VT01
Console and LCD Interface
(Real 4 colors or Virtual 16 colors)

Table of contents

Revision History:

Ver.	Contents
A1	Original Version
A2	Revise the CPU Instruction.

VT01
Console and LCD Interface
(Real 4 colors or Virtual 16 colors)

Features

System
- CPU: 6502
- Internal Program RAM: 2K Bytes
- Internal Video RAM: 2K Bytes
- DMA (Sprite)
- Multiple control of IRQ
- Programmable timer
- T.V. signal output (NTSC, PAL)

Peripheral Applications
- Joystick
- STN and TFT LCD interface built in.

Graphic Processor
- Resolution:
  TV: 256x240 pixels
  STN LCD:R/C check board 16colors 120x2x240.
  STN LCD:B/W 4 gray level 240x240.
  TFT LCD: 64 colors 160x3x240.
- 64 sprites in one frame
- Two pages of figure for Background.
- Sprites have 8X8, 8X16.
- Color palette has 28 colors.

Sound Generator
- 2 Rhythm channels,
- 2 special sound channels,
- 1 Data waveform synthesizer.

General Description

VT01 includes the CPU, Graphic Unit, Sound Unit, two internal 2KBytes SRAMs, and some I/O controller. There are two main systems in VT01, 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. VT01 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.

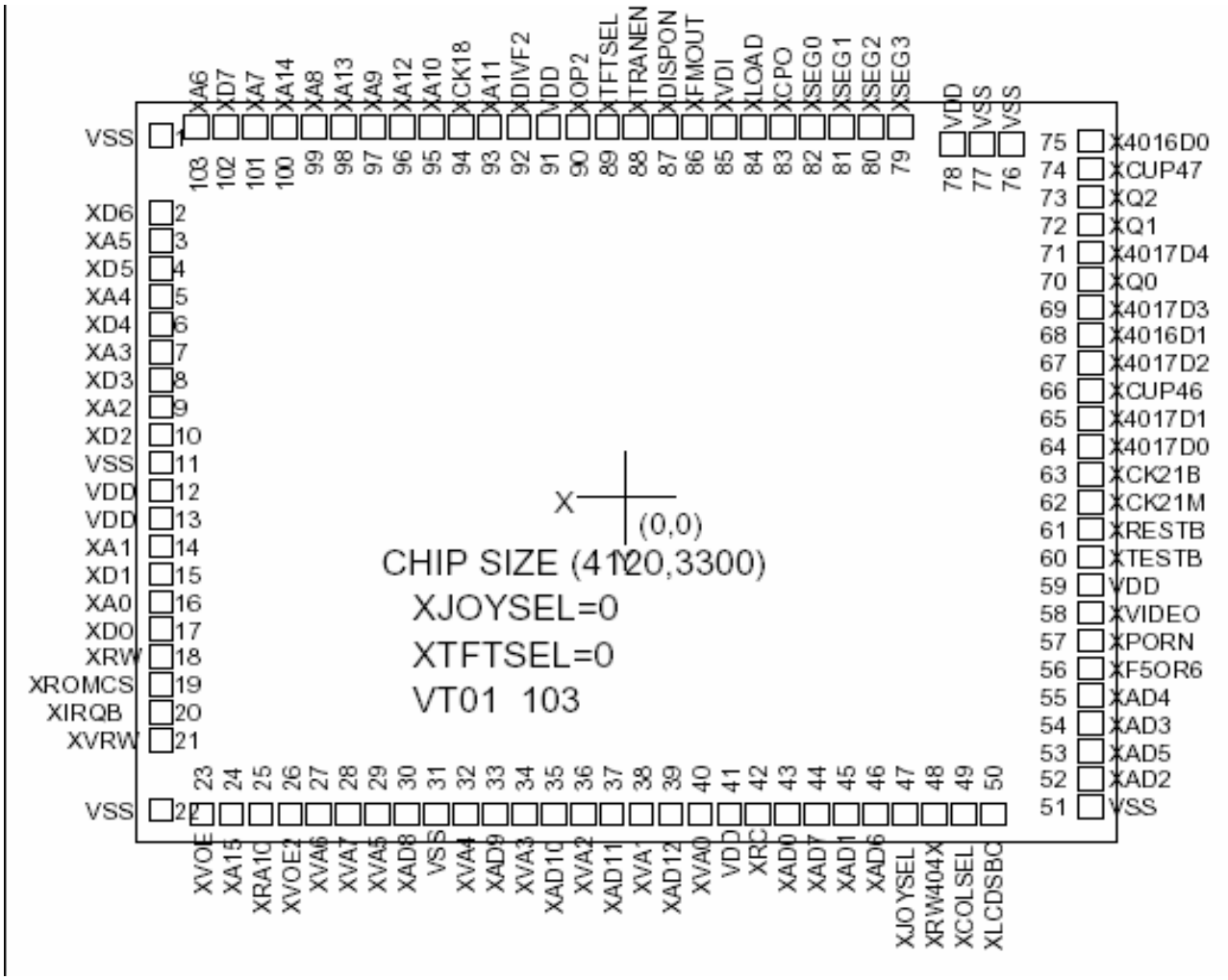
Graphic Unit is the main role of the video system. It can access the video memory automatically to display some figures. In addition to the internal program SRAM, VT01 is equipped the other 2KByte SRAM for Video RAM. Internal Video RAM stores pattern vectors for 2 pages of background. External Video memory stores the video characters to be pointed by the pattern vectors.

