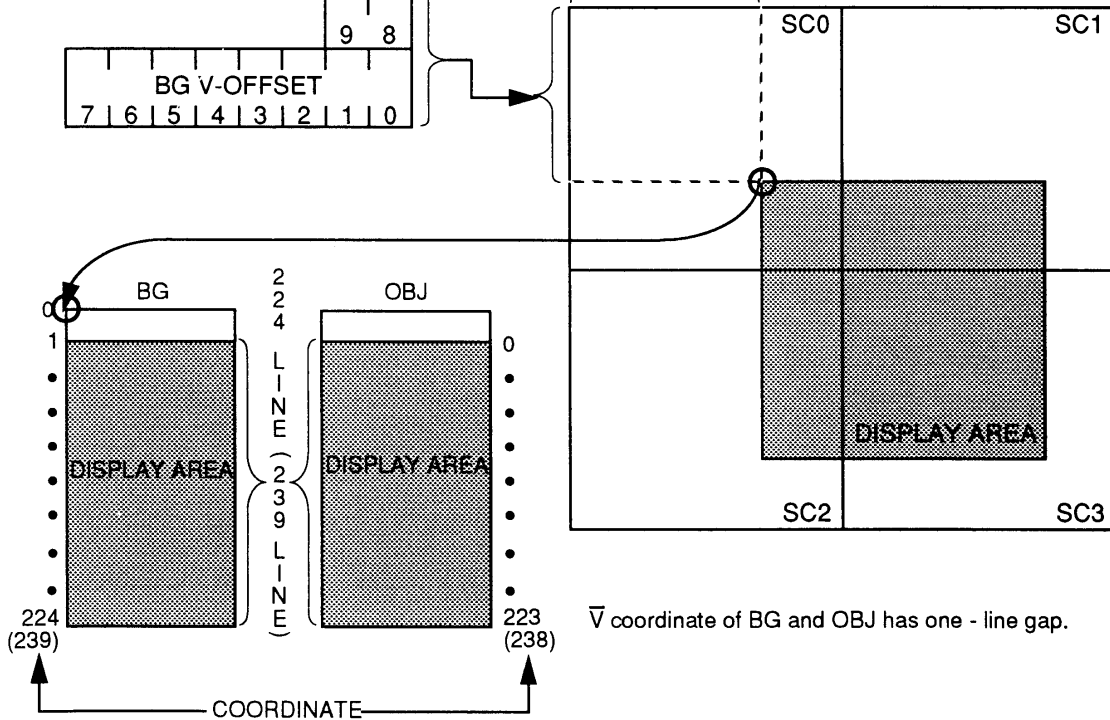


REGISTER <210EH> <2110H>
<2112H> <2114H>



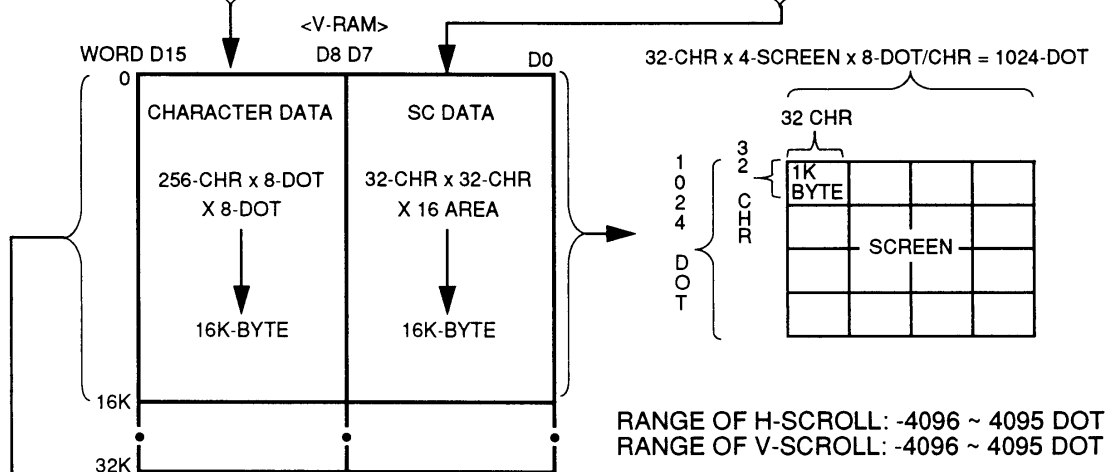
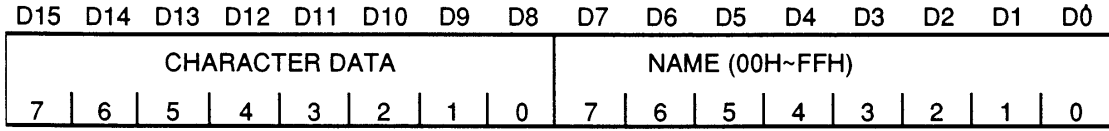
RANGE OF H-SCROLL
0 ~ 1023 DOT
RANGE OF V-SCROLL
0 ~ 1023 DOT

H/Vscroll range may be changed and 2-dot scroll may be possible, depending on the combination of the modes (512, 16-size, interlace, etc.). Also, SC size may be changed against the screen. (Page A-21 and A-22)



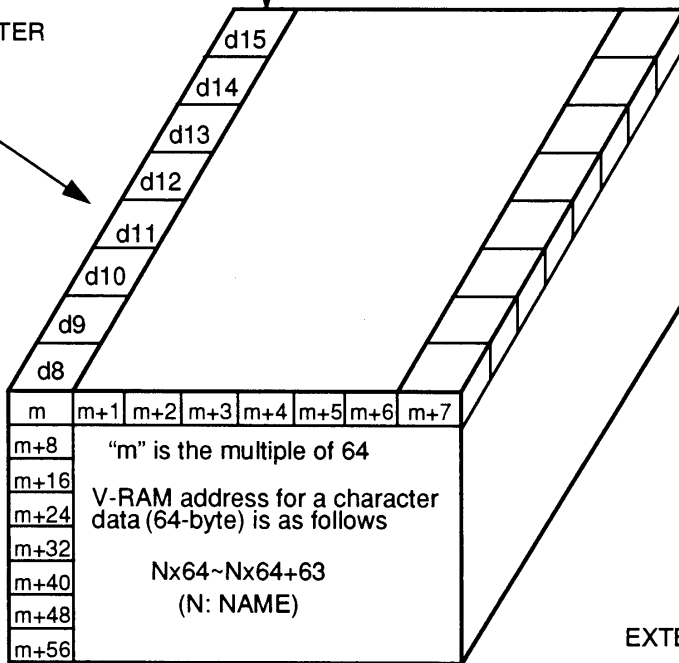
(NCL PG 75)

BG SC DATA (MODE 7)

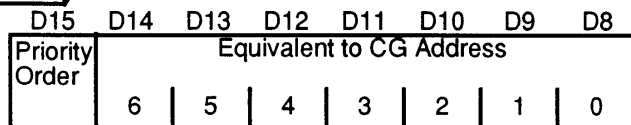


In case of EXTBG mode, D15 will be BG order (Display Priority Order).

DATA FORMAT FOR A CHARACTER



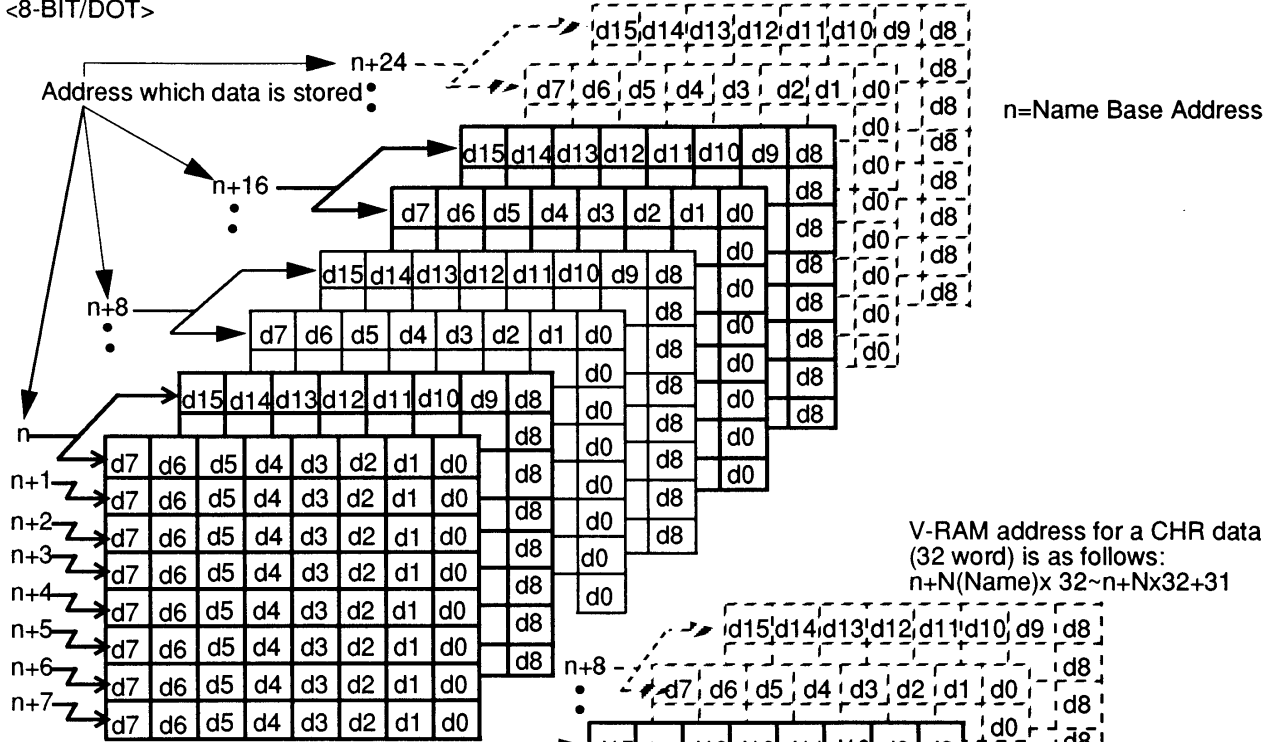
EXTBG MODE



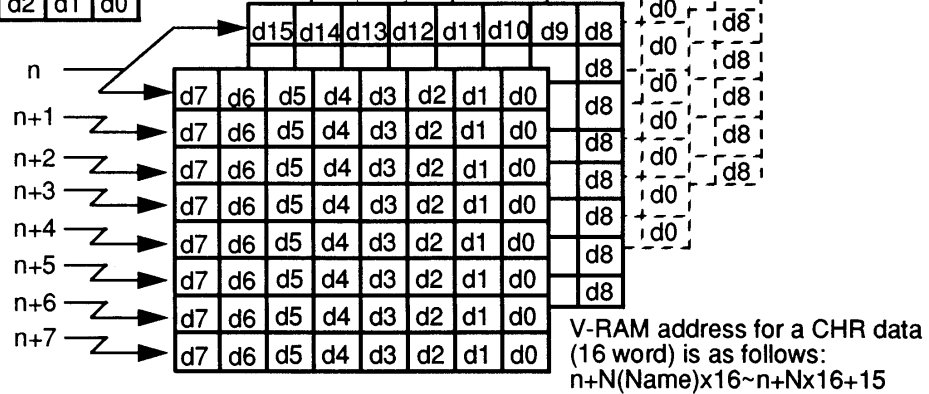
(NCL PG 76)

CHR DATA CONSTRUCTION

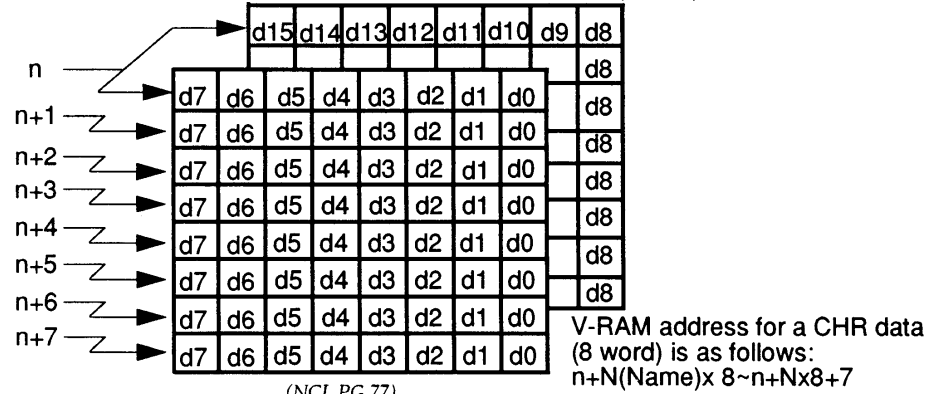
<8-BIT/DOT>



<4-BIT/DOT>



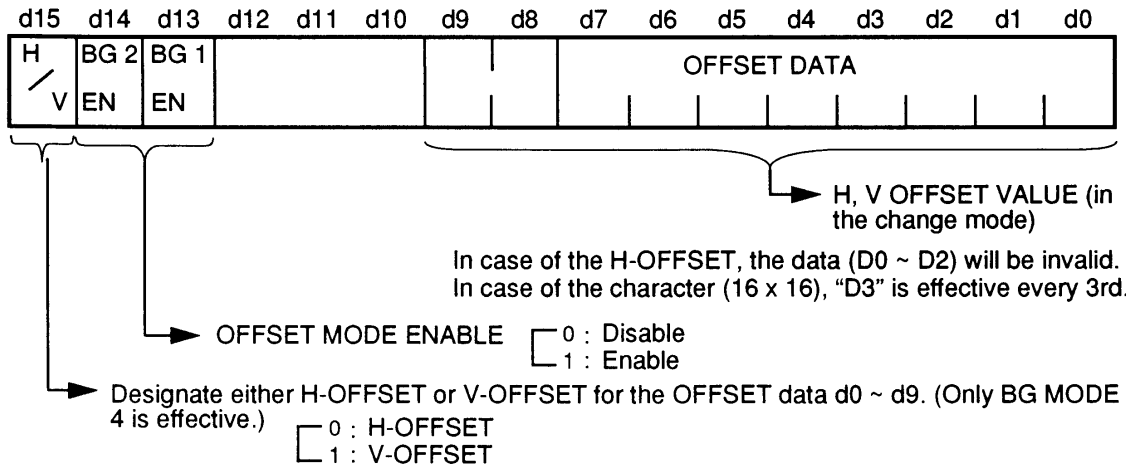
<2-BIT/DOT>



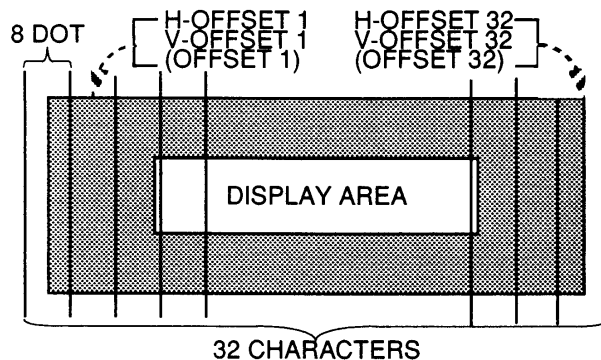
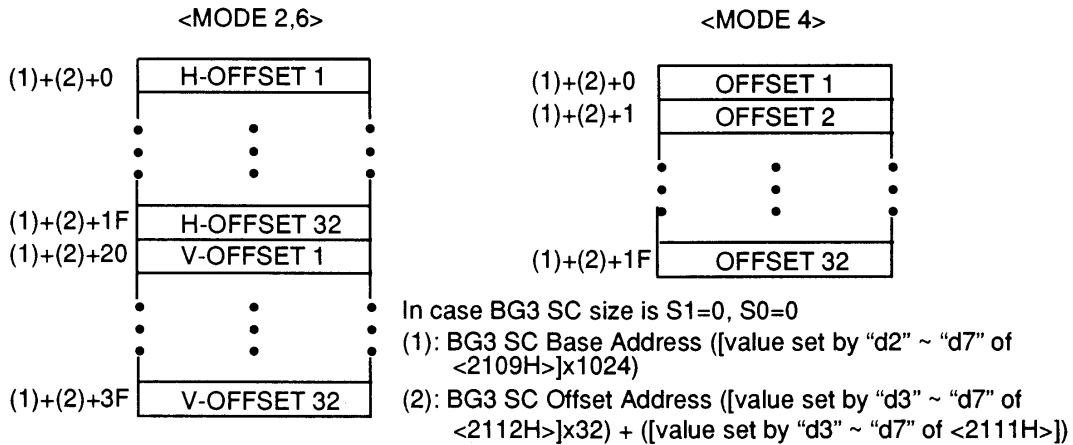
(NCL PG 77)

OFFSET CHANGE MODE

The offset change mode can be used in the BG mode 2, 4 and 6, and the following data is required in this mode.

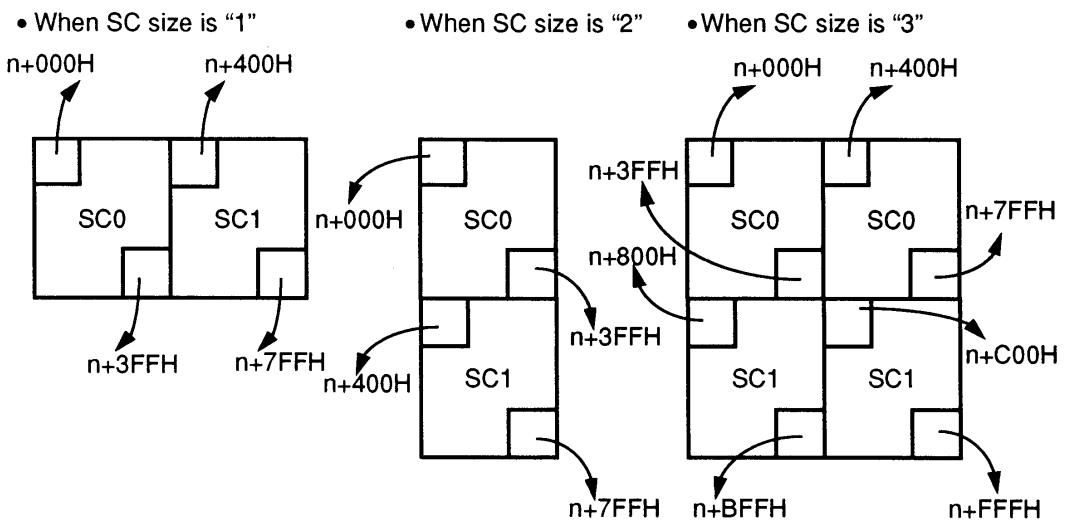
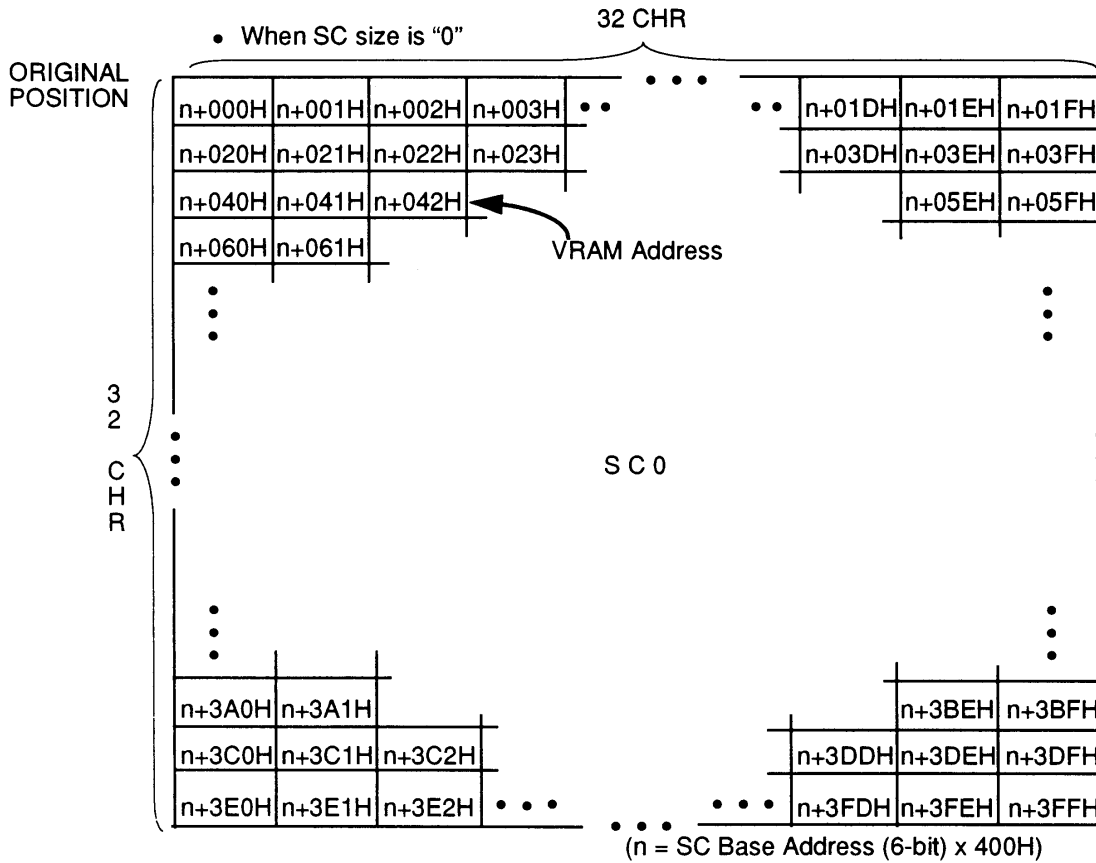


Write this data to VRAM of address designated at (1) and (2), using the BG Mode. (See below.)



