VT02
Console and One Bus System
(Real 4 colors or Virtual 16 colors)

Table of contents

Revision History:

Ver.	Contents
A1	Original Version
A2	1.Page 3 Feature ---Add 8 bits data bus mode has auxiliary 16 I/O pins and 16 bits data bus mode has auxiliary 8 I/O pins. 2.Page 6 Pin optional table ---Add Note: In one bus mode, when you want to use the extension I/O port. Please you refer page 17 Address ports of program unit to set up them. 3.Page 16 Block Diagram of Video address Multiplexer ---Modified the error BKEXTEN to BKEXTEN=1 4.Page 19,20 Functional description of Program Address ports --- #410B D3 add 0: $6000-$7FFF writing function will not active the XRWB. 1: $6000-$7FFF writing function will active XRWB . Note: When FEWN(D3) was High then the old program method will not active. ---Modified #410E:D7-D0 ⟶ Output to XVRW,XVOE,XRCB … To #410F :D7-D0 ⟶ Output to XVRW,XVOE,XRCB … ---Modified %410E: Input XVRW,XVOE,XRCB … To %410F: Input XVRW,XVOE,XRCB … ---Modified %4109 to %4119 --- Modified TIFLAG ⟶ D5 to RINGF ⟶ D5 5. Page 23 Function Description of Graphic Address ports Modified error of #2000 D5: SPRITE SIZE CONTROL From 1 → BIG SPRITE (8X16)or(16X16) To 1 ⟶ BIG SPRITE (8X16) From 0 → SMALL SPRITE (8X8)or(16X8) To 0 ⟶ SMALL SPRITE (8X8) 6. Page 24 Modified the error From #2014 : Video Bank0 register2 D7-D0 → RV07-RV20. To #2014 : Video Bank0 register2 D7-D0 → RV27-RV20. From #2015 : Video Bank0 register3 D7-D0 → RV07-RV30. To #2015 : Video Bank0 register3 D7-D0 → RV37-RV30. 7.Page 28 SPRITE RAM ---Modified From Sprite color and 8X8 or16X8 option To Sprite color and 8X8 option 8.Page 37 Programming Guide ---Add: When you connect an additional chip and you have to use the XRWB function to control them then you have to set up #410B function. The thing you have to know that when the FWEN was high then the old program method will not active. ---Add: Don't use the DMA copy to color palette in NTSC system. PAL system haven't this prohibition. 9. Add Page39,40:Program/Video Memory Bank Mapping Programming Guide:
A3	Add: item10, 11 to describe the PCM and RS232
A4	Revise all the format.
A5	Revise the CPU Instruction.

VT02
Console and One Bus System
(Real 4 colors or Virtual 16 colors)

Features

System

- CPU: 6502
- Internal Program RAM: 2K Bytes
- Internal Video RAM: 2K Bytes
- DMA (Sprite and Background)
- One Bus Mode, 8 bits data bus or 16 bits data bus
- Multiple control of IRQ
- Programmable timer
- Bank decoder for expandable memory up to 32M Bytes
- T.V. signal output (NTSC, PAL)
- 8 bits data bus mode has auxiliary 16 I/O pins and 16 bits data bus mode has auxiliary 8 I/O pins.

Peripheral Applications

- Joystick
- RS232 serial port built-in.

Graphic Processor

- Resolution: 256x240 pixels
- 64 sprites in one frame
- Background color can be 4 colors (4 color sets).
- Sprites with 4 colors (4 color sets), have 8X8 or 8X16 character size.
- Color palette has 25 colors.

Sound Generator

- 4 Rhythm channels,
- 2 Low frequency channels,
- 2 Noise channels ,
- PCM or DWS DMA built-in.

General Description

VT02 includes the CPU, Graphic Unit, Sound Unit, two internal 2KBytes SRAMs, and some I/O controller. There are two main systems in VT02, program system and video system.

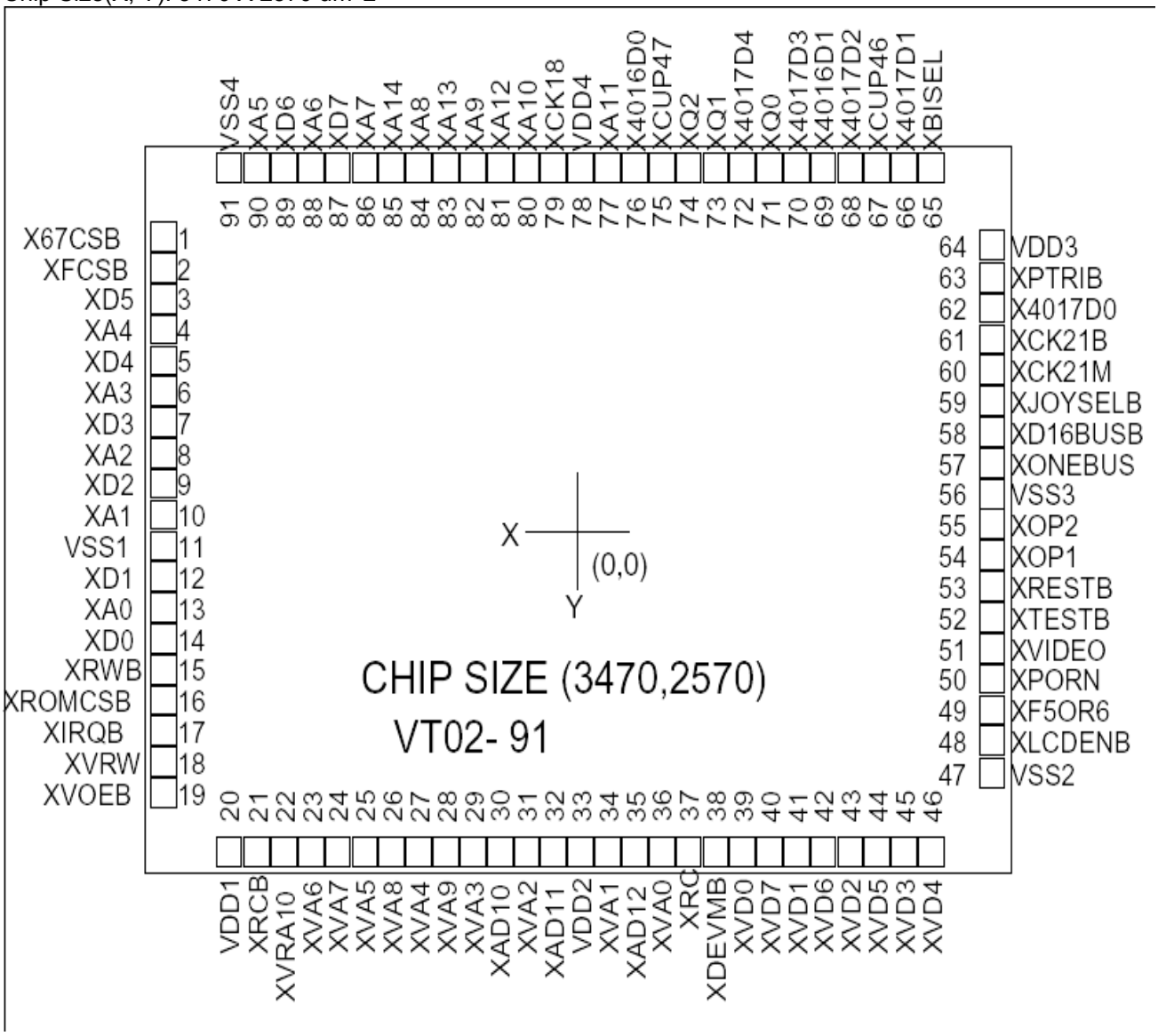
CPU plays the key role in program system. It can access the internal and external program memories. The program memory stores the program command, instructions, and sound data. VT02 is equipped with a 2KByte SRAM as internal program memory. This program RAM will be the zero page RAM, STACK and some memory of CPU. Program system controls the operations of Education machine, including figure, voice, and the title. It means CPU will control the video system to display the specified figure.

Graphic Unit is the main role of the video system. It can access the video memory automatically to display some figures. In addition to the internal program SRAM, VT02 is equipped the other 2KByte SRAM for Video RAM. Internal Video RAM stores pattern vectors for 2 pages of background. External Video memory stores the video characters to be pointed by the pattern vectors.

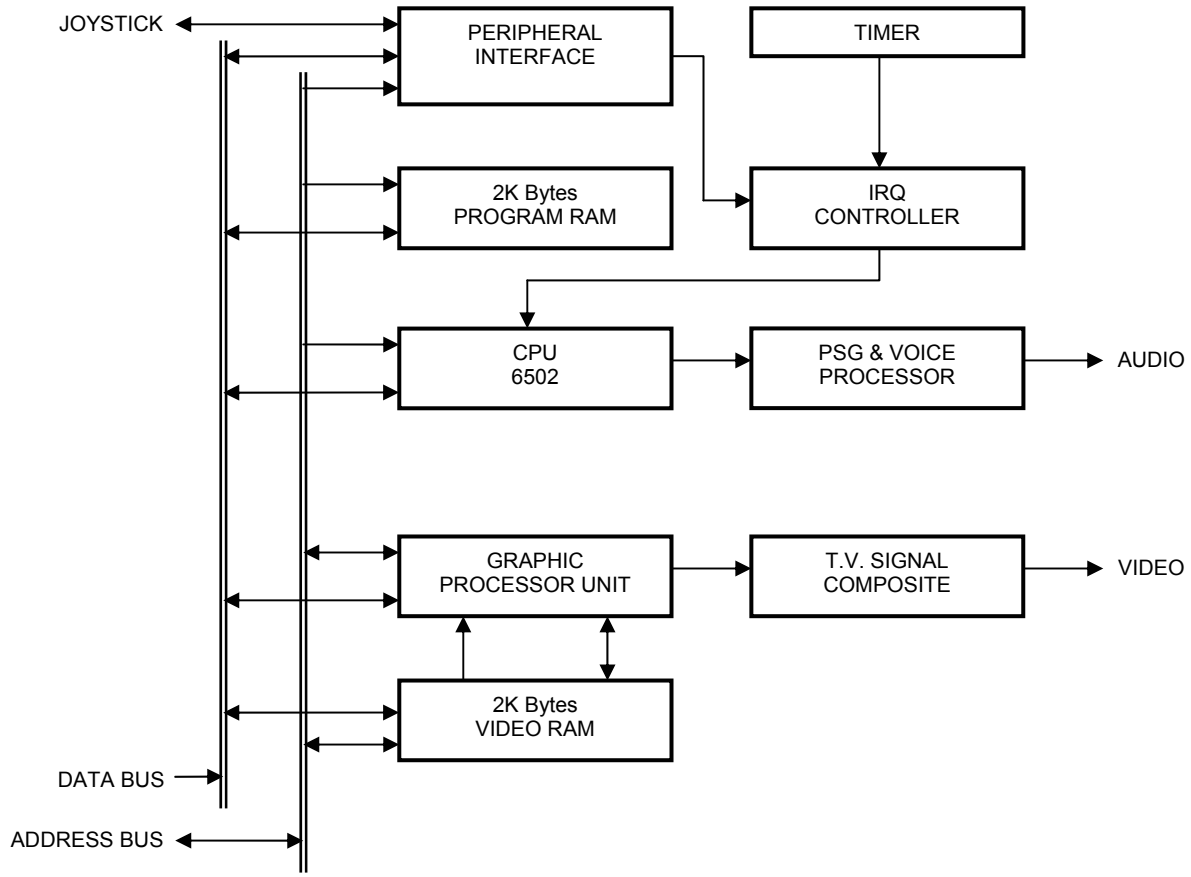
VT02 can combine program and video bus into one bus mode. Thus it needs only one memory IC as the program memory and video memory. Under one bus mode, programmer specifies the program and video bank individually in the same external memory and then VT02 will combine the two independent buses into one bus. External memory can be extended to 32Mbytes through the function decoder of VT02.

Pin configuration

Chip Size(X, Y): 3470 X 2570 um^2



Block diagram



Pin Description

SYMBOL	TYPE	DESCRIPTIONS
XA[14:0]	O	CPU address bus or one bus mode Address OA14-OA0.
XD[7:0]	I/O	CPU data bus or one bus mode data bus Bit7-0.
XCK18	O	CPU clock 1.8MHz.
XRW	O	CPU or One bus mode read/write signal.
XROMCS	O	ROM chip select signal or one bus mode ROM OEB.
XFCSB	O	ROM or flash chip selector in one bus mode, Low active, address $8000~$FFFF will be low.
X67CSB	O	Address $6000~7FFF chip selector, Low active.
XDEVMB	I	Swap the function of XFCSB and X67CSB when low.(PH)
XLCDENB	I	The enable of LCD signal output for testing, Low active. (PH)
XPTRIB	I	The control signal to force the bus tristate,Low active. (PH)
XBISEL	I	The selector pin of built-in function, Low: Built-in function will force the X67CSB and XFCSB to high.(PH)
XIRQB	I	CPU interrupt input signal. (PH)
XVRW	O	Video Read/Write signal or I/O in one bus mode.
XVOEB	O	Video data output enable or I/O in one bus mode.
XVRA10	I	Internal Video RAM Address bit 10 or I/O in one bus mode.
XRC	O	External ROM chip selector Low active or I/O in one bus mode.
XRCB	O	External ROM chip selector High active or Power on indicator or I/O in one bus mode.
XVA[9:0]	O	Video address bus or OA24-OA15 in one bus mode.
XAD[12:10]	O I/O	Video address bus A12-A10 or I/O in one bus mode.
XVD[7:0]	I/O	Video data bus or I/O in one bus 8 bits mode or data bus Bit15-8 in one bus 16 bits.
XTESTB	I	Wafer test pin. (PH)
XRESTB	I	System reset pin low active. (PH)
XCK21M	I	Clock input pin for crystal.
XCK21B	O	Clock output pin for crystal.
X4016 [1:0]	I	I/O interface input pins. (PH)
X4017 [4:0]	I	I/O interface input pins.(PH)
XQ[2:0]	O I/O	I/O interface output pins or Video extension address. XQ1,XQ0(O), XQ2(I/O)
XCUP46,XCUP47	I/O	Clock of I/O or XCUP47 can be video extension address. XCUP46(PH)
XVIDEO	O	Composite video signal.
XOP1,XOP2	O	Audio signal.
XJOYSELB	I	Internal Joystick enable when XJOYSEL=0. (PH)
XONEBUS	I	One bus mode selector High active. (PH)
XD16BUSB	I	16 bits data bus selector (low active) in one bus mode, A0 will decide the low byte (XD7-0) or high byte (XVD7-0) data. (PH)
XPORN,XF5OR6	O	TV system selector. All 0:NTSC, All 1:PAL.(PH)

Note: (I) input pin. (O) output pin. (I/O) input/output pin. (PH) pull high resistor 20K~50K inside, (PL) pull low resistor 20K~50K inside.

Pin optional table

Status	Register IOP0EN=1 XONEBUS=1 XD16BUSB=1	Register IOP0EN=0 XONEBUS=1 XD16BUSB=0	Register IOP0EN=0 XONEBUS=0 XD16BUSB=X
XVD0	IOP00	D8 of one bus	XVD0
XVD1	IOP01	D9 of one bus	XVD1
XVD2	IOP02	D10 of one bus	XVD2
XVD3	IOP03	D11 of one bus	XVD3

Status	Register IOP1EN=1 XONEBUS=1 XD16BUSB=1	Register IOP1EN=0 XONEBUS=1 XD16BUSB=0	Register IOP1EN=0 XONEBUS=0 XD16BUSB=X
XVD4	IOP10	D12 of one bus	XVD4
XVD5	IOP11	D13 of one bus	XVD5
XVD6	IOP12	D14 of one bus	XVD6
XVD7	IOP13	D15 of one bus	XVD7

Status	Register IOP2EN=1 XONEBUS=1	Register IOP2EN=0 XONEBUS=X
XVRA10	IOP20	XVRA10
XAD10	IOP21	XAD10
XAD11	IOP22	XAD11
XAD12	IOP23	XAD12

Status	Register IOP3EN=1	Register IOP3EN=0	
		XJOYSELB=1	XJOYSELB=0
XRC	IOP30	XRC	XRC
XRCB	IOP31	XRCB	POWON
XVOEB	IOP32	XVOEB	XVOEB
XVRW	IOP33	XVRW	XVRW

Status	XONEBUS=1	XONEBUS=0
XA[14:0]	One bus OA[14:0]	XA[14:0]
XVA[9:0]	One bus OA[24:15]	XVA[9:0]
XROMCSB	One bus ROM OEB	XROMCSB
XRWB	One bus MEMORY RWB	XRWB

Note: In one bus mode, when you want to use the extension I/O port. Please you refer page 26,27 Address ports of program unit to set up them.