The function of the decoder is to expendand the memory location. In general, no decoder, the education machine can handle 32K bytes of program ROM and 8K bytes of character ROM. Decoder can help education machine to handle 2M bytes program and even more than this.

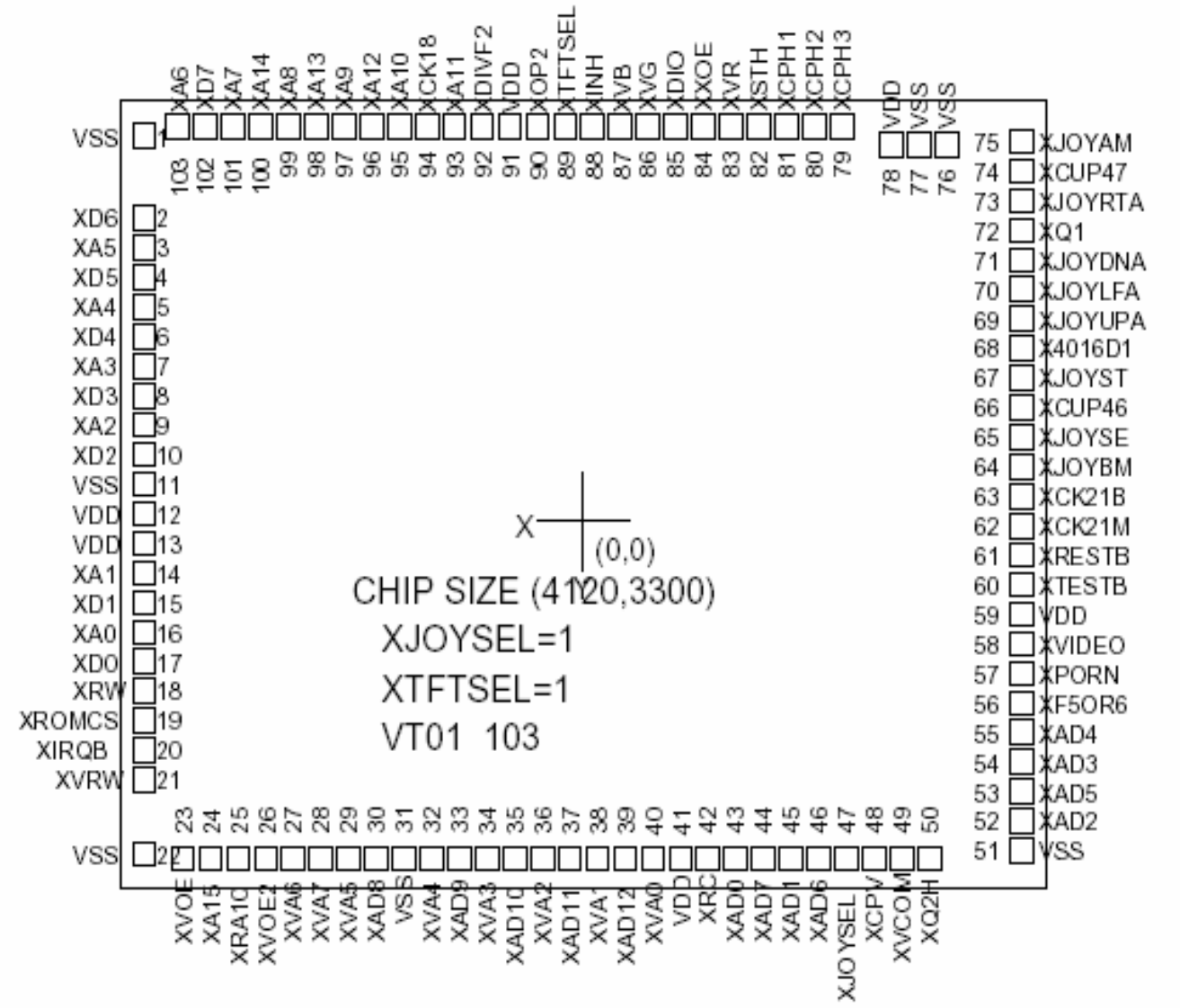
Pin configuration

(Chip size(X,Y):4120x3300 um^2)

