APPLICATION NOTE

(DOC No. HX8352-A(T)-AN)

HX8352-A(T)

240RGB x 480 dot, 262K color, with internal GRAM, TFT Mobile Single Chip Driver

Preliminary version 01 February, 2008

Himax Technologies, Inc.
http://www.himax.com.tw

List of Contents

February, 2008

List of Figures

List of Figures

List of Table

1. Introduction

This document describes Himax's HX8352-A is supports four types resolution driving controller. The HX8352-A is designed to provide a single-chip solution that combines a gate driver, a source driver, power supply circuit for 262,144 colors to drive a TFT panel with 240RGBx480 dots at maximum.

The HX8352-A can be operated in low-voltage (1.65V) condition for the interface and integrated internal boosters that produce the liquid crystal voltage, breeder resistance and the voltage follower circuit for liquid crystal driver. In addition, the HX8352-A also supports various functions to reduce the power consumption of a LCD system via software control.

The HX8352-A is suitable for any small portable battery-driven and long-term driving products, such as small PDAs, digital cellular phones and bi-directional pagers.

The HX8352-A supports three interface modes, include Command-Parameter interface mode, Register-Content interface mode and MDDI (Mobile Display Digital Interface) interface mode. The interface mode is selected by the external pins IFSEL0, P68, BS2~0 setting.