Status	XJOYSELB=0	XJOYSELB=1
X4016D0	JOYAM	X4016D0
X4016D1	JOYBM	X4016D1
X4017D0	JOYUPA	X4017D0
X4017D1, GUNPORT1	JOYST	X4017D1
X4017D2, GUNPORT2	JOYSE	X4017D2
X4017D3	JOYDNA	X4017D3
X4017D4	JOYLFA	X4017D4
XCUP46	JOYRTA	XCUP46

Status	XJOYSELB=0 && XONEBUS=0	XJOYSELB=1 || XONEBUS=1
XQ0	VIDEO ROM A10	XQ0
XQ1	VIDEO ROM A11	XQ1

Status	Register RS232EN=1 XONEBUS=X XJOYSELB=X	Register RS232EN=0 XONEBUS=0 && XJOYSELB=0	Register RS232EN=0 XONEBUS=1 || XJOYSELB=1
XQ2	RD	VIDEO ROM A12	XQ2
XCUP47	TD	VIDEO ROM A13	XCUP47

Functional description

Console chip is composed of CPU, video, sound function and I/O.

Video:

1. Video can handle two objects, SPRITE and BACKGROUND. SPRITE is the moving object as bullet, car, and man. BACKGROUND is the larger figure as tree, forest, house, scenery which can be scrolled.
2. On A TV screen, VIDEO can display 256 pixels on a horizontal coordinate and 240 pixels on a vertical coordinate.
3. Programmer can specify 64 SPRITE to display on a screen. One SPRITE needs four bytes to define.
4. The maximum SPRITE number on a horizontal scanning line is 8. If it is over 8, the rest will be careless and the message will be responded to CPU.
5. A basic SPRITE or BACKGROUND pattern is a character with 8X8 pixels, one pixel which show 4 kinds of color.
6. Programmer can choose SPRITE being (8X16), (8X8).
7. Two pages of figure for BACKGROUND can be immediately changed page or scrolled with horizontal or vertical way.
8. 25 colors in color plate can be defined. One color needs 6 bits to define.
9. Automatic TV Synchronized signal generation which is independent with program.
10. TV composite signal output.

Sound:

1. Providing maximally 256 bytes DMA function for graphic unit updated sprite, background vector and character data.
2. 2 ports for reading the status of sound generator.
3. Every sound channel gets 4 address ports to control its operation.
4. There are 4 Rhythm channels, 2 low frequency channels, 2 noise channels and PCM or DWS DMA built in.
5. Two independent sound DA output pin.

CPU:

CPU included in Console gets 16 bits program counter, 8 bits AL and Accumulator, status register, two general purposes registers X, Y, 8 bits stack pointer, 16 bits address bus and 8 bits data bus.

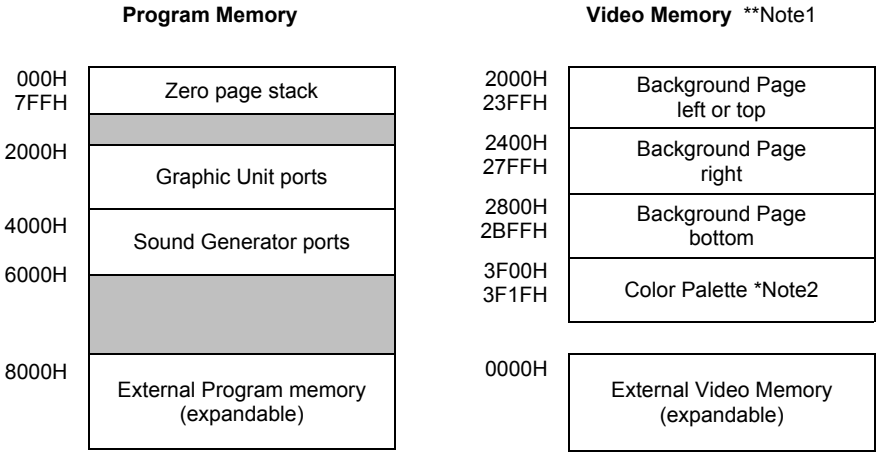
Internal RAM:

One 2K bytes RAM for VIDEO Memory, another for Program RAM.

I/O:

1. 7 pins for reading peripheral I/O, 3 pins for outputting peripheral I/O, 2 clock pins.
2. Built-in optionally 8 bit serial to parallel I/O for joystick.
3. In one bus mode, 8 bits data bus mode has auxiliary 16 I/O pins and 16 bits data bus mode has auxiliary 8 I/O pins.
4. Built-in optionally RS232 serial port.

Address Map of Program Memory and Video Memory



**Note1
Address of Video Memory should be asserted through 2006H of Graphic Unit ports. The details methods to access video memory are described in section: Access Video Memory and the Bank Mapping.

*Note2
When XRC = 1
3F00-3F1F is the old color mapping location of color palette, total 25 colors.
3F00 is transparent color, and 3F10, 3F04, 3F14, 3F08, 3F18, 3F0C, 3F1C can be ignored.

One Bus System

VT02 automatically combine the program address bus and video address bus into one bus. Under one bus mode a single external memory can be used as program memory and video memory. Although there is only one external memory physically, Video memory bank and program memory bank are set individually. Programmer must divide the signal external memory carefully to store program and video data. OA[24:0] are the output pins to address this external memory up to 32M Bytes. Based on the program address, XA, video address, AD, and other relative registers, VT02 will set the values of OA[24:0] to address the external memory. Please refer to the following two sections for detail mappings.

In one bus mode, the program initial address A24-A0 is 007FFFC, and the video initial address is 0000XXX.

Access Video Memory

Address of Video Memory should be asserted through 2006H of Graphic Unit ports. 2006H is a two-bytes-set-up port. D5 of the first byte output to XRC. The remain bits of 2006H set AD[12:0] as described in Table A1. When XRC=1, AD[12:0] are the address of the internal video memory. When XRC=0 under one bus mode, AD[12:0] together with the settings of Video Memory Bank decide the output pins OA[24:0] as the address of external video memory.

D7	D6	D5	D4	D3	D2	D1	D0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Second byte							

D7	D6	D5	D4	D3	D2	D1	D0
		XRC	AD12	AD11	AD10	AD9	AD8
First byte							

Table A1. Writing 2006H (two bytes set up)

Video memory Bank Mapping under One Bus Mode

Under one bus mode, VT02 can address up to 32M bytes external memory through 25 bits of address, OA[24:0]. A sketch map of Video Bank is described in Figure A1. To address such a big size of external memory, VT02 separate the 32M bytes into several blocks via Video Bank 2. Each block is then divided into several small blocks through Video Bank 1. By the same manner, Bank 0 divides each of those small blocks into smaller blocks.

Please compare to Table A3 for detail mappings. Please Note that, there is no Video Bank 1 in the extension mode, i.e. each block divided from Video Bank 2 are directly banking by Video Bank 0.

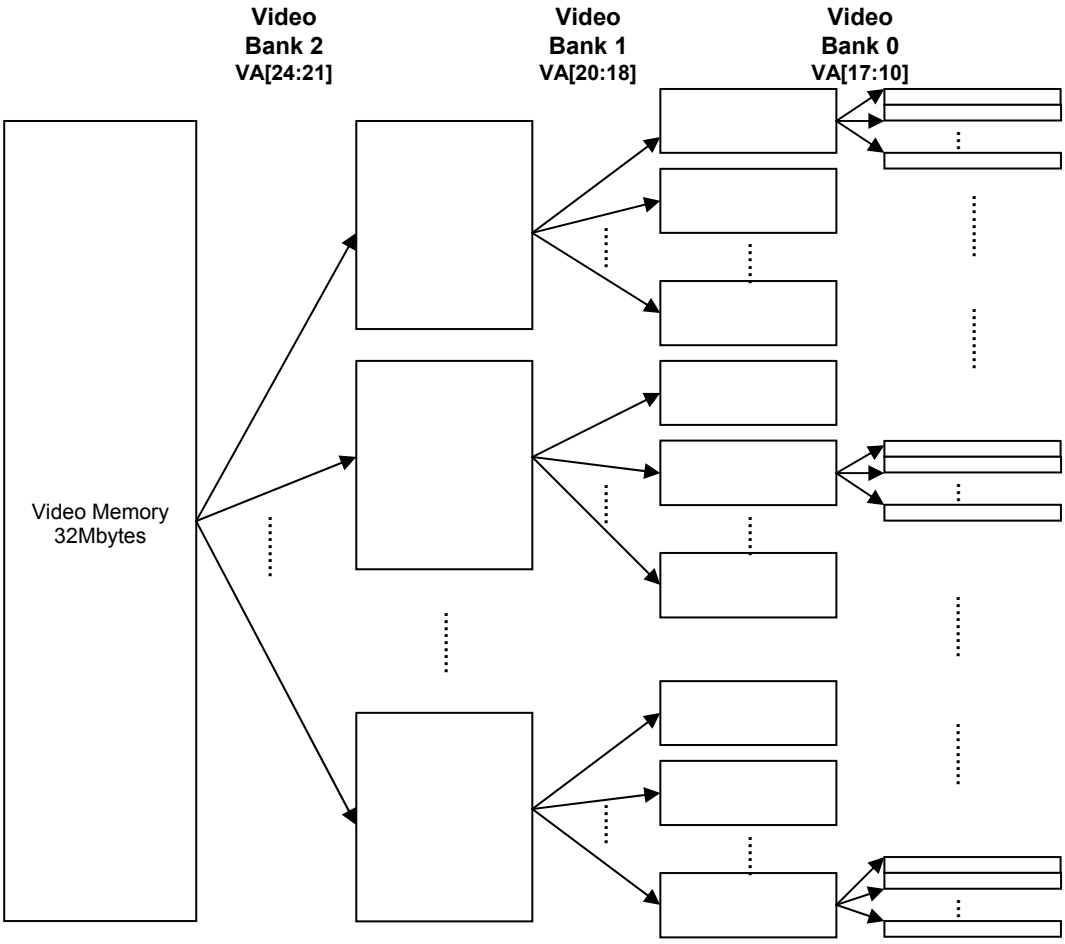
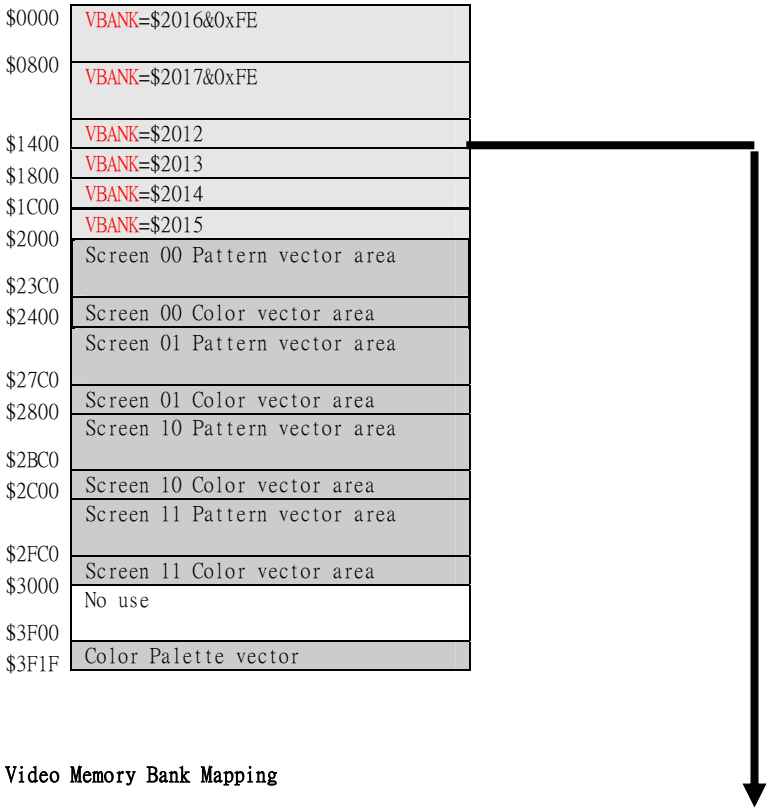


Figure A1. A sketch map of Video Memory Banking

PPU



Video Memory Bank Mapping

Video address normal mode：
Case ($201A & 0x07)
0： ($4100&0x0F)<<21+($2018&0x70)<<14+ VBANK<<10 (Default)
1： ($4100&0x0F)<<21+($2018&0x70)<<14+(($201A&0x80)|(VBANK&0x7F))<<10
2： ($4100&0x0F)<<21+($2018&0x70)<<14+(($201A&0xC0)|(VBANK&0x3F))<<10
4： ($4100&0x0F)<<21+($2018&0x70)<<14+(($201A&0xE0)|(VBANK&0x1F))<<10
5： ($4100&0x0F)<<21+($2018&0x70)<<14+(($201A&0xF0)|(VBANK&0x0F))<<10
6： ($4100&0x0F)<<21+($2018&0x70)<<14+(($201A&0xF8)|(VBANK&0x07))<<10
Video address extension mode：
Case ($201A & 0x07)
0： ($4100&0x0F)<<21+VBANK<<13+EVA<<10 (Default)
1： ($4100&0x0F)<<21+(($201A&0x80)|(VBANK&0x7F))<<13+EVA<<10
2： ($4100&0x0F)<<21+(($201A&0xC0)|(VBANK&0x3F))<<13+EVA<<10
4： ($4100&0x0F)<<21+(($201A&0xE0)|(VBANK&0x1F))<<13+EVA<<10
5： ($4100&0x0F)<<21+(($201A&0xF0)|(VBANK&0x0F))<<13+EVA<<10
6： ($4100&0x0F)<<21+(($201A&0xF8)|(VBANK&0x07))<<13+EVA<<10

- When $4105&0x80 isn't 0 , $0000-$0FFF and $1000-$1FFF exchang.
- EVA Table

	EVA2	EVA1	EVA0
Background display extension address mode，$2011&0x02=1	HV	BG4	BG3
Background display extension address mode，$2011&0x02=0	BKPAGE	BG4	BG3
Sprite display extension address mode	SPEVA2	SPEVA1	SPEVA0
R/W extension address mode	VRWB2	VRWB1	VRWB0

Video Memory Bank Mapping

Minimum Video bank 1K bytes
VA24-21 <- $4100(D3-0)
VA17-10 <- $2012-$2017(D7-0), $201A(D7-0)
EVA12-10 <- $2018(D2-0)
VA20-10 <- $2018(D6-4)

Video Address	4 colors	Extension	Video BANK0	Video BANK0
0000-000F	Character 0	EVA12-10=0 If Extension Mode active	VA17-10=0 If Extension Mode not active	VA17-10=0 If Extension Mode active
0010-001F	Character 1			
0020-003F	Character 2,3			
….	Character ..			
03E0-03FF	Character 63			
0400-07FF	64 Characters	EVA12-10=1	VA17-10=1	
….	.. Character	….		
1C00-1FFF	64 Characters	EVA12-10=7	VA17-10=7	
2000-3FFF	512 Characters		VA17-10=8-F	VA17-10=1
4000-5FFF	512 Characters		VA17-10=10-17	VA17-10=2
….	.. Character	….		
3E000-3FFFF	512 Characters		VA17-10=F8-FF	VA17-10=1F
40000-7FFFF	16K Character			VA17-10=20-3F
80000-BFFFF	16K Character			VA17-10=40-5F
C0000-FFFFF	16K Character			VA17-10=60-7F
100000-13FFFFF	16K Character			VA17-10=80-9F
140000-17FFFFF	16K Character			VA17-10=A0-BF
180000-1BFFFFF	16K Character			VA17-10=C0-DF
1C0000-1FFFFFF	16K Character			VA17-10=E0-FF

Video Address	Bank1 no extension	Bank2
00000-3FFFF	VA20-18=0	VA24-21=0
40000-7FFFF	VA20-18=1	
….		
1C0000-1FFFFF	VA20-18=7	
200000-3FFFFF	VA20-18=0-7	VA24-21=1
400000-5FFFFF	VA20-18=0-7	VA24-21=2
600000-7FFFFF	VA20-18=0-7	VA24-21=3
….		
1E00000-1FFFFFF	VA20-18=0-7	VA24-21=F

Address the Video memory under One Bus Mode

VT02 provide different function decoder to address video memory under different settings of background and sprite. There are two types of settings, as describe in Table A2. Programmer can set the background and sprite into different types. The graphic unit automatically changes to the relative mode of function decoder to access background characters or sprite characters.
When accessing video memory under one bus mode, address pins OA[24:0] can be asserted as Table A3, where VA[24:0], EVA[12:10] can be specified by different registers. VA[9:0] are assigned by AD[9:0] which are specified through 2006H. Under different mode of Video Bank 0 Selector, VA[17:10] are specified as described in Table A4. VA[20:18] are specified through 2018H(D[6:4)) for Video Bank 1. VA[24:21] are specified through 4100H (D[3:0]) for Video Bank 2. EVA[12:10] are specified as described in Table A5.