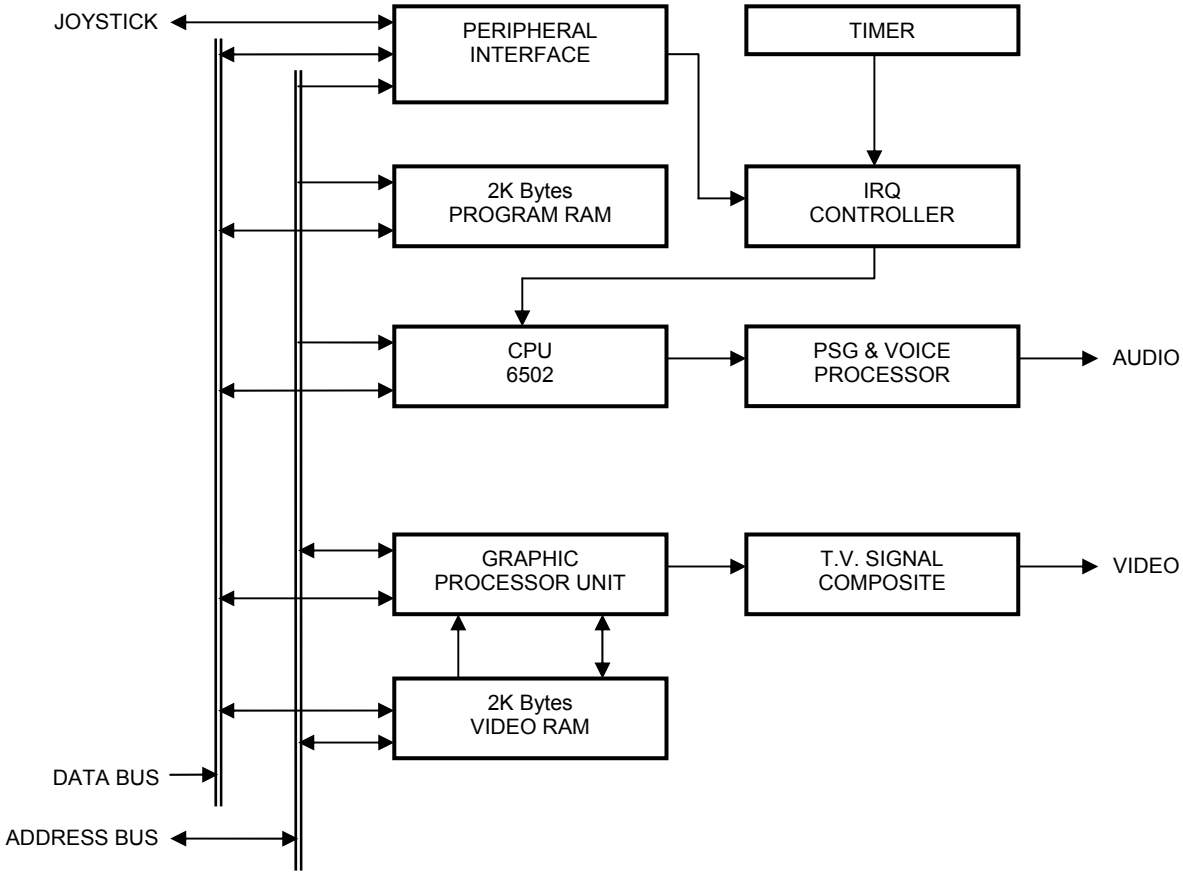
XJOYSEL=0 AND XTFTSEL=0



XJOYSEL=1 AND XTFTSEL=1



Block diagram



Pin Description

SYMBOL	TYPE	DESCRIPTIONS
XA[15:0]	O	CPU address bus.
XD[7:0]	I/O	CPU data bus.
XCK18	O	CPU clock 1.8MHz.
XRW	O	CPU Read/Write signal.
XROMCS	O	ROM chip select signal.
XIRQB	I	CPU interrupt input signal.(PH)
XVRW	O	Video Read/Write signal.
XVOE	O	Video output enable.
XRA10	I	Internal Video RAM address bit 10.
XRC	O	External ROM chip selector low active.
XAD[12:8],VA[7:0]	O	Video address bus.
XAD[7:0]	I/O	Video data bus.
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 interface output pins. (XQ2,XQ0:PH)
XVOE2	O	Power On signal, default high, low if $2001(D6)=1 or no any pushed for 5 minutes.
XVIDEO	O	Composited video signal.
XOP2	O	Audio signal.
XTRANEN	I	STN LCD Reflection(0) or Transparency(1).
XDISPON	O	LCD enable pin. XDISPON=1 -> enable LCD.
XJOYAM , XJOYBM	I	I/O interface.(PH)
XJOYSE , XJOYST	I	I/O interface.(PH)
XJOYLFA , XJOYRTA	I	I/O interface.(PH)
XJOYUPA , XJOYDNA	I	I/O interface.(PH)
XFMOUT	O	M signal of STN LCD driver.
XVDI	O	The first line signal of STN LCD driver.
XLOAD	O	Line clock of STN LCD driver.
XCPO	O	Dot clock of STN LCD driver.
XSEG [3:0]	O	Segment Data of STN LCD.
XTFTSEL	I	TFT LCD selector. TFT selected when XTFTSEL=1.
XJOYSEL	I	Internal Joystick enable when XJOYSEL=1. (PH)
XINH	O	Toggle enable for source driver of TFT LCD driver.
XVR	O	R signal of RGB output for TFT LCD Driver.
XVG	O	G signal of RGB output for TFT LCD Driver.
XVB	O	B signal of RGB output for TFT LCD Driver.
XDIO	O	Vertical start pulse of TFT LCD Driver.
XXOE	O	Output enable for gate driver of TFT LCD Driver.
XSTH	O	Start pulse for source driver of TFT LCD Driver.
XCP[3:1]	O	Sampling and shift clock for source driver of TFT LCD Driver.
XQ2H	O	Video input rotation control of TFT LCD Driver.
XVCOM	O	Common electrode voltage control of TFT LCD Driver.
XCPV	O	Clock input for gate driver of TFT LCD Driver.
XRW404X		R/W port 404X. Low active.
XLCDSBC		BW of RC check board selector for STN panel. 0: BW, 1: RC.
XCOLSEL		Color compitable for RC check board new color for STN panel. 1: Not compatible , 0: Compatible.
XDIVF2,XPORN,XF5OR6	O	TV system selector. ALL 0: NTSC ; ALL 1: PAL (PH)