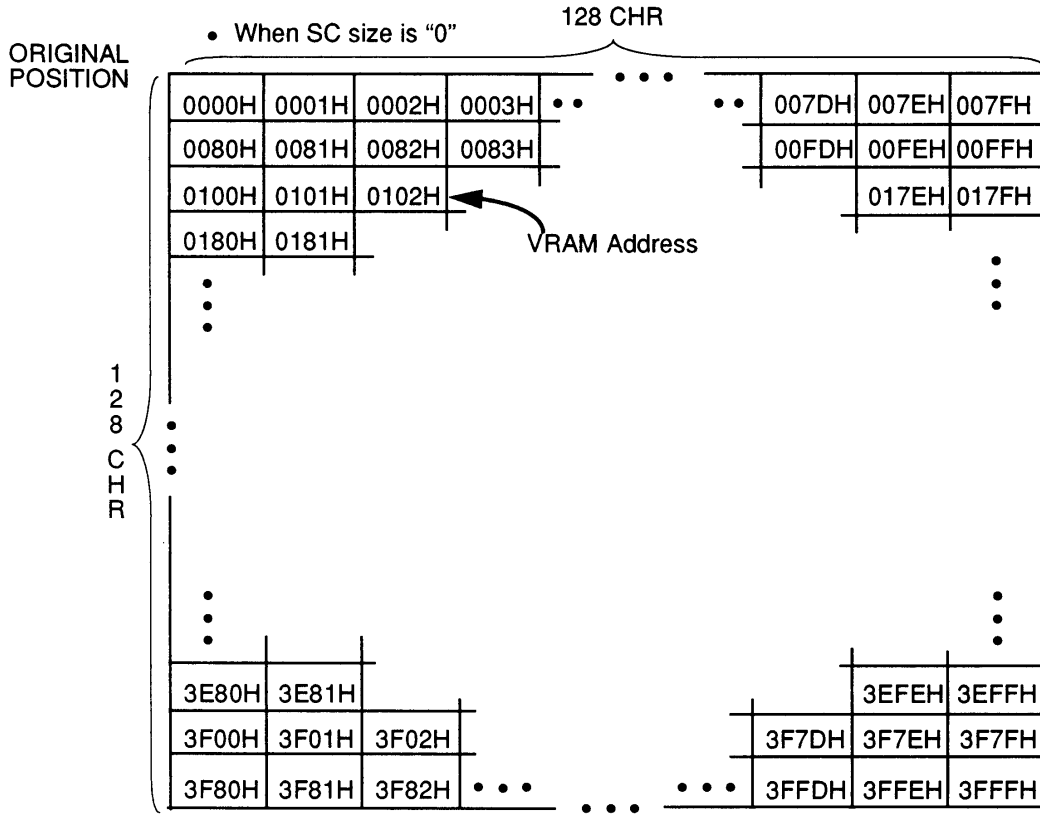
The offset value can be changed by each column (character unit).
 (Up to 3rd character can be seen horizontally on the screen by setting the offset value of the entire screen, but the offset can not be changed for 1st character (0 character).)
 (NCL PG 78)

BG SCREEN (BG Mode 0 ~ 6)



(NCL PG 79)

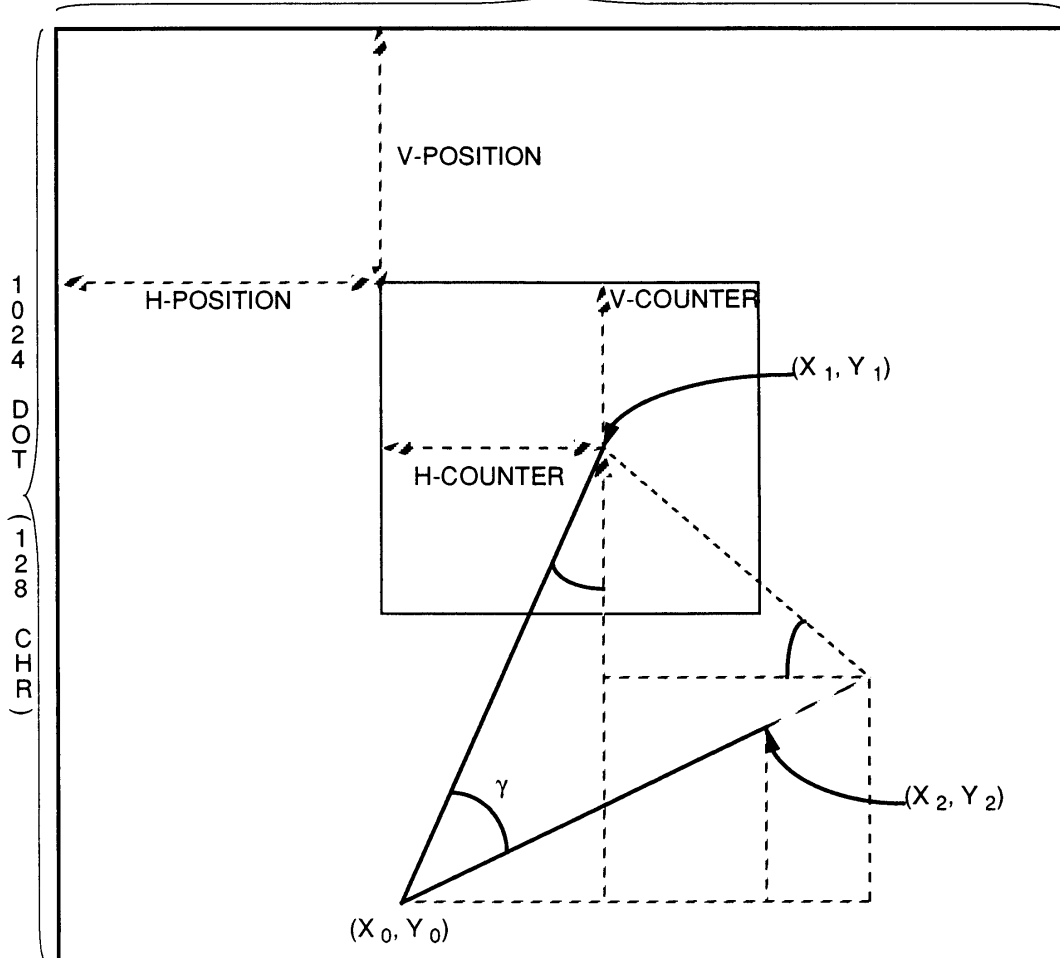
BG Screen (BG Mode 7)
Screen Size and Area are Fixed



(NCL PG 79a)

OPERATION (ROTATION/ENLARGEMENT/REDUCTION)

1024 DOT (128 CHR)



ROTATIONAL TRANSFORM FORMULA

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \cos \gamma & \sin \gamma \\ -\sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} X_1 - X_0 \\ Y_1 - Y_0 \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$X_0 \bullet Y_0$: Center Coordinate

$X_1 \bullet Y_1$: Display Coordinate

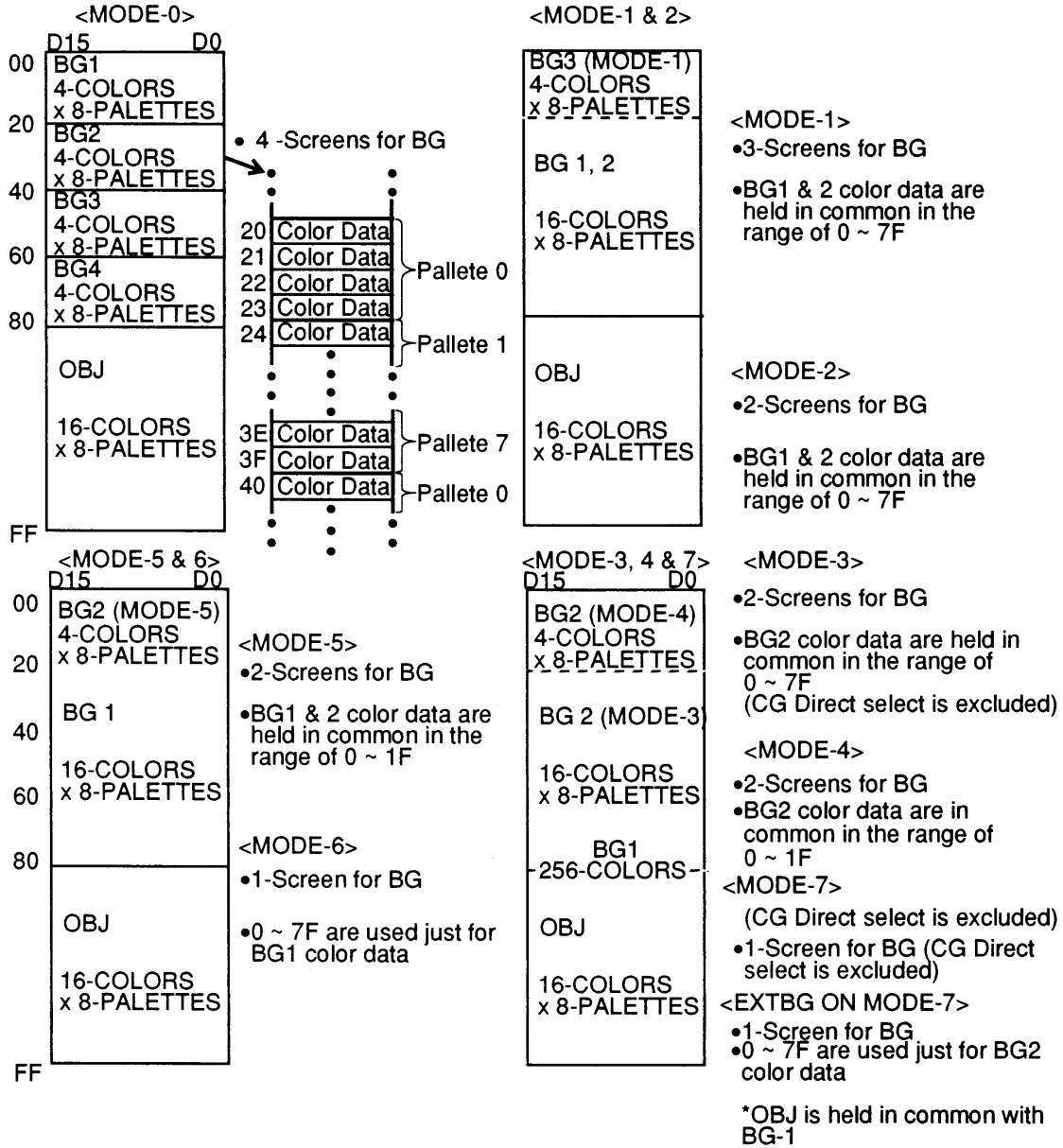
$X_2 \bullet Y_2$: Coordinate before calculation of display coordinate

If the reduction rates for X-dir (α) and the reduction rates for Y-dir (β) are considered, the formula described above will be as follows:

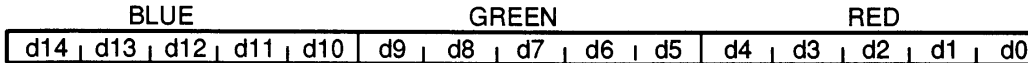
$$\begin{aligned} A &= \cos \gamma \times (1/\alpha), & B &= \sin \gamma \times (1/\alpha), \\ C &= -\sin \gamma \times (1/\beta), & D &= \cos \gamma \times (1/\beta), \end{aligned}$$

(NCL PG 80)

CG-RAM



CG-RAM COLOR DATA



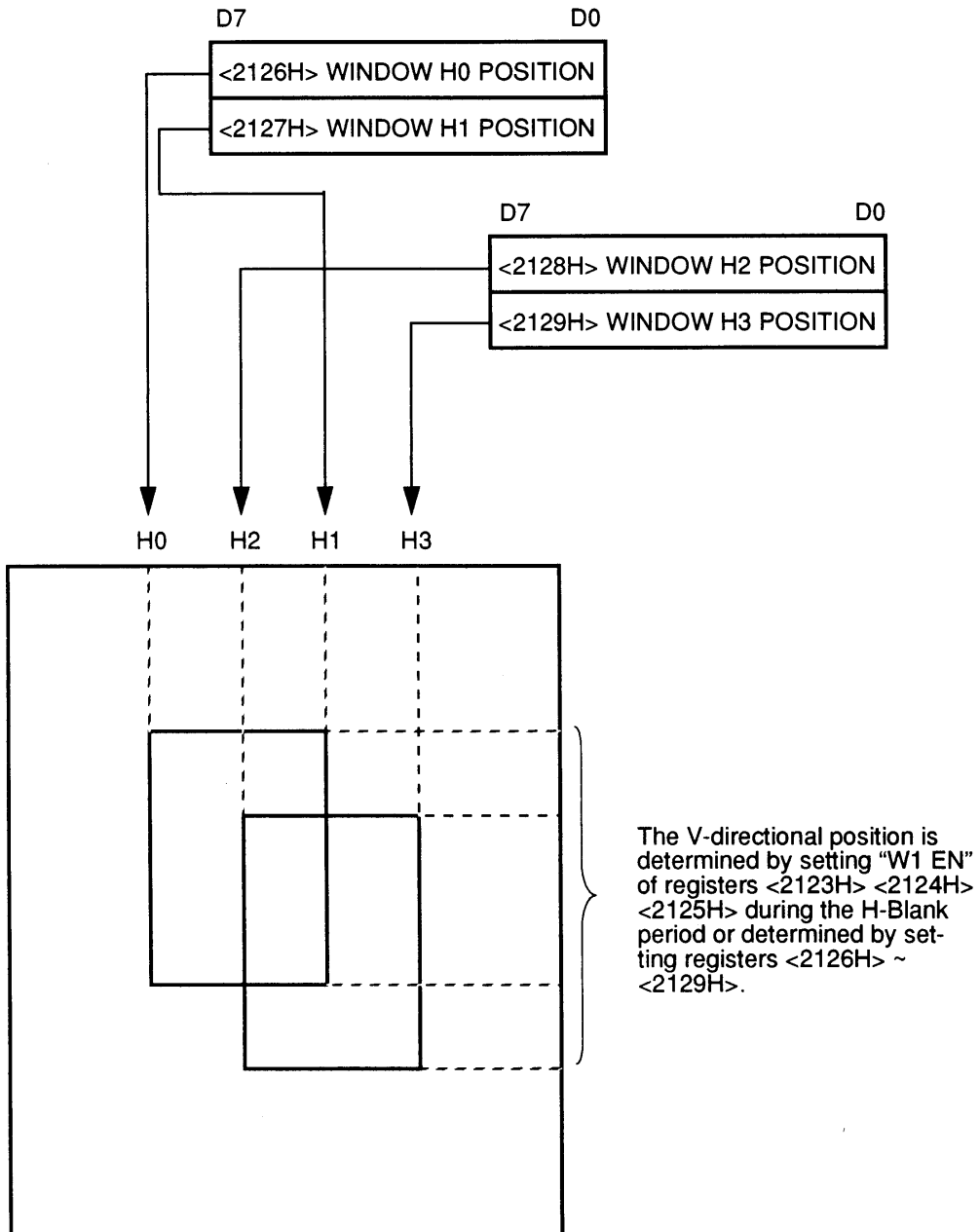
DIRECT SELECT COLOR DATA



NOTE: DA0 ~ DA7 are used for the character dot data. CL0 ~ CL2 are used for the BG-SC data of the color. (However, in case of Mode-7, CL0 ~ CL2 should be "0")
 NOTE: If they are "0," it becomes transparent. The color of CG-RAM address (00H) will be background.
 (NCL PG 81)

WINDOW

REGISTER <2126H> ~ <2129H>

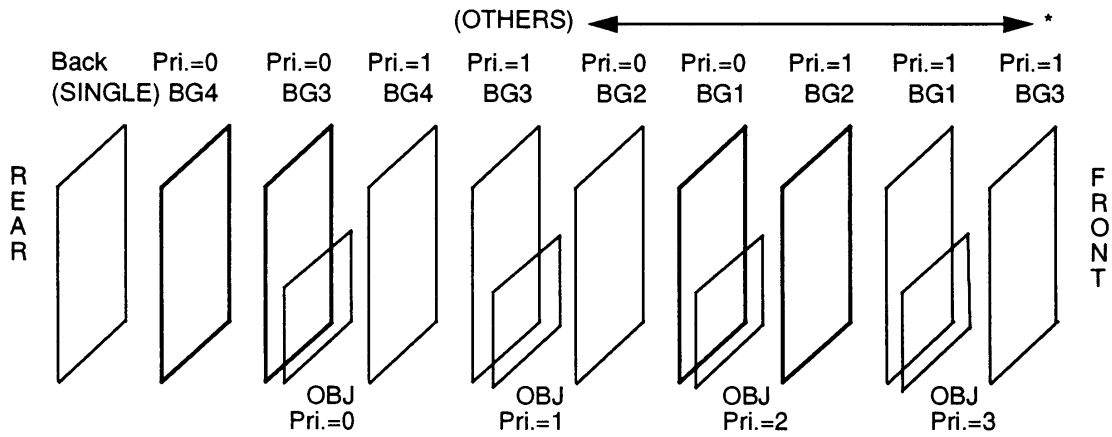


(NCL PG 82)

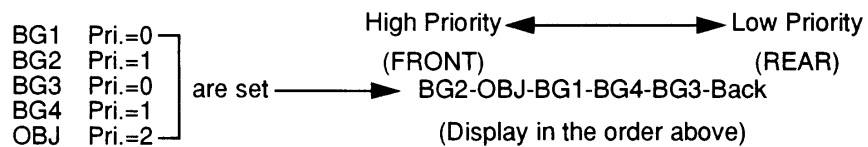
BG & OBJ PRIORITY

4-SCREEN/3-SCREEN MODE (In case Mode 0 and 1 are selected by register <2105H>)

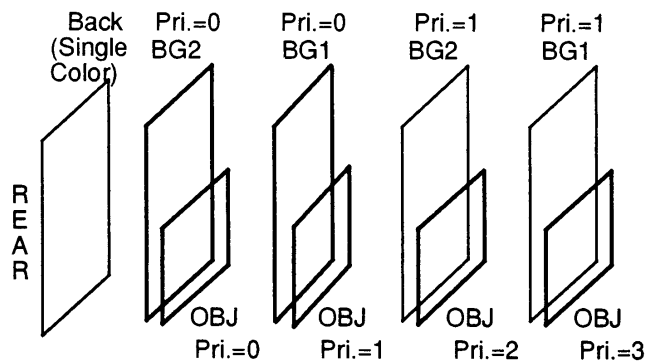
*In case "D3=1" is selected by register <2105H> in the mode-1



<Example of Display Priority (in case of mode 0)>



2-SCREEN/1-SCREEN MODE (in case Mode 2 ~ 7 is selected by register <2105H>)

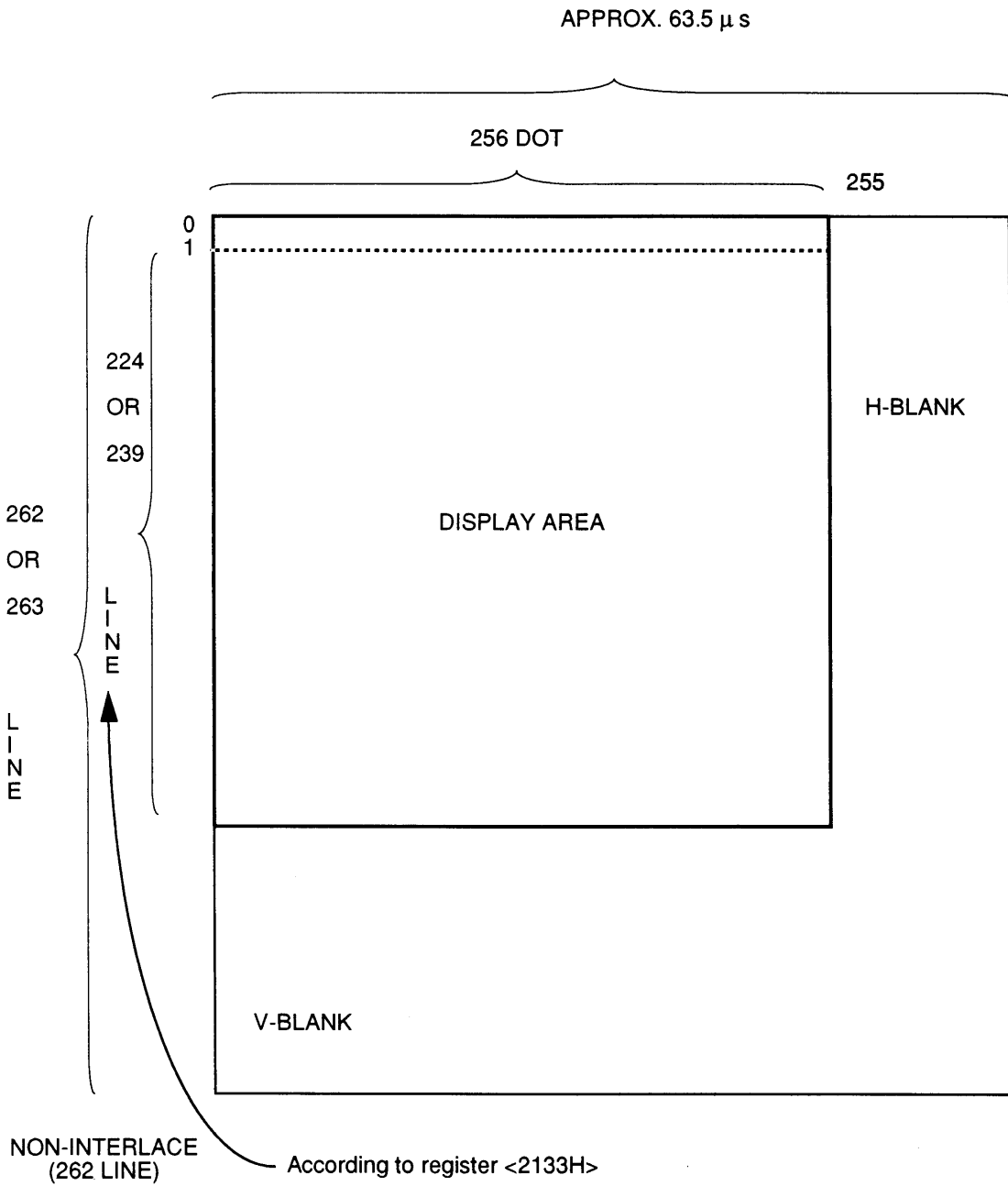


NOTE: In case of the display priority between the OBJ's, normally the lower numbered OBJ will be displayed as higher priority. (See page 1-20-2 for exception.)
This display priority will be determined before the priority between OBJ and BG is determined.

NOTE: In case of Mode 7, the priority is 0 at BG1.