Type of background or sprite char.	
Type1	Extension video address disable and 4 colors per pixel.
Type2	Extension video address enable and 4 colors per pixel.

Table A2. Different types of background or sprite characters.

VA[24:21] : Video Bank 2
VA[20:18] : Video Bank 1
VA[17:10] : Video Bank 0

Address output	Type of background or sprite char.	
	Type1	**Type2**
OA24	VA24	VA24
OA23	VA23	VA23
OA22	VA22	VA22
OA21	VA21	VA21
OA20	VA20	VA17
OA19	VA19	VA16
OA18	VA18	VA15
OA17	VA17	VA14
OA16	VA16	VA13
OA15	VA15	VA12
OA14	VA14	VA11
OA13	VA13	VA10
OA12	VA12	EVA12
OA11	VA11	EVA11
OA10	VA10	EVA10
OA9	VA9	VA9
OA8	VA8	VA8
OA7	VA7	VA7
OA6	VA6	VA6
OA5	VA5	VA5
OA4	VA4	VA4
OA3	VA3	VA3
OA2	VA2	VA2
OA1	VA1	VA1
OA0	VA0	VA0

Table A3. Specify OA[24:0] under different types of background or sprite characters.

VB0S[2:0] (201AH)	VA[17:10]							
	VA17	VA16	VA15	VA14	VA13	VA12	VA11	VA10
000	TVA17	TVA16	TVA15	TVA14	TVA13	TVA12	TVA11	TVA10
001	RV67	TVA16	TVA15	TVA14	TVA13	TVA12	TVA11	TVA10
010	RV67	RV66	TVA15	TVA14	TVA13	TVA12	TVA11	TVA10
100	RV67	RV66	RV65	TVA14	TVA13	TVA12	TVA11	TVA10
101	RV67	RV66	RV65	RV64	TVA13	TVA12	TVA11	TVA10
110	RV67	RV66	RV65	RV64	RV63	TVA12	TVA11	TVA10

Table A4. Specify VA[17:10] under different mode of Video Bank 0 Selector (VB0S).
NOTE: TVA[17:10] are specified as described in Table vvv05. RV[67:63] are specified through 201AH (D[7:3]).

	EVA12	EVA11	EVA10
BKEXTEN=1 & EVAS12=1 & Background Display Area	HV (4106H)	BG4	BG3
BKEXTEN=1 & EVAS12=0 & Background Display Area	BKPAGE (2018H)	BG4	BG3
SPEXTEN=1 & Horizontal Synchronized Read Charater Area	SPEVA2	SPEVA1	SPEVA0
CPU RW MODE in Vertical Synchronized Area or not Display	VRWB2	VRWB1	VRWB0

Table A5. EVA[12:10]

COMR7 (4105H,D7)	AD[12:10] (2006H)	TVA17	TVA16	TVA15	TVA14	TVA13	TVA12	TVA11	TVA10
0H or 1H or CH or DH		RV47	RV46	RV45	RV44	RV43	RV42	RV41	AD10
2H or 3H or EH or FH		RV57	RV56	RV55	RV54	RV53	RV52	RV51	AD10
4H or 8H		RV07	RV06	RV05	RV04	RV03	RV02	RV01	RV00
5H or 9H		RV17	RV16	RV15	RV14	RV13	RV12	RV11	RV10
6H or AH		RV27	RV26	RV25	RV24	RV23	RV22	RV21	RV20
7H or BH		RV37	RV36	RV35	RV34	RV33	RV32	RV31	RV30

Table A6. TVA[17:10].
NOTE: RV[17:10], RV[27:20], RV[37:30], RV[47:40], RV[57:50] are specified through 2012H~2017H.

Program Memory Bank Mapping under One Bus Mode

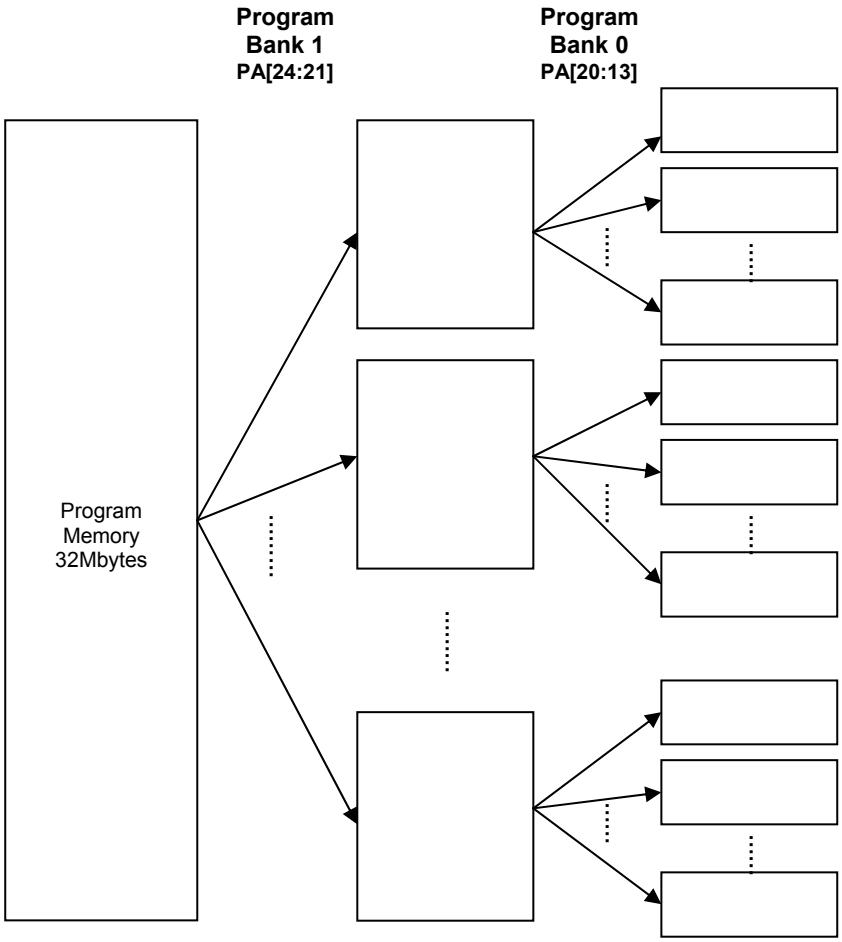
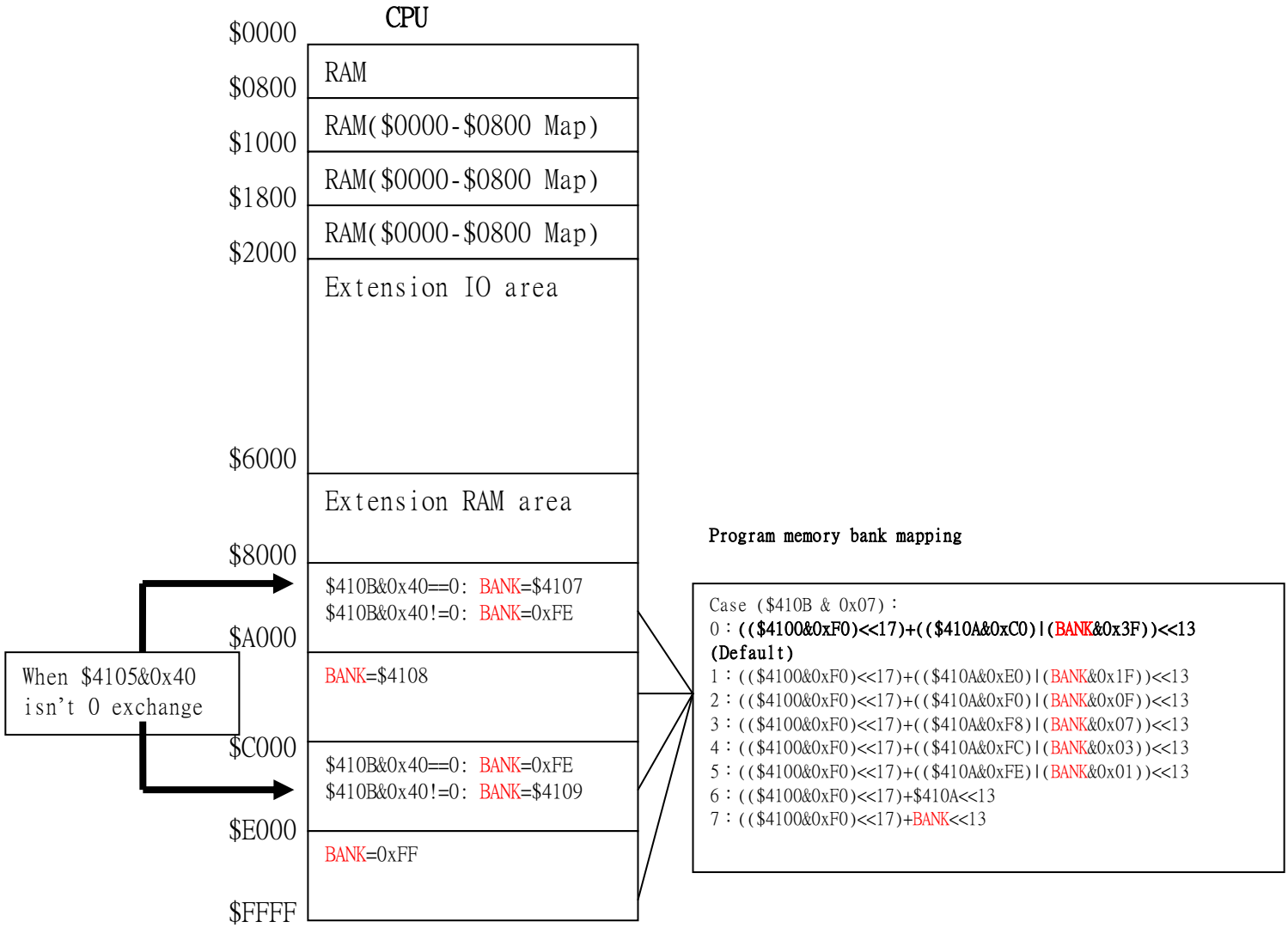


Figure B1. A sketch map of Program Memory Banking


CPU
$0000
$0800
$1000
$1800
$2000
$6000
$8000
$A000
$C000
$E000
$FFFF
RAM
RAM($0000-$0800 Map)
RAM($0000-$0800 Map)
RAM($0000-$0800 Map)
Extension IO area
Extension RAM area
$410B&0x40==0: BANK=$4107
$410B&0x40!=0: BANK=0xFE
BANK=$4108
$410B&0x40==0: BANK=0xFE
$410B&0x40!=0: BANK=$4109
BANK=0xFF
When $4105&0x40 isn't 0 exchange
Program memory bank mapping
Case ($410B & 0x07):
0:((($4100&0xF0)<<17)+(($410A&0xC0)|(BANK&0x3F))<<13
(Default)
1:((($4100&0xF0)<<17)+(($410A&0xE0)|(BANK&0x1F))<<13
2:((($4100&0xF0)<<17)+(($410A&0xF0)|(BANK&0x0F))<<13
3:((($4100&0xF0)<<17)+(($410A&0xF8)|(BANK&0x07))<<13
4:((($4100&0xF0)<<17)+(($410A&0xFC)|(BANK&0x03))<<13
5:((($4100&0xF0)<<17)+(($410A&0xFE)|(BANK&0x01))<<13
6:((($4100&0xF0)<<17)+$410A<<13
7:((($4100&0xF0)<<17)+BANK<<13

Program Memory Bank Mapping

Minimum program bank 8K bytes
PS2-0 <- $410B(D2-0)
PQ07-0 <- $4107(D7-0)
PQ17-0 <- $4108(D7-0)
PQ27-0 <- $4109(D7-0)
PQ37-0 <- $410A(D7-0)
PA24-21 <- $4100(D7-4)

<table>
<tr><td rowspan="2">Program Address</td><td colspan="3">Program Bank0 allocate 256 banks, 2M bytes</td><td rowspan="2">Program Bank1
16 Bank0 bank</td></tr>
<tr><td>PS2-0=0</td><td>PS2-0=6</td><td>PS2-0=7</td></tr>
<tr><td>0000-1FFF</td><td rowspan="6">PQ37-6=0
PQ25-0, PQ15-0,
PQ05-0 select
64 banks</td><td rowspan="15">PQ37-0 select
256 banks</td><td rowspan="15">PQ27-0, PQ17-0,
PQ07-0 select
256 banks</td><td rowspan="15">PA24-21=0</td></tr>
<tr><td>2000-3FFF</td></tr>
<tr><td>4000-5FFF</td></tr>
<tr><td>6000-7FFF</td></tr>
<tr><td>….</td></tr>
<tr><td>7E000-7FFFF</td></tr>
<tr><td>80000-81FFF</td><td rowspan="3">PQ37-6=1</td></tr>
<tr><td>….</td></tr>
<tr><td>FE000-FFFFF</td></tr>
<tr><td>100000-101FFF</td><td rowspan="3">PQ37-6=2</td></tr>
<tr><td>….</td></tr>
<tr><td>17E000-17FFFF</td></tr>
<tr><td>180000-181FFF</td><td rowspan="3">PQ37-6=3</td></tr>
<tr><td>….</td></tr>
<tr><td>1FE000-1FFFFF</td></tr>
<tr><td>200000-201FFF</td><td colspan="3" rowspan="3">2M Bytes</td><td rowspan="3">PA24-21=1</td></tr>
<tr><td>….</td></tr>
<tr><td>3FE000-3FFFFF</td></tr>
<tr><td>400000-401FFF</td><td colspan="3" rowspan="3">2M Bytes</td><td rowspan="3">PA24-21=2</td></tr>
<tr><td>….</td></tr>
<tr><td>5FE000-5FFFFF</td></tr>
<tr><td>600000-601FFF</td><td colspan="3" rowspan="3">2M Bytes</td><td rowspan="3">PA24-21=3</td></tr>
<tr><td>….</td></tr>
<tr><td>7FE000-7FFFFF</td></tr>
<tr><td>800000-801FFF</td><td colspan="3" rowspan="3">8M Bytes</td><td rowspan="3">PA24-21=4-7</td></tr>
<tr><td>….</td></tr>
<tr><td>FFE000-FFFFFF</td></tr>
<tr><td>1000000-1001FFF</td><td colspan="3" rowspan="3">16M Bytes</td><td rowspan="3">PA24-21=8-F</td></tr>
<tr><td>….</td></tr>
<tr><td>1FFE000-1FFFFFF</td></tr>
</table>

Address the Program memory under One Bus Mode

6502 is the CPU of VT02. By different addressing modes of 6502, programmer can access the program memory. Under one bus mode, the function decoder works with several registers to help programmer to address the external program memory up to 32 Mbytes. When accessing program memory under one bus mode, address pins OA[24:0] can be asserted as Table B1, where PA[24:13] are specified by different registers and A[12:0] are the low 12 bits address when programmer perform LDA or STA to program CPU6502. PA[24:21] are specified through 4100H for Program Bank 1. Under different settings of PS[2:0], PA[20:13] are specified for Program Bank0 as described in Table B2, where TPA[20:13] are specified as Table B3 and PQ[07:00], PQ[17:10], PQ[27:20] and PQ[37:30] are specified through 4107H to 410AH.

Address output	Value
OA24	PA24
OA23	PA23
OA22	PA22
OA21	PA21
OA20	PA20
OA19	PA19
OA18	PA18
OA17	PA17
OA16	PA16
OA15	PA15
OA14	PA14
OA13	PA13
OA12	A12
OA11	A11
OA10	A10
OA9	A9
OA8	A8
OA7	A7
OA6	A6
OA5	A5
OA4	A4
OA3	A3
OA2	A2
OA1	A1
OA0	A0

Table B1. Specify OA[24:0] to address external program memory.

PS[2:0] (410BH)	**PA20**	**PA19**	**PA18**	**PA17**	**PA16**	**PA15**	**PA14**	**PA13**
000	PQ37	PQ36	TPA18	TPA17	TPA16	TPA15	TPA14	TPA13
001	PQ37	PQ36	PQ35	TPA17	TPA16	TPA15	TPA14	TPA13
010	PQ37	PQ36	PQ35	PQ34	TPA16	TPA15	TPA14	TPA13
011	PQ37	PQ36	PQ35	PQ34	PQ33	TPA15	TPA14	TPA13
100	PQ37	PQ36	PQ35	PQ34	PQ33	PQ32	TPA14	TPA13
101	PQ37	PQ36	PQ35	PQ34	PQ33	PQ32	PQ31	TPA13
110	PQ37	PQ36	PQ35	PQ34	PQ33	PQ32	PQ31	PQ30
111	TPA20	TPA19	TPA18	TPA17	TPA16	TPA15	TPA14	TPA13

Table B2. Specify PA[20:13] for Program Bank 1.