Note1: (I) Input pin ; (O) Output pin ; (I/O) Input/Output pin ; (PH) Pull high resistor 20K~50K inside.

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. 28 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.
11. 8 address ports only.

Sound:

1. Providing 256 bytes DMA function for graphic unit.
2. Two address ports, 8 bits for reading peripheral I/O.
3. One address port, 3 bits for controlling peripheral I/O.
4. One port for reading the status of sound generator.
5. PSG gets 18 address ports to control its operation.
6. Every sound channel gets 4 address ports to control its operation.
7. There are 2 Rhythm channels, 2 specail sound channels, a data waveform synthesizer.
8. One 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.

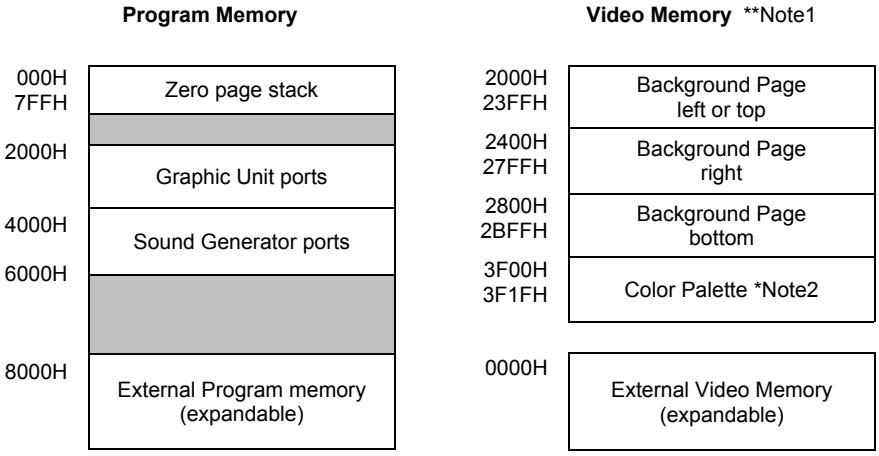
Internal RAM:

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. STN and TFT LCD interface built in.

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.

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.

VT01 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 bits color address which point the 28x6 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 = (1, 1) 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.