(NCL PG 83)

SCREEN



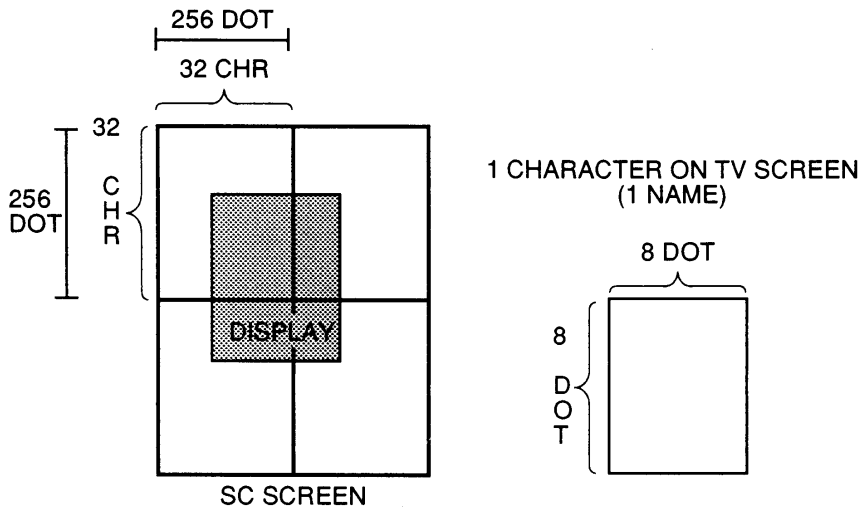
(NCL PG 84)

BG SCREEN

H/V SCROLL ① (Scroll range by the combination of modes and SC size against screen)
 <Example: in case SC size is "3" - refer to register 2107H ~ 210AH>

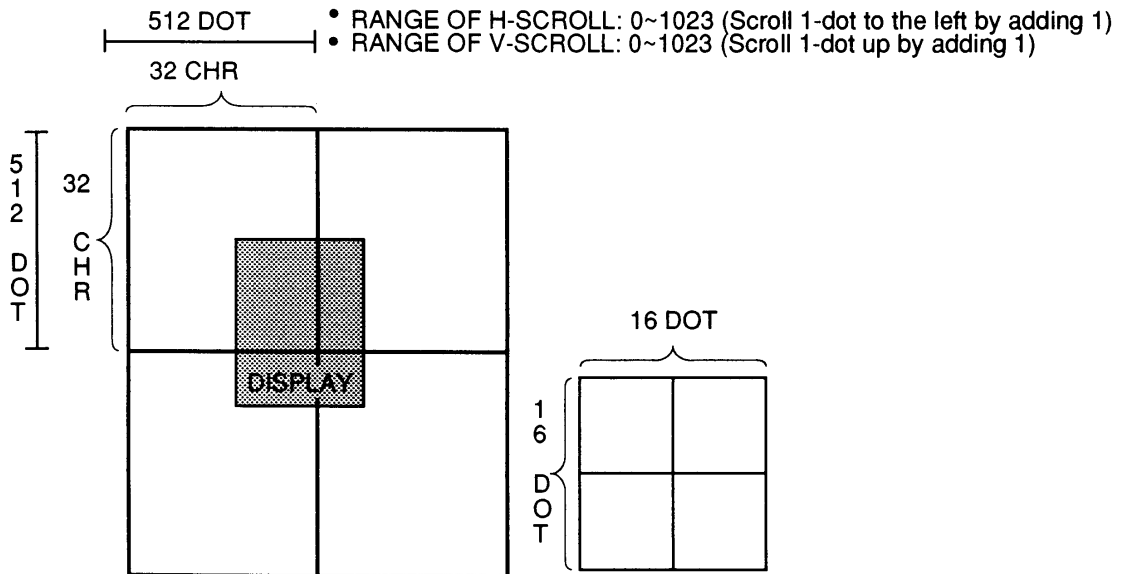
*In case of mode 0, 1, 2, 3, & 4

- BG SIZE (8 x 8)



- RANGE OF H-SCROLL: 0~511 (Scroll 1-dot to the left by adding 1)
- RANGE OF V-SCROLL: 0~511 (Scroll 1-dot up by adding 1)

- BG SIZE (16 x 16)



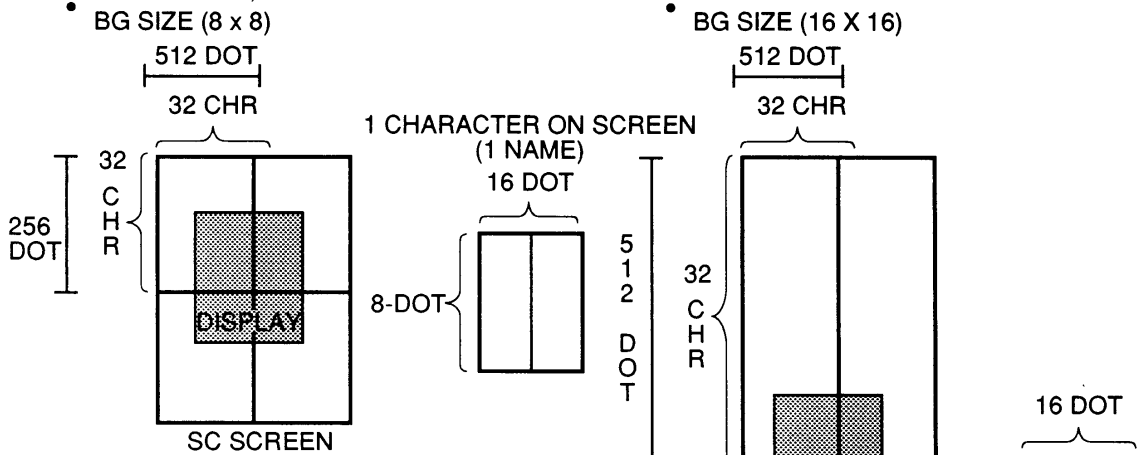
(NCL PG 85)

BG SCREEN

H/V SCROLL ② (Scroll range by the combination of modes and SC size against screen)

<Example: in case SC size is "3" - refer to register 2107H ~ 210AH>

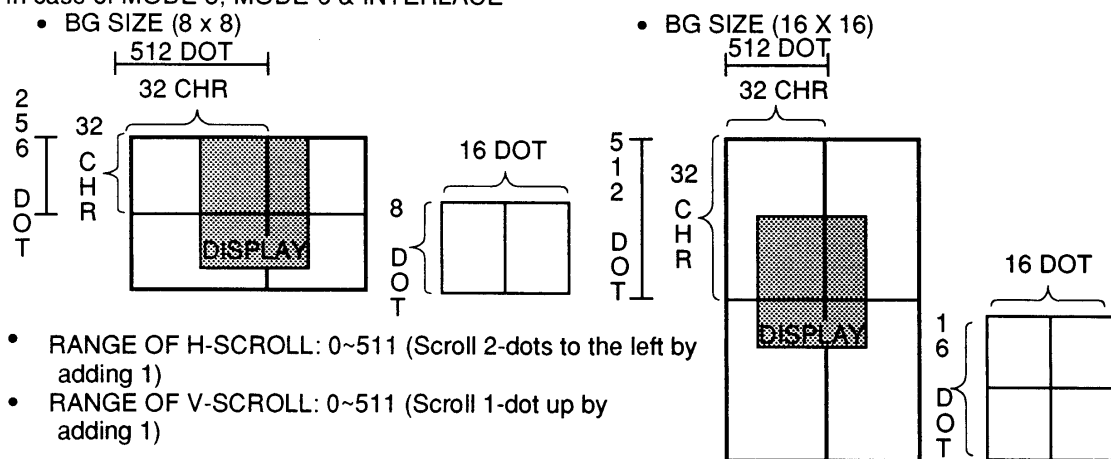
*In case of MODE-5, MODE-6 & NON-INTERLACE



- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~511 (Scroll 1-dot up by adding 1)

- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~1023 (Scroll 1-dot up by adding 1)

*In case of MODE-5, MODE-6 & INTERLACE

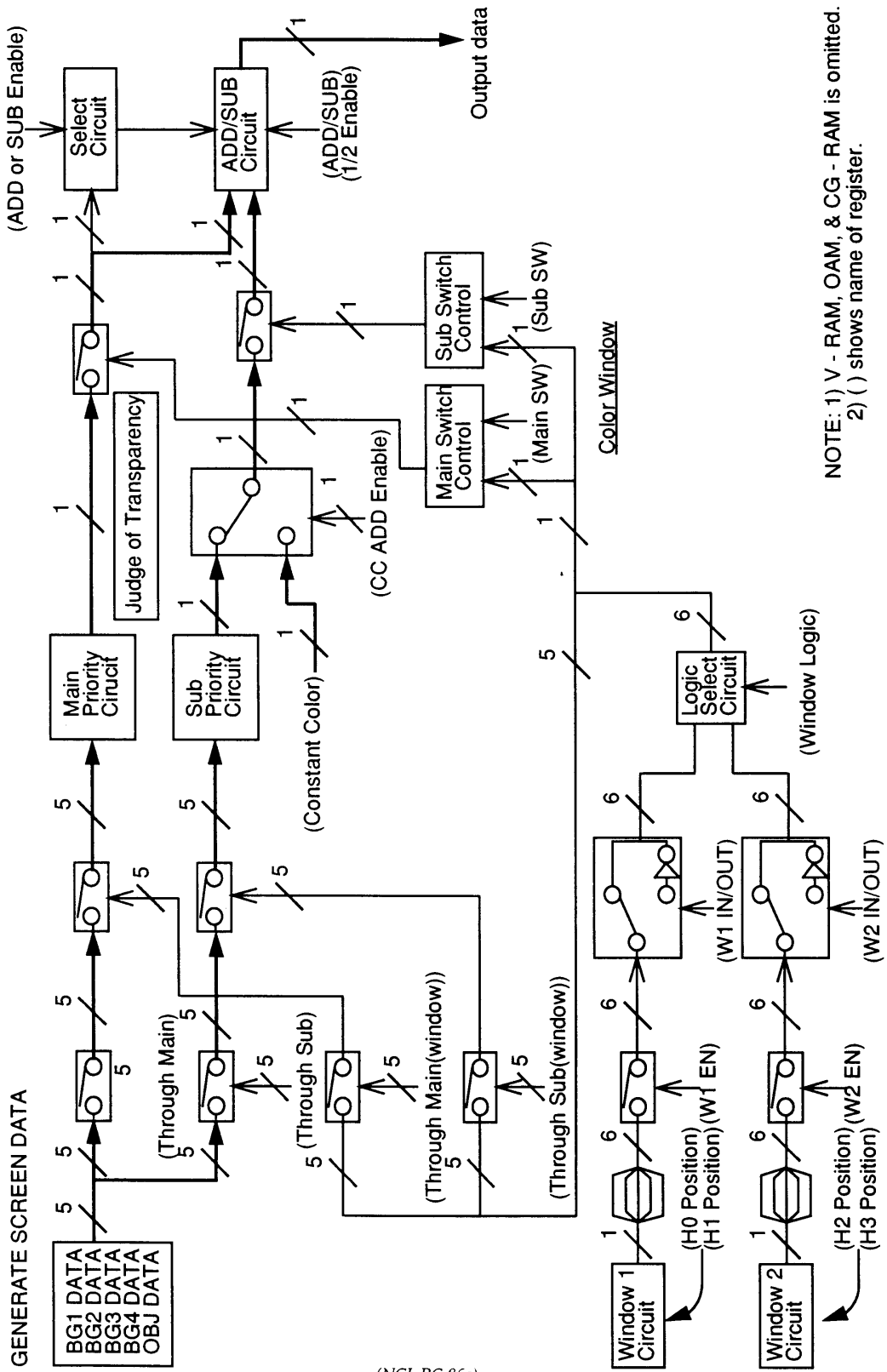


- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~511 (Scroll 1-dot up by adding 1)

- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~1023 (Scroll 1 dot up by adding 1)

(NCL PG 86)

Figure A-1 SNES PPU Main/Sub Screen Window



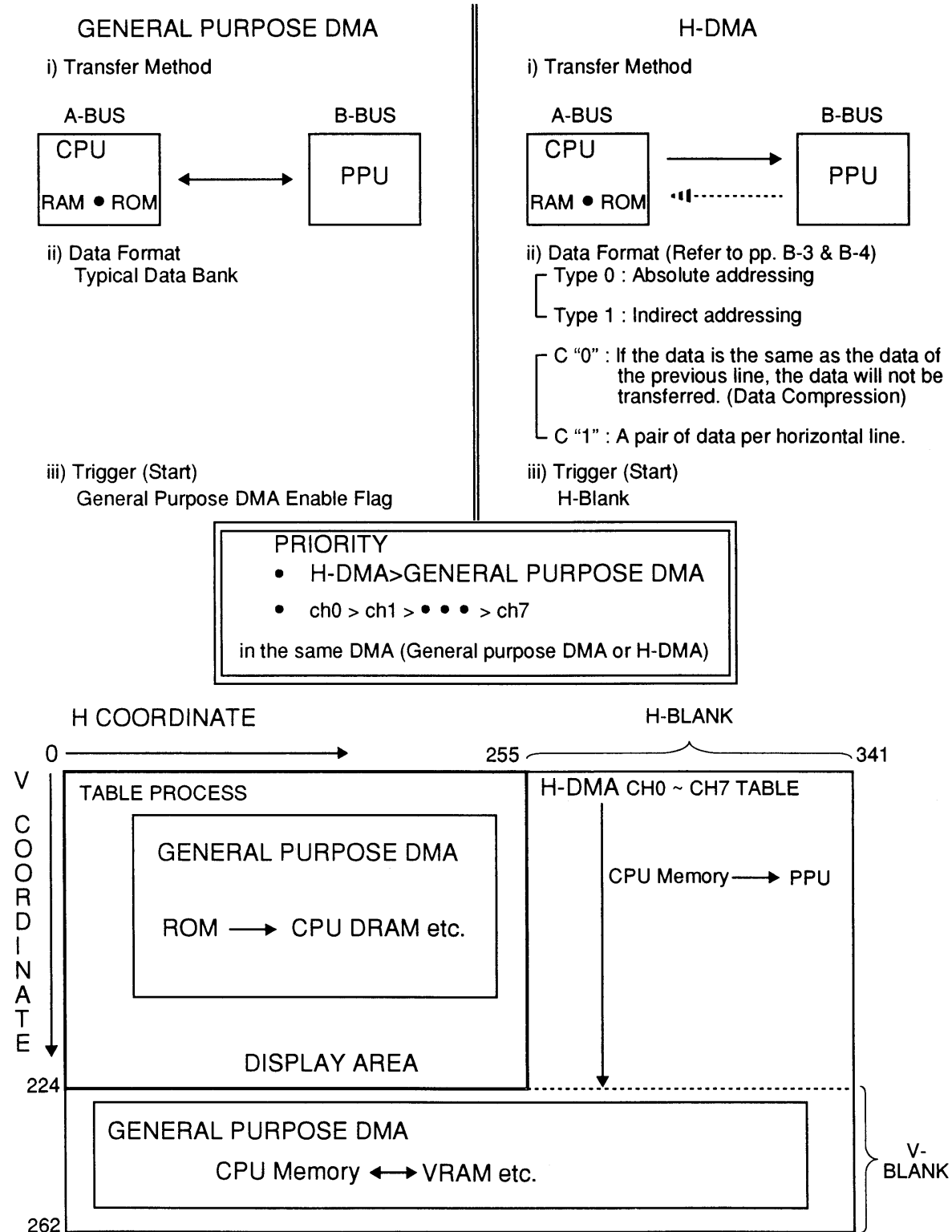
NOTE: 1) V - RAM, OAM, & CG - RAM is omitted.
 2) () shows name of register.

(NCL PG 86a)

Chapitre 21

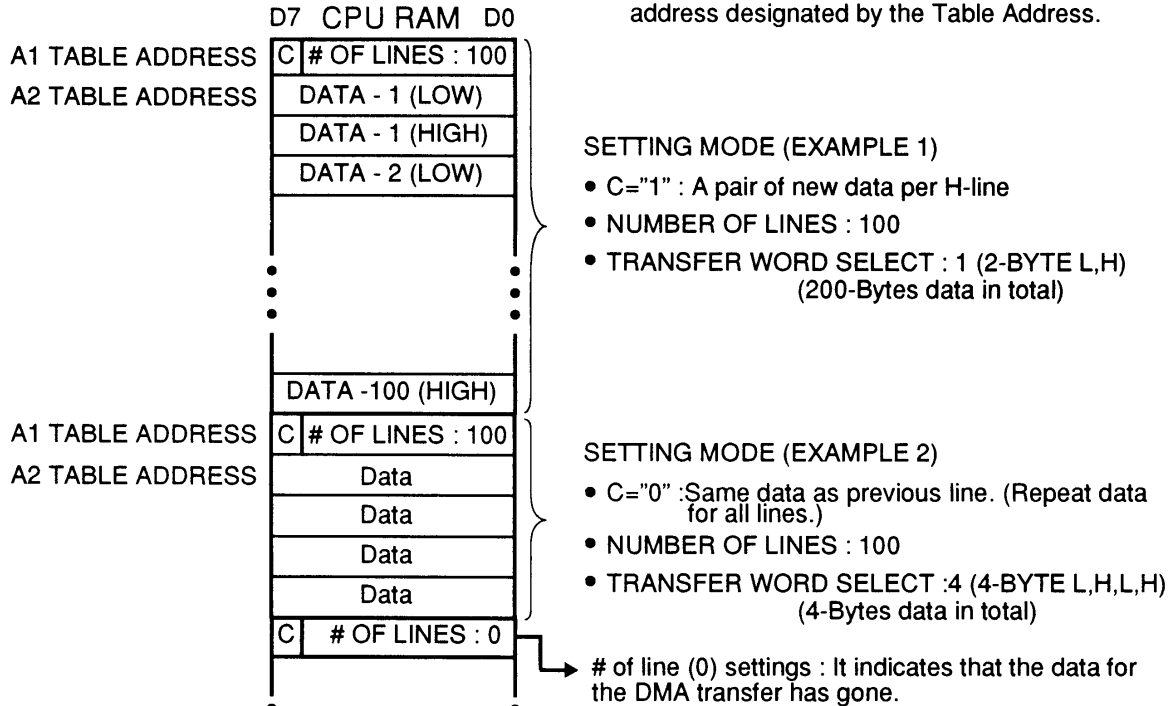
Registre CPU

Appendix B. CPU Registers

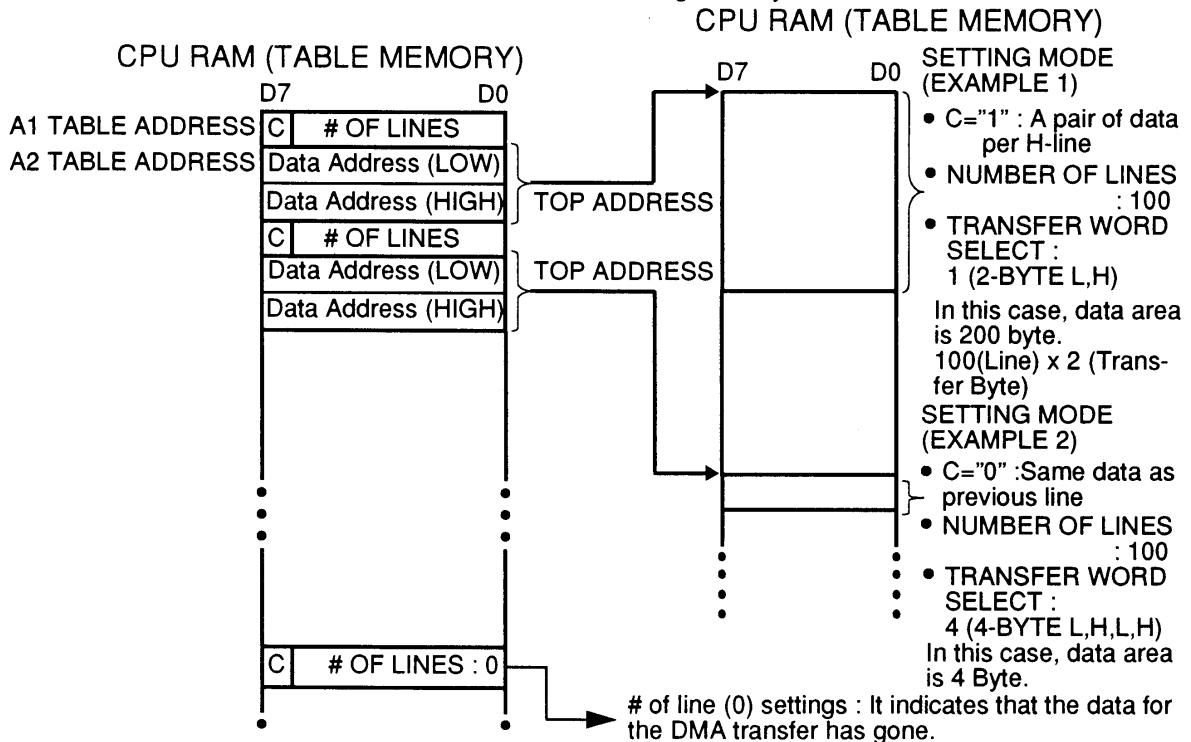


(NCL PG 101)

H-DMA ABSOLUTE ADDRESSING (TYPE-0) : This is a mode used to transfer the data of the address designated by the Table Address.

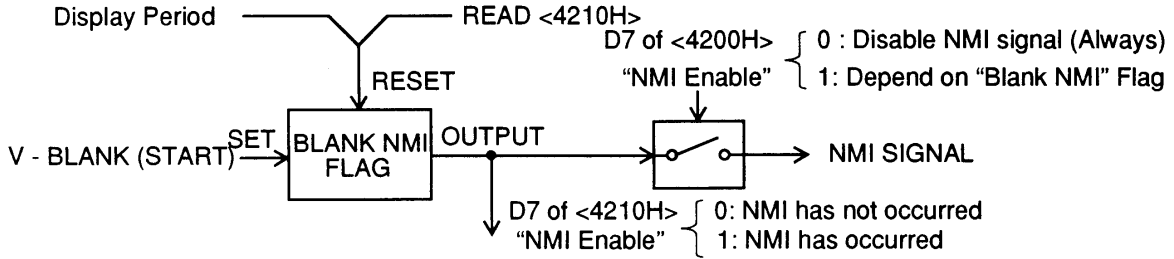


INDIRECT ADDRESSING (TYPE-1) : This is a mode used to transfer the data of the address designated by the Data Address, which is stored to the address designated by the Table Address.