PQ2EN (410B)	COMR6 (4105H)	A[14:13] (CPU)	TPA20	TPA19	TPA18	TPA17	TPA16	TPA15	TPA14	TPA13
0	0H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	1H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	2H		1	1	1	1	1	1	1	0
	3H		1	1	1	1	1	1	1	1
	4H		1	1	1	1	1	1	1	0
	5H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	6H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	7H		1	1	1	1	1	1	1	1
1	0H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	1H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	2H		PQ27	PQ26	PQ25	PQ24	PQ23	PQ22	PQ21	PQ20
	3H		1	1	1	1	1	1	1	1
	4H		PQ27	PQ26	PQ25	PQ24	PQ23	PQ22	PQ21	PQ20
	5H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	6H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	7H		1	1	1	1	1	1	1	1

Table B3. Specify TPA[20:13]

Background patterns and Internal Video RAM

In this system, 256x240 pixels are defined for one page graphic which contains 32x30 background patterns when displaying background. Each background patterns is 8x8 pixels.
Background patterns are stored in the external video memory. The internal video RAM stores vectors whose data is the addresses to point the background patterns. Each byte in the video RAM address corresponds to one position in one page. One byte in the internal video RAM points one background pattern in the external video memory. Thus, it needs 32x30=960 bytes to completely point one page of background. A simply mapping is described in Figure B1.

It only needs the low 960 bytes of 1Kbytes, the remain high 64Bytes store the 3rd, 4th color address bits of the same page.

VT02 group four adjacent patterns to share the same 3rd, 4th color address. Please refer to Figure B2 for details about 3rd, 4th color address bits. The 1st, 2nd color address bits consist each background pattern and are stored in the external video memory. The color of each pixel is decided by five (4 color mode) bits color palette which point the 25x6 SRAM. The SRAM is stored the chrominance and luminance data which will be transferred into video signal and outputted through video output pin. Color address bit 1, 2, decided the internal color of a pattern. A pattern can have three different colors to describe, bit 1, 2 = (0, 0) is transparent pixel. Color address bit 3, 4 can change the colors of the whole pattern; four sets of colors could be chose. Color address bit 5 decide the colors of sprite or colors of background, bit 5 = 1 for sprite and bit 5 = 0 for background.

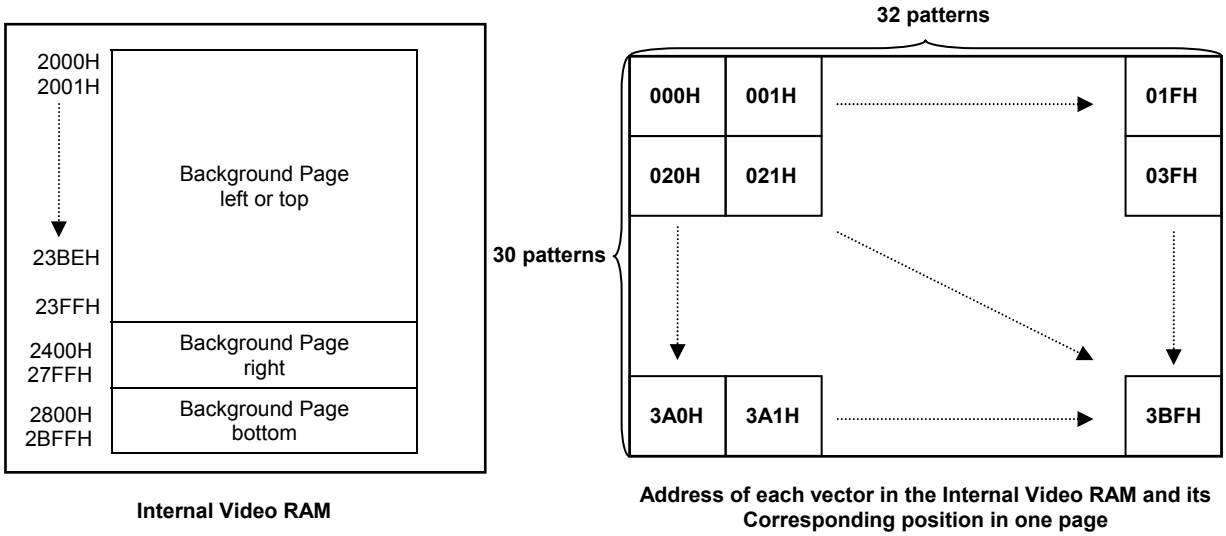


Figure B1. Mappings between Screen and Internal Video RAM

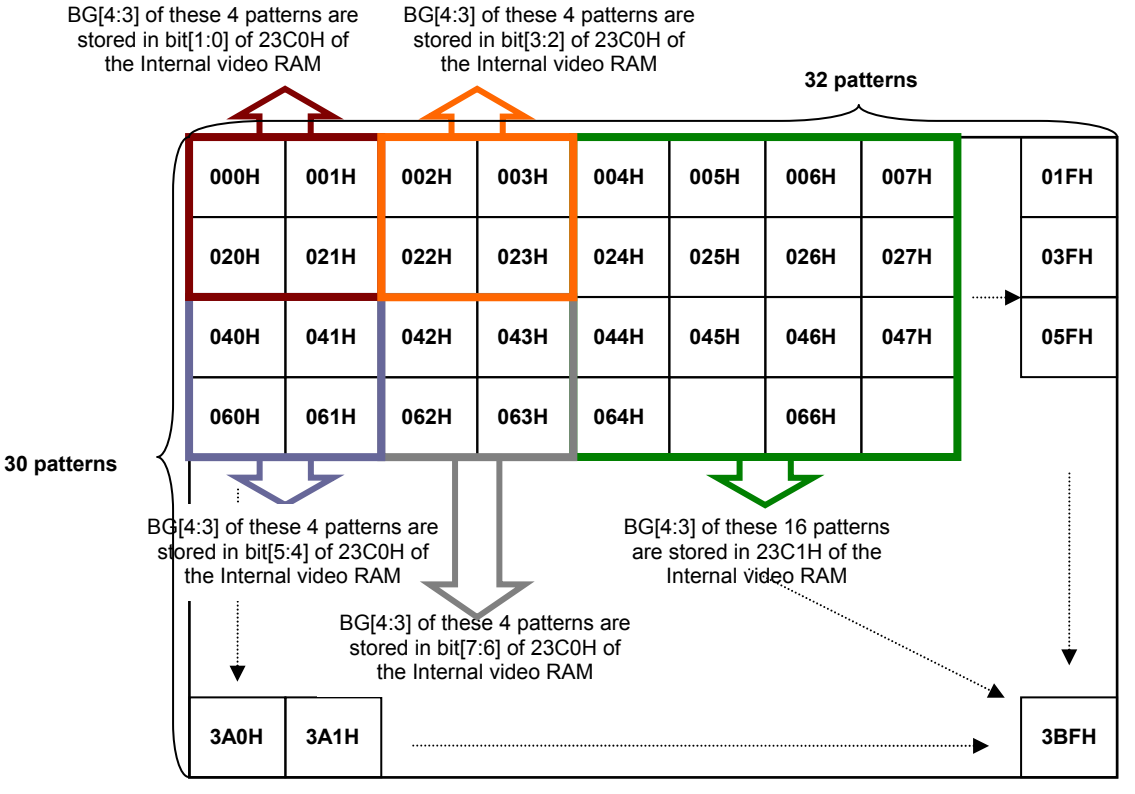


Figure B2. Four adjacent patterns to share the same 3rd, 4th color address

Two page for Background display

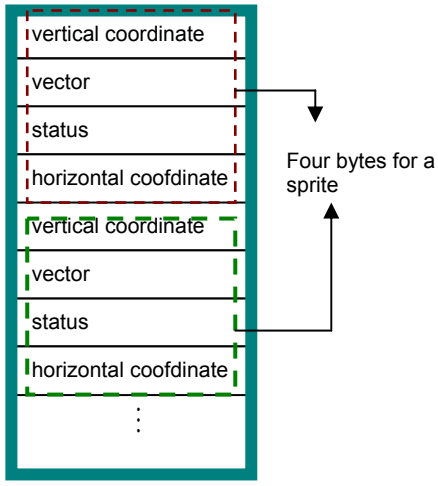
2Kbytes of internal video RAM are divided into 2 pages to moving screen effectively. Screen can be moved by horizontal or vertical way, that decided by every game card. In horizontal scroll, the AD10 of VIDEO and the A10 of the 2K RAM will be connected in game card. In vertical scroll, the AD11 of VIDEO and the A10 of the 2K RAM will be connected in game card.
In addition to the hardware connection, programmers also have to specify 4106H(D0) to decide horizontal scrolling or vertical scrolling. When horizontal, the left page is stored in 2000H to 23BEH and the righ page is stored in 2400H to 27FFH. When vertical, the top page is stored in 2000H to 23FFH and the bottom page is stored in 2800H to 2BFFH. Please also refer to Figure B1.

Sprite Pool

All of the sprites on screen stored in the sprite pool which has 256 bytes. Programmers can write data into the sprite pool through 2003H and 2004H or the DMA function of 4014H and 4034H. Programmers can specify 64 sprites on a screen, and no more 8 sprites on a row. It needs four bytes in sprite pool to describe each sprite. According to the order to store each sprite, they are the vertical coordinate, the 8-bit-vector, the status and the horizontal coordinate. The 8-bit-vector is used as the address to point the sprite patterns in the external video memory, just like the background vector stored in the internal video RAM. The function of the status byte is as follows:

D7:1: MIRROR AT X_AXIS, 0: NORMAL
D6:1: MIRROR AT Y_AXIS, 0: NORMAL
D5:1: BKGRND COVER SPRITE,
0: SPRITE COVER BKGRND
D4: Bit 2 of Sprite extension vector address, SPEVA2.
D3: Bit 1 of Sprite extension vector address, SPEVA1.
D2: Bit 0 of Sprite extension vector address, SPEVA0.
D1: Bit 4 of COLOR SET OF SPRITE (SP4).
D0: Bit 3 of COLOR SET OF SPRITE (SP3).

The function of SP[4:3] is just like the color address bits BG[4:3] of the background.



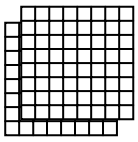
The Sprite Pool

Sprite Color and Size

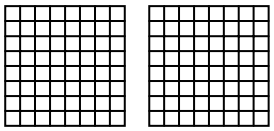
Programmers can choose the size and color mode of the sprite through 2000H, 2001H and 2010H. You have the following options:

Size 8x16 in 4 color mode
Size 8x8 in 4 color mode

In 4 color mode it needs two bits for a pixel. Take size 8x8 in 4 color mode as an example. One sprite pattern in the external video RAM is arranged as the following figure.

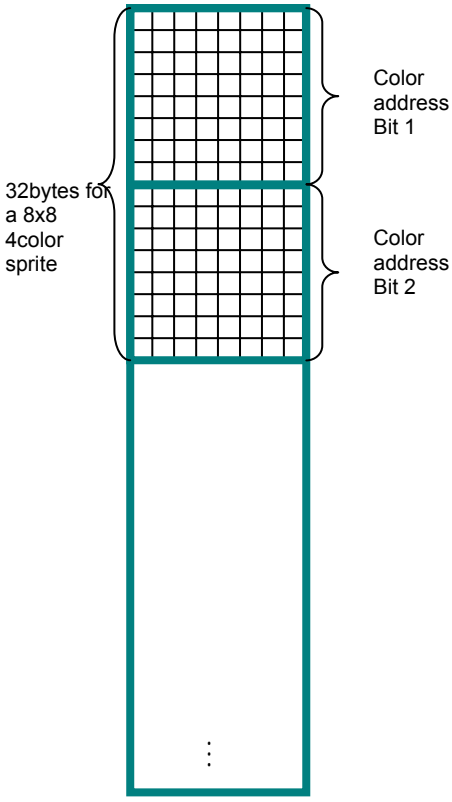


Sprite 8x8 in 4 color mode



Sprite 8x16 in 4 color mode

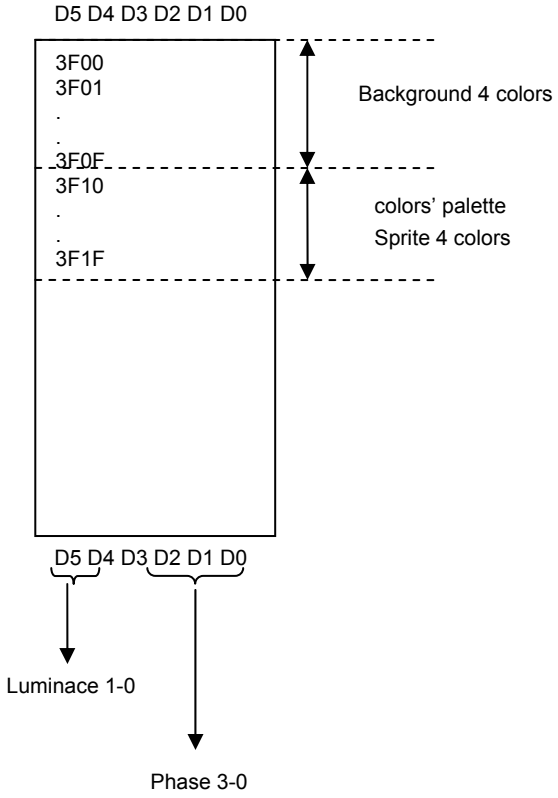
Bit 5 → Background or Sprite selector
Bit 4 → Bit 4 of Color set of Sprite
Bit 3 → Bit 3 of Color set of Sprite



Somewhere in the external video memory

Color Palette

Address being 3F00-3F1F programmer can program the color palette. There are 6 bits D5-D0, to specify the color.

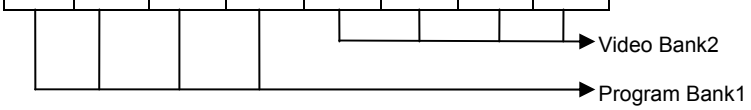


Register Description

Address Ports of Program Unit

4100H W Program Bank1, Video Bank2

D7	D6	D5	D4	D3	D2	D1	D0
PA24	PA23	PA22	PA21	VA24	VA23	VA22	VA21



4101H W Preload Times of timer interrupt

TSYNEN	D7	D6	D5	D4	D3	D2	D1	D0
0	The number of AD12 switching high low							
1	The number of HSYNC switching high low							

4102H W Load the preload interrupt timer data and start to count

D7	D6	D5	D4	D3	D2	D1	D0
Any value							

Writing any value to this register will load the data of 4101H to timer and start to count.

4103H W Disable the timer interrupt

D7	D6	D5	D4	D3	D2	D1	D0
Any value							

Writing any value to this register will disable the timer interrupt.

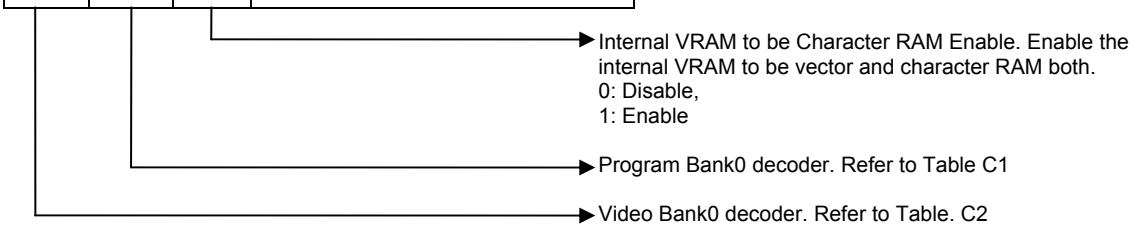
4104H W Enable the timer interrupt

D7	D6	D5	D4	D3	D2	D1	D0
Any value							

Writing any value to this register will enable the timer interrupt.

4105H W V Bank0 decode type, P Bank0 decode type, Inter Char VRAM

D7	D6	D5	D4	D3	D2	D1	D0
COMR7	COMR6	IVRCH	UNUSED				



PQ2EN (410BH)	COMR6	A[14:13] (CPU)	TPA20	TPA19	TPA18	TPA17	TPA16	TPA15	TPA14	TPA13
0	0H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	1H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	2H		1	1	1	1	1	1	1	0
	3H		1	1	1	1	1	1	1	1
	4H		1	1	1	1	1	1	1	0
	5H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	6H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	7H		1	1	1	1	1	1	1	1
1	0H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	1H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	2H		PQ27	PQ26	PQ25	PQ24	PQ23	PQ22	PQ21	PQ20
	3H		1	1	1	1	1	1	1	1
	4H		PQ27	PQ26	PQ25	PQ24	PQ23	PQ22	PQ21	PQ20
	5H		PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10
	6H		PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00
	7H		1	1	1	1	1	1	1	1

Table C1

COMR7	AD[12:10]	TVA17	TVA16	TVA15	TVA14	TVA13	TVA12	TVA11	TVA10
0H or 1H or CH or DH		RV47	RV46	RV45	RV44	RV43	RV42	RV41	AD10
2H or 3H or EH or FH		RV57	RV56	RV55	RV54	RV53	RV52	RV51	AD10
4H or 8H		RV07	RV06	RV05	RV04	RV03	RV02	RV01	RV00
5H or 9H		RV17	RV16	RV15	RV14	RV13	RV12	RV11	RV10
6H or AH		RV27	RV26	RV25	RV24	RV23	RV22	RV21	RV20
7H or BH		RV37	RV36	RV35	RV34	RV33	RV32	RV31	RV30

Table C2

4106H W Horizontal / Vertical Scrolling selector.