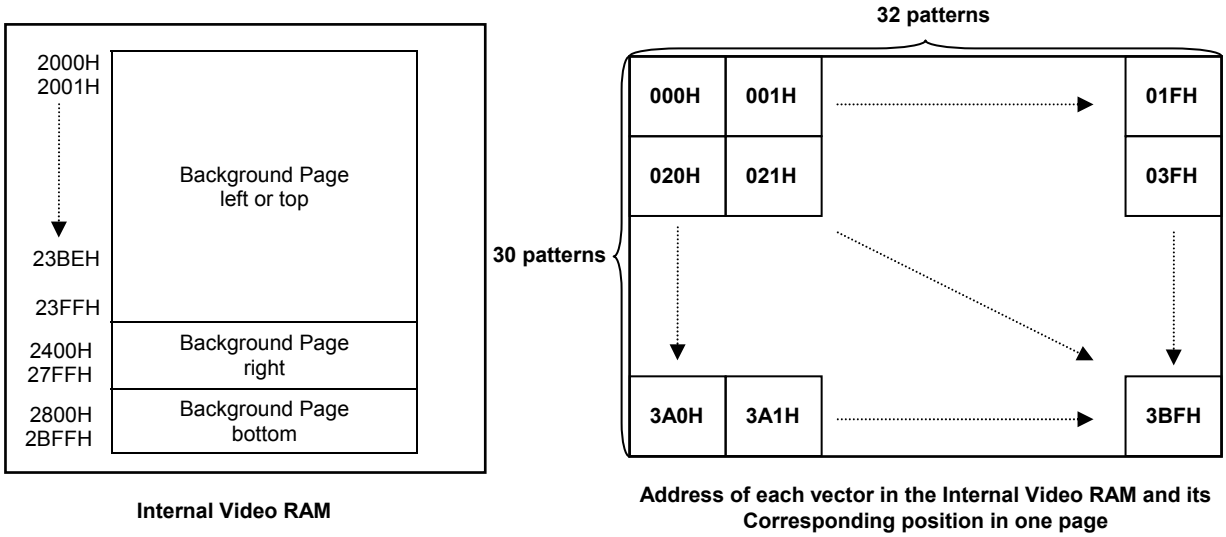
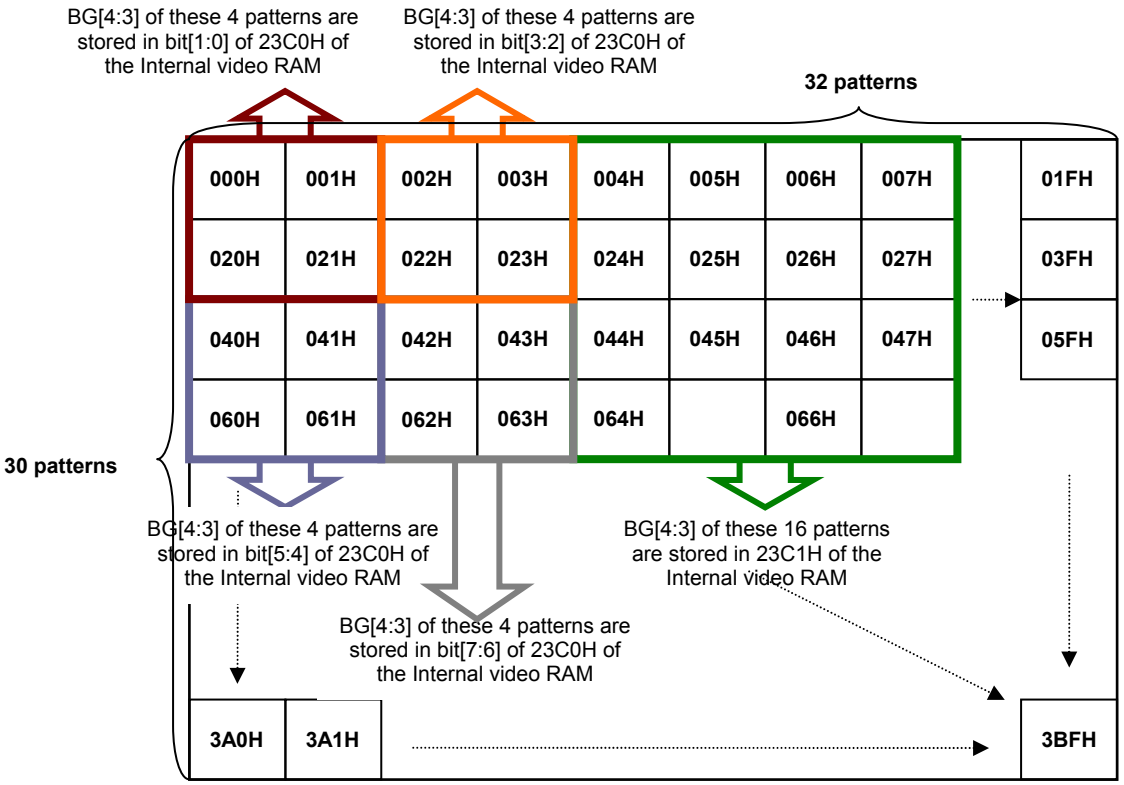


Figure B1. Mappings between Screen and Internal Video RAM



Address of each vector in the Internal Video RAM and its Corresponding position in one page

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

Two page for Background display

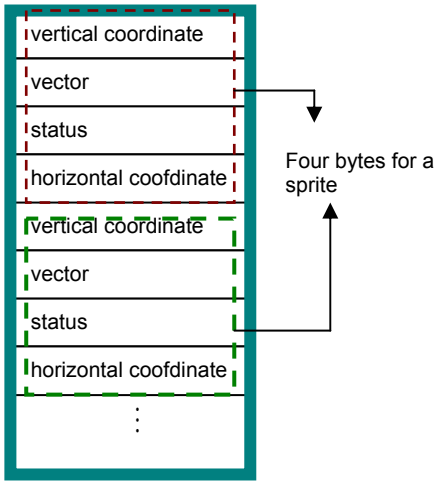
2Kbytes of internal video RAM are divided into 2 pages to moving screen effectively. Screen can be moved by horizontal or vertical way, that decided by every game card. In horizontal scroll, the AD10 of VIDEO and the A10 of the 2K RAM will be connected in game card. In vertical scroll, the AD11 of VIDEO and the A10 of the 2K RAM will be connected in game card. 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. 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
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.



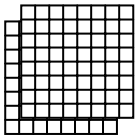
The Sprite Pool

Sprite Color and Size

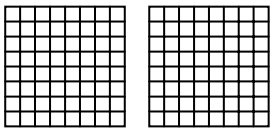
Programmers can choose the size and color mode of the sprite through 2000H and 2001H . You have the following options:

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

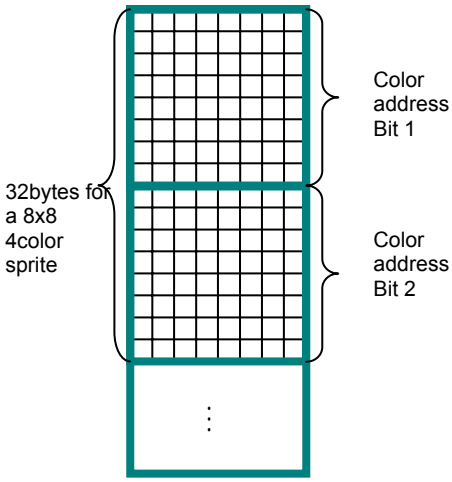
In 4 color mode it needs two bits for a pixel. Take size 8x8 in 4 color mode as an example. One sprite pattern in the external video RAM is arranged as the following figure.