(NCL PG 102)

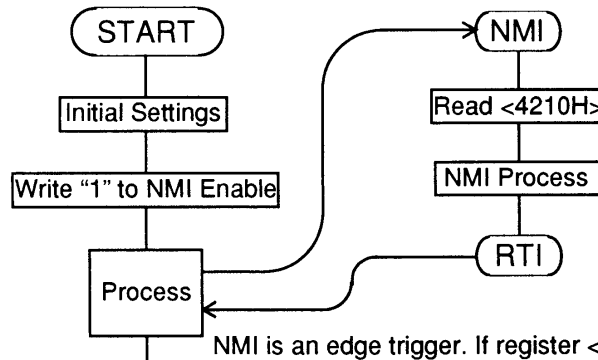
DETECT BEGINNING OF V - BLANK



The "Blank NMI" flag of register <4210H> will be set at the beginning of V - Blank and will reset at the end of V - Blank. It may also be reset by reading register <4210H>.

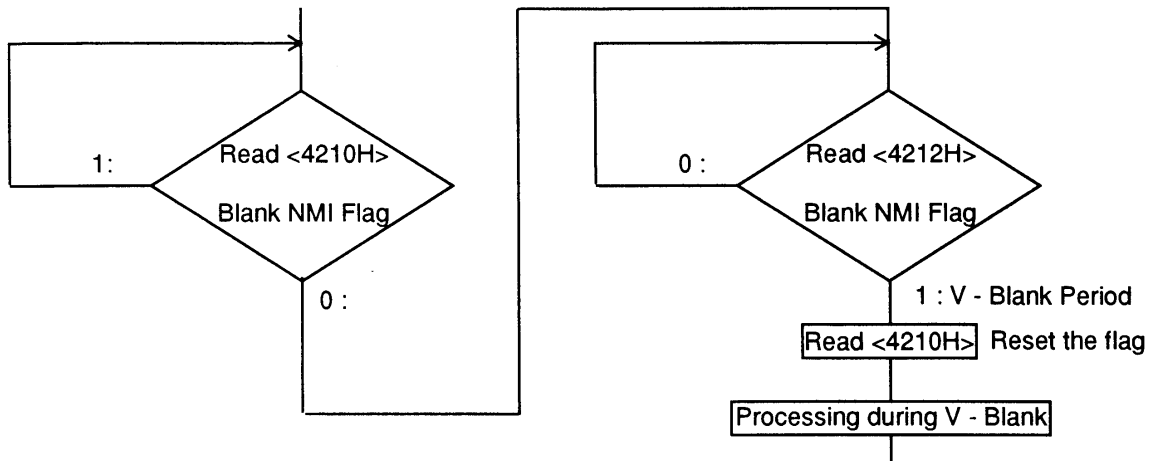
<EXAMPLE>

1. In case of detecting the beginning of V - Blank by NMI :



NMI is an edge trigger. If register <4210H> is not read during V - Blank and "NMI Enable" is set to "1", NMI will be duplicated.

2. In case of detecting the beginning of V - Blank by the flag :



(NCL PG 103)

SUMMARY OF REGISTERS

REGISTERS (WRITE) S - PPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2100H	Blanking				Fade IN/OUT (0 ~ 15)			
2101H	OBJ Size Select			OBJ Name Select	OBJ Name Base Address			
2102H	OAM Address							
2103H	OAM Priority Rotation						OAM Address MSB	
2104H	OAM Data (Low, High)							
2105H	BG Size BG4 BG3 BG2 BG1				BG 3 Priority	BG Mode (0 ~ 7)		
2106H	Mosaic Size				BG4	BG3	Mosaic Enable BG2 BG1	
2107H	BG1 SC Base Address						BG1 SC Size	
2108H	BG2 SC Base Address						BG2 SC Size	
2109H	BG3 SC Base Address						BG3 SC Size	
210AH	BG4 SC Base Address						BG4 SC Size	
210BH	BG2 Name Base Address				BG1 Name Base Address			
210CH	BG4 Name Base Address				BG3 Name Base Address			
210DH	BG1 H - Offset (Low, High)							
210EH	BG1V - Offset (Low, High)							
210FH	BG2 H - Offset (Low, High)							
2110H	BG2 V - Offset (Low, High)							
2111H	BG3 H - Offset (Low, High)							
2112H	BG3 V - Offset (Low, High)							
2113H	BG4 H - Offset (Low, High)							
2114H	BG4 V - Offset (Low, High)							
2115H	H/L Inc				V - RAM Address Sequence Mode Full Graphic SC Increment			
2116H	V - RAM Address (Low)							
2117H	V - RAM Address (High)							
2118H	V - RAM Data (Low)							
2119H	V - RAM Data (High)							
211AH	Screen Over						Screen Flip V H	

(NCL PG 104)

REGISTERS (WRITE) S - PPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0		
211BH	Matrix Parameter A (Low, High)									
211CH	Matrix Parameter B (Low, High)									
211DH	Matrix Parameter C (Low, High)									
211EH	Matrix Parameter D (Low, High)									
211FH	Matrix Parameter X (Low, High)									
2120H	Matrix Parameter Y (Low, High)									
2121H	CG - RAM Address									
2122H	CG - RAM Data (Low, High)									
2123H	BG2 Window W2 EN IN/OUT W1 EN IN/OUT				BG1 Window W2 EN IN/OUT W1 EN IN/OUT					
2124H	BG4 Window W2 EN IN/OUT W1 EN IN/OUT				BG3 Window W2 EN IN/OUT W1 EN IN/OUT					
2125H	Color Window W2 EN IN/OUT W1 EN IN/OUT				OBJ Window W2 EN IN/OUT W1 EN IN/OUT					
2126H	Window H0 Position (0 ~ 255)									
2127H	Window H1 Position (0 ~ 255)									
2128H	Window H2 Position (0 ~ 255)									
2129H	Window H3 Position (0 ~ 255)									
212AH	Window Logic									
212BH	BG4	BG3	BG2	BG1	Window Logic					
212CH					Color	OBJ				
212DH					Through Main	OBJ	BG4	BG3	BG2	BG1
212EH					Through Sub	OBJ	BG4	BG3	BG2	BG1
212FH					Through Main (Window)	OBJ	BG4	BG3	BG2	BG1
2130H					Through Sub (Window)	OBJ	BG4	BG3	BG2	BG1
2131H	Window ON/OFF				CG ADD				Direct	
	Main SW (A)		Sub SW (B)		Enable				Select	
2132H	ADD	1/2	ADD or SUB Enable							
	SUB	Enable	BACK	OBJ	BG4	BG3	BG2	BG1		
2133H	Color Constant Data									
	Blue	Green	Red	Color Brilliance Data						
	EXT. Sync.	EXT. Input		Pseudo 512	224/239	OBJ - V Select	Inter-lace			

S - PPU READ REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2134H	M P Y (Low)							
2135H	M P Y (Mid)							
2136H	M P Y (High)							
2137H	Soft Latch for H/V Counter							
2138H	OAM Data (Low, High)							
2139H	V - RAM Data (Low)							
213AH	V - RAM Data (High)							
213BH	CG Data (Low, High)							
213CH	Output Data of H - Counter (Low, High)							
213DH	Output Data of V - Counter (Low, High)							
213EH	Time Over	Range Over	Master /Slave		S - PPU1 Version Number			
213FH	Field	EXT. Latch		NTSC /PAL	S - PPU2 Version Number			

APU READ/WRITE REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2140H	APU I/O Port							
2141H	APU I/O Port							
2142H	APU I/O Port							
2143H	APU I/O Port							

WORK RAM READ/WRITE REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2180H	WORK RAM Data							

WORK RAM WRITE REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2181H	WORK RAM Address (Low)							
2182H	WORK RAM Address (Mid)							
2183H	WORK RAM Address (High)							

REGISTERS (WRITE) S - CPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
4200H	NMI Enable		Timer Enable V - EN H - EN					Joy - C Enable
4201H	I/O Port							
4202H	Multiplicand - A							
4203H	Multiplier - B							
4204H	Dividend - C (Low)							
4205H	Dividend - C (High)							
4206H	Divisor - B							
4207H	H - Counter Timer							
4208H								H - MSB
4209H	V - Counter Timer							
420AH								V - MSB
420BH	General Purpose DMA (Enable Flag)							
	CH7 EN	CH6 EN	CH5 EN	CH4 EN	CH3 EN	CH2 EN	CH1 EN	CH0 EN
420CH	H-DMA (Enable Flag)							
	CH7 EN	CH6 EN	CH5 EN	CH4 EN	CH3 EN	CH2 EN	CH1 EN	CH0 EN
420DH								2.68 /3.58

REGISTERS (READ) S - CPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
4210H	Blank NMI				SNES - CPU Version Number			
4211H	Timer IRQ							
4212H	V -Blank	H -Blank						Joy - C Enable
4213H	I/O Port							
4214H	Quotient - A (Low)							
4215H	Quotient - A (High)							
4216H	Product - C / Remainder (Low)							
4217H	Product - C / Remainder (High)							
4218H	Joy Controller I (Low)							
4219H	Joy Controller I (High)							
421AH	Joy Controller II (Low)							
421BH	Joy Controller II (High)							
421CH	Joy Controller III (Low)							
421DH	Joy Controller III (High)							
421EH	Joy Controller IV (Low)							
421FH	Joy Controller IV (High)							

REGISTERS (WRITE) S - CPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
43X0H	CHX *T-Org	CHX Type		A - Bus Address INC/DEC Fixed	CHX Transfer Word Select			
43X1H	CHX B - Address							
43X2H	CHX A1 Table Address (Low)							
43X3H	CHX A1 Table Address (High)							
43X4H	CHX A Table Bank							
43X5H	CHX Data Address (H-DMA) / Number of Bytes to be Transferred (General Purpose DMA) (Low)							
43X6H	CHX Data Address (H-DMA) / Number of Bytes to be Transferred (General Purpose DMA) (High)							
43X7H	CHX Data Bank (H - DMA)							
43X8H	CHX A2 Table Address (Low)							
43X9H	CHX A2 Table Address (High)							
43XAH	Continue	Number of Lines						

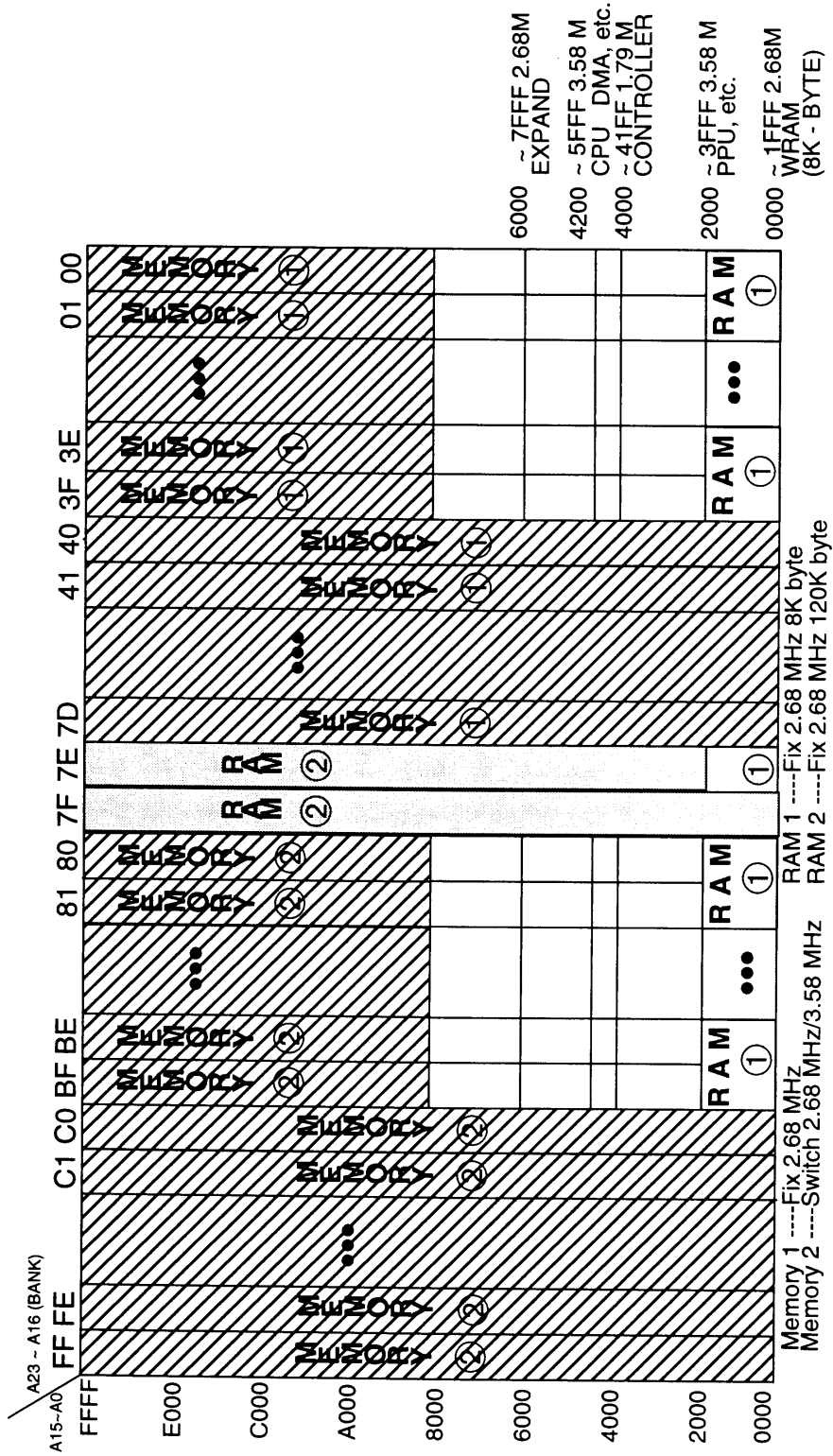
* T - Org means the "Transfer Orientation".

(NCL PG 106)

Chapitre 22

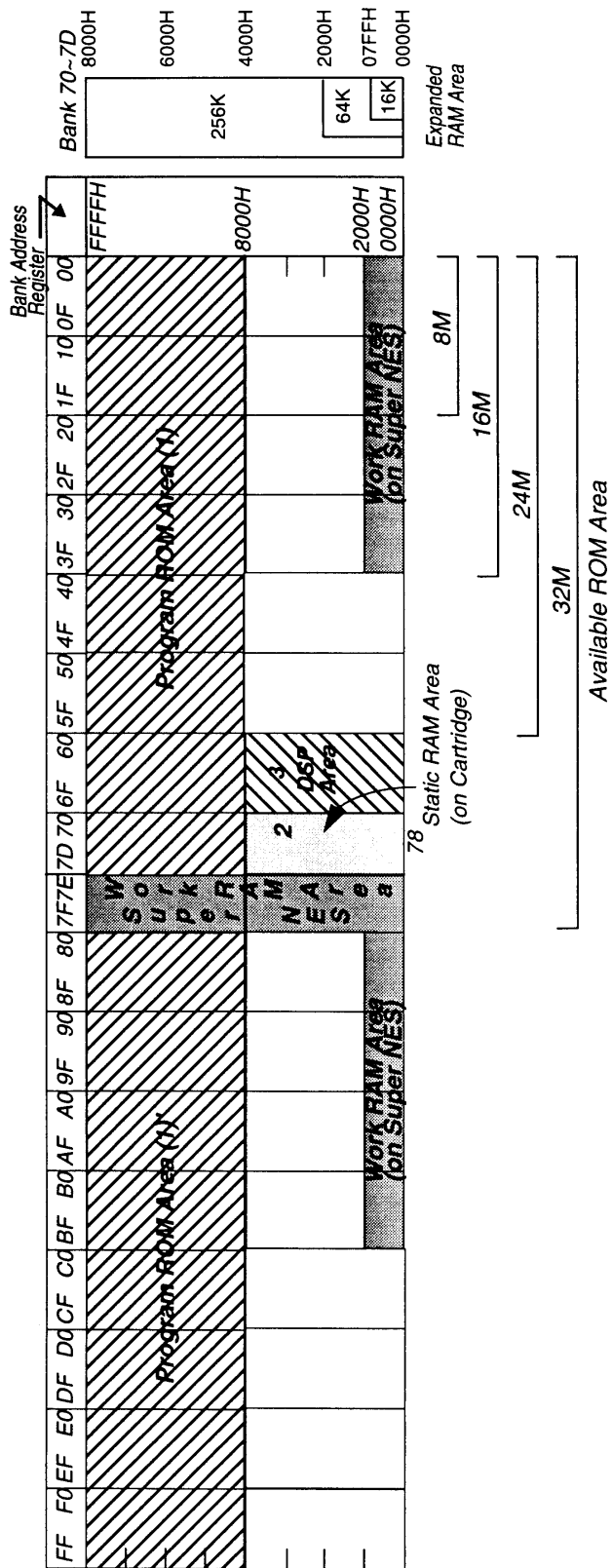
Mapping mémoire (MM)

Figure 2-21-1 Super NES CPU Memory Map



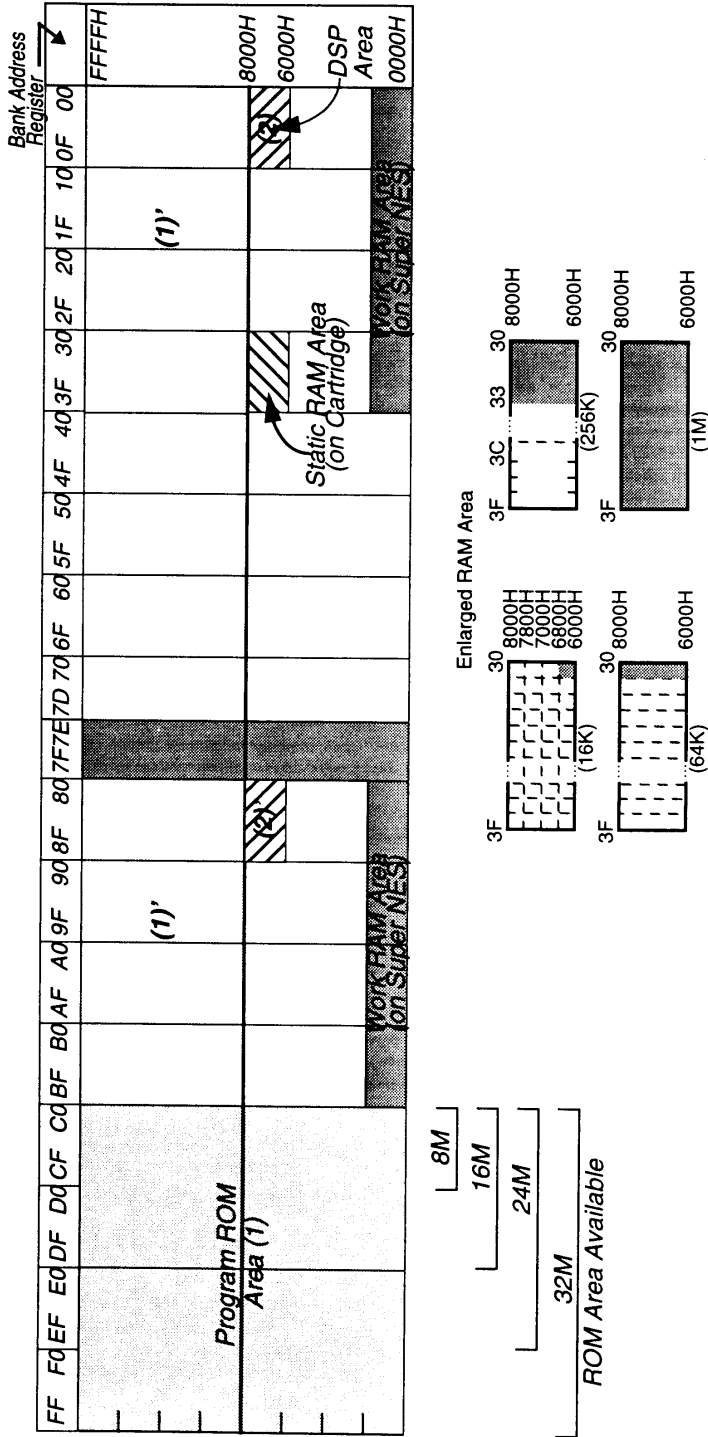
(NCL PG 37)

Figure 2-21-2 Super NES Memory Map (Mode 20)



- Note 1: The ROM image for bank 00H ~ 7DH is generated in bank 80H ~ FDH.
- Note 2: Set start vector and general registration area at address FFC0H ' (ROM address 00:7FC0H) in bank 00H.
- Note 3: Programs located in the ROM area of bank 80H ~ FFH can be executed in the high speed mode.
Specify the need for the high speed mode in the submission form.
- Note 4: When the program ROM is less than 8M, the DSP area is from 8000H ~ FFFFH in bank 30H ~ 3FH.