2. HX8352-A Chip Block Diagram

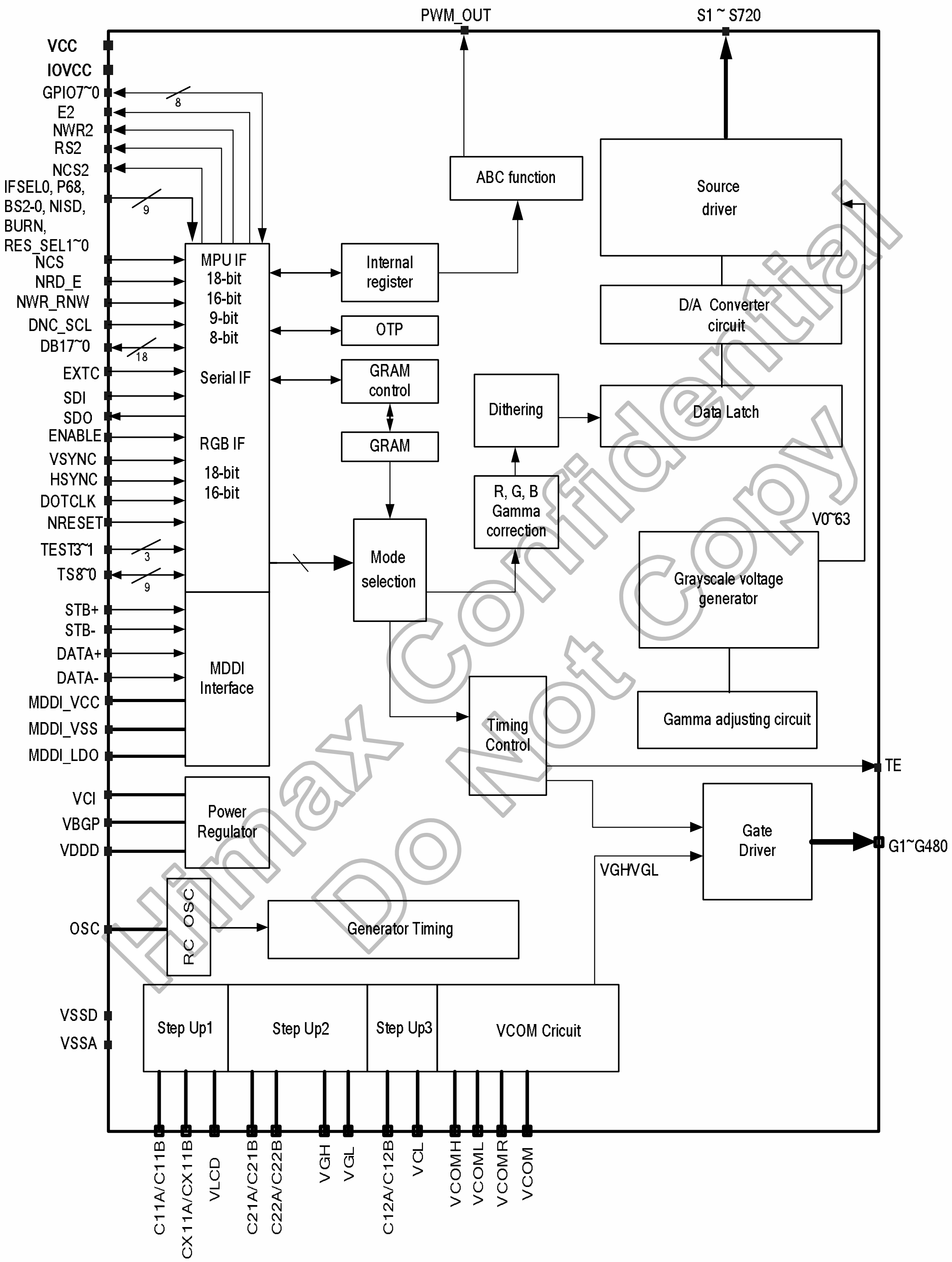


Figure 2. 1 HX8352-A block diagram

3. HX8352-A PAD Assignment

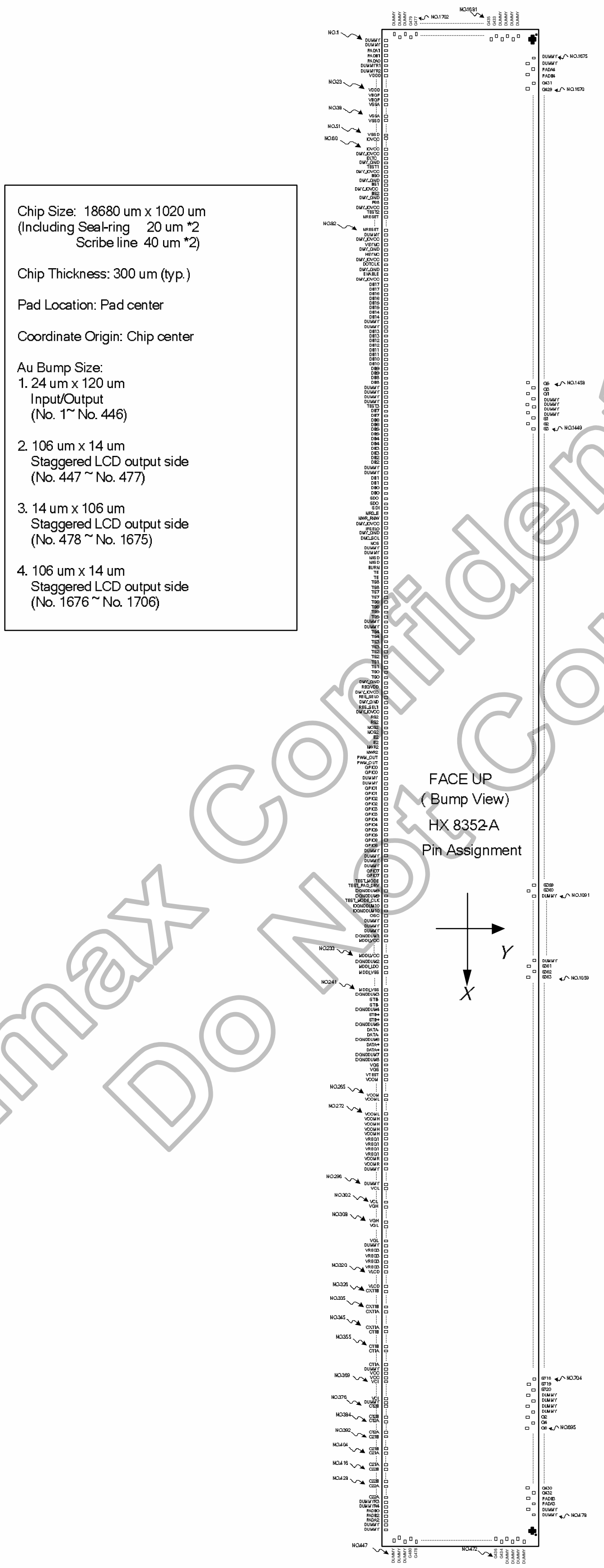
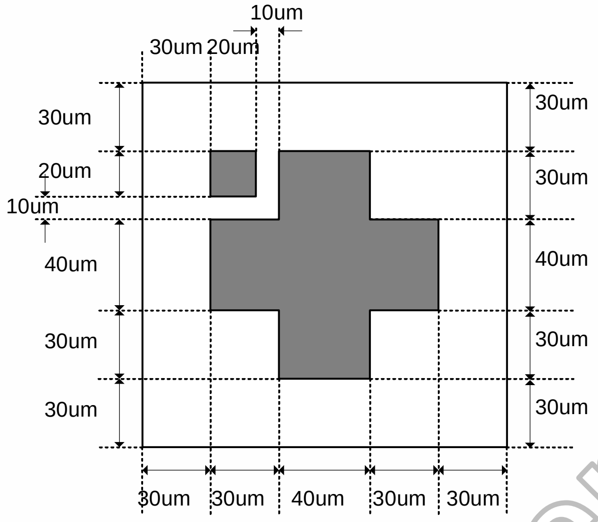


Figure 3. 1 HX8352-A pad assignment

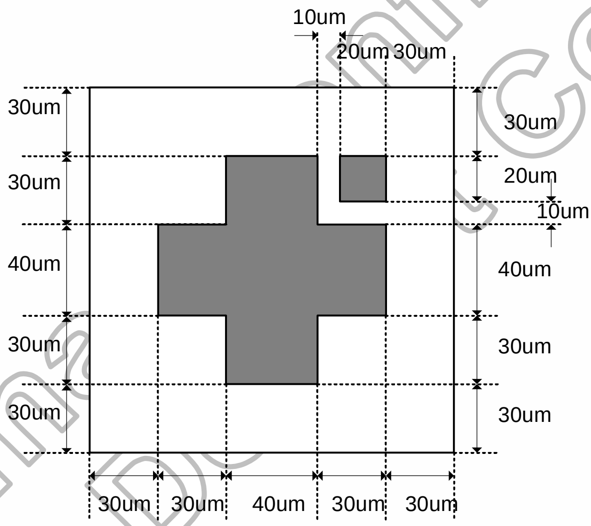
3.1 Alignment mark

A_MARK (A1)



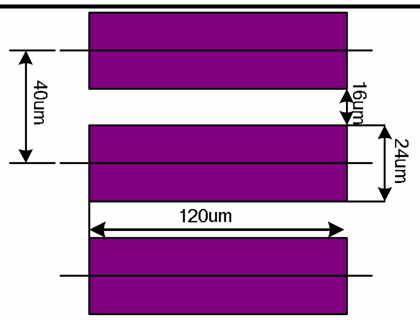
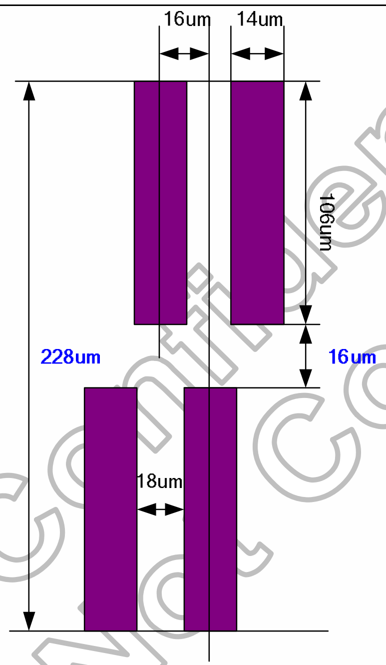
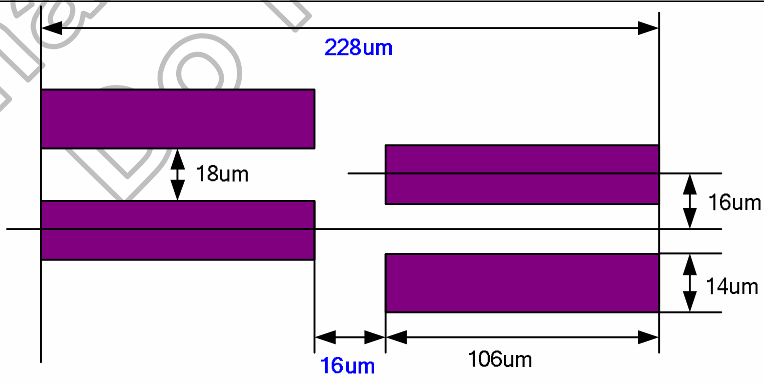
A1 (-9200,+370)

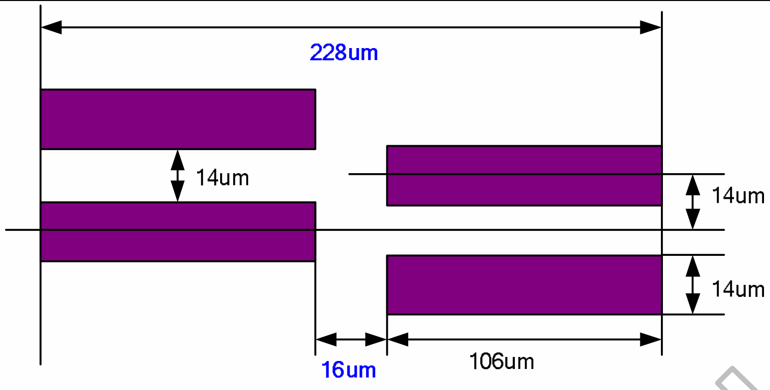
A_MARK (A2)