D7	D6	D5	D4	D3	D2	D1	D0
UNUSED							HV

Horizontal Scrolling or Vertical Scrolling selector.
0: Horizontal,
1: Vertical.

4107H W Program Bank0 register0

D7	D6	D5	D4	D3	D2	D1	D0
PQ07	PQ06	PQ05	PQ04	PQ03	PQ02	PQ01	PQ00

4108H W Program Bank0 register1

D7	D6	D5	D4	D3	D2	D1	D0
PQ17	PQ16	PQ15	PQ14	PQ13	PQ12	PQ11	PQ10

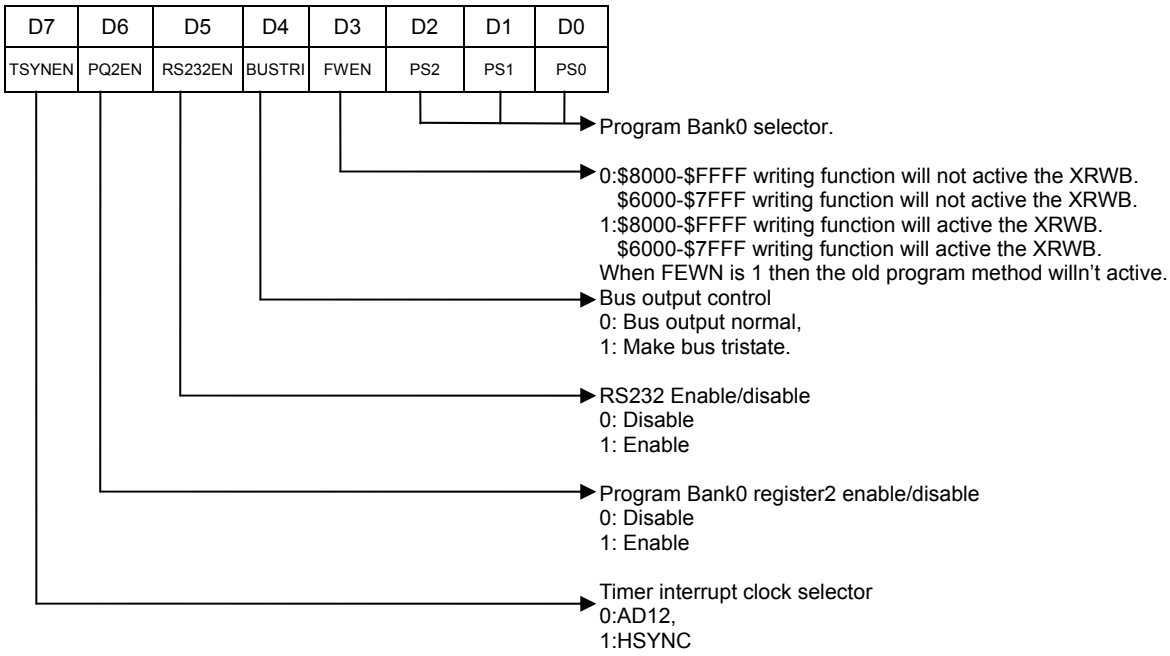
4109H W Program Bank0 register2

D7	D6	D5	D4	D3	D2	D1	D0
PQ27	PQ26	PQ25	PQ24	PQ23	PQ22	PQ21	PQ20

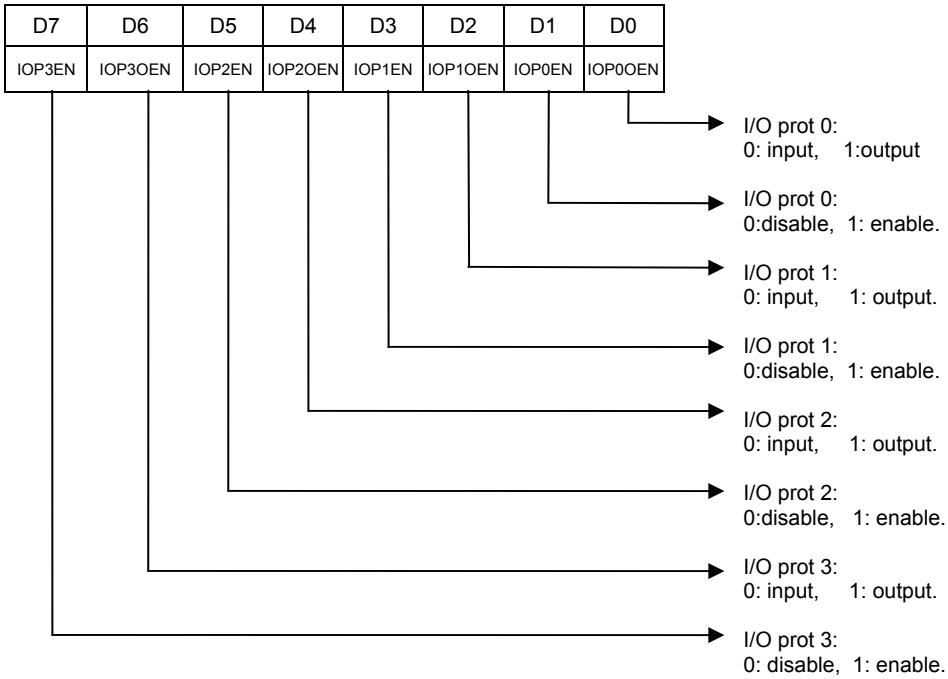
410AH W Program Bank0 register3

D7	D6	D5	D4	D3	D2	D1	D0
PQ37	PQ36	PQ35	PQ34	PQ33	PQ32	PQ31	PQ30

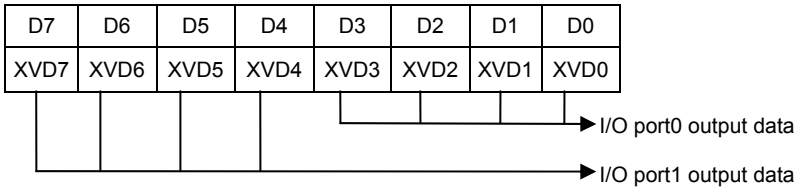
410BH W Timer interrupt clock selector, Program Bank0 register2 enable/disable, RS232 enable/disable, Bus output normal/ tristate, Program Bank0 selector



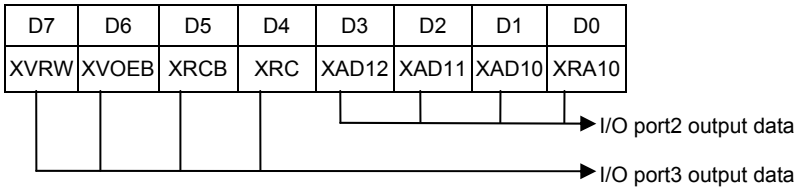
410DH W I/O port control



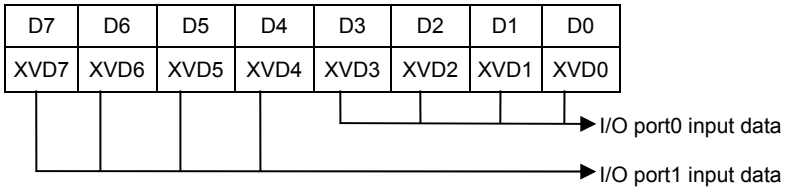
410EH W I/O port 0, 1 output data



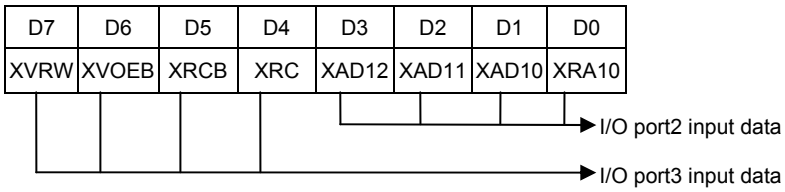
410FH W I/O port 2, 3 output data



410EH R I/O port 0, 1 input data



410FH R I/O port 2, 3 input data



4114H W Low byte of RS232 Timer

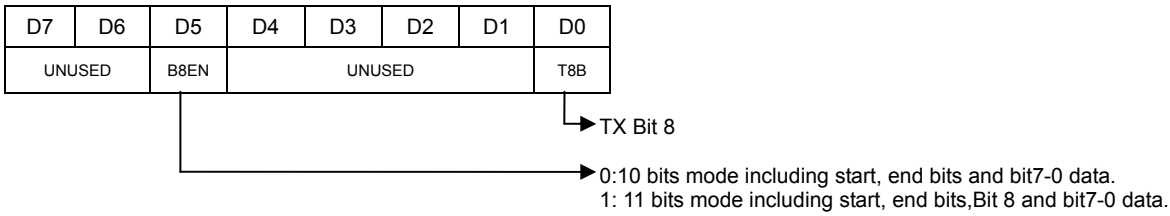
D7	D6	D5	D4	D3	D2	D1	D0
Low byte of RS232 Timer							

4115H W High byte of RS232 Timer

D7	D6	D5	D4	D3	D2	D1	D0
High byte of RS232 Timer							

In PAL system, CK21M is 26.601712MHz, in NTSC system is 21.47727MHz. RS232T=#4115,#4114 data. Baud rate will be CK21M/((RS232T+2)*2). For example, In PAL system, the baud rate 9600, RS232T=0567.

4119H W RS232 Register



4119H R RS232 Flags

D7	D6	D5	D4	D3	D2	D1	D0
RIFLAG	TIFLAG	RINGF	XF5OR6	XPORN		RERRF	R8B

- R8B → RX bit 8
- RERRF → RERRF=1 is the status of receiving error
- XF5OR6, XPORN → Output pins: TV system selector
- RINGF → RINGF=1 is the status of receiving data.
- TIFLAG → TIFLAG=1 is the status of completed sending data.
- RIFLAG → RIFLAG=1 is the status of completed receiving data.

411AH W TX data of RS232

D7	D6	D5	D4	D3	D2	D1	D0
TX data of RS232							

411BH R RX data of RS232

D7	D6	D5	D4	D3	D2	D1	D0
RX data of RS232							

Address Ports of Graph Unit

2000H W NMI, Sprite Size, Background AD12, Sprite AD12, Video data updated sequence, Vertical page specified, Horizontal page specified

D7	D6	D5	D4	D3	D2	D1	D0
NMI EN	UNUSED	SP SIZE	BK AD12	SP AD12	V W SEQ	VCOOR6	HCOOR6

- HCOOR6 → Page selection under horizontal mode.
  0: Page 0 is dispayed.
  1: Page 1 is displayed
- VCOOR6 → Page selection under vertical mode.
  0: Page 0 is dispayed.
  1: Page 1 is displayed
- V W SEQ → Video data updated sequency
  0: Horizontal
  1: Vertical
- SP AD12 → Sprite character in ROM address XAD12.
- BK AD12 → Background character in ROM address XAD12.
- SP SIZE → SPRITE SIZE CONTROL,
  1: BIG SPRITE (8X16))
  0: SMALL SPRITE (8X8)
- UNUSED → TEST pin control I/O
- NMI EN → NMI control
  0: enable NMI
  1: disable NMI

2001H W Sprite enable/disable, Background enable/disable, Sprite initial coordinate, Background initial coordinate, B/W or color control

D7	D6	D5	D4	D3	D2	D1	D0
UNUSED			SP EN	BK EN	SP INI	BK INI	B/W

B/W or Color control.
0: Color,
1: B/W

Background initial coordinate.
0: Righter,
1: Lefter

Sprite initial coordinate.
0:Righter,
1:Lefter

Background enable.
0: DISABLE,
1: ENABLE

Sprite enable.
0: DISABLE,
1: ENABLE

2002H R Blanking indicator, Priority indicator, Sprite overload indicator.

D7	D6	D5	D4	D3	D2	D1	D0
VSYN	B,S 0V	OVLOAD	UNUSED				

Sprite overload indicator.

Priority indicator

Blanking indicator.
0: Display,
1: Blanking

Read 2002H will also reset the command sequence for accessing 2005H and 2006H, without affecting the connect of 2005H and 2006H. An example is given after the description of register 2006H

2003H W Initial values of the Sprite pool counter (address)

D7	D6	D5	D4	D3	D2	D1	D0
The initial adderss to store the sprite data							

Set sprite pool counter initial data by this register.

2004H W Data of the sprite pool

D7	D6	D5	D4	D3	D2	D1	D0
Write the data of into DRAM							

Write data into sprite pool and increment sprite counter

2005H W Horizontal/Vertical coordinate of the display original mapping in RAM (two bytes set up).

D7	D6	D5	D4	D3	D2	D1	D0
Horizontal/Vertical coordinate							

Set the horizontal/vertical coordinate of the display original mapping in RAM (two bytes set up). The first write will set the horizontal coordinate and the second write will set the vertical coordinate. Before writing this register, read 2002H can reset the command sequence. (After reading 2002H, the first write to 2005H will set the horizontal coordinate and the next write will set the vertical coordinate.)

2006H W Initial Address of the Video RAM or ROM (two bytes set up)

D7	D6	D5	D4	D3	D2	D1	D0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Second byte							

D7	D6	D5	D4	D3	D2	D1	D0
		XRC	AD12	AD11	AD10	AD9	AD8
First byte							

Two bytes are needed to set the initial address of the Video RAM or ROM. Set the height byte first and then the low byte. The initial address will be incremented by one automatically, after every read/write to 2007H. Befor writing this register, read 2002H can reset the command sequence. After reading 2002H, the first write to 2006H will set the high byte address and the next write will set the low byte address. An example is given after the register description of 2007H.

2007H R/W Data read from/written to the Video RAM or ROM

D7	D6	D5	D4	D3	D2	D1	D0
Data read from/written to the Video RAM or ROM							

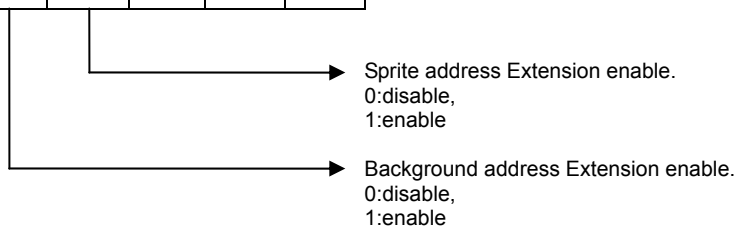
To access the Video RAM or ROM, fill the address into 2006H first and then read or write data from 2007H. Note: While reading data, the first data of 2007H is unknown. The next read will get the previous data pointed by 2006H.

Ex: read data from the video ram or rom at address 2010H and 2011H.

```
LDA  $2002  ;reset the command sequence
LDA  #20
STA  $2006  ;set high byte address
LDA  #10
STA  $2006  ;set low byte address
LDA  $2007  ;first byte is ignored
LDA  $2007  ;data of $2010
LDA  $2007  ;data of $2011
```

2010H W Background/Sprite address Extension enable.

D7	D6	D5	D4	D3	D2	D1	D0
	UNUSED		BKEXTEN	SPEXTEN			



2011H W Option of Vertical line number of LCD display, B/W 2 color mode., Composted Video DA Enable, Video

Extension Address EVA12 selector, Enable the internal VRAM or not

D7	D6	D5	D4	D3	D2	D1	D0
UNUSED		VLS1	VLS0	EVRAMEN	PIX2EN	VDAEN	EVA12S

Video Extension Address EVA12 selector.
0 ' reg. BKPAGE
1 ' HV

Composited Video DA Enable.
0 ' Enable
1 ' Disable

B/W 2 color mode.
0 ' Disable
1 ' Enable

Enable the internal VRAM or not.
0 ' Enable
1 ' Disable (Don't use it)

option of Vertical line number of LCD display

VLS1	VLS0	function
0	0	240 lines can be displayed.
0	1	160 lines can be displayed.
1	0	120 lines can be displayed.
1	1	80 lines can be displayed.