Figure 2-21-3 Super NES Memory Map (Mode 21)

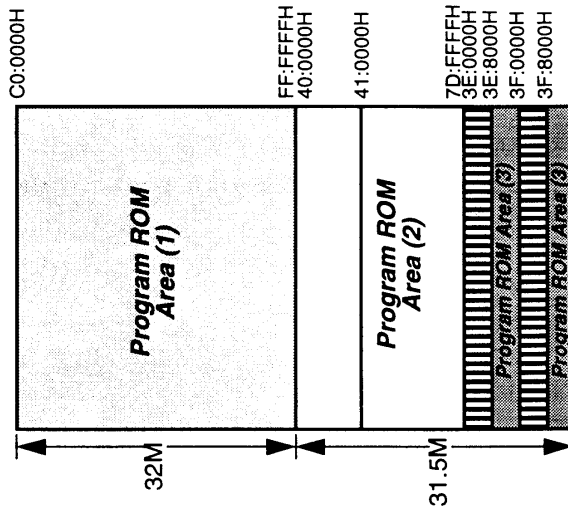
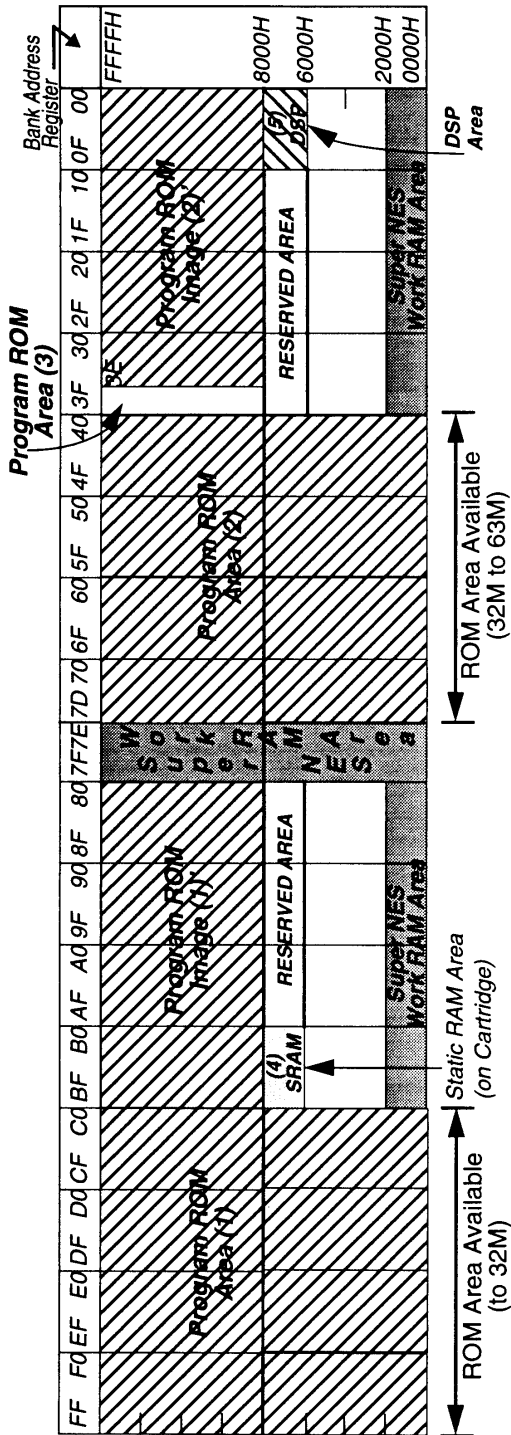


Note 1: In memory Mode 21, memory can be added from bank C0H to bank FFH (maximum is 32M bit). The ROM image of "8000H - FFFFH of bank C0H - FFH" will appear on "bank 00H - 3FH". Set vectors (i.e., Reset Vector) in the vector area of bank C0H".

Note 2: The ROM image from address 8000H ~ FFFFH of bank C0H ~ FFH is generated in bank 00H ~ 3FH and bank 80H ~ BFH.

Note 3: Programs located in the area of bank 80H ~ FFH can be executed in the high speed mode. Specify the need for the high speed mode in the submission form.

Figure 2-21-4 Super NES Memory Map (Mode 25, ROM Size Greater than 32 Mbits only)



- Note 1: The ROM image of 8000H~FFFFH of bank 40H~7DH will appear at bank 00H~3DH. Set vectors and registration data in FFB0H of bank 40H (ROM address 40:FFB0H).
- Note 2: Programs located in the area of bank 80H~FFH can be executed in the high speed mode (3.58 MHz).
- Note 3: Don't access null area.
- Note 4: Use the area of bank 3E and 3F as the program ROM area of bank 7E and 7F.

Chapitre 23

Le Programme

23.1 WLA.bat

```
@echo off
c:
cd \Projets_SNES\Projet01
echo [objects] > temp.prj
echo snos.obj >> temp.prj

echo on
wla-65816 -vo snos.asm snos.obj
wlalink -vr temp.prj snes.smc
@echo off

del snos.obj
del temp.prj

PAUSE
```

23.2 MAIN.inc

```
;  
; Super Nintendo ROM bank map  
;  
  
; == Mode 20 LoRom ==  
;  
MEMORYMAP  
SLOTSIZE $8000 ; Taille de la zone mémoire de la Super Nintendo  
DEFAULTSLOT 0 ; Slot par défaut, sur la Super Nintendo il n'y en a qu  
 ; 'un.  
SLOT 0 $8000 ; adresse de Départ de la zone mémoire de la Snes  
.ENDME  
  
.ROMBANKSIZE $8000 ; Taille de chaque Banque dans laquelle on peut écrire.  
.ROMBANKS 8 ; Nombre de banques que vous voulez utiliser pour votre  
 ; programme. Mode 20, 125 banques max : 31.2 Mbits  
;  
  
; === Cartridge Header – part 1 –  
SNESHEADER  
ID "SNES" ; Identification en 4 lettres  
  
NAME "SNOS byOlorin113V0r00" ; Titre du programme en 21 bits/lettres (utiliser  
 ; des espaces pour les bits non utilisés)  
 ; "123456789012345678901"  
  
SLOWROM  
LOROM ; Ici on c'est mis en lowrom et slowrom  
  
CARTRIDGETYPE $00 ; Type de la cartouche / $00 = ROM seulement  
ROMSIZE $08 ; Taille de la rom / $08 = 2 Mbits  
SRAMSIZE $00 ; Pas de SRAM  
COUNTRY $01 ; Localisation, U.S, Japon, europe etc...  
LICENSEECODE $00 ; Utiliser $00 tout le temps ;)  
VERSION $00 ; La version ($00=1.0, $01=1.1 etc...)  
.ENDSNES  
  
; === Interrupt Vector Table  
SNESEMUVECTOR  
COP EmptyHandler  
UNUSED EmptyHandler  
ABORT EmptyHandler  
NMI VBlank  
RESET Main  
IRQBRK EmptyHandler  
ENDEMUVECTOR  
  
.SNESNATIVEVECTOR  
COP EmptyHandler  
BRK EmptyHandler  
ABORT EmptyHandler  
NMI VBlank  
UNUSED EmptyHandler  
IRQ EmptyHandler  
ENDNATIVEVECTOR  
  
.EMPTYFILL $00 ; fill unused areas with $00, opcode for BRK.  
 ; BRK will crash the Super Nintendo if executed.
```

23.3 INTERRUPTIONS.asm

```
.BANK 0 SLOT 0
.ORG 0
.SECTION "INTERRUPTIONS" SEMIFREE
COPHandler :
    RTI                ;Retour d'interruption

BRKHandler :
    RTI                ;Retour d'interruption

VBlank :
    RTI                ;Retour d'interruption

IRQBRKHandler :
    RTI                ;Retour d'interruption

ABORThandler :
    RTI                ;Retour d'interruption

EmptyHandler :
    RTI                ;Retour d'interruption

.ENDS
```

23.4 InitSNES.asm

```

.BANK 0 SLOT 0
.ORG 0
.SECTION "Snes_Init" SEMIFREE
InitSnes:
    SEI          ; Disabled interrupts
    CLC          ; clear carry to switch to native mode
    XCE          ; Xchange carry & emulation bit. native mode
    REP         #$18 ; Binary mode (decimal mode off), X/Y 16 bit
    LDX         #$1FFF ; set stack to $1FFF
    SEP         #$30 ; X,Y,A are 8 bit numbers
    LDA         #$8F ; screen off, full brightness
    STA         $2100 ; brightness + screen enable register
    STZ         $2101 ; Sprite register (size + address in VRAM)
    STZ         $2102 ; Sprite registers (address of sprite memory [OAM])
    STZ         $2103 ; "" ""
    STZ         $2105 ; Mode 0, = Graphic mode register
    STZ         $2106 ; noplanes, no mosaic, = Mosaic register
    STZ         $2107 ; Plane 0 map VRAM location
    STZ         $2108 ; Plane 1 map VRAM location
    STZ         $2109 ; Plane 2 map VRAM location
    STZ         $210A ; Plane 3 map VRAM location
    STZ         $210B ; Plane 0+1 Tile data location
    STZ         $210C ; Plane 2+3 Tile data location
    STZ         $210D ; Plane 0 scroll x (first 8 bits)
    STZ         $210D ; Plane 0 scroll x (last 3 bits) #$0 - #$07ff
    STZ         $210E ; Plane 0 scroll y (first 8 bits)
    STZ         $210E ; Plane 0 scroll y (last 3 bits) #$0 - #$07ff
    STZ         $210F ; Plane 1 scroll x (first 8 bits)
    STZ         $210F ; Plane 1 scroll x (last 3 bits) #$0 - #$07ff
    STZ         $2110 ; Plane 1 scroll y (first 8 bits)
    STZ         $2110 ; Plane 1 scroll y (last 3 bits) #$0 - #$07ff
    STZ         $2111 ; Plane 2 scroll x (first 8 bits)
    STZ         $2111 ; Plane 2 scroll x (last 3 bits) #$0 - #$07ff
    STZ         $2112 ; Plane 2 scroll y (first 8 bits)
    STZ         $2112 ; Plane 2 scroll y (last 3 bits) #$0 - #$07ff
    STZ         $2113 ; Plane 3 scroll x (first 8 bits)
    STZ         $2113 ; Plane 3 scroll x (last 3 bits) #$0 - #$07ff
    STZ         $2114 ; Plane 3 scroll y (first 8 bits)
    STZ         $2114 ; Plane 3 scroll y (last 3 bits) #$0 - #$07ff
    LDA         #$80 ; increase VRAM address after writing to $2119
    STA         $2115 ; VRAM address increment register
    STZ         $2116 ; VRAM address low
    STZ         $2117 ; VRAM address high
    STZ         $211A ; Initial Mode 7 setting register
    STZ         $211B ; Mode 7 matrix parameter A register (low)
    LDA         #$01
    STA         $211B ; Mode 7 matrix parameter A register (high)
    STZ         $211C ; Mode 7 matrix parameter B register (low)
    STZ         $211C ; Mode 7 matrix parameter B register (high)
    STZ         $211D ; Mode 7 matrix parameter C register (low)
    STZ         $211D ; Mode 7 matrix parameter C register (high)
    STZ         $211E ; Mode 7 matrix parameter D register (low)
    STA         $211E ; Mode 7 matrix parameter D register (high)
    STZ         $211F ; Mode 7 center position X register (low)
    STZ         $211F ; Mode 7 center position X register (high)

```

```

STZ     $2120 ; Mode 7 center position Y register (low)
STZ     $2120 ; Mode 7 center position Y register (high)

```

```

STZ    $2121    ; Color number register ($0–ff)
STZ    $2123    ; BG1 & BG2 Window mask setting register
STZ    $2124    ; BG3 & BG4 Window mask setting register
STZ    $2125    ; OBJ & Color Window mask setting register
STZ    $2126    ; Window 1 left position register
STZ    $2127    ; Window 2 left position register
STZ    $2128    ; Window 3 left position register
STZ    $2129    ; Window 4 left position register
STZ    $212A    ; BG1, BG2, BG3, BG4 Window Logic register
STZ    $212B    ; OBJ, Color Window Logic Register (or,and,xor,xnor)
STA    $212C    ; Main Screen designation (planes, sprites enable)
STZ    $212D    ; Sub Screen designation
STZ    $212E    ; Window mask for Main Screen
STZ    $212F    ; Window mask for Sub Screen
LDA    #$30
STA    $2130    ; Color addition & screen addition init setting
STZ    $2131    ; Add/Sub sub designation for screen, sprite, color
LDA    #$E0
STA    $2132    ; color data for addition/subtraction
STZ    $2133    ; Screen setting (interlace x,y/enable SFX data)
STZ    $4200    ; Enable V-blank, interrupt, Joypad register
LDA    #$FF
STA    $4201    ; Programmable I/O port
STZ    $4202    ; Multiplicand A
STZ    $4203    ; Multiplier B
STZ    $4204    ; Multiplier C
STZ    $4205    ; Multiplicand C
STZ    $4206    ; Divisor B
STZ    $4207    ; Horizontal Count Timer
STZ    $4208    ; Horizontal Count Timer MSB (most significant bit)
STZ    $4209    ; Vertical Count Timer
STZ    $420A    ; Vertical Count Timer MSB
STZ    $420B    ; General DMA enable (bits 0–7)
STZ    $420C    ; Horizontal DMA (HDMA) enable (bits 0–7)
STZ    $420D    ; Access cycle designation (slow/fast rom)
CLI    ; Enable interrupts
RTS

```

.ENDS

23.5 MAIN.asm

```
; Super Nintendo Initialization Tutorial code
; This code is in the public domain.

;=== Init Mapping Mémoire ===
INCLUDE "main.inc"

;=== Init routines d'interruptions ===
INCLUDE "interruptions.inc"

;=== Init Registres CPU et PPU ===
INCLUDE "initSnes.asm"

;*****
;                               MAIN PROGRAM
;*****
.BANK 0 SLOT 0
.ORG 0
.SECTION "MainCode" FORCE

Main:
    ; Initialisation de la Super Nintendo
    JSR InitSnes

    ; Mettre le fond d'écran en blanc
    SEP    #$20          ; Mettre le registre Accumulateur à 8 bits.
    LDA    #%10000000   ; On force le VBlank en éteignant l'écran.
    STA    $2100
    LDA    #%00000000   ; Charger l'octet bas de la couleur blanche.
    STA    $2122
    LDA    #%00000000   ; Charger l'octet haut de la couleur blanche
    STA    $2122
    LDA    #%00001111   ; find du VBlank, mettre la luminance à 15 (100%).
    STA    $2100

    ; boucler à l'infinie.
Forever:
    JMP Forever

.ENDS
```

23.6 Tbl.asm

```
;*****  
;                                     TABULAR PROGRAMME  
;*****  
  
.BANK 1 SLOT 0  
.ORG 0  
.SECTION "tabularCode"  
    ;-----  
    ;TABULAR SEARCH  
  
SEARCH:  
    LDA #TABASE  
    STA POINTR  
    LDX #TABLEN  
    BEQ OUT  
ENTRY:  
    LDY #0  
    LDA OBJECT,Y  
    CMP (POINTR),Y  
    BNE NOGOOD  
    INY  
    LDA OBJECT,Y  
    CMP (POINTR),Y  
    BEQ FOUND  
NOGOOD:  
    DEX  
    BEQ OUT  
    LDA #ENTLEN  
    CLC  
    ADC POINTR  
    STA POINTR  
    BRA ENTRY  
FOUND:  
    LDA #$FFFF  
OUT:  
    RTS  
  
    ;-----  
    ;TABULAR INSERTION  
  
.ENDS
```


Chapitre 24

Les Modes d'adressages

24.1 Classements pas code opération (opcode)

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcod</i> <i>e</i> <i>(Hex)</i>	<i>Disponible sur</i>		<i>nb d'</i> <i>Octets</i>	<i>nb de</i> <i>Cycles</i>
			<i>6502</i>	<i>65816</i>		
Stack/Interrupt	BRK	00	x	x	2 ⁺	7 ⁷
DP indirectement indexé par X	ORA (<i>dp,X</i>)	01	x	x	2	6 ^{1,3}
Stack/Interrupt	COP	02		x	2 ⁺⁺	7 ⁷
Stack relative (SR)	ORA <i>sr,S</i>	03		x	2	4 ¹
Direct Page(DP)	TSB <i>dp</i>	04		x	2	5 ^{2,3}
Direct Page(DP)	ORA <i>dp</i>	05	x	x	2	3 ^{1,3}
Direct Page(DP)	ASL <i>dp</i>	06	x	x	2	5 ^{2,3}
DP indirect Long	ORA [<i>dp</i>]	07		x	2	6 ^{1,3}
Stack (push)	PHP	08	x	x	1	3
Immédiat	ORA <i>#const</i>	09	x	x	2	2 ¹
Accumulateur	ASL A	0A	x	x	1	2
Stack (push)	PHD	0B		x	1	4
Absolue	TSB <i>addr</i>	0C		x	3	6 ²
Absolue	ORA <i>addr</i>	0D	x	x	3	4 ¹
Absolue	ASL <i>addr</i>	0E	x	x	3	6 ²
Absolue Long	ORA <i>long</i>	0F		x	4	5 ¹
Compteur de Programme relatif	BPL <i>nearlabel</i>	10	x	x	2	2 ^{5,6}
DP indirectement indexé par Y	ORA (<i>dp</i>), <i>Y</i>	11	x	x	2	5 ^{1,3,4}
DP indirect	ORA (<i>dp</i>)	12		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	ORA (<i>sr,S</i>), <i>Y</i>	13		x	2	7 ¹
Direct Page(DP)	TRB <i>dp</i>	14		x	2	5 ^{2,3}
DP indexé par X	ORA <i>dp,X</i>	15	x	x	2	4 ^{1,3}
DP indexé par X	ASL <i>dp,X</i>	16	x	x	2	6 ^{2,3}
DP long indirectement indexé par Y	ORA [<i>dp</i>], <i>Y</i>	17		x	2	6 ^{1,3}
Implicite	CLC	18	x	x	1	2
Absolue indexé par Y	ORA <i>addr,Y</i>	19	x	x	3	4 ^{1,4}
Accumulateur	INC A	1A		x	1	2
Implicite	TCS	1B		x	1	2
Absolue	TRB <i>addr</i>	1C		x	3	6 ²
Absolue indexé par X	ORA <i>addr,X</i>	1D	x	x	3	4 ^{1,4}
Absolue indexé par X	ASL <i>addr,X</i>	1E	x	x	3	7 ^{2,4}
Absolue Long indexé par X	ORA <i>long,X</i>	1F		x	4	5 ¹
Absolue	JSR <i>addr</i>	20		x	3	6
DP indirectement indexé par X	AND (<i>dp,X</i>)	21	x	x	2	6 ^{1,3}
Absolue Long	JSR <i>long</i>	22		x	4	8
Absolue Long	JSL <i>long</i>	22		x	4	8
Stack relative (SR)	AND <i>sr,S</i>	23		x	2	4 ¹
Direct Page(DP)	BIT <i>dp</i>	24	x	x	2	3 ^{1,3}
Direct Page(DP)	AND <i>dp</i>	25	x	x	2	3 ^{1,3}
Direct Page(DP)	ROL <i>dp</i>	26	x	x	2	5 ^{2,3}
DP indirect Long	AND [<i>dp</i>]	27		x	2	6 ^{1,3}
Stack (pull)	PLP	28	x	x	1	4
Immédiat	AND <i>#const</i>	29	x	x	2	2 ¹
Accumulateur	ROL A	2A		x	1	2

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Stack (pull)	PLD	2B		x	1	5
Absolute	BIT <i>addr</i>	2C	x	x	3	4 ¹
Absolute	AND <i>addr</i>	2D	x	x	3	4 ¹
Absolute	ROL <i>addr</i>	2E	x	x	3	6 ²
Absolute Long	AND <i>long</i>	2F		x	4	5 ¹
Compteur de Programme relatif	BMI <i>nearlabel</i>	30	x	x	2	2 ^{5,6}
DP indirectement indexé par Y	AND (<i>dp</i>), <i>Y</i>	31	x	x	2	5 ^{1,3,4}
DP indirect	AND (<i>dp</i>)	32		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	AND (<i>sr,S</i>), <i>Y</i>	33		x	2	7 ¹
DP indexé par X	BIT <i>dp,X</i>	34		x	2	6 ^{1,3}
DP indexé par X	AND <i>dp,X</i>	35	x	x	2	4 ^{1,3}
DP indexé par X	ROL <i>dp,X</i>	36	x	x	2	6 ^{2,3}
DP long indirectement indexé par Y	AND [<i>dp</i>], <i>Y</i>	37		x	2	6 ^{1,3}
Implicite	SEC	38	x	x	1	2
Absolute indexé par Y	AND <i>addr,Y</i>	39	x	x	3	4 ^{1,4}
Accumulateur	DEC A	3A		x	1	2
Implicite	TSC	3B		x	1	2
Implicite	TSA	3B		x	1	2
Absolute indexé par X	BIT <i>addr,X</i>	3C		x	3	4 ^{1,4}
Absolute indexé par X	AND <i>addr,X</i>	3D	x	x	3	4 ^{1,4}
Absolute indexé par X	ROL <i>addr,X</i>	3E	x	x	3	7 ^{2,4}
Absolute Long indexé par X	AND <i>long,X</i>	3F		x	4	5 ¹
Stack (RTI)	RTI	40	x	x	1	6 ⁷
DP indirectement indexé par X	EOR (<i>dp,X</i>)	41	x	x	2	6 ^{1,3}
	WDM	42		x	2 ^x	x
Stack relative (SR)	EOR <i>sr,S</i>	43		x	2	4 ¹
Block Move	MVP	44		x	3	9
	<i>srcbk,destbk</i>					
Direct Page(DP)	EOR <i>dp</i>	45	x	x	2	3 ^{1,3}
Direct Page(DP)	LSR <i>dp</i>	46	x	x	2	5 ^{2,3}
DP indirect Long	EOR [<i>dp</i>]	47		x	2	6 ^{1,3}
Stack (push)	PHA	48	x	x	1	3 ¹
Immédiat	EOR <i>#const</i>	49	x	x	2	2 ¹
Accumulateur	LSR A	4A		x	1	2
Stack (push)	PHK	4B		x	1	3
Absolute	JMP <i>addr</i>	4C	x	x	3	3
Absolute	EOR <i>addr</i>	4D	x	x	3	4 ¹
Absolute	LSR <i>addr</i>	4E	x	x	3	6 ²
Absolute Long	EOR <i>long</i>	4F		x	4	5 ¹
Compteur de Programme relatif	BVC <i>nearlabel</i>	50	x	x	2	2 ^{5,6}
DP indirectement indexé par Y	EOR (<i>dp</i>), <i>Y</i>	51	x	x	2	5 ^{1,3,4}
DP indirect	EOR (<i>dp</i>)	52		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	EOR (<i>sr,S</i>), <i>Y</i>	53		x	2	7 ¹