Sprite 8x8 in 4 color mode



Sprite 8x16 in 4 color mode



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. The following is the color mapping of the data D5-D0. D4-D5 will be corresponding to the R color of LCD panel and D2-D3 is to the C color.

	MSB LSB	Dark			Bright Red
R	(D4-D5)	0	1	2	3
	MSB LSB	Dark			Bright Cyan
C	(D2-D3)	0	1	2	3

Register Description

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

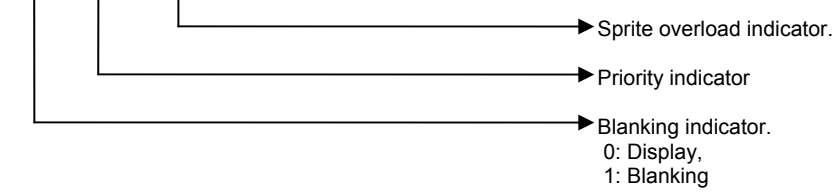
2001H W Blue/Green/Red intensity video output control ,Sprite enable/disable, Background enable/disable, Sprite initial coordinate, Background initial coordinate, B/W or color control

D7	D6	D5	D4	D3	D2	D1	D0
BLUE	GREEN	RED	SP EN	BK EN	SP INI	BK INI	B/W

- B/W: B/W or Color control.
  0: Color,
  1: B/W
- BK INI: Background initial coordinate.
  0: Righter,
  1: Lefter
- SP INI: Sprite initial coordinate.
  0:Righter,
  1:Lefter
- BK EN: Background enable.
  0: DISABLE,
  1: ENABLE
- SP EN: Sprite enable.
  0: DISABLE,
  1: ENABLE
- RED: Red Intensity video output control
  0: Normal ; 1: Intensity
- GREEN: Green Intensity video output control
  0: Normal ; 1: Intensity
- BLUE: Blue Intensity video output control
  0: Normal ; 1: Intensity