2012H W Video Bank0 register0

D7	D6	D5	D4	D3	D2	D1	D0
RV07	RV06	RV05	RV04	RV03	RV02	RV01	RV00

2013H W Video Bank0 register1

D7	D6	D5	D4	D3	D2	D1	D0
RV17	RV16	RV15	RV14	RV13	RV12	RV11	RV10

2014H W Video Bank0 register2

D7	D6	D5	D4	D3	D2	D1	D0
RV27	RV26	RV25	RV24	RV23	RV22	RV21	RV20

2015H W Video Bank0 register3

D7	D6	D5	D4	D3	D2	D1	D0
RV37	RV36	RV35	RV34	RV33	RV32	RV31	RV30

2016H W Video Bank0 register4

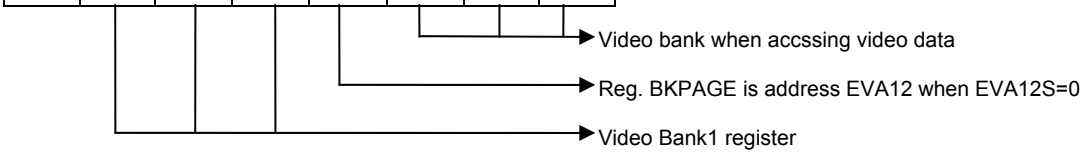
D7	D6	D5	D4	D3	D2	D1	D0
RV47	RV46	RV45	RV44	RV43	RV42	RV41	RV40

2017H W Video Bank0 register5

D7	D6	D5	D4	D3	D2	D1	D0
RV57	RV56	RV55	RV54	RV53	RV52	RV51	RV50

2018H W Video Bank1 register, BKPAGE, Video RW Bank

D7	D6	D5	D4	D3	D2	D1	D0
UNUSED	VA20	VA19	VA18	BKPAGE	VRWB2	VRWB1	VRWB0



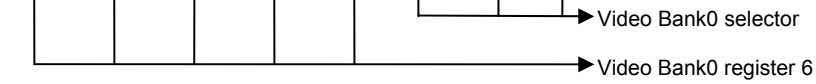
2019H W Reset the Gun port

D7	D6	D5	D4	D3	D2	D1	D0
Any value							

Writing any value to this register clear the X, Y coordinates of Gun port 1, 2

201AH W Video Bank0 register6, Video Bank0 selector

D7	D6	D5	D4	D3	D2	D1	D0
RV67	RV66	RV65	RV64	RV63	VB0S2	VB0S1	VB0S0



201CH R X coordinate of Gun port 1

D7	D6	D5	D4	D3	D2	D1	D0
X coordinate of Gun port 1							

Get the X coordinate of Gun port 1.

201DH R Y coordinates of Gun port 1

D7	D6	D5	D4	D3	D2	D1	D0
Y coordinate of Gun port 1							

Get the Y coordinate of Gun port 1.

201EH R X coordinate of Gun port 2

D7	D6	D5	D4	D3	D2	D1	D0
X coordinate of Gun port 2							

Get the X coordinate of Gun port 2.

201FH R Y coordinates of Gun port 2

D7	D6	D5	D4	D3	D2	D1	D0
Y coordinate of Gun port 2							

Get the Y coordinate of Gun port 2.

Sound Generator

Sound Generator XOP1 Address Port

Address	R/W	CHANNEL		Register								Note
				ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
4000H	W	A	RHYTHM A	1DY2	1DY1	1SC	1IW	1WI3	1WI2	1WI1	1WI0	Envelop Control
4001H	W	A	RHYTHM A	1AT	1ST2	1ST1	1ST0	1SG	1AD2	1AD1	1AD0	Auto Tune Control
4002H	W	A	RHYTHM A	1FT7	1FT6	1FT5	1FT4	1FT3	1FT2	1FT1	1FT0	Fine Tune Control
4003H	W	A	RHYTHM A	1SL4	1SL3	1SL2	1SL1	1SL0	1FTA	1FT9	1FT8	Coarse Tune & Single Sound Control
4004H	W	B	RHYTHM B	2DY2	2DY1	2SC	2IW	2WI3	2WI2	2WI1	2WI0	Envelop Control
4005H	W	B	RHYTHM B	2AT	2ST2	2ST1	2ST0	2SG	2AD2	2AD1	2AD0	Auto Tune Control
4006H	W	B	RHYTHM B	2FT7	2FT6	2FT5	2FT4	2FT3	2FT2	2FT1	2FT0	Fine Tune Control
4007H	W	B	RHYTHM B	2SL4	2SL3	2SL2	2SL1	2SL0	2FTA	2FT9	2FT8	Coarse Tune & Single Sound Control
4008H	W	C	ENVELOP	3EN	3EL6	3EL5	3EL4	3EL3	3EL2	3EL1	3EL0	Single Sound Enable
400AH	W	C	ENVELOP	3FT7	3FT6	3FT5	3FT4	3FT3	3FT2	3FT1	3FT0	Fine Tune Value
400BH	W	C	ENVELOP	3SL4	3SL3	3SL2	3SL1	3SL0	3FTA	3FT9	3FT8	Coarse Tune & Single Sound Control
400CH	W	D	NOISE			4SC	4IW	4WI3	4WI2	4WI1	4WI0	Envelope Control
400EH	W	D	NOISE	4NS				4BF3	4BF2	4BF1	4BF0	Control Base Frequency
400FH	W	D	NOISE	4SL4	4SL3	4SL2	4SL1	4SL0				Channel Enable & Single Sound Control
4010H	W	E	DWS DMA	DIRQ	DREP			SD3	SD2	SD1	SD0	Amplitude
4011H	W	E	DWS DMA		IA6	IA5	IA4	IA3	IA2	IA1	IA0	Initial Amplitude
4012H	W	E	DWS DMA	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	Starting add. of DWS data
4013H	W	E	DWS DMA	DL11	DL10	DL9	DL8	DL7	DL6	DL5	DL4	Data length of DWS data

Sound Generator XOP2 Address Port

Address	R/W	CHANNEL		Register								Note
				ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
4020H	W	A	RHYTHM A	1DY2	1DY1	1SC	1IW	1WI3	1WI2	1WI1	1WI0	Envelop Control
4021H	W	A	RHYTHM A	1AT	1ST2	1ST1	1ST0	1SG	1AD2	1AD1	1AD0	Auto Tune Control
4022H	W	A	RHYTHM A	1FT7	1FT6	1FT5	1FT4	1FT3	1FT2	1FT1	1FT0	Fine Tune Control
4023H	W	A	RHYTHM A	1SL4	1SL3	1SL2	1SL1	1SL0	1FTA	1FT9	1FT8	Coarse Tune & Single Sound Control
4024H	W	B	RHYTHM B	2DY2	2DY1	2SC	2IW	2WI3	2WI2	2WI1	2WI0	Envelop Control
4025H	W	B	RHYTHM B	2AT	2ST2	2ST1	2ST0	2SG	2AD2	2AD1	2AD0	Auto Tune Control
4026H	W	B	RHYTHM B	2FT7	2FT6	2FT5	2FT4	2FT3	2FT2	2FT1	2FT0	Fine Tune Control
4027H	W	B	RHYTHM B	2SL4	2SL3	2SL2	2SL1	2SL0	2FTA	2FT9	2FT8	Coarse Tune & Single Sound Control
4028H	W	C	ENVELOP	3EN	3EL6	3EL5	3EL4	3EL3	3EL2	3EL1	3EL0	Single Sound Enable
402AH	W	C	ENVELOP	3FT7	3FT6	3FT5	3FT4	3FT3	3FT2	3FT1	3FT0	Fine Tune Value
402BH	W	C	ENVELOP	3SL4	3SL3	3SL2	3SL1	3SL0	3FTA	3FT9	3FT8	Coarse Tune & Single Sound Control
402CH	W	D	NOISE			4SC	4IW	4WI3	4WI2	4WI1	4WI0	Envelope Control
402EH	W	D	NOISE	4NS				4BF3	4BF2	4BF1	4BF0	Control Base Frequency
402FH	W	D	NOISE	4SL4	4SL3	4SL2	4SL1	4SL0				Channel Enable & Single Sound Control
4030H	W	DWS/PCM					DP	DA2	DA1	~A15	~A14	Dws/ PCM selector, DA control
4031H	W	PCM		PCM7	PCM6	PCM5	PCM4	PCM3	PCM2	PCM1	PCM0	Write PCM Data

Parameter description:

xDY2, xDY1: Specify the duty cycle of the square wave of channel 1, 2. The mapping is described as the following table.

xDY2	xDY1	Duty
0	0	1/8
1	0	1/4
0	1	1/2
1	1	3/4

xSC:

Set the sound output to be continuous or one time only.
0: single sound (one time only)
1: continuous

xIW:

Envelop setting
0: The envelope decays from the FH to 0H with the slop specified by xWI[3:0].
1: The envelope is kept at a constant value specified by xWI3:0.

xWI[3:0]:

When xIW = 0, xWI[3:0] specify the decay time of the envelop from Fh to 0h as 4.16ms*(xWI[3:0]).
When xIW=1, xWI[3:0] specify the envelop level as full scale*(xWI3:0)/15d.

xAT:

Sound effect control of pitch-band
0: disable
1: enable; as enable, the frequency of the channel will smoothly shift from the setting value to maximum or minimum frequency. The function is used for special sound effect, like machine gun. And the modulation rate of pitch band is set by xSTx.

xST[2:0]:

Set the modulation time. Modulation time means the time of frequency change of each modulation, i.e., the change rate is inverse-proportion to modulation time.
Modulation time = 8.33ms*(xST[2:0])

xSG:

Specify the sign in front of 2^{-m} in equation for changing ratio.
0:"+"
1:"-"

xAD[2:0]:

m=xAD[2:0], a parameter to set the changing ratio of the frequency.
When xSG=0, $F_{n+1}=F_n*(1+2^{-m})$.
When xSG=1, $F_{n+1}=F_n*(1-2^{-m})$.
F_{n+1}: next frequency
F_n: current frequency

xFT[A:0]:

Frequency=111,860Hz/(xFTA:0), the minimum value of xFT[A:0] is 08H.

xSL[4:0]:

Sound duration of single sound.(Beat length decoder input))

xSL[4:0]		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Sound duration (ms)	BCLK2=120Hz	72	2024	152	8	312	24	632	40	1272	56	472	72	104	88	112	104
	BCLK2=100Hz	90	2530	190	10	390	30	790	50	1590	70	590	90	130	110	250	130
xSL[4:0]		**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**	**1B**	**1C**	**1D**	**1E**	**1F**
Sound duration (ms)	BCLK2=120Hz	88	120	184	136	376	152	760	168	1528	184	568	200	120	216	248	232
	BCLK2=100Hz	110	150	230	170	470	190	950	210	1910	230	710	250	150	270	310	290

BCLK2 is set by 4017H.

3EN:

0: Enable (Beat length 1)
1: Disable

3EL[6:0]:

Beat length 1 =BLCK1*3EL[6:0]
Through 4017H BLCK1 can be set as 250Hz or 200Hz.

4NS:

Noise band of channel 4 setting

0: wide band
1: narrow band

xBF[3:0]:
Specify the noise frequency.

DIRQ:
0: Disable DWS IRQ
1: Enable DWS IRQ

DREP:
0: No repeat
1: Repeat DWS data access

SD[3:0]:
Input of slop decoder.

SD[3:0]	FH	EH	DH	CH	BH	AH	9H	8H
Sample rate(Hz)	33K	25K	21K	17K	14K	13K	11K	9K
SD[3:0]	**7H**	**6H**	**5H**	**4H**	**3H**	**2H**	**1H**	**0H**
Sample rate(Hz)	8.4K	7.9k	7K	6.2K	5.5K	5.3K	4.7K	4.2K

IA[6:0]:
DWS Initial amplitude

SA[13:6]:
DWS Data start address #11xxxxxxxx000000, (SA[13:6]=xxxxxxxx)

DL[11:4]:
DWS or PCM data length #xxxxxxxx0000, (DL[11:4]=xxxxxxxx)

DP:
Speech synthesis DWS or PCM selector
0: DWS
1:PCM

DA2:
XOP2 DA enable/ disable
0: Disable (default)
1: Enable

DA1:
XOP1 DA enable/ disable
0: Enable (default)
1: Disable

~A15:
DWS or PCM DMA address A15's complement.

~A14:
DWS or PCM DMA address A14's complement.

PCM[7:0]:
Update the PCM data by CPU.
We have two ways to control the PCM data, one is updated by CPU, and another is DMA similar as DWS. PCM DMA is controlled by 4010H, 4012H and 4013H to specify the starting address, data length, slope and repeat or not.

Miscellaneous Address Port .

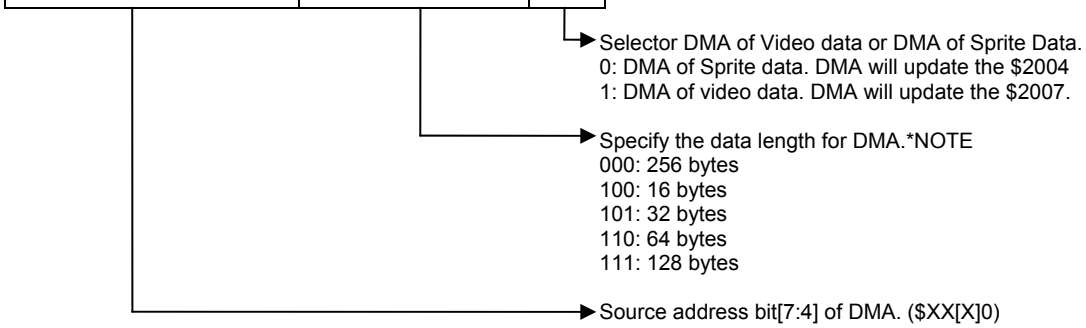
4014H W High byte Address of Source to start the DMA of Video data or Sprite data

D7	D6	D5	D4	D3	D2	D1	D0
High byte Address of Source							

It needs two bytes to specify the source address during DMA of video data or sprite data. 4014H specifies the high byte address ($[XX]X0). Writing 4014H also starts the DMA. VT02 is equipped the DMA of both video and sprite data. Please refer to 4034H for relevant settings.

4034H W Settings for the DMA of Video data or Sprite data

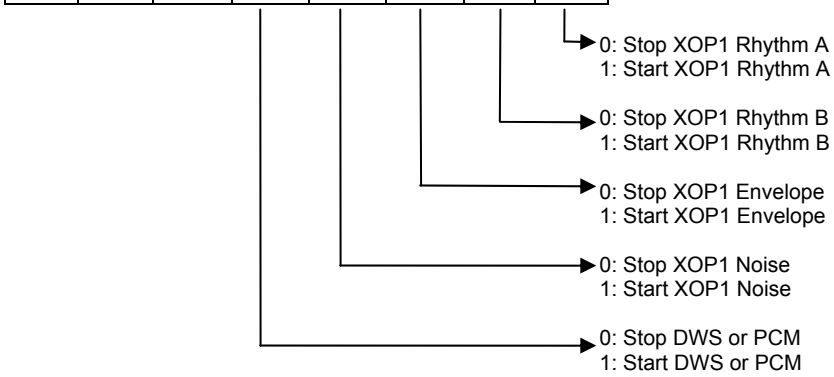
D7	D6	D5	D4	D3	D2	D1	D0
Source add. Bit[7:4] of DMA				Max. data length of DMA			SEL47



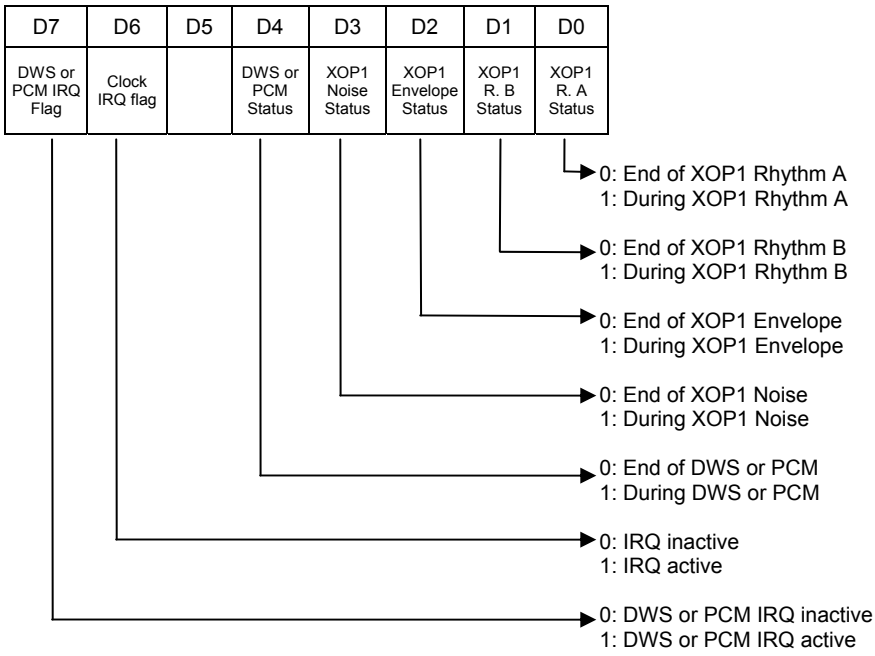
*NOTE: Under 64byte mode, VT02 cut the memory into 4 pieces. If you want to access complete 64 bytes. The low byte of address must be 00H, 40H, 80H or C0H, because VT02 will stop accessing when address counted to 3FH, 7FH, BFH or FFH respectively. Under 16 byte mode, VT02 cut the memory into 16 pieces. Under 128 byte mode, VT02 cut the memory into 2 pieces.