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Block Move	MVN <i>srcbk,destbk</i>	54	x	x	3	9
DP indexé par X	EOR <i>dp,X</i>	55	x	x	2	4 ^{1,3}
DP indexé par X	LSR <i>dp,X</i>	56	x	x	2	6 ^{2,3}
DP long indirectement indexé par Y	EOR <i>[dp],Y</i>	57		x	2	6 ^{1,3}
Implicite	CLI	58	x	x	1	2
Absolute indexé par Y	EOR <i>addr,Y</i>	59	x	x	3	4 ^{1,4}
Stack (push)	PHY	5A		x	1	3 ⁸
Implicite	TCD	5B		x	1	2
Absolute Long	JMP <i>long</i>	5C		x	4	4
Absolute indexé par X	EOR <i>addr,X</i>	5D	x	x	3	4 ^{1,4}
Absolute indexé par X	LSR <i>addr,X</i>	5E	x	x	3	7 ^{2,4}
Absolute Long indexé par X	EOR <i>long,X</i>	5F		x	4	5 ¹
Stack (RTS)	RTS	60	x	x	1	6
DP indirectement indexé par X	ADC <i>(dp,X)</i>	61	x	x	2	6 ^{1,3}
Stack (Compteur de programme relatif long)	PER <i>label</i>	62			3	6
Stack relative (SR)	ADC <i>sr,S</i>	63		x	2	4 ¹
Direct Page(DP)	STZ <i>dp</i>	64	x	x	2	3 ^{1,3}
Direct Page(DP)	ADC <i>dp</i>	65	x	x	2	3 ^{1,3}
Direct Page(DP)	ROR <i>dp</i>	66	x	x	2	5 ^{2,3}
DP indirect Long	ADC <i>[dp]</i>	67		x	2	6 ^{1,3}
Stack (pull)	PLA	68	x	x	1	4 ⁸
Immédiat	ADC <i>#const</i>	69	x	x	2	2 ¹
Accumulateur	ROR A	6A	x	x	1	2
Stack (RTL)	RTL	6B		x	1	6
Absolute indirect	JMP <i>(addr)</i>	6C	x	x	3	5
Absolute	ADC <i>addr</i>	6D	x	x	3	4 ¹
Absolute	ROR <i>addr</i>	6E	x	x	3	6 ²
Absolute Long	ADC <i>long</i>	6F		x	4	5 ¹
Compteur de Programme relatif	BVS <i>nearlabel</i>	70	x	x	2	2 ^{5,6}
DP indirectement indexé par Y	ADC <i>(dp),Y</i>	71	x	x	2	5 ^{1,3,4}
DP indirect	ADC <i>(dp)</i>	72		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	ADC <i>(sr,S),Y</i>	73		x	2	7 ¹
DP indexé par X	STZ <i>dp,X</i>	74	x	x	2	4 ^{2,3}
DP indexé par X	ADC <i>dp,X</i>	75	x	x	2	4 ^{1,3}
DP indexé par X	ROR <i>dp,X</i>	76	x	x	2	6 ^{2,3}
DP long indirectement indexé par Y	ADC <i>[dp],Y</i>	77		x	2	6 ^{1,3}
Implicite	SEI	78	x	x	1	2
Absolute indexé par Y	ADC <i>addr,Y</i>	79	x	x	3	4 ^{1,4}
Stack (pull)	PLY	7A		x	1	4 ⁸
Implicite	TDC	7B		x	1	2
Absolute indirect indexé par X	JMP <i>(addr,X)</i>	7C		x	3	6
Absolute indexé par X	ADC <i>addr,X</i>	7D	x	x	3	4 ^{1,4}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Absolute indexé par X	ROR <i>addr,X</i>	7E	x	x	3	7 ^{2,4}
Absolute Long indexé par X	ADC <i>long,X</i>	7F		x	4	5 ¹
Compteur de Programme relatif	BRA <i>nearlabel</i>	80	x	x	2	3 ⁶
DP indirectement indexé par X	STA (<i>dp,X</i>)	81	x	x	2	6 ^{1,3}
Compteur de Programme relatif Long	BRL <i>label</i>	82		x	3	4
Stack relative (SR)	STA <i>sr,S</i>	83		x	2	4 ¹
Direct Page(DP)	STY <i>dp</i>	84	x	x	2	3 ^{8,3}
Direct Page(DP)	STA <i>dp</i>	85	x	x	2	3 ^{1,3}
Direct Page(DP)	STX <i>dp</i>	86	x	x	2	3 ^{8,3}
DP indirect Long	STA [<i>dp</i>]	87		x	2	6 ^{1,3}
Implicite	DEY	88	x	x	1	2
Immédiat	BIT <i>#const</i>	89		x	2	2 ¹
Implicite	TXA	8A	x	x	1	2
Stack (push)	PHB	8B		x	1	3
Absolute	STY <i>addr</i>	8C	x	x	3	4 ⁸
Absolute	STA <i>addr</i>	8D	x	x	3	4 ¹
Absolute	STX <i>addr</i>	8E	x	x	3	4 ⁸
Absolute Long	STA <i>long</i>	8F		x	4	5 ¹
Compteur de Programme relatif	BTL <i>nearlabel</i>	90	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BCC <i>nearlabel</i>	90	x	x	2	2 ^{5,6}
DP indirectement indexé par Y	STA (<i>dp</i>), <i>Y</i>	91	x	x	2	6 ^{1,3,4}
DP indirect	STA (<i>dp</i>)	92		x	2	5 ^{1,3}
SR indirectement indexé indirecté par Y	STA (<i>sr,S</i>), <i>Y</i>	93		x	2	7 ¹
DP indexé par Y	STY <i>dp,Y</i>	94	x	x	2	4 ^{8,3}
DP indexé par X	STA <i>dp,X</i>	95	x	x	2	4 ^{1,3}
DP indexé par Y	STX <i>dp,Y</i>	96	x	x	2	4 ^{8,3}
DP long indirectement indexé par Y	STA [<i>dp</i>], <i>Y</i>	97		x	2	6 ^{1,3}
Implicite	TYA	98	x	x	1	2
Absolute indexé par Y	STA <i>addr,Y</i>	99	x	x	3	5 ¹
Implicite	TXS	9A	x	x	1	2
Implicite	TXY	9B		x	1	2
Absolute	STZ <i>addr</i>	9C	x	x	3	4 ¹
Absolute indexé par X	STA <i>addr,X</i>	9D	x	x	3	5 ¹
Absolute indexé par X	STZ <i>addr,X</i>	9E	x	x	3	5 ¹
Absolute Long indexé par X	STA <i>long,X</i>	9F		x	4	5 ¹
Immédiat	LDY <i>#const</i>	A0		x	2	2 ⁸
DP indirectement indexé par X	LDA (<i>dp,X</i>)	A1	x	x	2	6 ^{1,3}
Immédiat	LDX <i>#const</i>	A2		x	2	2 ⁸
Stack relative (SR)	LDA <i>sr,S</i>	A3		x	2	4 ¹
Direct Page(DP)	LDY <i>dp</i>	A4	x	x	2	3 ^{8,3}
Direct Page(DP)	LDA <i>dp</i>	A5	x	x	2	3 ^{1,3}
Direct Page(DP)	LDX <i>dp</i>	A6	x	x	2	3 ^{8,3}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
DP indirect Long	LDA <i>[dp]</i>	A7		x	2	6 ^{1,3}
Implicite	TAY	A8	x	x	1	2
Immédiat	LDA <i>#const</i>	A9	x	x	2	2 ¹
Implicite	TAX	AA	x	x	1	2
Stack (pull)	PLB	AB		x	1	4
Absolue	LDY <i>addr</i>	AC	x	x	3	4 ⁸
Absolue	LDA <i>addr</i>	AD	x	x	3	4 ¹
Absolue	LDX <i>addr</i>	AE	x	x	3	4 ⁸
Absolue Long	LDA <i>long</i>	AF		x	4	5 ¹
Compteur de Programme relatif	BCS <i>nearlabel</i>	B0	x	x	2	2 ^{5,6}
DP indirectement indexé par Y	LDA <i>(dp),Y</i>	B1	x	x	2	5 ^{1,3,4}
DP indirect	LDA <i>(dp)</i>	B2		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	LDA <i>(sr,S),Y</i>	B3		x	2	7 ¹
DP indirectement indexé par X	LDY <i>(dp,X)</i>	B4		x	2	4 ^{8,3}
DP indexé par X	LDA <i>dp,X</i>	B5	x	x	2	4 ^{1,3}
DP indirectement indexé par X	LDX <i>(dp,X)</i>	B6		x	2	4 ^{8,3}
DP long indirectement indexé par Y	LDA <i>[dp],Y</i>	B7		x	2	6 ^{1,3}
Implicite	CLV	B8	x	x	1	2
Absolue indexé par Y	LDA <i>addr,Y</i>	B9	x	x	3	4 ^{1,4}
Implicite	TSX	BA	x	x	1	2
Implicite	TYX	BB		x	1	2
Absolue indexé par X	LDY <i>addr,X</i>	BC		x	3	4 ^{8,4}
Absolue indexé par X	LDA <i>addr,X</i>	BD	x	x	3	4 ^{1,4}
Absolue indexé par X	LDX <i>addr,X</i>	BE		x	3	4 ^{8,4}
Absolue Long indexé par X	LDA <i>long,X</i>	BF		x	4	5 ¹
Immédiat	CPY <i>#const</i>	C0	x	x	2	2 ⁸
DP indirectement indexé par X	CMP <i>(dp,X)</i>	C1	x	x	2	6 ^{1,3}
Immédiat	REP <i>#const</i>	C2		x	2	3
Stack relative (SR)	CMP <i>sr,S</i>	C3		x	2	4 ¹
Direct Page(DP)	CPY <i>dp</i>	C4	x	x	2	3 ^{8,3}
Direct Page(DP)	CMP <i>dp</i>	C5	x	x	2	3 ^{1,3}
Direct Page(DP)	DEC <i>dp</i>	C6	x	x	2	5 ^{2,3}
DP indirect Long	CMP <i>[dp]</i>	C7		x	2	6 ^{1,3}
Implicite	INY	C8	x	x	1	2
Immédiat	CMP <i>#const</i>	C9	x	x	2	2 ¹
Implicite	DEX	CA	x	x	1	2
Implicite	WAI	CB		x	1	3 ¹¹
Absolue	CPY <i>addr</i>	CC	x	x	3	4 ⁸
Absolue	CMP <i>addr</i>	CD	x	x	3	4 ¹
Absolue	DEC <i>addr</i>	CE	x	x	3	6 ²
Absolue Long	CMP <i>long</i>	CF		x	4	5 ¹
Compteur de Programme relatif	BNE <i>nearlabel</i>	D0	x	x	2	2 ^{5,6}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
DP indirectement indexé par Y	CMP (<i>dp</i>), <i>Y</i>	D1	x	x	2	5 ^{1,3,4}
DP indirect	CMP (<i>dp</i>)	D2		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	CMP (<i>sr,S</i>), <i>Y</i>	D3		x	2	7 ¹
Stack(Direct Page Indirect)	PEI (<i>dp</i>)	D4		x	2	6 ³
DP indexé par X	CMP <i>dp,X</i>	D5	x	x	2	4 ^{1,3}
DP indexé par X	DEC <i>dp,X</i>	D6	x	x	2	6 ^{2,3}
DP long indirectement indexé par Y	CMP [<i>dp</i>], <i>Y</i>	D7		x	2	6 ^{1,3}
Implicite	CLD	D8	x	x	1	2
Absolute indexé par Y	CMP <i>addr,Y</i>	D9	x	x	3	4 ^{1,4}
Stack (push)	PHX	DA		x	1	3 ⁸
Implicite	STP	DB		x	1	3 ¹⁰
Absolute indirect long	JMP [<i>addr</i>]	DC		x	3	6
Absolute indirect	JML (<i>addr</i>)	DC		x	3	4
Absolute indexé par X	CMP <i>addr,X</i>	DD	x	x	3	4 ^{1,4}
Absolute indexé par X	DEC <i>addr,X</i>	DE	x	x	3	7 ^{2,4}
Absolute Long indexé par X	CMP <i>long,X</i>	DF		x	4	5 ¹
Immédiat	CPX <i>#const</i>	E0	x	x	2	2 ⁸
DP indirectement indexé par X	SBC (<i>dp,X</i>)	E1	x	x	2	6 ^{1,3}
Immédiat	SEP <i>#const</i>	E2		x	2	3
Stack relative (SR)	SBC <i>sr,S</i>	E3		x	2	4 ¹
Direct Page(DP)	CPX <i>dp</i>	E4	x	x	2	3 ^{8,3}
Direct Page(DP)	SBC <i>dp</i>	E5	x	x	2	3 ^{1,3}
Direct Page(DP)	INC <i>dp</i>	E6	x	x	2	5 ^{2,3}
DP indirect Long	SBC [<i>dp</i>]	E7		x	2	6 ^{1,3}
Implicite	INX	E8	x	x	1	2
Immédiat	SBC <i>#const</i>	E9	x	x	2	2 ¹
Implicite	NOP	EA	x	x	1	2
Implicite	XBA	EB		x	1	3
Implicite	SWA	EB		x	1	3
Absolute	CPX <i>addr</i>	EC	x	x	3	4 ⁸
Absolute	SBC <i>addr</i>	ED	x	x	3	4 ¹
Absolute	INC <i>addr</i>	EE	x	x	3	6 ²
Absolute Long	SBC <i>long</i>	EF		x	4	5 ¹
Compteur de Programme relatif	BEQ <i>nearlabel</i>	F0	x	x	2	2 ^{5,6}
DP indirectement indexé par Y	SBC (<i>dp</i>), <i>Y</i>	F1	x	x	2	5 ^{1,3,4}
DP indirect	SBC (<i>dp</i>)	F2		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	SBC (<i>sr,S</i>), <i>Y</i>	F3		x	2	7 ¹
Stack absolute	PEA <i>addr</i>	F4		x	3	5
DP indexé par X	SBC <i>dp,X</i>	F5	x	x	2	4 ^{1,3}
DP indexé par X	INC <i>dp,X</i>	F6	x	x	2	6 ^{2,3}
DP long indirectement indexé par Y	SBC [<i>dp</i>], <i>Y</i>	F7		x	2	6 ^{1,3}
Implicite	SED	F8	x	x	1	2

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d'</i>	<i>nb de</i>
			<i>6502</i>	<i>65816</i>	<i>Octets</i>	<i>Cycles</i>
Absolute indexé par Y	SBC <i>addr, Y</i>	F9	x	x	3	4 ^{1,4}
Stack (pull)	PLX	FA		x	1	4 ⁸
Implicite	XCE	FB		x	1	2
Absolute indirect indexé par X	JSR (<i>addr, X</i>)	FC		x	3	8
Absolute indexé par X	SBC <i>addr, X</i>	FD	x	x	3	4 ^{1,4}
Absolute indexé par X	INC <i>addr, X</i>	FE	x	x	3	7 ^{2,4}
Absolute Long indexé par X	SBC <i>long, X</i>	FF		x	4	5 ¹

ADC,AND,CMD,EOR,LDA,ORA,SBC,STA, une instruction de groupe primaire, a disponible tous les modes d'adressage du groupe primaires et les modèles binaires

+ - BRK est sur 1 Octet, mais la valeur du compteur de programme poussé dans la stak(pile) est incrémenté de 2 suivant

l'octet signé facultatif

++ - COP est sur 1 Octet, mais la valeur du compteur de programme poussé dans la stak(pile) est incrémenté de 2 suivant

l'octet codé facultatif

* - Ajoutez 1 octet si m=0 (mémoire et accumulateur à 16bits)

** - Ajoutez 1 octet si x=0 (registre d'index à 16bits)

1- Ajoutez 1 cycle si m=0 (mémoire et accumulateur à 16bits)

2- Ajoutez 2 cycle si m=0 (mémoire et accumulateur à 16bits)

3- Ajoutez 1 cycle si l'octet bas du Registre de Page Zéro (D) est autre que 0 (DL <>0)

4- Ajoutez 1 cycle si l'ajout d'index dépasse la page courante (si \$00:0000 à \$01:0000 par ex)

5- Ajoutez 1 cycle si la condition est vrai, si l'on branche

6- Ajoutez 1 cycle si vous êtes en mode émulation (e=1)

7- Ajoutez 1 cycle si vous êtes en mode native (e=0)

8- Ajoutez 1 cycle si x =0 (registre d'index à 16 bits)

9- Ajoutez 7 cycles par octets déplacé

10- Utilise 3 cycles pour éteindre le processeur, des cycles additionnels sont requis pour le redémarrer par reset

11- Utilise 3 cycles pour éteindre le processeur, des cycles additionnels sont requis pour le redémarrer par interruption

24.2 Classements par Instruction

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
DP indirectement indexé par X	ADC (<i>dp,X</i>)	61	x	x	2	6 ^{1,3}
Stack relative (SR)	ADC <i>sr,S</i>	63		x	2	4 ¹
Direct Page(DP)	ADC <i>dp</i>	65	x	x	2	3 ^{1,3}
DP indirect Long	ADC [<i>dp</i>]	67		x	2	6 ^{1,3}
Immédiat	ADC <i>#const</i>	69	x	x	2	2 ¹
Absolute	ADC <i>addr</i>	6D	x	x	3	4 ¹
Absolute Long	ADC <i>long</i>	6F		x	4	5 ¹
DP indirectement indexé par Y	ADC (<i>dp</i>), <i>Y</i>	71	x	x	2	5 ^{1,3,4}
DP indirect	ADC (<i>dp</i>)	72		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	ADC (<i>sr,S</i>), <i>Y</i>	73		x	2	7 ¹
DP indexé par X	ADC <i>dp,X</i>	75	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	ADC [<i>dp</i>], <i>Y</i>	77		x	2	6 ^{1,3}
Absolute indexé par Y	ADC <i>addr,Y</i>	79	x	x	3	4 ^{1,4}
Absolute indexé par X	ADC <i>addr,X</i>	7D	x	x	3	4 ^{1,4}
Absolute Long indexé par X	ADC <i>long,X</i>	7F		x	4	5 ¹
DP indirectement indexé par X	AND (<i>dp,X</i>)	21	x	x	2	6 ^{1,3}
Stack relative (SR)	AND <i>sr,S</i>	23		x	2	4 ¹
Direct Page(DP)	AND <i>dp</i>	25	x	x	2	3 ^{1,3}
DP indirect Long	AND [<i>dp</i>]	27		x	2	6 ^{1,3}
Immédiat	AND <i>#const</i>	29	x	x	2	2 ¹
Absolute	AND <i>addr</i>	2D	x	x	3	4 ¹
Absolute Long	AND <i>long</i>	2F		x	4	5 ¹
DP indirectement indexé par Y	AND (<i>dp</i>), <i>Y</i>	31	x	x	2	5 ^{1,3,4}
DP indirect	AND (<i>dp</i>)	32		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	AND (<i>sr,S</i>), <i>Y</i>	33		x	2	7 ¹
DP indexé par X	AND <i>dp,X</i>	35	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	AND [<i>dp</i>], <i>Y</i>	37		x	2	6 ^{1,3}
Absolute indexé par Y	AND <i>addr,Y</i>	39	x	x	3	4 ^{1,4}
Absolute indexé par X	AND <i>addr,X</i>	3D	x	x	3	4 ^{1,4}
Absolute Long indexé par X	AND <i>long,X</i>	3F		x	4	5 ¹
Direct Page(DP)	ASL <i>dp</i>	06	x	x	2	5 ^{2,3}
Accumulateur	ASL A	0A	x	x	1	2
Absolute	ASL <i>addr</i>	0E	x	x	3	6 ²
DP indexé par X	ASL <i>dp,X</i>	16	x	x	2	6 ^{2,3}
Absolute indexé par X	ASL <i>addr,X</i>	1E	x	x	3	7 ^{2,4}
Compteur de Programme relatif	BCC <i>nearlabel</i>	90	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BCS <i>nearlabel</i>	B0	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BEQ <i>nearlabel</i>	F0	x	x	2	2 ^{5,6}
Direct Page(DP)	BIT <i>dp</i>	24	x	x	2	3 ^{1,3}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Absolute	BIT <i>addr</i>	2C	x	x	3	4 ¹
DP indexé par X	BIT <i>dp,X</i>	34		x	2	6 ^{1,3}
Absolute indexé par X	BIT <i>addr,X</i>	3C		x	3	4 ^{1,4}
Immédiat	BIT <i>#const</i>	89		x	2	2 ¹
Compteur de Programme relatif	BMI <i>nearlabel</i>	30	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BNE <i>nearlabel</i>	D0	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BPL <i>nearlabel</i>	10	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BRA <i>nearlabel</i>	80	x	x	2	3 ⁶
Stack/Interrupt	BRK	00	x	x	2 ⁺	7 ⁷
Compteur de Programme relatif Long	BRL <i>label</i>	82		x	3	4
Compteur de Programme relatif	BTL <i>nearlabel</i>	90	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BVC <i>nearlabel</i>	50	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BVS <i>nearlabel</i>	70	x	x	2	2 ^{5,6}
Implicite	CLC	18	x	x	1	2
Implicite	CLD	D8	x	x	1	2
Implicite	CLI	58	x	x	1	2
Implicite	CLV	B8	x	x	1	2
DP indirectement indexé par X	CMP (<i>dp,X</i>)	C1	x	x	2	6 ^{1,3}
Stack relative (SR)	CMP <i>sr,S</i>	C3		x	2	4 ¹
Direct Page(DP)	CMP <i>dp</i>	C5	x	x	2	3 ^{1,3}
DP indirect Long	CMP [<i>dp</i>]	C7		x	2	6 ^{1,3}
Immédiat	CMP <i>#const</i>	C9	x	x	2	2 ¹
Absolute	CMP <i>addr</i>	CD	x	x	3	4 ¹
Absolute Long	CMP <i>long</i>	CF		x	4	5 ¹
DP indirectement indexé par Y	CMP (<i>dp</i>), <i>Y</i>	D1	x	x	2	5 ^{1,3,4}
DP indirect	CMP (<i>dp</i>)	D2		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	CMP (<i>sr,S</i>), <i>Y</i>	D3		x	2	7 ¹
DP indexé par X	CMP <i>dp,X</i>	D5	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	CMP [<i>dp</i>], <i>Y</i>	D7		x	2	6 ^{1,3}
Absolute indexé par Y	CMP <i>addr,Y</i>	D9	x	x	3	4 ^{1,4}
Absolute indexé par X	CMP <i>addr,X</i>	DD	x	x	3	4 ^{1,4}
Absolute Long indexé par X	CMP <i>long,X</i>	DF		x	4	5 ¹
Stack/Interrupt	COP	02		x	2 ⁺⁺	7 ⁷
Immédiat	CPX <i>#const</i>	E0	x	x	2	2 ⁸
Direct Page(DP)	CPX <i>dp</i>	E4	x	x	2	3 ^{8,3}
Absolute	CPX <i>addr</i>	EC	x	x	3	4 ⁸
Immédiat	CPY <i>#const</i>	C0	x	x	2	2 ⁸
Direct Page(DP)	CPY <i>dp</i>	C4	x	x	2	3 ^{8,3}
Absolute	CPY <i>addr</i>	CC	x	x	3	4 ⁸