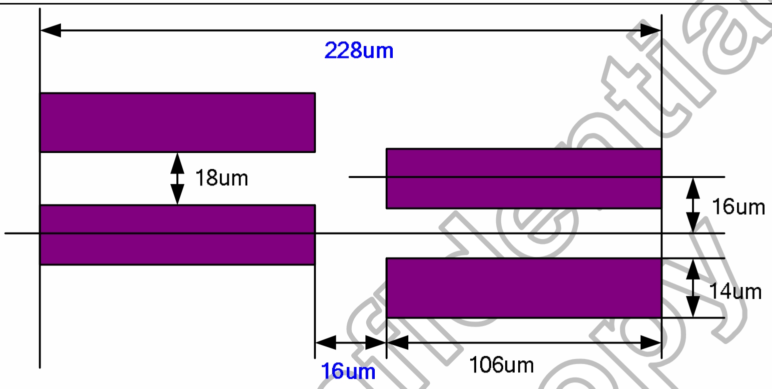
A2 (+9200,+370)

3.2 Bump size

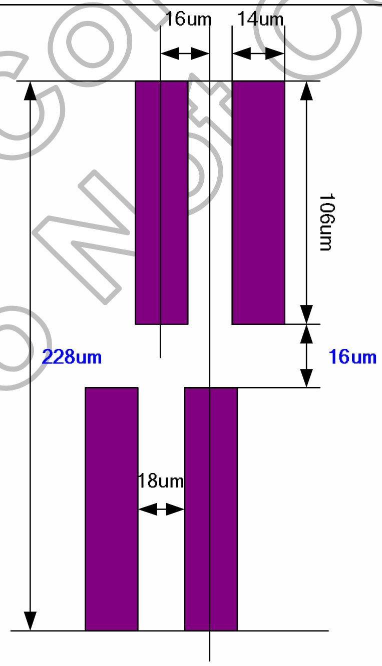
Input PAD	 NO.1~NO.446
Output PAD	 NO.447~ NO.477
	 NO.478~NO.699



NO.700~NO.1453



NO.1454~NO.1675



NO.1676~ NO.1706

4. Pin Description

<table>
<tr><th colspan="5">Input Parts</th></tr>
<tr><th>Signals</th><th>I/O</th><th>Pin Number</th><th>Connected with</th><th>Description</th></tr>
<tr><td>P68, BS2,BS1,BS0</td><td>I</td><td>4</td><td>VSSD/ IOVCC</td><td>Select the MPU interface mode as listed below
Use with IFSEL0=1 Register-content interface mode or MDDI interface mode.
<table>
<tr><th>P68</th><th>BS2</th><th>BS1</th><th>BS0</th><th>Interface mode</th><th>DB pins</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>16-bit bus interface, 80-system, 65K-Color</td><td>D17-D16: Unused, D15-D0: Data</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>16-bit bus interface, 80-system, 262K-color</td><td>D17-D16: Unused, D15-D0: Data</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>18-bit bus interface, 80-system, 262K-color</td><td>D17-D0: Data</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>8-bit bus interface, 80-system, 262K-Color</td><td>D17-D8: Unused D7-D0: Data</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>8-bit bus interface, 80-system, 65K-Color</td><td>D17-D8: Unused D7-D0: Data</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>16-bit bus interface, 68-system, 65K-Color</td><td>D17-D16: Unused, D15-D0: Data</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>16-bit bus interface, 68-system, 262K-color</td><td>D17-D16: Unused, D15-D0: Data</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>18-bit bus interface, 68-system, 262K-Color</td><td>D17-D0: Data</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>8-bit bus interface, 68-system, 262K-color</td><td>D17-D8:Unused D7-D0: Data</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>8-bit bus interface, 68-system, 65K-color</td><td>D17-D8:Unused D7-D0: Data</td></tr>
<tr><td>X</td><td>1</td><td>0</td><td>1</td><td>MDDI interface.</td><td>STB+, STB-, DATA+, DATA-</td></tr>
<tr><td>X</td><td>1</td><td>1</td><td>ID</td><td>Serial bus IF + RGB interface</td><td>DNC_SCL, SDO,SDI, VSYNC, HSYNC, ENABLE, DOTCLK, DB17-0</td></tr>
</table></td></tr>
<tr><td>IFSEL0</td><td>I</td><td>1</td><td>MPU</td><td>Interface format select pin
<table>
<tr><th>IFSEL0</th><th>Interface Format Selection</th></tr>
<tr><td>0</td><td>Command-Parameter interface mode</td></tr>
<tr><td>1</td><td>Register-content interface mode or MDDI interface mode</td></tr>
</table></td></tr>
<tr><td>EXTC</td><td>I</td><td>1</td><td>MPU</td><td>Extended command set enable. (Only support Command-Parameter Interface mode à IFSEL0=0)
Low: extended command set is discarded
High: extended command set is accepted
If operate in Register-content interface mode, the EXTC can be connected to IOVCC or VSSD.</td></tr>
<tr><td>RES_SEL1~0</td><td>I</td><td>2</td><td>MPU</td><td>Panel Resolution select pin.
<table>
<tr><th>RES_SEL1</th><th>RES_SEL0</th><th>Panel Resolution</th></tr>
<tr><td>0</td><td>0</td><td>240RGB x 320 dot</td></tr>
<tr><td>0</td><td>1</td><td>240RGB x 400 dot</td></tr>
<tr><td>1</td><td>0</td><td>240RGB x 432 dot</td></tr>
<tr><td>1</td><td>1</td><td>240RGB x 480 dot</td></tr>
</table></td></tr>
<tr><td>NCS</td><td>I</td><td>1</td><td>MPU</td><td>Chip select signal.
Low: chip can be accessed;
High: chip cannot be accessed. Must be connected to VSSD if not in use.</td></tr>
<tr><td>NWR_RNW</td><td>I</td><td>1</td><td>MPU</td><td>I80 system: Serves as a write signal and writes data at the rising edge.
M68 system: 0: Write, 1: Read.
Fix it to IOVCC or VSSD level when using serial buss interface.</td></tr>
<tr><td>NRD_E</td><td>I</td><td>1</td><td>MPU</td><td>I80 system: Serves as a read signal and read data at the low level.
M68 system: 0: Read/Write disable, 1: Read/Write enable.
Fix it to IOVCC or VSSD level when using serial buss interface.</td></tr>
</table>

Input Parts				
Signals	**I/O**	**Pin Number**	**Connected with**	**Description**
DNC_SCL	I	1	MPU	The signal for command or parameter select under parallel mode(i.e. Not serial interface): Low: command. High: parameter. When under serial interface, it servers as SCL.
BURN	I	1	MPU	Free Running mode If BURN=Hi, this can enable free running mode for burn in test. The display data alternates between full black and full white independent of input data in free running mode.
SDI	I	1	MPU	Serial data input. If not used, please let it connected to IOVCC or VSSD.
VSYNC	I	1	MPU	Frame synchronizing signal. Has to be fixed to IOVCC level if is not used.
HSYNC	I	1	MPU	Frame synchronizing signal. Has to be fixed to IOVCC level if is not used.
ENABLE	I	1	MPU	A data ENABLE signal in RGB I/F mode. Has to be fixed to VSSD level if unused (High active, if EPL=0).
DOTCLK	I	1	MPU	Dot clock signal. Has to be fixed to VSSD level if is not used.
NRESET	I	1	MPU or reset circuit	Reset pin. Setting either pin low initializes the LSI. Must be reset after power is supplied.
OSC	I	1	Oscillation Resistor	Oscillator input for test purpose. If not used, please let it open or connected to VSSD.
VCOMR	I	1	Resistor or open	A VcomH reference voltage. When adjusting VcomH externally, set registers to halt the VcomH internal adjusting circuit and place a variable resistor between VREG1 and VSSD. Otherwise, leave this pin open and adjust VcomH by setting the internal register of the HX8352-A.
VGS	I	1	VSSD or external resistor	Connect to a variable resistor to adjusting internal gamma reference voltage for matching the characteristic of different panel used.