4015H W Enable/ disable XOP1 & DWS IRQ

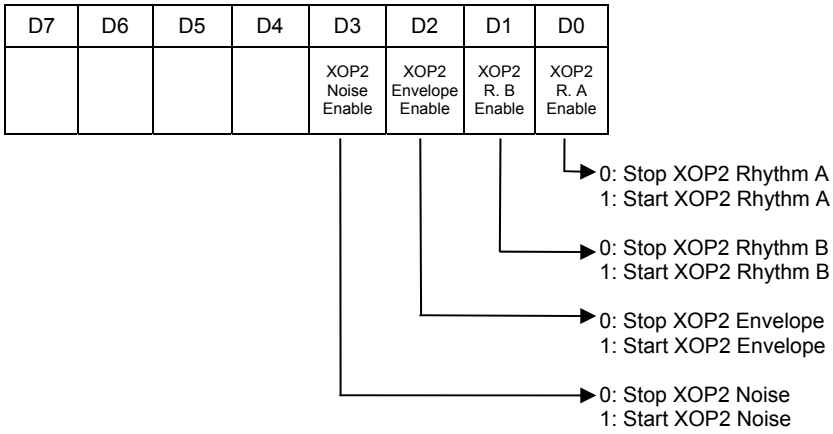
D7	D6	D5	D4	D3	D2	D1	D0
			DWS or PCM Enable	XOP1 Noise Enable	XOP1 Envelope Enable	XOP1 R. B Enable	XOP1 R. A Enable



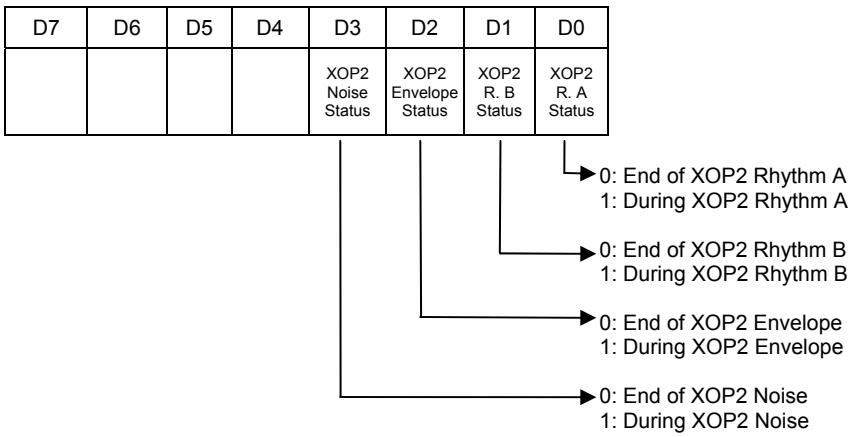
4015H R Read XOP1 FLAG



4035H W Enable/ disable XOP2



4035H R Read XOP2 FLAG



4016H W Set output pin XQ[2:0]

D7	D6	D5	D4	D3	D2	D1	D0
UNUSED					XQ2	XQ1	XQ0

Set output pin XQ0
Set output pin XQ1
Set output pin XQ2

4016H R Read peripheral data

D7	D6	D5	D4	D3	D2	D1	D0
UNUSED					Micro-Phone	Floppy	Major joystick

Read the data from major joystick
Read the data from floppy
Read the data from Microphone

4017H W Clock for beat Length 1, 2 and Clock IRQ Control

D7	D6	D5	D4	D3	D2	D1	D0
BLCK1/ BLCK2	Clock IRQ Enable						

0: Enable Clock IRQ (60Hz)
1: Disable Clock IRQ

0: BLCK1=250Hz and BLCK2=120Hz
1: BLCK1=200Hz and BLCK2=100Hz

4017H R Read Peripheral Data

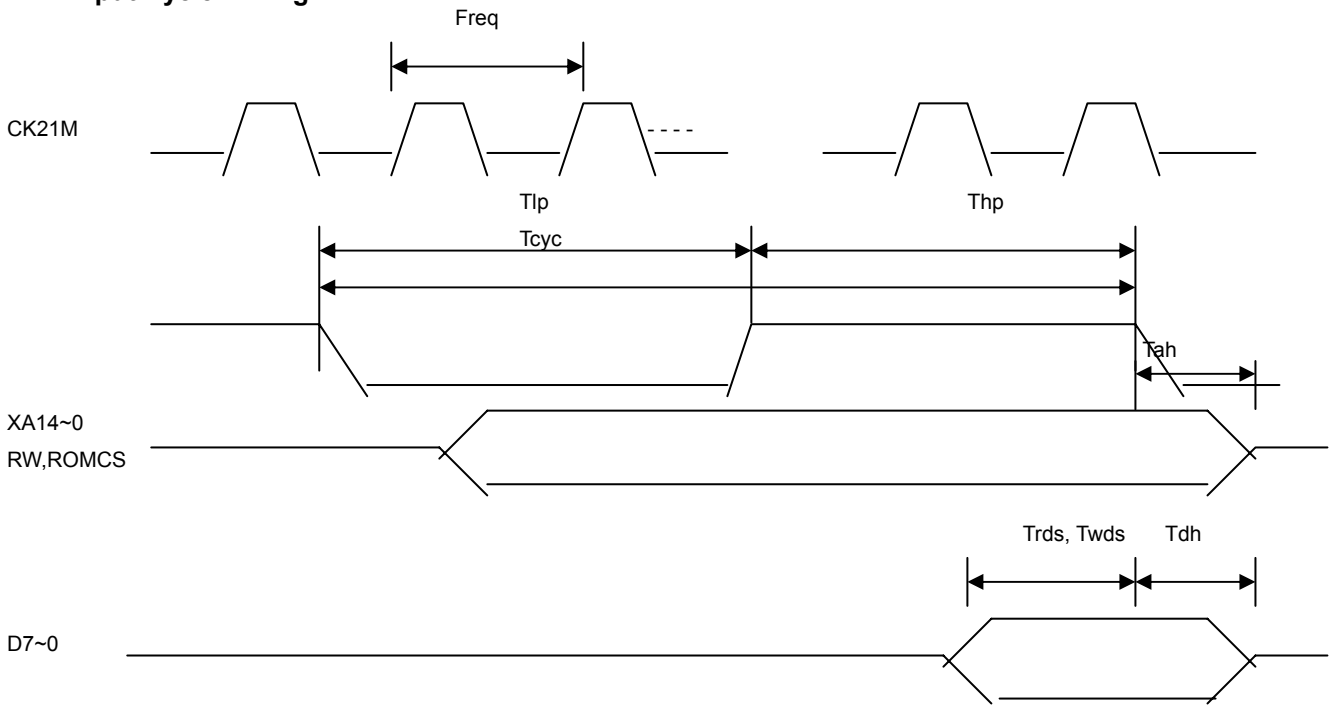
D7	D6	D5	D4	D3	D2	D1	D0
			Floppy Disk	Floppy Disk	Floppy Disk	Floppy Disk	Second Joystick

Read the data from second joystick
Read the data from floppy
Read the data from floppy
Read the data from floppy
Read the data from floppy

Timing Waveforms

Timing Spec. of Program Unit In Application Mode

Input Cycle Timing

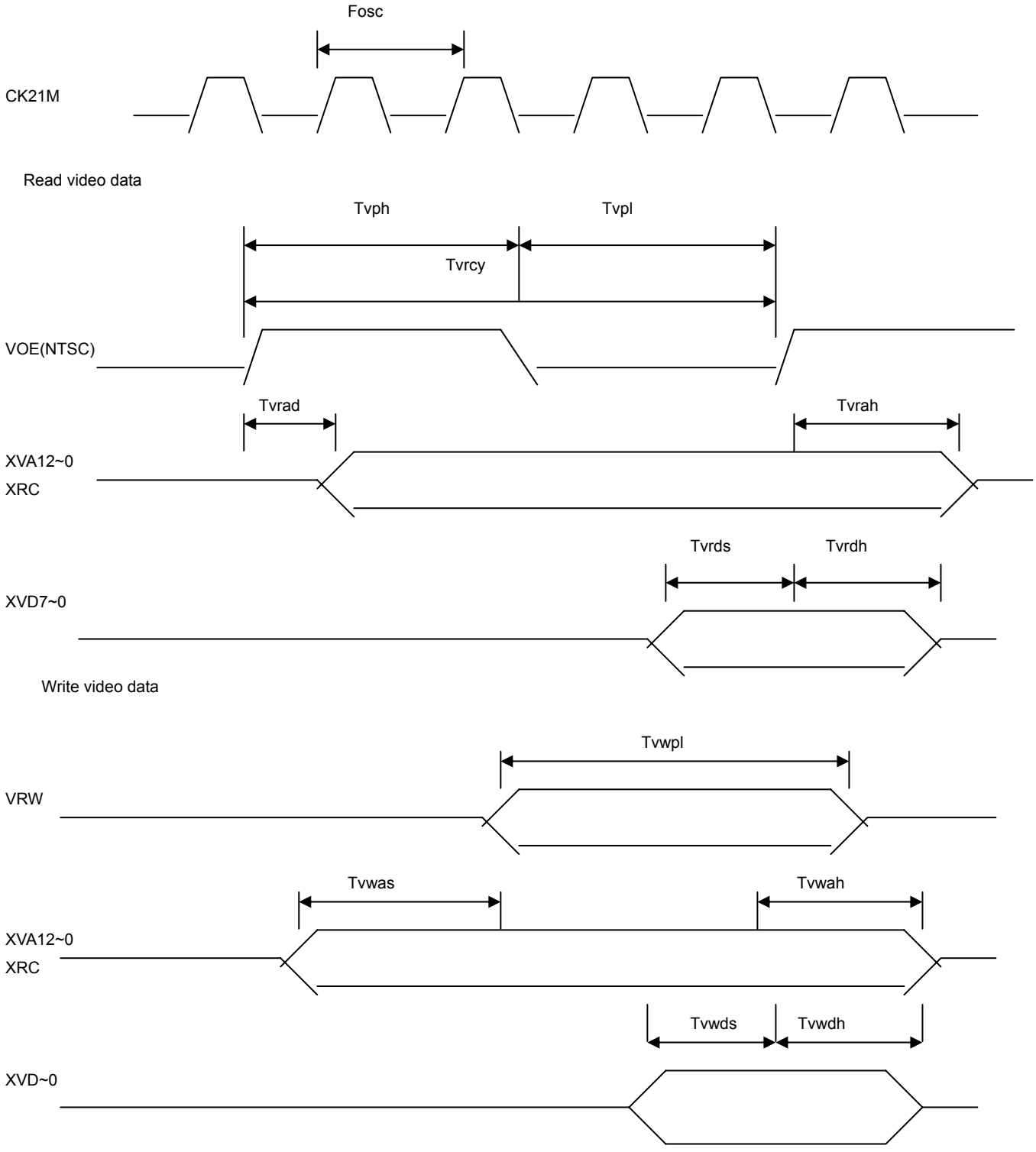


AC Characteristics : TA = 0`C to 70`C, VCC = 3.0V ~ 3.6V, GND = 0V

Symbol	Parameter	Min	Max	Unit	Condition
Fpal	Frequency of PAL B option	26.601712		MHz	
Fntsc	Frequency of NTSC option	21.47727		MHz	
Tcyc	Program cycle time	70	450	ns	
Tph	Cycle High Pulse Width	240	300	ns	
Tpl	Cylce Low Pulse Width	100	150	ns	
Tah	Program Address Hold time	10		ns	
Tdh	Program Data Hold time	10		ns	
Trds	Program Read Data Set up time	10		ns	
Twds	Program Write Data Set up time	10		ns	

Timing Spec of Graphic Unit In Application Mode

Input Cycle Timing



AC Characteristics: TA = 0`C to 70`C, VCC = 3.0V ~ 3.6V, GND = 0V

Symbol	Parameter	Min	Max	Unit	Condition
Fpal	Frequency of PAL B option	26.601712		MHz	
Fntsc	Frequency of NTSC option	21.47727		MHz	
Tvrcyc	Video Read cycle time	120	285	ns	
Tvph	Video Read High Pulse Width	120	150	ns	
Tvpl	Video Read Low Pulse Width	120	150	ns	
Tvrad	Video Read Address Delay time	7	35	ns	
Tvrah	Video Read Address Hold time	0		ns	
Tvrds	Video Read Data Set up time	10		ns	
Tvrdh	Video Read Data Set up time	10		ns	
Tvwpl	Video Write Pulse time	40	150	ns	
Tvwas	Video Write Address Set up time	10		ns	
Tvwah	Video Write Address Hold time	10	90	ns	
Tvwds	Video Write Data Set up time	10	70	ns	
Tvwdh	Video Write Data Hold time	10	90	ns	

DC Characteristics : TA = 0`C to 70`C, VCC = 3.0V ~ 3.6V, GND = 0V

Symbol	Parameter	Min	Max	Unit	Condition
VIL	Input Low Voltage	-0.5	0.8	V	
VIL	Input High Voltage	2.4	VCC+0.4	V	
VOL	Output Low Voltage		0.8	V	
VOH	Output High Voltage	2.4		V	
VCL	Clock Low Voltage	-0.7	0.4	V	
VCH	Clock High Voltage	2.5	3.5	V	
ICC	Power Supply Current		30	mA	
IIL	Input Leakage Current		10	uA	
ICL	Clock Leakage		10	uA	
ITL	Tri_state Leakage		20	uA	
IRL	Reset pin Leakage (pull high R)		1	mA	
IOL	Output Low Current	2	10	mA	
IOH	Output High Current	2	10	mA	

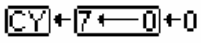
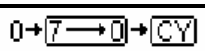
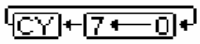
Programming Guide

1. For avoiding the unexpected IRQ interrupt happen all the time. It's better first to set $4017 = #$C0 or #$40.
2. If programmer didn't set the new address ports, the old compatible mode will be chose default.
3. In single bus mode, the program initial address A24-A0 is 007FFFC, and the video initial address is 0000XXX. You can individually specify the program and video bank; the hardware will combine the two independent bus into one bus. Study the program address multiplexer and video address multiplexer before the programming. The original decoder address ports is still reserved, and $4102 ~ $410A can also control the decoder address ports. A special port $4109 is the program bank0 register 2 make the program bank register increased from 2 to 3, but it must specified by $4109.
4. In the background extension address enable, the BG4, BG3 and character vector will be a 10 bits character vector; the character size will be 16X16. The background color set function will be disable, and color set (BG4,BG3) be fixed 00.
5. In reading gun ports, it's better to control the gun ports in NMI interrupt service routine. Read $201C to get gun one X coordinate, and read $201D to get Y coordinate. Read $201E to get gun two X coordinate, and read $201F to get Y coordinate. Finally, write $2019 to reset all of the gun ports.
6. PCM only outputted to XOP2. Programmer should set $4030 = #$18 to turn on the DA of channel and switch to PCM mode (channel DA 2 is off default). Setting the $4031 any 8 bits value will directly output to the XOP2 DA. If you want to use the PCM DMA mode, you should set $4010, $4012, $4013, these ports function is similar as DWS. The PCM data length is maximally 4081 bytes. If the PCM data is over than 4081 bytes, you should enable the PCM IRQ, set the unit waveform mode $4010 initially and modify the $4012 and Bank register to point the suitable PCM data address and start PCM DMA by $4015 = #$10 in interrupt service routine. PCM DMA mode is exclusive from DWS mode; you can only select one of these two modes.
7. Video DMA can update the data of the address $2004 or $2007. If you set $4034 = #$58, and $4014 = #$02, the video DMA will be started and update the $2004 from $0250~$025F 16 bytes. If you set the $4034 = #$AD, and $4014 = #$03, the video DMA will be start and update the $2007 from $03A0~$03BF 32 bytes. If you set the $4034 = #$0D, and $4014 = #$03, the video DMA will be start and update the $2007 from $0300~$033F 64 bytes.
8. When you connect an additional chip and you have to use the XRWB function to control them then you have to set up #410B function. The thing you have to know that when the FWEN was high then the old program method will not active.
9. Don't use the DMA copy to color palette in NTSC system. PAL system hasn't this prohibition. If you want to use the DMA copy to color palette in NTSC system, please follow up the below method. Otherwise your color will not correct.
   --- Use the DMA copy to color palette, but read the $4119(D3,D4) to check NTSC or PAL. If it was NTSC then shift the data one byte.
   For example: IN NTSC, put the data of $3F00 into $0301 and the data of 3F01 in to $0302 , $2006=$3F00,$4034=01,$4014=03.
   Finally $2006=$3FFF , $2007= data of $3FFF
10. Because of the gun game program will access the color palette address $3F20 data. So please you add a short program to initialize the $3F20 to #$2D in the menu program. Otherwise your gun game will not normal operation in this system.
11. The PCM data should be the multiple of 64 bytes. If your data haven't fill this capacity then it will have some noise issue in the PCM playing step. The PCM DMA data length is different from our previous understanding as the datasheet. As we set the $4013=#$FF, the data length isn't 4096 bytes. Physically, it only play 4081 bytes, data fetched from address $000 to $FF0. The data of address from $FF1 to $FFF can't be fetched from PCM DMA.
12. When you want to use the RS232 function function of VT02. Please you follow up this notice.If RS232 is necessary, please make the first command $410B(D5)=1. Because of TXDP will output data through XCUP47. In order to avoid the $4017 active this pin and make communicated error, we should set XCUP47 is TXDP first.