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Accumulateur	DEC A	3A		x	1	2
Direct Page(DP)	DEC <i>dp</i>	C6	x	x	2	5 ^{2,3}
Absolue	DEC <i>addr</i>	CE	x	x	3	6 ²
DP indexé par X	DEC <i>dp,X</i>	D6	x	x	2	6 ^{2,3}
Absolue indexé par X	DEC <i>addr,X</i>	DE	x	x	3	7 ^{2,4}
Implicite	DEX	CA	x	x	1	2
Implicite	DEY	88	x	x	1	2
DP indirectement indexé par X	EOR (<i>dp,X</i>)	41	x	x	2	6 ^{1,3}
Stack relative (SR)	EOR <i>sr,S</i>	43		x	2	4 ¹
Direct Page(DP)	EOR <i>dp</i>	45	x	x	2	3 ^{1,3}
DP indirect Long	EOR [<i>dp</i>]	47		x	2	6 ^{1,3}
Immédiat	EOR <i>#const</i>	49	x	x	2	2 ¹
Absolue	EOR <i>addr</i>	4D	x	x	3	4 ¹
Absolue Long	EOR <i>long</i>	4F		x	4	5 ¹
DP indirectement indexé par Y	EOR (<i>dp</i>), Y	51	x	x	2	5 ^{1,3,4}
DP indirect	EOR (<i>dp</i>)	52		x	2	5 ^{1,3}
SR indirectement indexé in- dexé par Y	EOR (<i>sr,S</i>), Y	53		x	2	7 ¹
DP indexé par X	EOR <i>dp,X</i>	55	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	EOR [<i>dp</i>], Y	57		x	2	6 ^{1,3}
Absolue indexé par Y	EOR <i>addr, Y</i>	59	x	x	3	4 ^{1,4}
Absolue indexé par X	EOR <i>addr,X</i>	5D	x	x	3	4 ^{1,4}
Absolue Long indexé par X	EOR <i>long,X</i>	5F		x	4	5 ¹
Accumulateur	INC A	1A		x	1	2
Direct Page(DP)	INC <i>dp</i>	E6	x	x	2	5 ^{2,3}
Absolue	INC <i>addr</i>	EE	x	x	3	6 ²
DP indexé par X	INC <i>dp,X</i>	F6	x	x	2	6 ^{2,3}
Absolue indexé par X	INC <i>addr,X</i>	FE	x	x	3	7 ^{2,4}
Implicite	INX	E8	x	x	1	2
Implicite	INY	C8	x	x	1	2
Absolute indirect	JML (<i>addr</i>)	DC		x	3	4
Absolue	JMP <i>addr</i>	4C	x	x	3	3
Absolue Long	JMP <i>long</i>	5C		x	4	4
Absolute indirect	JMP (<i>addr</i>)	6C	x	x	3	5
Absolute indirect indexé par X	JMP (<i>addr,X</i>)	7C		x	3	6
Absolute indirect long	JMP [<i>addr</i>]	DC		x	3	6
Absolue Long	JSL <i>long</i>	22		x	4	8
Absolue	JSR <i>addr</i>	20		x	3	6
Absolue Long	JSR <i>long</i>	22		x	4	8
Absolute indirect indexé par X	JSR (<i>addr,X</i>)	FC		x	3	8
DP indirectement indexé par X	LDA (<i>dp,X</i>)	A1	x	x	2	6 ^{1,3}
Stack relative (SR)	LDA <i>sr,S</i>	A3		x	2	4 ¹
Direct Page(DP)	LDA <i>dp</i>	A5	x	x	2	3 ^{1,3}
DP indirect Long	LDA [<i>dp</i>]	A7		x	2	6 ^{1,3}
Immédiat	LDA <i>#const</i>	A9	x	x	2	2 ¹
Absolue	LDA <i>addr</i>	AD	x	x	3	4 ¹

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Absolute Long	LDA <i>long</i>	AF		x	4	5 ¹
DP indirectement indexé par Y	LDA (<i>dp</i>), Y	B1	x	x	2	5 ^{1,3,4}
DP indirect	LDA (<i>dp</i>)	B2		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	LDA (<i>sr,S</i>), Y	B3		x	2	7 ¹
DP indexé par X	LDA <i>dp,X</i>	B5	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	LDA [<i>dp</i>], Y	B7		x	2	6 ^{1,3}
Absolute indexé par Y	LDA <i>addr,Y</i>	B9	x	x	3	4 ^{1,4}
Absolute indexé par X	LDA <i>addr,X</i>	BD	x	x	3	4 ^{1,4}
Absolute Long indexé par X	LDA <i>long,X</i>	BF		x	4	5 ¹
Immédiat	LDX <i>#const</i>	A2		x	2	2 ⁸
Direct Page(DP)	LDX <i>dp</i>	A6	x	x	2	3 ^{8,3}
Absolute	LDX <i>addr</i>	AE	x	x	3	4 ⁸
DP indirectement indexé par X	LDX (<i>dp,X</i>)	B6		x	2	4 ^{8,3}
Absolute indexé par X	LDX <i>addr,X</i>	BE		x	3	4 ^{8,4}
Immédiat	LDY <i>#const</i>	A0		x	2	2 ⁸
Direct Page(DP)	LDY <i>dp</i>	A4	x	x	2	3 ^{8,3}
Absolute	LDY <i>addr</i>	AC	x	x	3	4 ⁸
DP indirectement indexé par X	LDY (<i>dp,X</i>)	B4		x	2	4 ^{8,3}
Absolute indexé par X	LDY <i>addr,X</i>	BC		x	3	4 ^{8,4}
Direct Page(DP)	LSR <i>dp</i>	46	x	x	2	5 ^{2,3}
Accumulateur	LSR A	4A		x	1	2
Absolute	LSR <i>addr</i>	4E	x	x	3	6 ²
DP indexé par X	LSR <i>dp,X</i>	56	x	x	2	6 ^{2,3}
Absolute indexé par X	LSR <i>addr,X</i>	5E	x	x	3	7 ^{2,4}
Block Move	MVN	54	x	x	3	9
Block Move	MVP <i>srcbk,destbk</i>	44		x	3	9
Implicite	NOP	EA	x	x	1	2
DP indirectement indexé par X	ORA (<i>dp,X</i>)	01	x	x	2	6 ^{1,3}
Stack relative (SR)	ORA <i>sr,S</i>	03		x	2	4 ¹
Direct Page(DP)	ORA <i>dp</i>	05	x	x	2	3 ^{1,3}
DP indirect Long	ORA [<i>dp</i>]	07		x	2	6 ^{1,3}
Immédiat	ORA <i>#const</i>	09	x	x	2	2 ¹
Absolute	ORA <i>addr</i>	0D	x	x	3	4 ¹
Absolute Long	ORA <i>long</i>	0F		x	4	5 ¹
DP indirectement indexé par Y	ORA (<i>dp</i>), Y	11	x	x	2	5 ^{1,3,4}
DP indirect	ORA (<i>dp</i>)	12		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	ORA (<i>sr,S</i>), Y	13		x	2	7 ¹
DP indexé par X	ORA <i>dp,X</i>	15	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	ORA [<i>dp</i>], Y	17		x	2	6 ^{1,3}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Absolute indexé par Y	ORA <i>addr, Y</i>	19	x	x	3	4 ^{1,4}
Absolute indexé par X	ORA <i>addr, X</i>	1D	x	x	3	4 ^{1,4}
Absolute Long indexé par X	ORA <i>long, X</i>	1F		x	4	5 ¹
Stack absolute	PEA <i>addr</i>	F4		x	3	5
Stack(Direct Page Indirect)	PEI (<i>dp</i>)	D4		x	2	6 ³
Stack (Compteur de programme relatif long)	PER <i>label</i>	62			3	6
Stack (push)	PHA	48	x	x	1	3 ¹
Stack (push)	PHB	8B		x	1	3
Stack (push)	PHD	0B		x	1	4
Stack (push)	PHK	4B		x	1	3
Stack (push)	PHP	08	x	x	1	3
Stack (push)	PHX	DA		x	1	3 ⁸
Stack (push)	PHY	5A		x	1	3 ⁸
Stack (pull)	PLA	68	x	x	1	4 ⁸
Stack (pull)	PLB	AB		x	1	4
Stack (pull)	PLD	2B		x	1	5
Stack (pull)	PLP	28	x	x	1	4
Stack (pull)	PLX	FA		x	1	4 ⁸
Stack (pull)	PLY	7A		x	1	4 ⁸
Immédiat	REP <i>#const</i>	C2		x	2	3
Direct Page(DP)	ROL <i>dp</i>	26	x	x	2	5 ^{2,3}
Accumulateur	ROL A	2A		x	1	2
Absolute	ROL <i>addr</i>	2E	x	x	3	6 ²
DP indexé par X	ROL <i>dp, X</i>	36	x	x	2	6 ^{2,3}
Absolute indexé par X	ROL <i>addr, X</i>	3E	x	x	3	7 ^{2,4}
Direct Page(DP)	ROR <i>dp</i>	66	x	x	2	5 ^{2,3}
Accumulateur	ROR A	6A	x	x	1	2
Absolute	ROR <i>addr</i>	6E	x	x	3	6 ²
DP indexé par X	ROR <i>dp, X</i>	76	x	x	2	6 ^{2,3}
Absolute indexé par X	ROR <i>addr, X</i>	7E	x	x	3	7 ^{2,4}
Stack (RTI)	RTI	40	x	x	1	6 ⁷
Stack (RTL)	RTL	6B		x	1	6
Stack (RTS)	RTS	60	x	x	1	6
DP indirectement indexé par X	SBC (<i>dp, X</i>)	E1	x	x	2	6 ^{1,3}
Stack relative (SR)	SBC <i>sr, S</i>	E3		x	2	4 ¹
Direct Page(DP)	SBC <i>dp</i>	E5	x	x	2	3 ^{1,3}
DP indirect Long	SBC [<i>dp</i>]	E7		x	2	6 ^{1,3}
Immédiat	SBC <i>#const</i>	E9	x	x	2	2 ¹
Absolute	SBC <i>addr</i>	ED	x	x	3	4 ¹
Absolute Long	SBC <i>long</i>	EF		x	4	5 ¹
DP indirectement indexé par Y	SBC (<i>dp</i>), Y	F1	x	x	2	5 ^{1,3,4}
DP indirect	SBC (<i>dp</i>)	F2		x	2	5 ^{1,3}
SR indirectement indexé indirectement par Y	SBC (<i>sr, S</i>), Y	F3		x	2	7 ¹
DP indexé par X	SBC <i>dp, X</i>	F5	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	SBC [<i>dp</i>], Y	F7		x	2	6 ^{1,3}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Absolute indexé par Y	SBC <i>addr, Y</i>	F9	x	x	3	4 ^{1,4}
Absolute indexé par X	SBC <i>addr, X</i>	FD	x	x	3	4 ^{1,4}
Absolute Long indexé par X	SBC <i>long, X</i>	FF		x	4	5 ¹
Implicite	SEC	38	x	x	1	2
Implicite	SED	F8	x	x	1	2
Implicite	SEI	78	x	x	1	2
Immédiat	SEP <i>#const</i>	E2		x	2	3
DP indirectement indexé par X	STA <i>(dp, X)</i>	81	x	x	2	6 ^{1,3}
Stack relative (SR)	STA <i>sr, S</i>	83		x	2	4 ¹
Direct Page(DP)	STA <i>dp</i>	85	x	x	2	3 ^{1,3}
DP indirect Long	STA <i>[dp]</i>	87		x	2	6 ^{1,3}
Absolute	STA <i>addr</i>	8D	x	x	3	4 ¹
Absolute Long	STA <i>long</i>	8F		x	4	5 ¹
DP indirectement indexé par Y	STA <i>(dp), Y</i>	91	x	x	2	6 ^{1,3,4}
DP indirect	STA <i>(dp)</i>	92		x	2	5 ^{1,3}
SR indirectement indexé indirectement indexé par Y	STA <i>(sr, S), Y</i>	93		x	2	7 ¹
DP indexé par X	STA <i>dp, X</i>	95	x	x	2	4 ^{1,3}
DP long indirectement indexé par Y	STA <i>[dp], Y</i>	97		x	2	6 ^{1,3}
Absolute indexé par Y	STA <i>addr, Y</i>	99	x	x	3	5 ¹
Absolute indexé par X	STA <i>addr, X</i>	9D	x	x	3	5 ¹
Absolute Long indexé par X	STA <i>long, X</i>	9F		x	4	5 ¹
Implicite	STP	DB		x	1	3 ¹⁰
Direct Page(DP)	STX <i>dp</i>	86	x	x	2	3 ^{8,3}
Absolute	STX <i>addr</i>	8E	x	x	3	4 ⁸
DP indexé par Y	STX <i>dp, Y</i>	96	x	x	2	4 ^{8,3}
Direct Page(DP)	STY <i>dp</i>	84	x	x	2	3 ^{8,3}
Absolute	STY <i>addr</i>	8C	x	x	3	4 ⁸
DP indexé par Y	STY <i>dp, Y</i>	94	x	x	2	4 ^{8,3}
Direct Page(DP)	STZ <i>dp</i>	64	x	x	2	3 ^{1,3}
DP indexé par X	STZ <i>dp, X</i>	74	x	x	2	4 ^{2,3}
Absolute	STZ <i>addr</i>	9C	x	x	3	4 ¹
Absolute indexé par X	STZ <i>addr, X</i>	9E	x	x	3	5 ¹
Implicite	SWA	EB		x	1	3
Implicite	TAX	AA	x	x	1	2
Implicite	TAY	A8	x	x	1	2
Implicite	TCD	5B		x	1	2
Implicite	TCS	1B		x	1	2
Implicite	TDC	7B		x	1	2
Direct Page(DP)	TRB <i>dp</i>	14		x	2	5 ^{2,3}
Absolute	TRB <i>addr</i>	1C		x	3	6 ²
Implicite	TSA	3B		x	1	2
Direct Page(DP)	TSB <i>dp</i>	04		x	2	5 ^{2,3}
Absolute	TSB <i>addr</i>	0C		x	3	6 ²
Implicite	TSC	3B		x	1	2
Implicite	TSX	BA	x	x	1	2
Implicite	TXA	8A	x	x	1	2

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Implicite	TXS	9A	x	x	1	2
Implicite	TXY	9B		x	1	2
Implicite	TYA	98	x	x	1	2
Implicite	TYX	BB		x	1	2
Implicite	WAI	CB		x	1	3 ¹¹
	WDM	42		x	2 ^x	x
Implicite	XBA	EB		x	1	3
Implicite	XCE	FB		x	1	2

ADC,AND,CMD,EOR,LDA,ORA,SBC,STA, une instruction de groupe primaire, a disponible tous les modes d'adressage du groupe primaires et les modèles binaires

+ - BRK est sur 1 Octet, mais la valeur du compteur de programme poussé dans la stak(pile) est incrémenté de 2 suivant

l'octet signé facultatif

++ - COP est sur 1 Octet, mais la valeur du compteur de programme poussé dans la stak(pile) est incrémenté de 2 suivant

l'octet codé facultatif

* - Ajoutez 1 octet si m=0 (mémoire et accumulateur à 16bits)

** - Ajoutez 1 octet si x=0 (registre d'index à 16bits)

1- Ajoutez 1 cycle si m=0 (mémoire et accumulateur à 16bits)

2- Ajoutez 2 cycle si m=0 (mémoire et accumulateur à 16bits)

3- Ajoutez 1 cycle si l'octet bas du Registre de Page Zéro (D) est autre que 0 (DL <>0)

4- Ajoutez 1 cycle si l'ajout d'index dépasse la page courante (si \$00:0000 à \$01:0000 par ex)

5- Ajouter 1 cycle si la condition est vrai, si l'on branche

6- Ajouter 1 cycle si vous êtes en mode émulation (e=1)

7- Ajouter 1 cycle si vous êtes en mode native (e=0)

8- Ajouter 1 cycle si x =0 (registre d'index à 16 bits)

9- Ajouter 7 cycles par octets déplacé

10- Utilise 3 cycles pour éteindre le processeur, des cycles additionnels sont requis pour le redémarrer par reset

11- Utilise 3 cycles pour éteindre le processeur, des cycles additionnels sont requis pour le redémarrer par interruption