Output Part				
Signals	**I/O**	**Pin Number**	**Connected with**	**Description**
S1~S720	O	720	LCD	Output voltages applied to the liquid crystal.
G1~G480	O	480	LCD	Gate driver output pins. These pins output VGH, VGL.(If not used, should be open)
VCOM	O	1	TFT common electrode	The power supply of common voltage in TFT driving. The voltage amplitude between VCOMH and VCOML is output. Connect this pin to the common electrode in TFT panel.
TE	O	1	MPU	Tearing effect output. If not used, please open this pin.
SDO	O	1	MPU	Serial data output. If not use, let it to open.
NISD	O	1	Open	Image Sticking Discharge signal. This pin is used for monitoring image sticking discharge phenomena. When the NISD goes low, the VGL, Source and VCOM would be discharged to VSSA. When the NISD goes high, the VGL, Source and VCOM are normal operation.
PWM_OUT	O	1	-	Backlight On/Off control pin. If use ABC function, the pin can connect to external LED driver IC. The output voltage rage = 0~ IOVCC.
NWR2	O	1	Sub Panel	80-interface NWR signal output pin for Sub Panel
E2	O	1	Sub Panel	68-interface Enable signal output pin for Sub Panel
NCS2	O	1	Sub Panel	The signal is Chip select for Sub Panel.
RS2	O	1	Sub Panel	The signal is register index or register parameter select for Sub Panel

MDDI Interface Parts				
Signals	**I/O**	**Pin Number**	**Connected with**	**Description**
STB+ STB-	-	2	MDDI Host	MDDI Strobe differential signal input pins. STB+ pin for Strobe+, STB- pin for Strobe-. Connect to a terminal resistance (100Ω) between STB+ and STB-.
DATA+ DATA-	-	2	MDDI Host	MDDI Data differential signal input pins. DATA+ pin for Data+, DATA- pin for Data-. Connect to a terminal resistance (100Ω) between DATA+ and DATA-.
MDDI_VCC	P	1	Power Supply	MDDI I/O power supply pin, 2.5V~3.3V.
MDDI_VSS	P	1	Ground	MDDI I/O ground pin.
MDDI_LDO	O	1	Capacitor	MDDI regulator output pin. Connect to a stabilizing capacitor between MDDI_VSS and MDDI_LDO

Input/Output Part				
Signals	**I/O**	**Pin Number**	**Connected with**	**Description**
C11A,C11B C12A,C12B	I/O	4	Step-up Capacitor	Connect to the step-up capacitors according to the step-up factor. Leave this pin open if the internal step-up circuit is not used.
CX11A, CX11B	I/O	2	Step-up Capacitor	Connect to the step-up capacitors for step up circuit 1 operation. Leave this pin open if the internal step-up circuit is not used.
C21A,C21B C22A,C22B	I/O	4	Step-up Capacitor	Connect these pins to the capacitors for the step-up circuit 2. According to the step-up rate. When not using the step-up circuit2, disconnect them.
DB17~0/ PD17~0 (DBS17~0)	I/O	18	MPU	When Operates in system interface mode, it is used liked an 18-bit bi-directional data bus. 8-bit bus: use DB7-DB0 16-bit bus: use DB15-DB0 18-bit bus: use DB17-DB0 When Operation in RGB interface mode, it is an 18-bit bus RGB data bus. 16-bit bus: use PD15-PD0 18-bit bus: use PD17-PD0 If no used, please open the pins. If use MDDI interface, these pins are sub panel data bus (DBS17~DBS0). If no used, please open these pins.
GPIO7~0	I/O	8	-	Standard Input/Output pin As for GPIO7 to 0 terminal, setting of an input and output direction is possible. If no used, please open these pins.

Power Part				
Signals	**I/O**	**Pin Number**	**Connected with**	**Description**
IOVCC	P	1	Power Supply	Digital IO Pad power supply, 1.65V~3.3V
VCC	P	1	Power Supply	Digital power supply, 2.3V~3.3V
VCI	P	1	Power Supply	Analog power supply, 2.3V~3.3V
VSSD	P	1	Ground	Digital ground
VSSA	P	1	Ground	Analog ground
VDDD	O	1	Stabilizing Capacitor	Output from internal logic voltage (1.6V). Connect to a stabilizing capacitor