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

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



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  ;
LDA  $2007  ;
```

200BH W Option of Vertical line number of LCD display,Enable the internal VRAM or not

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

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.

Sound Generator

Sound Generator XOP 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

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

Miscellaneous Address Port .

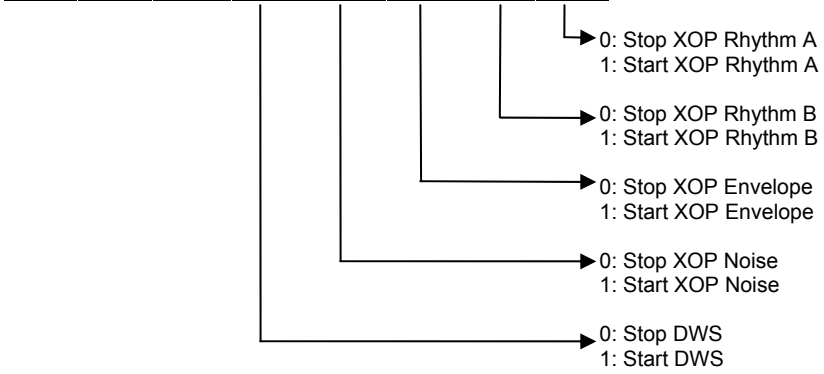
4014H W High byte Address of Source to start the DMA of Video data or Sprite data

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.

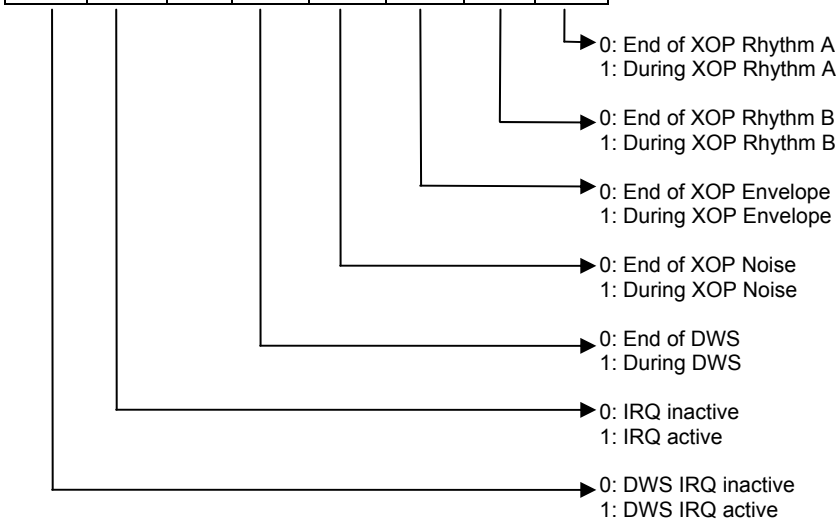
4015H W Enable/ disable XOP & DWS IRQ

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



4015H R Read XOP FLAG

D7	D6	D5	D4	D3	D2	D1	D0
DWS IRQ Flag	Clock IRQ flag		DWS Status	XOP Noise Status	XOP Envelope Status	XOP R. B Status	XOP R. A Status



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

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

- XQ0 → Set output pin XQ0
- XQ1 → Set output pin XQ1
- XQ2 → Set output pin XQ2

4016H R Read peripheral data

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

- Major joystick → Read the data from major joystick
- Floppy → Read the data from floppy
- Micro-Phone → 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						

- Clock IRQ Enable → 0: Enable Clock IRQ (60Hz)
1: Disable Clock IRQ
- BLCK1/ BLCK2 → 0: BLCK1=250Hz and BLCK2=120Hz
1: BLCK1=200Hz and BLCK2=100Hz

4017H R Read Peripheral Data

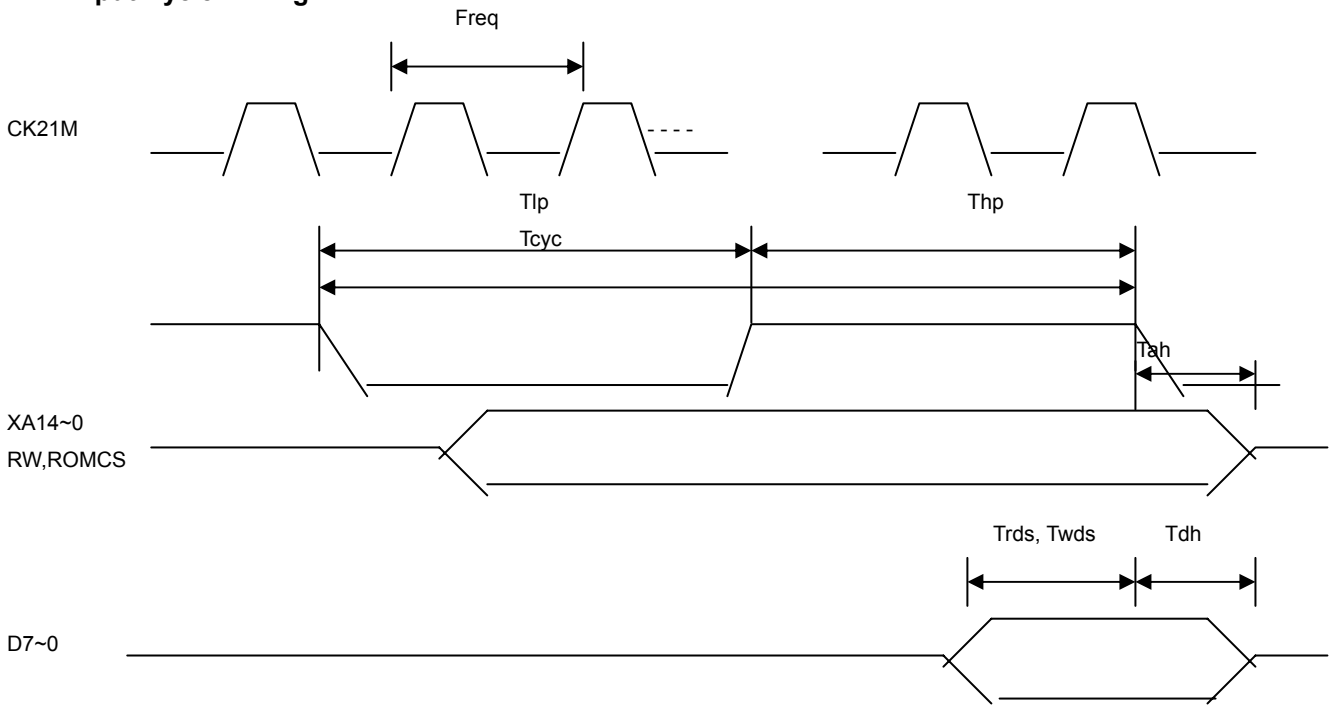
D7	D6	D5	D4	D3	D2	D1	D0
			Floppy Disk	Floppy Disk	Floppy Disk	Floppy Disk	Second Joystick