24.3 Classements par Mode d'adressage

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Absolue	ADC <i>addr</i>	6D	x	x	3	4 ¹
Absolue	AND <i>addr</i>	2D	x	x	3	4 ¹
Absolue	ASL <i>addr</i>	0E	x	x	3	6 ²
Absolue	BIT <i>addr</i>	2C	x	x	3	4 ¹
Absolue	CMP <i>addr</i>	CD	x	x	3	4 ¹
Absolue	CPX <i>addr</i>	EC	x	x	3	4 ⁸
Absolue	CPY <i>addr</i>	CC	x	x	3	4 ⁸
Absolue	DEC <i>addr</i>	CE	x	x	3	6 ²
Absolue	EOR <i>addr</i>	4D	x	x	3	4 ¹
Absolue	INC <i>addr</i>	EE	x	x	3	6 ²
Absolue	JMP <i>addr</i>	4C	x	x	3	3
Absolue	JSR <i>addr</i>	20		x	3	6
Absolue	LDA <i>addr</i>	AD	x	x	3	4 ¹
Absolue	LDX <i>addr</i>	AE	x	x	3	4 ⁸
Absolue	LDY <i>addr</i>	AC	x	x	3	4 ⁸
Absolue	LSR <i>addr</i>	4E	x	x	3	6 ²
Absolue	ORA <i>addr</i>	0D	x	x	3	4 ¹
Absolue	ROL <i>addr</i>	2E	x	x	3	6 ²
Absolue	ROR <i>addr</i>	6E	x	x	3	6 ²
Absolue	SBC <i>addr</i>	ED	x	x	3	4 ¹
Absolue	STA <i>addr</i>	8D	x	x	3	4 ¹
Absolue	STX <i>addr</i>	8E	x	x	3	4 ⁸
Absolue	STY <i>addr</i>	8C	x	x	3	4 ⁸
Absolue	STZ <i>addr</i>	9C	x	x	3	4 ¹
Absolue	TRB <i>addr</i>	1C		x	3	6 ²
Absolue	TSB <i>addr</i>	0C		x	3	6 ²
Asolue indirect	JML (<i>addr</i>)	DC		x	3	4
Asolue indirect	JMP (<i>addr</i>)	6C	x	x	3	5
Asolue indirect long	JMP [<i>addr</i>]	DC		x	3	6
Absolue Long	ADC <i>long</i>	6F		x	4	5 ¹
Absolue Long	AND <i>long</i>	2F		x	4	5 ¹
Absolue Long	CMP <i>long</i>	CF		x	4	5 ¹
Absolue Long	EOR <i>long</i>	4F		x	4	5 ¹
Absolue Long	JMP <i>long</i>	5C		x	4	4
Absolue Long	JSL <i>long</i>	22		x	4	8
Absolue Long	JSR <i>long</i>	22		x	4	8
Absolue Long	LDA <i>long</i>	AF		x	4	5 ¹
Absolue Long	ORA <i>long</i>	0F		x	4	5 ¹
Absolue Long	SBC <i>long</i>	EF		x	4	5 ¹
Absolue Long	STA <i>long</i>	8F		x	4	5 ¹
Absolue indexé par X	ADC <i>addr,X</i>	7D	x	x	3	4 ^{1,4}
Absolue indexé par X	AND <i>addr,X</i>	3D	x	x	3	4 ^{1,4}
Absolue indexé par X	ASL <i>addr,X</i>	1E	x	x	3	7 ^{2,4}
Absolue indexé par X	BIT <i>addr,X</i>	3C		x	3	4 ^{1,4}
Absolue indexé par X	CMP <i>addr,X</i>	DD	x	x	3	4 ^{1,4}
Absolue indexé par X	DEC <i>addr,X</i>	DE	x	x	3	7 ^{2,4}
Absolue indexé par X	EOR <i>addr,X</i>	5D	x	x	3	4 ^{1,4}
Absolue indexé par X	INC <i>addr,X</i>	FE	x	x	3	7 ^{2,4}
Absolue indexé par X	LDA <i>addr,X</i>	BD	x	x	3	4 ^{1,4}
Absolue indexé par X	LDX <i>addr,X</i>	BE		x	3	4 ^{8,4}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Absolute indexé par X	LDY <i>addr,X</i>	BC		x	3	4 ^{8,4}
Absolute indexé par X	LSR <i>addr,X</i>	5E	x	x	3	7 ^{2,4}
Absolute indexé par X	ORA <i>addr,X</i>	1D	x	x	3	4 ^{1,4}
Absolute indexé par X	ROL <i>addr,X</i>	3E	x	x	3	7 ^{2,4}
Absolute indexé par X	ROR <i>addr,X</i>	7E	x	x	3	7 ^{2,4}
Absolute indexé par X	SBC <i>addr,X</i>	FD	x	x	3	4 ^{1,4}
Absolute indexé par X	STA <i>addr,X</i>	9D	x	x	3	5 ¹
Absolute indexé par X	STZ <i>addr,X</i>	9E	x	x	3	5 ¹
Absolute Long indexé par X	ADC <i>long,X</i>	7F		x	4	5 ¹
Absolute Long indexé par X	AND <i>long,X</i>	3F		x	4	5 ¹
Absolute Long indexé par X	CMP <i>long,X</i>	DF		x	4	5 ¹
Absolute Long indexé par X	EOR <i>long,X</i>	5F		x	4	5 ¹
Absolute Long indexé par X	LDA <i>long,X</i>	BF		x	4	5 ¹
Absolute Long indexé par X	ORA <i>long,X</i>	1F		x	4	5 ¹
Absolute Long indexé par X	SBC <i>long,X</i>	FF		x	4	5 ¹
Absolute Long indexé par X	STA <i>long,X</i>	9F		x	4	5 ¹
Absolute indirect indexé par X	JMP (<i>addr,X</i>)	7C		x	3	6
Absolute indirect indexé par X	JSR (<i>addr,X</i>)	FC		x	3	8
Absolute indexé par Y	ADC <i>addr,Y</i>	79	x	x	3	4 ^{1,4}
Absolute indexé par Y	AND <i>addr,Y</i>	39	x	x	3	4 ^{1,4}
Absolute indexé par Y	CMP <i>addr,Y</i>	D9	x	x	3	4 ^{1,4}
Absolute indexé par Y	EOR <i>addr,Y</i>	59	x	x	3	4 ^{1,4}
Absolute indexé par Y	LDA <i>addr,Y</i>	B9	x	x	3	4 ^{1,4}
Absolute indexé par Y	ORA <i>addr,Y</i>	19	x	x	3	4 ^{1,4}
Absolute indexé par Y	SBC <i>addr,Y</i>	F9	x	x	3	4 ^{1,4}
Absolute indexé par Y	STA <i>addr,Y</i>	99	x	x	3	5 ¹
Accumulateur	ASL A	0A	x	x	1	2
Accumulateur	DEC A	3A		x	1	2
Accumulateur	INC A	1A		x	1	2
Accumulateur	LSR A	4A		x	1	2
Accumulateur	ROL A	2A		x	1	2
Accumulateur	ROR A	6A	x	x	1	2
Block Move	MVN	54	x	x	3	9
Block Move	<i>srcbk,destbk</i> MVP	44		x	3	9
Direct Page(DP)	ADC <i>dp</i>	65	x	x	2	3 ^{1,3}
Direct Page(DP)	AND <i>dp</i>	25	x	x	2	3 ^{1,3}
Direct Page(DP)	ASL <i>dp</i>	06	x	x	2	5 ^{2,3}
Direct Page(DP)	BIT <i>dp</i>	24	x	x	2	3 ^{1,3}
Direct Page(DP)	CMP <i>dp</i>	C5	x	x	2	3 ^{1,3}
Direct Page(DP)	CPX <i>dp</i>	E4	x	x	2	3 ^{8,3}
Direct Page(DP)	CPY <i>dp</i>	C4	x	x	2	3 ^{8,3}
Direct Page(DP)	DEC <i>dp</i>	C6	x	x	2	5 ^{2,3}
Direct Page(DP)	EOR <i>dp</i>	45	x	x	2	3 ^{1,3}
Direct Page(DP)	INC <i>dp</i>	E6	x	x	2	5 ^{2,3}
Direct Page(DP)	LDA <i>dp</i>	A5	x	x	2	3 ^{1,3}
Direct Page(DP)	LDX <i>dp</i>	A6	x	x	2	3 ^{8,3}
Direct Page(DP)	LDY <i>dp</i>	A4	x	x	2	3 ^{8,3}
Direct Page(DP)	LSR <i>dp</i>	46	x	x	2	5 ^{2,3}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Direct Page(DP)	ORA <i>dp</i>	05	x	x	2	3 ^{1,3}
Direct Page(DP)	ROL <i>dp</i>	26	x	x	2	5 ^{2,3}
Direct Page(DP)	ROR <i>dp</i>	66	x	x	2	5 ^{2,3}
Direct Page(DP)	SBC <i>dp</i>	E5	x	x	2	3 ^{1,3}
Direct Page(DP)	STA <i>dp</i>	85	x	x	2	3 ^{1,3}
Direct Page(DP)	STX <i>dp</i>	86	x	x	2	3 ^{8,3}
Direct Page(DP)	STY <i>dp</i>	84	x	x	2	3 ^{8,3}
Direct Page(DP)	STZ <i>dp</i>	64	x	x	2	3 ^{1,3}
Direct Page(DP)	TRB <i>dp</i>	14		x	2	5 ^{2,3}
Direct Page(DP)	TSB <i>dp</i>	04		x	2	5 ^{2,3}
DP indexé par X	ADC <i>dp,X</i>	75	x	x	2	4 ^{1,3}
DP indexé par X	AND <i>dp,X</i>	35	x	x	2	4 ^{1,3}
DP indexé par X	ASL <i>dp,X</i>	16	x	x	2	6 ^{2,3}
DP indexé par X	BIT <i>dp,X</i>	34		x	2	6 ^{1,3}
DP indexé par X	CMP <i>dp,X</i>	D5	x	x	2	4 ^{1,3}
DP indexé par X	DEC <i>dp,X</i>	D6	x	x	2	6 ^{2,3}
DP indexé par X	EOR <i>dp,X</i>	55	x	x	2	4 ^{1,3}
DP indexé par X	INC <i>dp,X</i>	F6	x	x	2	6 ^{2,3}
DP indexé par X	LDA <i>dp,X</i>	B5	x	x	2	4 ^{1,3}
DP indexé par X	LSR <i>dp,X</i>	56	x	x	2	6 ^{2,3}
DP indexé par X	ORA <i>dp,X</i>	15	x	x	2	4 ^{1,3}
DP indexé par X	ROL <i>dp,X</i>	36	x	x	2	6 ^{2,3}
DP indexé par X	ROR <i>dp,X</i>	76	x	x	2	6 ^{2,3}
DP indexé par X	SBC <i>dp,X</i>	F5	x	x	2	4 ^{1,3}
DP indexé par X	STA <i>dp,X</i>	95	x	x	2	4 ^{1,3}
DP indexé par X	STZ <i>dp,X</i>	74	x	x	2	4 ^{2,3}
DP indirectement indexé par X	ADC (<i>dp,X</i>)	61	x	x	2	6 ^{1,3}
DP indirectement indexé par X	AND (<i>dp,X</i>)	21	x	x	2	6 ^{1,3}
DP indirectement indexé par X	CMP (<i>dp,X</i>)	C1	x	x	2	6 ^{1,3}
DP indirectement indexé par X	EOR (<i>dp,X</i>)	41	x	x	2	6 ^{1,3}
DP indirectement indexé par X	LDA (<i>dp,X</i>)	A1	x	x	2	6 ^{1,3}
DP indirectement indexé par X	LDX (<i>dp,X</i>)	B6		x	2	4 ^{8,3}
DP indirectement indexé par X	LDY (<i>dp,X</i>)	B4		x	2	4 ^{8,3}
DP indirectement indexé par X	ORA (<i>dp,X</i>)	01	x	x	2	6 ^{1,3}
DP indirectement indexé par X	SBC (<i>dp,X</i>)	E1	x	x	2	6 ^{1,3}
DP indirectement indexé par X	STA (<i>dp,X</i>)	81	x	x	2	6 ^{1,3}
DP indexé par Y	STX <i>dp,Y</i>	96	x	x	2	4 ^{8,3}
DP indexé par Y	STY <i>dp,Y</i>	94	x	x	2	4 ^{8,3}
DP indirectement indexé par Y	ADC (<i>dp</i>), Y	71	x	x	2	5 ^{1,3,4}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
DP indirectement indexé par Y	AND (<i>dp</i>), Y	31	x	x	2	5 ^{1,3,4}
DP indirectement indexé par Y	CMP (<i>dp</i>), Y	D1	x	x	2	5 ^{1,3,4}
DP indirectement indexé par Y	EOR (<i>dp</i>), Y	51	x	x	2	5 ^{1,3,4}
DP indirectement indexé par Y	LDA (<i>dp</i>), Y	B1	x	x	2	5 ^{1,3,4}
DP indirectement indexé par Y	ORA (<i>dp</i>), Y	11	x	x	2	5 ^{1,3,4}
DP indirectement indexé par Y	SBC (<i>dp</i>), Y	F1	x	x	2	5 ^{1,3,4}
DP indirectement indexé par Y	STA (<i>dp</i>), Y	91	x	x	2	6 ^{1,3,4}
DP long indirectement indexé par Y	ADC [<i>dp</i>], Y	77		x	2	6 ^{1,3}
DP long indirectement indexé par Y	AND [<i>dp</i>], Y	37		x	2	6 ^{1,3}
DP long indirectement indexé par Y	CMP [<i>dp</i>], Y	D7		x	2	6 ^{1,3}
DP long indirectement indexé par Y	EOR [<i>dp</i>], Y	57		x	2	6 ^{1,3}
DP long indirectement indexé par Y	LDA [<i>dp</i>], Y	B7		x	2	6 ^{1,3}
DP long indirectement indexé par Y	ORA [<i>dp</i>], Y	17		x	2	6 ^{1,3}
DP long indirectement indexé par Y	SBC [<i>dp</i>], Y	F7		x	2	6 ^{1,3}
DP long indirectement indexé par Y	STA [<i>dp</i>], Y	97		x	2	6 ^{1,3}
DP indirect	ADC (<i>dp</i>)	72		x	2	5 ^{1,3}
DP indirect	AND (<i>dp</i>)	32		x	2	5 ^{1,3}
DP indirect	CMP (<i>dp</i>)	D2		x	2	5 ^{1,3}
DP indirect	EOR (<i>dp</i>)	52		x	2	5 ^{1,3}
DP indirect	LDA (<i>dp</i>)	B2		x	2	5 ^{1,3}
DP indirect	ORA (<i>dp</i>)	12		x	2	5 ^{1,3}
DP indirect	SBC (<i>dp</i>)	F2		x	2	5 ^{1,3}
DP indirect	STA (<i>dp</i>)	92		x	2	5 ^{1,3}
DP indirect Long	ADC [<i>dp</i>]	67		x	2	6 ^{1,3}
DP indirect Long	AND [<i>dp</i>]	27		x	2	6 ^{1,3}
DP indirect Long	CMP [<i>dp</i>]	C7		x	2	6 ^{1,3}
DP indirect Long	EOR [<i>dp</i>]	47		x	2	6 ^{1,3}
DP indirect Long	LDA [<i>dp</i>]	A7		x	2	6 ^{1,3}
DP indirect Long	ORA [<i>dp</i>]	07		x	2	6 ^{1,3}
DP indirect Long	SBC [<i>dp</i>]	E7		x	2	6 ^{1,3}
DP indirect Long	STA [<i>dp</i>]	87		x	2	6 ^{1,3}
Immédiat	ADC # <i>const</i>	69	x	x	2	2 ¹
Immédiat	AND # <i>const</i>	29	x	x	2	2 ¹
Immédiat	BIT # <i>const</i>	89		x	2	2 ¹
Immédiat	CMP # <i>const</i>	C9	x	x	2	2 ¹

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Immédiat	CPX <i>#const</i>	E0	x	x	2	2 ⁸
Immédiat	CPY <i>#const</i>	C0	x	x	2	2 ⁸
Immédiat	EOR <i>#const</i>	49	x	x	2	2 ¹
Immédiat	LDA <i>#const</i>	A9	x	x	2	2 ¹
Immédiat	LDX <i>#const</i>	A2		x	2	2 ⁸
Immédiat	LDY <i>#const</i>	A0		x	2	2 ⁸
Immédiat	ORA <i>#const</i>	09	x	x	2	2 ¹
Immédiat	REP <i>#const</i>	C2		x	2	3
Immédiat	SBC <i>#const</i>	E9	x	x	2	2 ¹
Immédiat	SEP <i>#const</i>	E2		x	2	3
Implicite	CLC	18	x	x	1	2
Implicite	CLD	D8	x	x	1	2
Implicite	CLI	58	x	x	1	2
Implicite	CLV	B8	x	x	1	2
Implicite	DEX	CA	x	x	1	2
Implicite	DEY	88	x	x	1	2
Implicite	INX	E8	x	x	1	2
Implicite	INY	C8	x	x	1	2
Implicite	NOP	EA	x	x	1	2
Implicite	SEC	38	x	x	1	2
Implicite	SED	F8	x	x	1	2
Implicite	SEI	78	x	x	1	2
Implicite	STP	DB		x	1	3 ¹⁰
Implicite	SWA	EB		x	1	3
Implicite	TAX	AA	x	x	1	2
Implicite	TAY	A8	x	x	1	2
Implicite	TCD	5B		x	1	2
Implicite	TCS	1B		x	1	2
Implicite	TDC	7B		x	1	2
Implicite	TSA	3B		x	1	2
Implicite	TSC	3B		x	1	2
Implicite	TSX	BA	x	x	1	2
Implicite	TXA	8A	x	x	1	2
Implicite	TXS	9A	x	x	1	2
Implicite	TXY	9B		x	1	2
Implicite	TYA	98	x	x	1	2
Implicite	TYX	BB		x	1	2
Implicite	WAI	CB		x	1	3 ¹¹
Implicite	XBA	EB		x	1	3
Implicite	XCE	FB		x	1	2
Compteur de Programme relatif	BCC <i>nearlabel</i>	90	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BCS <i>nearlabel</i>	B0	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BEQ <i>nearlabel</i>	F0	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BMI <i>nearlabel</i>	30	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BNE <i>nearlabel</i>	D0	x	x	2	2 ^{5,6}

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcode (Hex)</i>	<i>Disponible sur</i>		<i>nb d' Octets</i>	<i>nb de Cycles</i>
			<i>6502</i>	<i>65816</i>		
Compteur de Programme relatif	BPL <i>nearlabel</i>	10	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BRA <i>nearlabel</i>	80	x	x	2	3 ⁶
Compteur de Programme relatif	BTL <i>nearlabel</i>	90	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BVC <i>nearlabel</i>	50	x	x	2	2 ^{5,6}
Compteur de Programme relatif	BVS <i>nearlabel</i>	70	x	x	2	2 ^{5,6}
Compteur de Programme relatif Long	BRL <i>label</i>	82		x	3	4
Stack absolute	PEA <i>addr</i>	F4		x	3	5
Stack (Compteur de programme relatif long)	PER <i>label</i>	62			3	6
Stack(Direct Page Indirect)	PEI (<i>dp</i>)	D4		x	2	6 ³
Stack/Interrupt	BRK	00	x	x	2 ⁺	7 ⁷
Stack/Interrupt	COP	02		x	2 ⁺⁺	7 ⁷
Stack (push)	PHA	48	x	x	1	3 ¹
Stack (push)	PHB	8B		x	1	3
Stack (push)	PHD	0B		x	1	4
Stack (push)	PHK	4B		x	1	3
Stack (push)	PHP	08	x	x	1	3
Stack (push)	PHX	DA		x	1	3 ⁸
Stack (push)	PHY	5A		x	1	3 ⁸
Stack (pull)	PLA	68	x	x	1	4 ⁸
Stack (pull)	PLB	AB		x	1	4
Stack (pull)	PLD	2B		x	1	5
Stack (pull)	PLP	28	x	x	1	4
Stack (pull)	PLX	FA		x	1	4 ⁸
Stack (pull)	PLY	7A		x	1	4 ⁸
Stack relative (SR)	ADC <i>sr,S</i>	63		x	2	4 ¹
Stack relative (SR)	AND <i>sr,S</i>	23		x	2	4 ¹
Stack relative (SR)	CMP <i>sr,S</i>	C3		x	2	4 ¹
Stack relative (SR)	EOR <i>sr,S</i>	43		x	2	4 ¹
Stack relative (SR)	LDA <i>sr,S</i>	A3		x	2	4 ¹
Stack relative (SR)	ORA <i>sr,S</i>	03		x	2	4 ¹
Stack relative (SR)	SBC <i>sr,S</i>	E3		x	2	4 ¹
Stack relative (SR)	STA <i>sr,S</i>	83		x	2	4 ¹
SR indirectement indexé indexé par Y	ADC (<i>sr,S</i>),Y	73		x	2	7 ¹
SR indirectement indexé indexé par Y	AND (<i>sr,S</i>),Y	33		x	2	7 ¹
SR indirectement indexé indexé par Y	CMP (<i>sr,S</i>),Y	D3		x	2	7 ¹
SR indirectement indexé indexé par Y	EOR (<i>sr,S</i>),Y	53		x	2	7 ¹
SR indirectement indexé indexé par Y	LDA (<i>sr,S</i>),Y	B3		x	2	7 ¹
SR indirectement indexé indexé par Y	ORA (<i>sr,S</i>),Y	13		x	2	7 ¹

<i>Mode d'Adressage</i>	<i>Syntax</i>	<i>Opcodé</i> (Hex)	<i>Disponible sur</i>		<i>nb d'</i> <i>Octets</i>	<i>nb de</i> <i>Cycles</i>
			<i>6502</i>	<i>65816</i>		
SR indirectement indexé indexé par Y	SBC (<i>sr,S</i>),Y	F3		x	2	7 ¹
SR indirectement indexé indexé par Y	STA (<i>sr,S</i>),Y	93		x	2	7 ¹
Stack (RTI)	RTI	40	x	x	1	6 ⁷
Stack (RTL)	RTL	6B		x	1	6
Stack (RTS)	RTS	60	x	x	1	6
	WDM	42		x	2 ^x	x

ADC,AND,CMD,EOR,LDA,ORA,SBC,STA, une instruction de groupe primaire, a disponible tous les modes d'adressage du groupe primaires et les modèles binaires

+ - BRK est sur 1 Octet, mais la valeur du compteur de programme poussé dans la stak(pile) est incrémenté de 2 suivant

l'octet signé facultatif

++ - COP est sur 1 Octet, mais la valeur du compteur de programme poussé dans la stak(pile) est incrémenté de 2 suivant

l'octet codé facultatif

* - Ajoutez 1 octet si m=0 (mémoire et accumulateur à 16bits)

** - Ajoutez 1 octet si x=0 (registre d'index à 16bits)

1- Ajoutez 1 cycle si m=0 (mémoire et accumulateur à 16bits)

2- Ajoutez 2 cycle si m=0 (mémoire et accumulateur à 16bits)

3- Ajoutez 1 cycle si l'octet bas du Registre de Page Zéro (D) est autre que 0 (DL <>0)

4- Ajoutez 1 cycle si l'ajout d'index dépasse la page courante (si \$00:0000 à \$01:0000 par ex)

5- Ajouter 1 cycle si la condition est vrai, si l'on branche

6- Ajouter 1 cycle si vous êtes en mode émulation (e=1)

7- Ajouter 1 cycle si vous êtes en mode native (e=0)

8- Ajouter 1 cycle si x =0 (registre d'index à 16 bits)

9- Ajouter 7 cycles par octets déplacé

10- Utilise 3 cycles pour éteindre le processeur, des cycles additionnels sont requis pour le redémarrer par reset

11- Utilise 3 cycles pour éteindre le processeur, des cycles additionnels sont requis pour le redémarrer par interruption

Liste des tableaux

2.1	versions SNMP 1/5	29
2.2	versions SNMP 2/5	30
2.3	versions SNMP 3/5	31
2.4	versions SNMP 4/5	32
2.5	versions SNMP 5/5	33
9.1	Table des différence entre e=1& e=0	115
10.1	Instructions utilisant l'adressage inhérent	118
10.2	Notation du mode d'adressage	128
11.1	Valeurs possible avec ADC	133
11.2	Valeurs possible avec AND	134
11.3	Valeurs possible avec ASL	136
11.4	Valeurs possible avec CMP	156
11.5	Valeurs possible avec DEC	161
11.6	Valeurs possible avec DEX	162
11.7	Valeurs possible avec DEY	163
11.8	Valeurs possible avec EOR	164
11.9	Valeurs possible avec INC	166
11.10	Valeurs possible avec INX	168
11.11	Valeurs possible avec INY	169
11.12	Valeurs possible avec LDA	174
11.13	Valeurs possible avec LDX	176
11.14	Valeurs possible avec LDY	177
11.15	Valeurs possible avec LSR	179
11.16	Valeurs possible avec PLA	196
11.17	Valeurs possible avec PLB	197
11.18	Valeurs possible avec PLD	198
11.19	Valeurs possible avec PLX	200
11.20	Valeurs possible avec PLY	201
11.21	Valeurs possible avec ROL	204
11.22	Valeurs possible avec ROR	206
11.23	Valeurs possible avec SBC	211
11.24	Valeurs possible avec TAX	221
11.25	Valeurs possible avec TAY	222
11.26	Valeurs possible avec TCD	223
11.27	Valeurs possible avec TRB	226
11.28	Valeurs possible avec TSB	227
11.29	Valeurs possible avec TSX	229
11.30	Valeurs possible avec TXA	230
11.31	Valeurs possible avec TXY	232
11.32	Valeurs possible avec TYA	233
11.33	Valeurs possible avec TYX	234
11.34	Valeurs possible avec XBA	237
16.1	Information Cartouche	257
16.2	Code ASCII pour le titre du jeu	259
16.3	Mapping mémoire	260
16.4	Type de cartouche sans co-processeur	260

16.5 Type de cartouche Avec co-processeur	260
16.6 Taille de la ROM	261
16.7 Taille de la RAM CPU	261
16.8 Code Pays	262

Table des figures

4.1	Les opérandes pour addition sur 16 bits	61
4.2	Stockage d'opérandes de 16 bits dans le 65816	62
4.3	Stockage d'opérandes de 32 bits	63
4.4	Stockage de chiffres BCD	64
4.5	Addition en BCD compacté $N1 \leftarrow N2 + N1$	66
4.6	Ordinogramme de l'algorithme de la multiplication	68
4.7	Registre de Multiplication 16x16	70
4.8	Ordinogramme de la division binaire 16 bits	72
4.9	Registre de la division 16 par 16	73
7.1	Élément d'un tableau	95
7.2	Tableau Simple	96
7.3	Recherche d'un élément dans un tableau simple	97
7.4	Ajout d'un élément dans un tableau simple	99
7.5	Suppression d'un élément dans un tableau simple	101
8.1	Registres du 65816 mode Native	106
8.2	Accumulateur	110
8.3	Stack Pointer vide	111
8.4	Pointeur de pile après chargement	111
8.5	Registres du 65816 mode Emulation	113
10.1	Mode d'adressage de base	118
10.2	Adressage - Pré-Indexation	123
10.3	Adressage indexé indirect Post-Indexation	123
10.4	Adressage indirect.	124
10.5	Registres pour adresses longues.	126
14.1	Diagramme des blocs des fonctions de la Super Nintendo	250



COPYRIGHT © 2008 - 2009