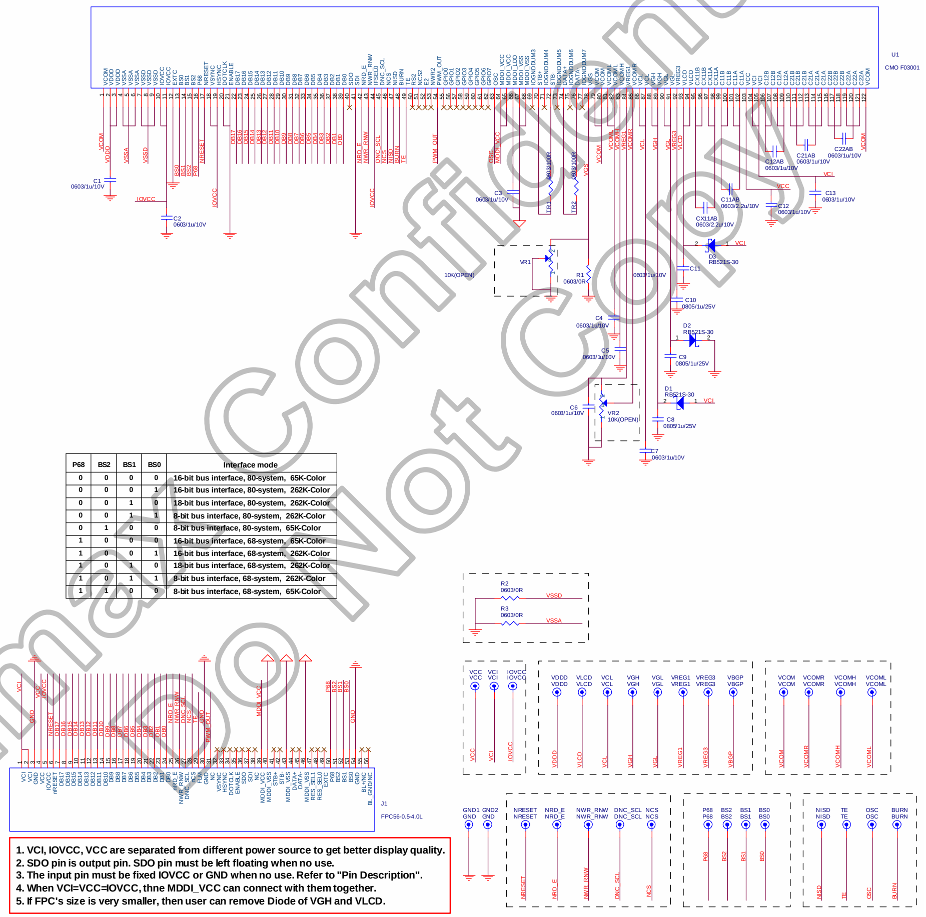
Power Part				
Signals	**I/O**	**Pin Number**	**Connected with**	**Description**
REGVDD	I	1	MPU	If REGVDD = high, the internal VDDD regulator will be turned on. If REGVDD = low, the internal VDDD regulator will be turned off, VDDD should connect to external power supply, the voltage range 1.65~1.95V. Must be connected to IOVCC or VSSD.
VBGP	-	1	Open	Band Gap Voltage. Let it to be open.
VREG1	P	1	Stabilizing Capacitor	Internal generated stable power for source driver unit.
VREG3	P	1	Stabilizing Capacitor	A reference voltage for VGH&VGL.
VCOMH	P	1	Stabilizing capacitor	Connect this pin to the capacitor for stabilization. This pin indicates a high level of VCOM amplitude generated in driving the VCOM alternation.
VCOML	P	1	Stabilizing capacitor	When the VCOM alternation is driven, this pin indicates a low level of VCOM amplitude. Connect this pin to a capacitor for stabilization.
VCL	P	1	Stabilizing capacitor	A negative voltage for VCOML circuit, VCL=-VCI
VLCD	P	1	Stabilizing capacitor	An output from the step-up circuit1. Connect to a stabilizing capacitor between VSSA and VLCD. Place a schotkey barrier diode (see “configuration of the power supply”).
VGH	P	1	Stabilizing capacitor	An output from the step-up circuit2.or 4 ~ 6 time the VCI level. The step-up rate is determined with BT2-0 bits. Connect to a stabilizing capacitor between VSSD and VGH. Place a schottkey barrier diode between VCI and VGH. Place a schottkey barrier diode (see “configuration of the power supply”).
VGL	P	1	Stabilizing capacitor	An output from the step-up circuit2.or –3~ -5 time the VCI level. The step-up rate is determined with BT2-0 bits. Connect to a stabilizing capacitor between VSSD and VGL. Place a schottkey barrier diode between VSSD and VGL. Place a schottkey barrier diode (see “configuration of the power supply”).

Test Pin and Others				
Signals	**I/O**	**Pin Number**	**Connected with**	**Description**
TEST3-1	I	3	GND	Test pin input (Internal pull low)
TS8~0	O	9	Open	A test pin. Disconnect it.
VTEST	O	1	Open	A test pin. Disconnect it.
TEST_MODE	I	1	Open	MDDI test pin. Must be left open.
TEST_PAD_DRV	I	1	Open	MDDI test pin. Must be left open.
TEST_MODE_CLK	I	1	Open	MDDI test pin. Must be left open.
DUMMYR1-2	-	2	Open	Dummy pads. Available for measuring the COG contact resistance. DUMMYR1 and DUMMYR2 are short-circuited within the chip.
DUMMYR3-4	-	2	Open	Dummy pads. Available for measuring the COG contact resistance. DUMMYR3 and DUMMYR4 are short-circuited within the chip.
DUMMY	-	112	Open	Dummy pads
IOGNDDUM1-10	O	10	Open	Dummy pin between MDDI pin, Leave them open.
PADA0,PADB0	I	2	MPU	Test pin for display glass break detection. If not used, please open these pins.
PADA1~ PADA4, PADB1~ PADB4	I	8	MPU	Test pin for chip attachment detection. If not used, please open these pins.
DMY_IOVCC	O	10	-	Dummy IOVCC output pads, Internal connected to IOVCC and only for external Hardware setting pin use. If not used, please open these pins.
DMY_GND	O	8	-	Dummy GND output pads, Internal connected to VSSD and only for external Hardware setting pin use. If not used, please open these pins.