The detail Instruction table

● According to the function of instruction set

According to the function of instruction set							
Assembly Language Form	Addressing Mode	Assembly Language Form	Operation	Flag NV● BDIZC	OP. Code	No. Bytes	No. Cycles
Access Instruction							
LDA	Immediate	LDA #Oper	A ← M	N●●●●●Z●	A9	2	2
	Zero page	LDA Oper			A5	2	3
	Zero page,X	LDA Oper,X			B5	2	4
	Absolute	LDA Oper			AD	3	4
	Absolute,X	LDA Oper,X			BD	3	4**
	Absolute,Y	LDA Oper,Y			B9	3	4**
	(Indirect,X)	LDA (Oper,X)			A1	2	6
	(Indirect),Y	LDA (Oper),Y			B1	2	5**
LDX	Immediate	LDX # Oper	X ← M	N●●●●●Z●	A2	2	2
	Zero page	LDX Oper			A6	2	3
	Zero page,Y	LDX Oper,Y			B6	2	4
	Absolute	LDX Oper			AE	3	4
	Absolute,Y	LDX Oper,Y			BE	3	4**
LDY	Immediate	LDY # Oper	Y ← M	N●●●●●Z●	A0	2	2
	Zero page	LDY Oper			A4	2	3
	Zero page,X	LDY Oper,X			B4	2	4
	Absolute	LDY Oper			AC	3	4
	Absolute,X	LDY Oper,X			BC	3	4**
STA	Zero page	STA Oper	M ← A	●●●●●●●●	85	2	3
	Zero page,X	STA Oper,X			95	2	4
	Absolute	STA Oper			8D	3	4
	Absolute,X	STA Oper,X			9D	3	5
	Absolute,Y	STA Oper,Y			99	3	5
	(Indirect,X)	STA (Oper,X)			81	2	6
	(Indirect),Y	STA (Oper),Y			91	2	6
STX	Zero page	STX Oper	M ← X	●●●●●●●●	86	2	3
	Zero page,Y	STX Oper,Y			96	2	4
	Absolute	STX Oper			8E	3	4
STY	Zero page	STY Oper	M ← Y	●●●●●●●●	84	2	3
	Zero page,X	STY Oper,X			94	2	4
	Absolute	STY Oper			8C	3	4
Push processor status on stack							
PHA	Implied	PHA	(S)←A, S←S-1	●●●●●●●●	48	1	3

Assembly Language Form	Addressing Mode	Assembly Language Form	Operation	Flag NV● BDIZC	OP. Code	No. Bytes	No. Cycles
PHP	Implied	PHP	(S)←P, S←S-1	●●●●●●●●	08	1	3
PLA	Implied	PLA	S←S+1, A←(S)	N●●●●●Z●	68	1	4
PLP	Implied	PLP	S←S+1, P←(S)	(Stack)	28	1	4
Decrement/Increment memory by one							
DEC	Zero page	DEC Oper	M ← M-1	N●●●●●Z●	C6	2	5
	Zero page,X	DEC Oper,X			D6	2	6
	Absolute	DEC Oper			CE	3	6
	Absolute,X	DEC Oper,X			DE	3	7
DEX	Implied	DEX	X ← X - 1	N●●●●●Z●	CA	1	2
DEY	Implied	DEY	Y ← Y - 1	N●●●●●Z●	88	1	2
INC	Zero page	INC Oper	M ← M + 1	N●●●●●Z●	E6	2	5
	Zero page,X	INC Oper,X			F6	2	6
	Absolute	INC Oper			EE	3	6
	Absolute,X	INC Oper,X			FE	3	7
INX	Implied	INX	X ← X + 1	N●●●●●Z●	E8	1	2
INY	Implied	INY	Y ← Y + 1	N●●●●●Z●	C8	1	2
Shift/Rotate Left/Right one bit							
ASL	Accumulator	ASL A		N●●●●●ZC	0A	1	2
	Zero page	ASL Oper			06	2	5
	Zero page,X	ASL Oper,X			16	2	6
	Absolute	ASL Oper			0E	3	6
	Absolute,X	ASL Oper,X			1E	3	7
LSR	Accumulator	LSR A		0●●●●●ZC	4A	1	2
	Zero Page	LSR Oper			46	2	5
	Zero page,X	LSR Oper,X			56	2	6
	Absolute	LSR Oper			4E	3	6
	Absolute,X	LSR Oper,X			5E	3	7
ROL	Accumulator	ROL A		N●●●●●ZC	2A	1	2
	Zero Page	ROL Oper			26	2	5
	Zero page,X	ROL Oper,X			36	2	6
	Absolute	ROL Oper			2E	3	6
	Absolute,X	ROL Oper,X			3E	3	7
ROR	Accumulator	ROR A		N●●●●●ZC	6A	1	2
	Zero Page	ROR Oper			66	2	5
	Zero page,X	ROR Oper,X			76	2	6
	Absolute	ROR Oper			6E	3	6
	Absolute,X	ROR Oper,X			7E	3	7

Assembly Language Form	Addressing Mode	Assembly Language Form	Operation	Flag NV● BDIZC	OP. Code	No. Bytes	No. Cycles
Logical operation instruction							
AND	Immediate	AND #Oper	A←A AND M	N●●●●●Z●	29	2	2
	Zero page	AND Oper			25	2	3
	Zero page,X	AND Oper,X			35	2	4
	Absolute	AND Oper			2D	3	4
	Absolute,X	AND Oper,X			3D	3	4**
	Absolute,Y	AND Oper,Y			39	3	4**
	(Indirect,X)	AND (Oper,X)			21	2	6
	(Indirect),Y	AND (Oper),Y			31	2	5**
BIT	Zero page	BIT Oper	$N \leftarrow M_7, V \leftarrow M_6$		24	2	3
	Absolute	BIT Oper			2C	3	4
CMP	Immediate	CMP #Oper	A - M	N●●●●●ZC	C9	2	2
	Zero page	CMP Oper			C5	2	3
	Zero page,X	CMP Oper			D5	2	4
	Absolute	CMP Oper			CD	3	4
	Absolute,X	CMP Oper, X			DD	3	4**
	Absolute,Y	CMP Oper, Y			D9	3	4**
	(Indirect,X)	CMP (Oper,X)			C1	2	6
	(Indirect),Y	CMP (Oper),Y			D1	2	5**
CPX	Immediate	CPX #Oper	X - M	N●●●●●ZC	E0	2	2
	Zero page	CPX Oper			E4	2	3
	Absolute	CPX Oper			EC	3	4
CPY	Immediate	CPY #Oper	Y - M	N●●●●●ZC	C0	2	2
	Zero page	CPY Oper			C4	2	3
	Absolute	CPY Oper			CC	3	4
EOR	Immediate	EOR #Oper	A ← A XOR M	N●●●●●Z●	49	2	2
	Zero page	EOR Oper			45	2	3
	Zero page,X	EOR Oper, X			55	2	4
	Absolute	EOR Oper			4D	3	4
	Absolute,X	EOR Oper, X			5D	3	4**
	Absolute,Y	EOR Oper, Y			59	3	4**
	(Indirect,X)	EOR (Oper,X)			41	2	6
	(Indirect),Y	EOR (Oper),Y			51	2	5**

Assembly Language Form	Addressing Mode	Assembly Language Form	Operation	Flag NV● BDIZC	OP. Code	No. Bytes	No. Cycles
ORA	Immediate	ORA #Oper	A ← A OR M	N●●●●●Z●	09	2	2
	Zero page	ORA Oper			05	2	3
	Zero page,X	ORA Oper, X			15	2	4
	Absolute	ORA Oper			0D	3	4
	Absolute,X	ORA Oper, X			1D	3	4**
	Absolute,Y	ORA Oper, Y			19	3	4**
	(Indirect,X)	ORA (Oper,X)			01	2	6
	(Indirect),Y	ORA (Oper),Y			11	2	5**
Arithmetic operation instruction							
ADC	Immediate	ADC #Oper	A ← A + M+C	NV●●●●ZC	69	2	2
	Zero page	ADC Oper			65	2	3
	Zero page,X	ADC Oper, X			75	2	4
	Absolute	ADC Oper			6D	3	4
	Absolute,X	ADC Oper, X			7D	3	4**
	Absolute,Y	ADC Oper, Y			79	3	4**
	(Indirect,X)	ADC (Oper,X)			61	2	6
	(Indirect),Y	ADC (Oper),Y			71	2	5**
SBC	Immediate	SBC #Oper	A ←A-M-1+C	NV●●●●ZC	E9	2	2
	Zero page	SBC Oper			E5	2	3
	Zero page,X	SBC Oper, X			F5	2	4
	Absolute	SBC Oper			ED	3	4
	Absolute,X	SBC Oper, X			FD	3	4**
	Absolute,Y	SBC Oper, Y			F9	3	4**
	(Indirect,X)	SBC (Oper,X)			E1	2	6
	(Indirect),Y	SBC (Oper),Y			F1	2	5**
Assembly Language Form	Addressing Mode	Assembly Language Form	Operation	Flag NV● BDIZC	OP. Code	No. Bytes	No. Cycles
BCC[2]	Relative	BCC Oper	When C = 0 jump	●●●●●●●●	90	2	2***
BCS[2]	Relative	BCS Oper	When C = 1 jump	●●●●●●●●	B0	2	2***
BEQ	Relative	BEQ Oper	When Z = 1 jump	●●●●●●●●	F0	2	2***
BMI	Relative	BMI Oper	When N = 1 jump	●●●●●●●●	30	2	2***
BNE	Relative	BNE Oper	When Z = 0 jump	●●●●●●●●	D0	2	2***
BPL	Relative	BPL Oper	When N = 0 jump	●●●●●●●●	10	2	2***
BVC	Relative	BVC Oper	When V = 0 jump	●●●●●●●●	50	2	2***
BVS	Relative	BVS Oper	When V = 1 jump	●●●●●●●●	70	2	2***
JMP	Absolute	JMP Oper	PC ← Addr	●●●●●●●●	4C	3	3
	Indirect absolute	JMP(Oper)			6C	3	5
	Absolute,X	JMP (Oper , X)			7C	3	6
JSR	Absolute	JSR Oper	PC←PC+2	●●●●●●●●	20	3	6
			(S)←PCH,S←S-1				
			(S)←PCL,S←S-1				
			PC←Oper				
RTI	Implied	RTI	S←S+1, P←(S)	(Stack)	40	1	6
			S←S+1, PCL←(S)				
			S←S+1,PCH←(S)				
RTS	Implied	RTS	S←S+1, PCL←(S)	●●●●●●●●	60	1	6

			S←S+1,PCH←(S) PC ← PC+1,				
Processor flag instruction							
CLC	Implied	CLC	C ← 0	●●●●●●●1	18	1	2
CLD	Implied	CLD	D ← 0	●●●●1●●●	D8	1	2
CLI	Implied	CLI	I ← 0	●●●●●1●●	58	1	2
CLV	Implied	CLV	V ← 0	●1●●●●●●	B8	1	2
SEC	Implied	SEC	C ← 0	●●●●●●●0	38	1	2
SED	Implied	SED	D ← 0	●●●●0●●●	F8	1	2
SEI	Implied	SEI	I ← 0	●●●●●0●●	78	1	2

Assembly Language Form	Addressing Mode	Assembly Language Form	Operation	Flag NV● BDIZC	OP. Code	No. Bytes	No. Cycles
Register transfer instruction							
TAX	Implied	TAX	X ← A	N●●●●●Z●	AA	1	2
TAY	Implied	TAY	Y ← A	N●●●●●Z●	A8	1	2
TSX	Implied	TSX	X ← S	N●●●●●Z●	BA	1	2
TXA	Implied	TXA	A ← X	N●●●●●Z●	8A	1	2
TXS	Implied	TXS	S ← X	●●●●●●●●	9A	1	2
TYA	Implied	TYA	A ← Y	N●●●●●Z●	98	1	2
Other special instruction							
BRK	Implied	BRK	PC←PC+2 B ← 1, I←1 (S)←PCH,S←S-1 (S)←PCL,S←S-1 (S) ← P ,S←S-1	●●●1●1●●	00	1	7
NOP	Implied	NOP	No operation	●●●●●●●●	EA	1	2

Note：

** Add one cycle, if indexing across page boundary.

*** Add one cycle if branch is taken, add one additional if branching operation crosses page boundary.

1 BIT instruction copy the bit6 of test byte to flag V，Copy the bit7 of test byte to flag N，But if you use the immediate address mode then you cann't change the value of flag V and flag N。The value of Flag Z was based on the accumulator and operation result。

2 BBC and BCS instruction is BLT（Branch Less Than) and BGE（Branch Greater or Equal）instruction，the difference between these condition of jump instruction is only the assembly language form。

● According to the OP. code of instruction table.

According to the OP. code																	
Low \ High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Low \ High
0	BRK imp	ORA inx				ORA zpg	ASL zpg		PHP imp	ORA imm	ASL acc			ORA abs	ASL abs		0
1	BPL rla	ORA iny				ORA zpx	ASL zpx		CLC imp	ORA aby				ORA abx	ASL abx		1
2	JSR abs	AND inx			BIT zpg	AND zpg	ROL zpg		PLP imp	AND imm	ROL acc		BIT abs	AND abs	ROL abs		2
3	BMI rla	AND iny				AND zpx	ROL zpx		SEC imp	AND aby				AND abx	ROL abx		3
4	RTI imp	EOR inx				EOR zpg	LSR zpg		PHA imp	EOR imm	LSR acc		JMP abs	EOR abs	LSR abs		4
5	BVC rla	EOR iny				EOR zpx	LSR zpx		CLI imp	EOR aby				EOR abx	LSR abx		5
6	RTS imp	ADC inx				ADC zpg	ROR zpg		PLA imp	ADC imm	ROR acc		JMP abi	ADC abs	ROR abs		6
7	BVS rla	ADC iny				ADC zpx	ROR zpx		SEI imp	ADC aby				ADC abx	ROR abx		7
8		STA inx			STY zpg	STA zpg	STX zpg		DEY imp		TXA imp		STY abs	STA abs	STX abs		8
9	BCC rla	STA iny			STY zpx	STA zpx	STX zpy		TYA imp	STA aby	TXS imp			STA abx			9
A	LDY imm	LDA inx	LDX imm		LDY zpg	LDA zpg	LDX zpg		TAY imp	LDA imm	TAX imp		LDY abs	LDA abs	LDX abs		A
B	BCS rla	LDA iny			LDY zpx	LDA zpx	LDX zpx		CLV imp	LDA aby	TSX imp		LDY abx	LDA abx	LDX aby		B
C	CPY imm	CMP inx			CPY zpg	CMP zpg	DEC zpg		INY imp	CMP imm	DEX imp		CPY abs	CMP abs	DEC abs		C
D	BNE rla	CMP iny				CMP zpx	DEC zpx		CLD imp	CMP aby				CMP abx	DEC abx		D
E	CPX imm	SBC inx			CPX zpg	SBC zpg	INC zpg		INX imp	SBC imm	NOP imp		CPX abs	SBC abs	INC abs		E
F	BEQ rla	SBC iny				SBC zpx	INC zpx		SED imp	SBC aby				SBC abx	INC abx		F
Low \ High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Low \ High

Note：

Immediate address mode	imm
Absolute address mode	abs
Zero page address mode	zpg
Accumulator address mode	acc
Implied address mode	imp
Absolute ,X address mode	abx
Absolute ,Y address mode	aby
Zero page,X address mode	zpx
Zero page,Y address mode	zpy
Indirect address mode	abi
Relative address mode	rla
(Indirect,X) address mode	inx
(Indirect) ,Y address mode	iny
Abs. Indirect address mode	ina
Zero page Indirect address mode	inz