- Second Joystick → Read the data from second joystick
- Floppy Disk → Read the data from floppy
- Floppy Disk → Read the data from floppy
- Floppy Disk → Read the data from floppy
- Floppy Disk → Read the data from floppy

Timing Waveforms

Timing Spec. of Program Unit In Application Mode

Input Cycle Timing

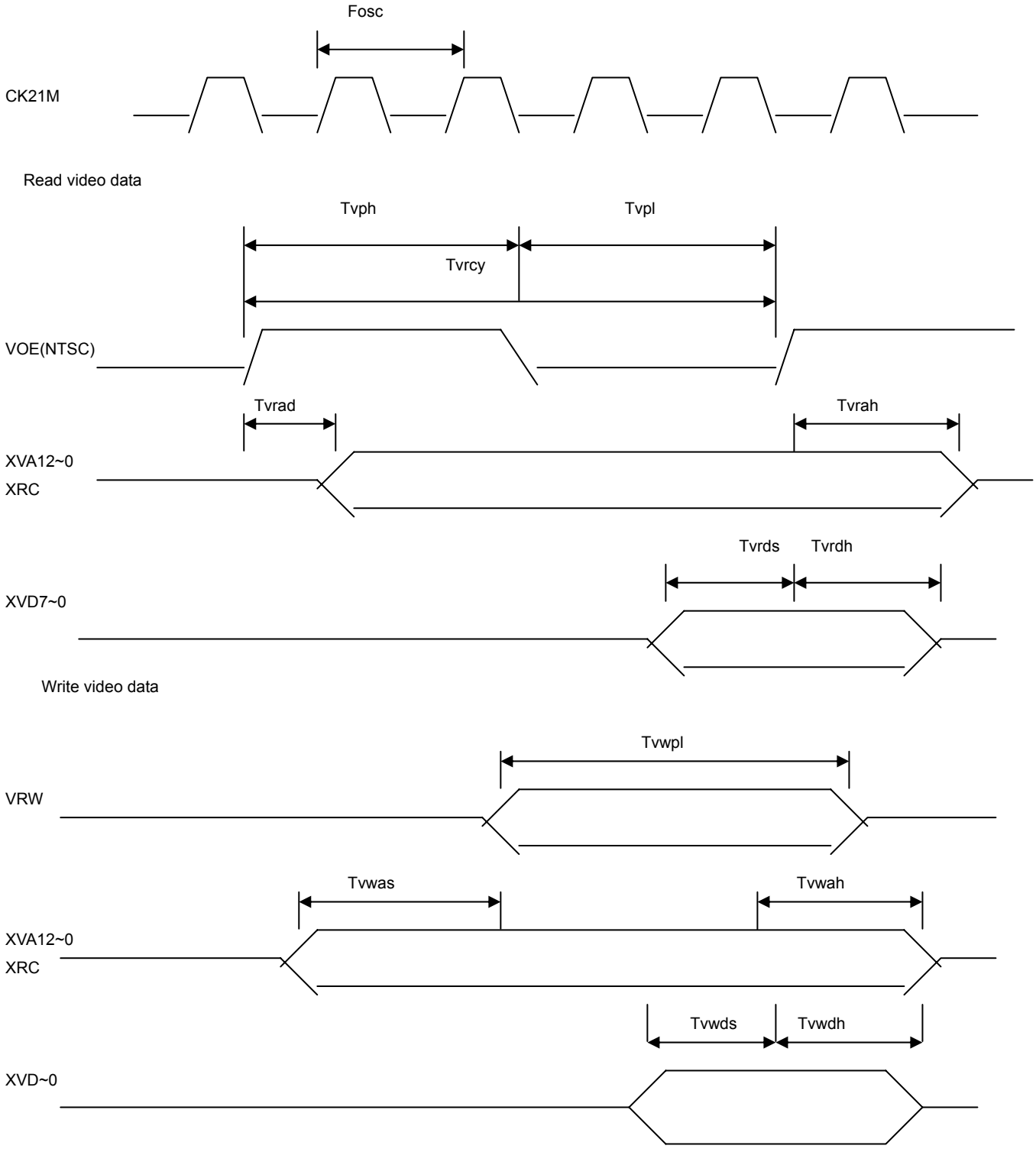


AC Characteristics : TA = 0`C to 70`C, VCC = 4.75V ~ 5.25V, GND = 0V

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	380	450	ns	
Tph	Cycle High Pulse Width	240	300	ns	
Tpl	Cylce Low Pulse Width	100	150	ns	
Tah	Program Address Hold time	15	70	ns	
Tdh	Program Data Hold time	15	225	ns	
Trds	Program Read Data Set up time	75		ns	
Twds	Program Write Data Set up time	112		ns	

Timing Spec of Graphic Unit In Application Mode

Input Cycle Timing



AC Characteristics: TA = 0`C to 70`C, VCC = 4.75V ~ 5.25V, 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	255	285	ns	
Tvph	Video Read High Pulse Width	127	150	ns	
Tvpl	Video Read Low Pulse Width	127	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	30		ns	
Tvrdh	Video Read Data Set up time	0		ns	
Tvwpl	Video Write Pulse time	127	150	ns	
Tvwas	Video Write Address Set up time	75		ns	
Tvwah	Video Write Address Hold time	45	90	ns	
Tvwds	Video Write Data Set up time	36	70	ns	
Tvwdh	Video Write Data Hold time	30	90	ns	

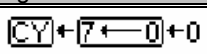
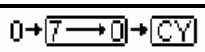
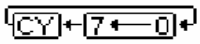
DC Characteristics : TA = 0`C to 70`C, VCC = 4.75V ~ 5.25V, GND = 0V

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	

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	●●●1●1●●	00	1	7
			B ← 1, I←1				
			(S)←PCH,S←S-1				
			(S)←PCL,S←S-1				
			(S) ← P ,S←S-1				
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