5. HX8352-A Reference FPC circuit (For CMO 3.0" LCD Panel)

5.1 Register-content interface mode

5.1.1 MPU interface

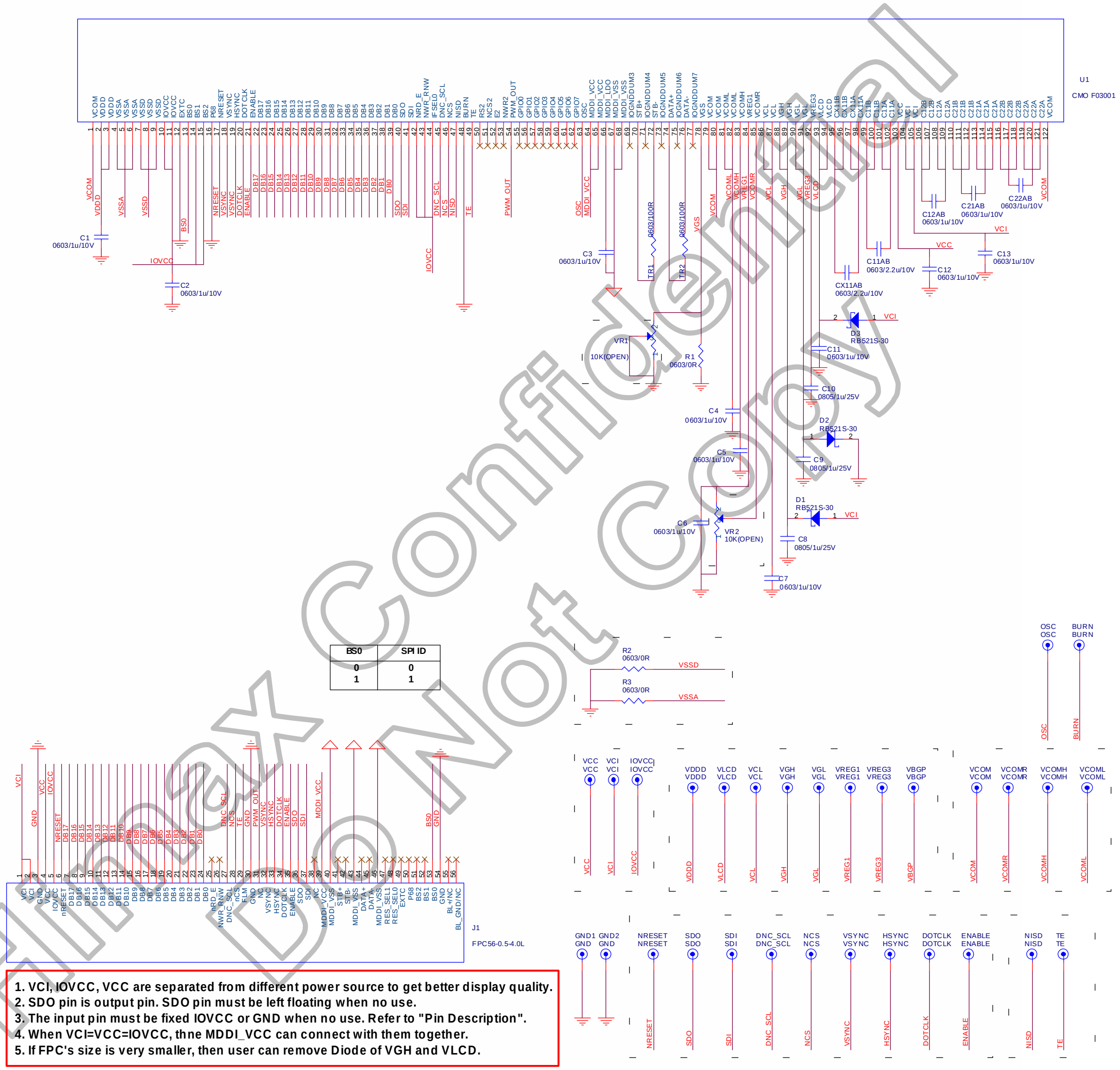


P68	BS2	BS1	BS0	Interface mode
0	0	0	0	16-bit bus interface, 80-system, 65K-Color
0	0	0	1	16-bit bus interface, 80-system, 262K-Color
0	0	1	0	18-bit bus interface, 80-system, 262K-Color
0	0	1	1	8-bit bus interface, 80-system, 262K-Color
0	1	0	0	8-bit bus interface, 80-system, 65K-Color
1	0	0	0	16-bit bus interface, 68-system, 65K-Color
1	0	0	1	16-bit bus interface, 68-system, 262K-Color
1	0	1	0	18-bit bus interface, 68-system, 262K-Color
1	0	1	1	8-bit bus interface, 68-system, 262K-Color
1	1	0	0	8-bit bus interface, 68-system, 65K-Color

1. VCI, IOVCC, VCC are separated from different power source to get better display quality.
2. SDO pin is output pin. SDO pin must be left floating when no use.
3. The input pin must be fixed IOVCC or GND when no use. Refer to "Pin Description".
4. When VCI=VCC=IOVCC, thne MDDI_VCC can connect with them together.
5. If FPC's size is very smaller, then user can remove Diode of VGH and VLCD.

Figure 5. 1 Reference FPC circuit of Register-content interface mode's MPU interface

5.1.2 RGB with Serial interface

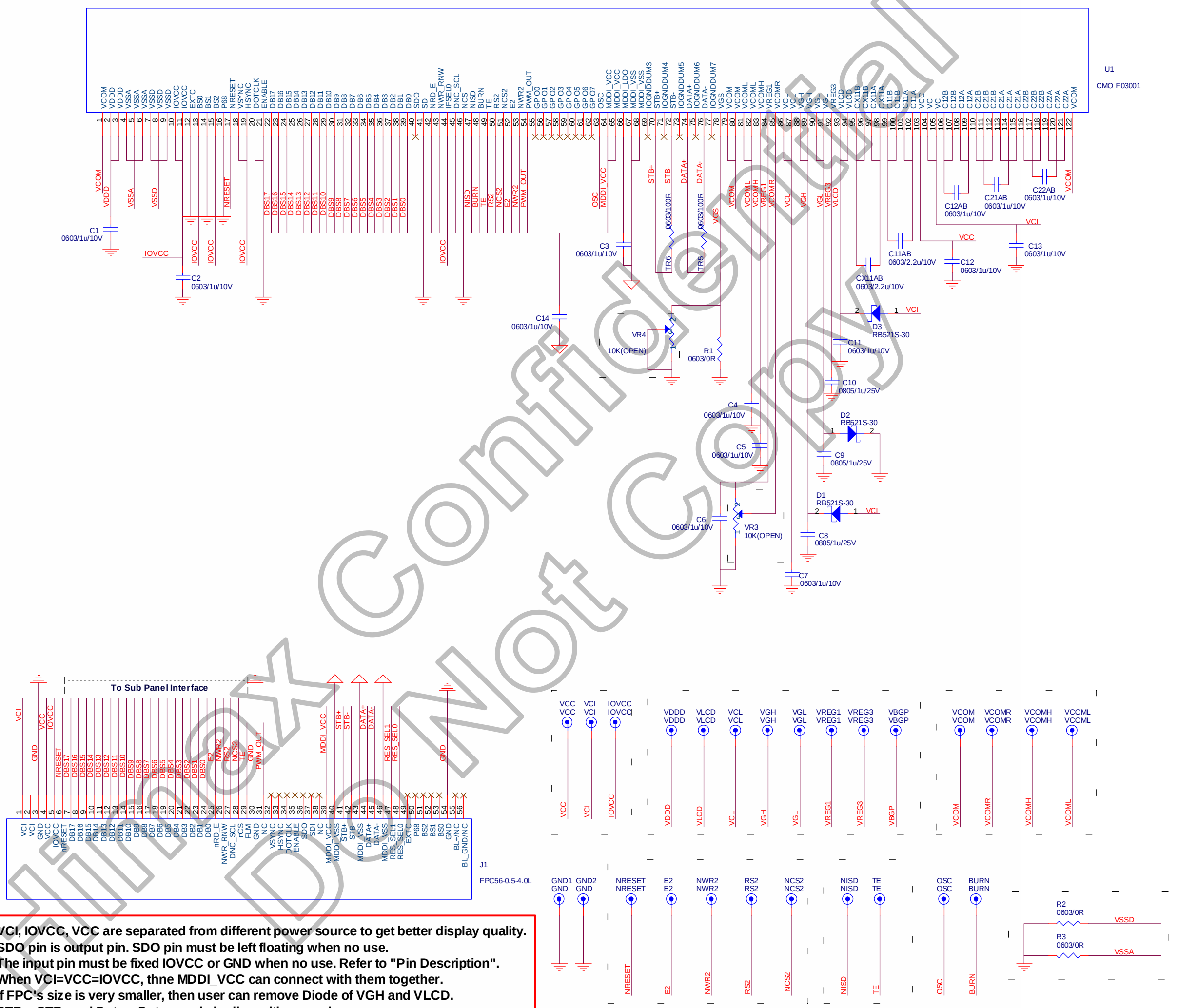


BS0	SPI ID
0 1	0 1

1. VCI, IOVCC, VCC are separated from different power source to get better display quality.
2. SDO pin is output pin. SDO pin must be left floating when no use.
3. The input pin must be fixed IOVCC or GND when no use. Refer to "Pin Description".
4. When VCI=VCC=IOVCC, thne MDDI_VCC can connect with them together.
5. If FPC's size is very smaller, then user can remove Diode of VGH and VLCD.

Figure 5. 2 Reference FPC circuit of Register-content interface mode's RGB interface

5.1.3 MDDI interface



1. VCI, IOVCC, VCC are separated from different power source to get better display quality.
2. SDO pin is output pin. SDO pin must be left floating when no use.
3. The input pin must be fixed IOVCC or GND when no use. Refer to "Pin Description".
4. When VCI=VCC=IOVCC, thne MDDI_VCC can connect with them together.
5. If FPC's size is very smaller, then user can remove Diode of VGH and VLCD.
6. STB+, STB- and Data+, Data- need shading with ground.
7. STB+, STB- and Data+, Data- need layout the same length.

Figure 5. 3 Reference FPC circuit of Register-content interface mode's MDDI interface

The specification of FPC circuit and pins connection is shown as following table:

Pad Name	Connection	Typical capacitance value (B characteristics)
VCOMH	Connect to Capacitor (Max 6V): VCOMH---(+)----| |--- (-)------ VSSA	1.0µF
VCOML	Connect to Capacitor (Max 3V): VCOML ---(-)----| |--- (+)----- VSSA	1.0µF
VGL	Connect to Capacitor (Max 16V): VGL ---(-)----| |--- (+)----- VSSA	1.0µF
VGH	Connect to Capacitor (Max 21V): VGH ---(+)----| |--- (-)----- VSSA	1.0µF
VCL	Connect to Capacitor (Max 5V): VCL ---(-)----| |--- (+)----- VSSA	1.0µF
C22A,C22B	Connect to Capacitor (Max 7V): C22A ---(+)----| |--- (-)-----C22B	1.0µF
C21A,C21B	Connect to Capacitor (Max 7V): C21A ---(+)----| |--- (-)-----C21B	1.0µF
CX11A,CX11B	Connect to Capacitor (Max 7V): CX11A ---(+)----| |--- (-)-----CX11B	2.2µF
C11A, C11B	Connect to Capacitor (Max 5V): C11A ---(+)----| |--- (-)-----C11B	2.2µF
C12A, C12B	Connect to Capacitor (Max 5V): C12A ---(+)----| |--- (-)-----C12B	1.0µF
VREG1	Connect to Capacitor (Max 6V): VREG1 ---(+)----| |--- (-)-----VSSA	1.0µF
VREG3	Connect to Capacitor (Max 16V): VREG3 ---(+)----| |--- (-)-----VSSA	1.0µF
VDDD	Connect to Capacitor (Max 6V): VDDD ---(+)----| |--- (-)-----VSSA	1.0µF
DDVDH	Connect to Capacitor (Max 6V): DDVDH ---(+)----| |--- (-)-----VSSA	1.0µF
VCI	Connect to Capacitor (Max 6V): VCI ---(+)----| |--- (-)-----VSSA	2.2µF
IOVCC	Connect to Capacitor (Max 6V): IOVCC ---(+)----| |--- (-)-----VSSA	1.0µF

Note: The aforementioned capacitor must be connected otherwise it will cause poor display quality.

Table 5. 1 Connected Capacitor

Pins connection	Feature
a. VCI – VLCD b. VCI – VGH c. VSSD – VGL	VF < 0.4V / 20mA at 25°C, VR ≥30V (Recommended diode: RB521S-30)

Table 5. 2 Connected Schottkey diode

6. LCD Power Generation

6.1 LCD Power Generation Scheme

The boost voltage generated is shown as below.

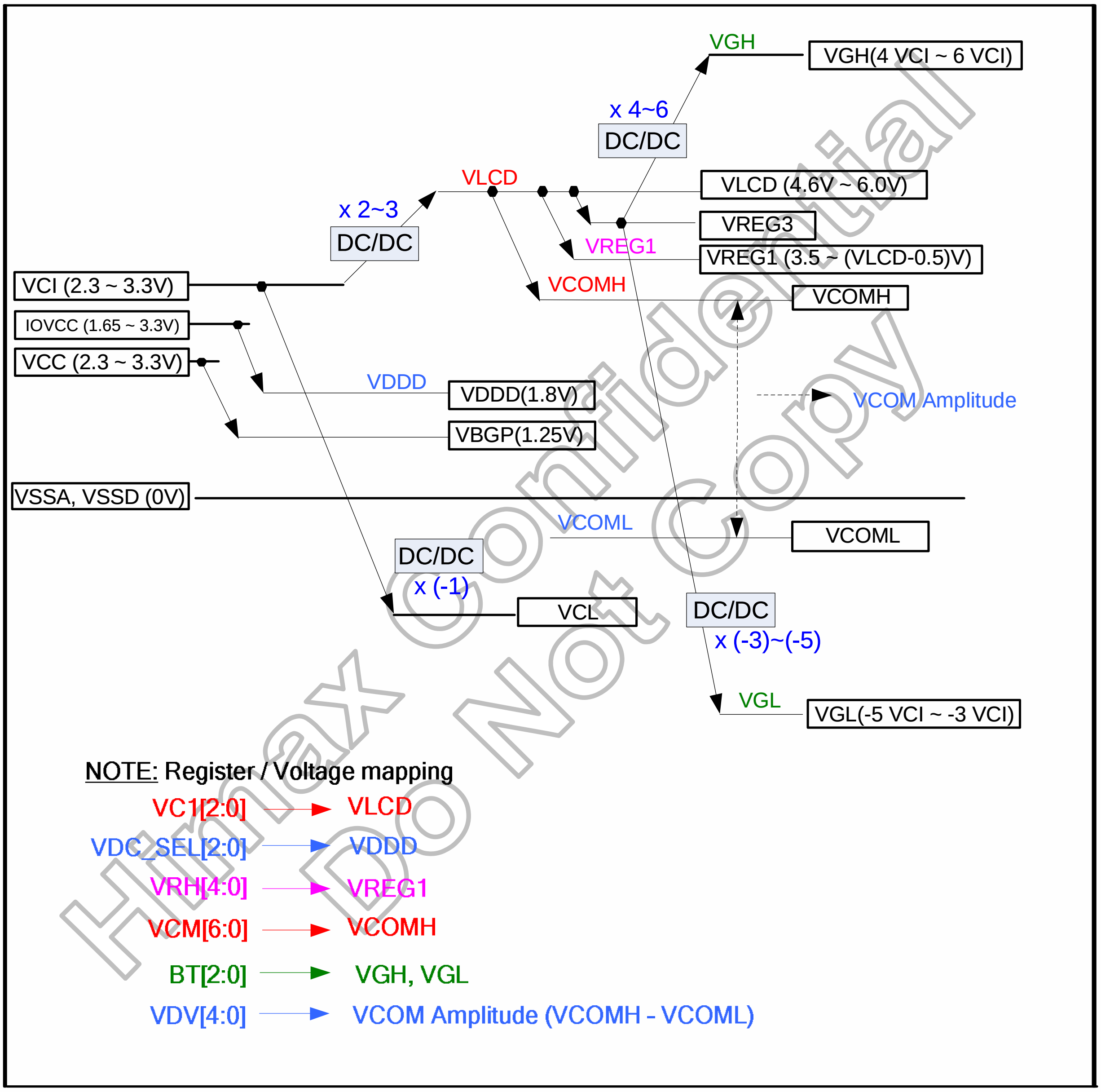


Figure 6. 1 LCD power generation scheme

6.2 Various Boosting Steps

The boost steps of each boosting voltage are selected according to how the external capacitors are connected. Different booster applications are shown as below.

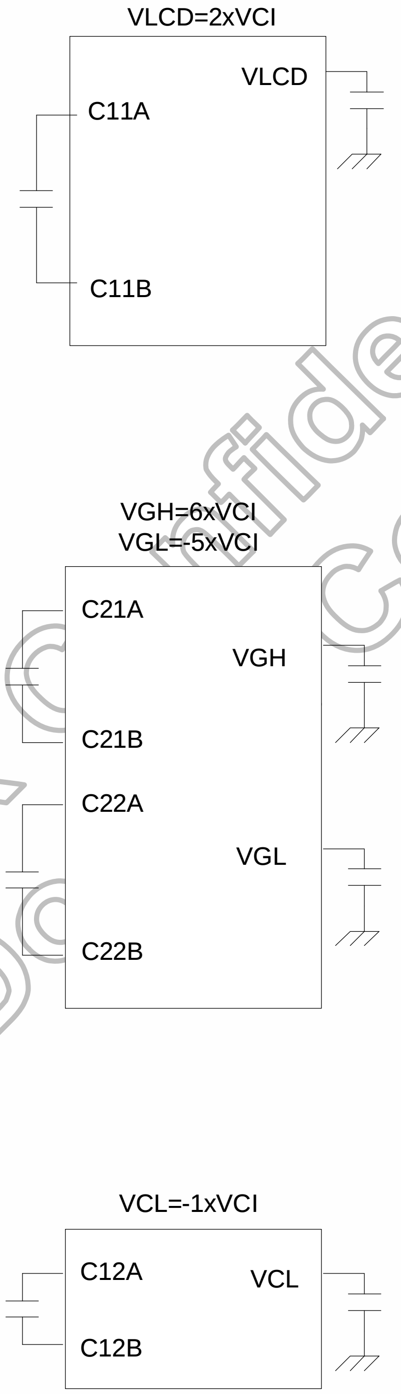


Figure 6. 2 Various boosting steps

7. Software Configuration

7.1 Features

7.1.1 Display

- Resolution: 240(H) x RGB(H) x 480(V)
- Display Color modes
  - A. Normal Display Mode On
    - a. Register-Content interface mode
      - i. 262,144(R(6),G(6),B(6)) colors
      - ii. 65,536(R(5),G(6),B(5)) colors
  - B. Idle Mode On
    - a. 8 (R(1),G(1),B(1)) colors.

7.1.2 Display module

- AM-LCD glass 240xRGBx320
- On module VCOM control (-2.0 to 5.5V Common electrode output voltage range)
- On module DC/DC converter
  - A. DDVDH = 4.6 to 6.0V (Source output voltage range)
  - B. VGH = +9.0 to +16.5V (Positive Gate output voltage range)
  - C. VGL = -6.0 to -13.5V (Negative Gate output voltage range)
- Frame Memory area 240 (H) x 480 (V) x 18 bit

7.1.3 Display/Control interface

- Display Interface types supported
  - A. Register-Content interface mode
    - 8-/16-/18-bit MPU parallel interface.
    - Serial data transfer interface.
    - 16, 18 data lines parallel video (RGB) interface.
    - MDDI (Mobile Display Digital Interface) interface.
    - Control Interface types supported
  - B. Register-Content interface mode (IFSEL0 = 1)

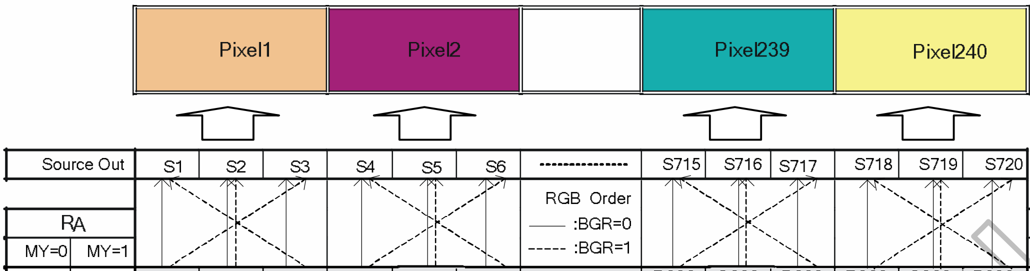
- Logic voltage:
  - A. Register-Content interface mode: (IOVCC): 1.65V ~ 3.3V
    (VCC): 2.3V ~ 3.3V.

- Driver power supply (VCI): 2.3 ~ 3.3V
- Color modes
  - A. 16 bit/pixel: R(5), G(6), B(5)
  - B. 18 bit/pixel: R(6), G(6), B(6)

7.1.4 Others

- Low power consumption, suitable for battery operated systems
- Image sticking eliminated function
- CMOS compatible inputs
- Optimized layout for COG assembly
- Temperature range: -40 ~ +85 °C
- Proprietary multi phase driving for lower power consumption
- Support external VDD for lower power consumption (such as 1.8 volts input)
- Support RGB through mode with lower power consumption
- Support Gamma correction of RGB independence
- Support normal black/normal white LCD
- Support wide view angle display
- Support burn-in mode for efficient test in module production
- On-chip OTP (one-time-programming) non-volatile memory

7.2 GRAM mapping



RA MY=0	RA MY=1	S1	S2	S3	S4	S5	S6	------	S715	S716	S717	S718	S719	S720	Gate GS=0	Gate GS=1
0	431	R0 5-0	G0 5-0	B0 5-0	R1 5-0	G1 5-0	B1 5-0	–	R238 5-0	G238 5-0	B238 5-0	R239 5-0	G239 5-0	B239 5-0	G1	G432
1	430							–							G2	G431
2	429							–							G3	G430
3	428							–							G4	G429
4	427							–							G5	G428
5	426							–							G6	G427
6	425							–							G7	G426
7	424							–							G8	G425
8	423							–							G9	G424
9	422							–							G10	G423
10	421							–							G11	G422
11	420							–							G12	G421
:	:	:	:	:	:	:	:	-- Display Pattern Data	:	:	:	:	:	:	:	:
424	7							–							G425	G8
425	6							–							G426	G7
426	5							–							G427	G6
427	4							–							G428	G5
428	3							–							G429	G4
429	2							–							G430	G3
430	1							–							G431	G2
431	0							–				RN 7-0	GN 7-0	BN 7-0	G432	G1
CA	MX=0	0			1				238			239				
CA	MX=1	239			238				1			0				

Note: RA=Row Address,
CA=Column Address,
MX=Mirror X-axis (Column address direction parameter), D6 parameter of Memory Access Control (R16h) command
MY=Mirror Y-axis (Row address direction parameter), D7 parameter of Memory Access Control (R16h) command
GS=Scan direction parameter, D4 parameter of Memory Access Control (R16h) command
BGR=Red, Green and Blue pixel position change, D3 parameter of Memory Access Control (R16h) command

Figure 7. 1 Memory Map. (240RGBx320)

7.3 Scan Function

The data is written in the order illustrated above. The Counter which dictates where in the physical memory the data is to be written is controlled by “Memory Data Access Control” Command, Bits MY, MX, MV as described below.

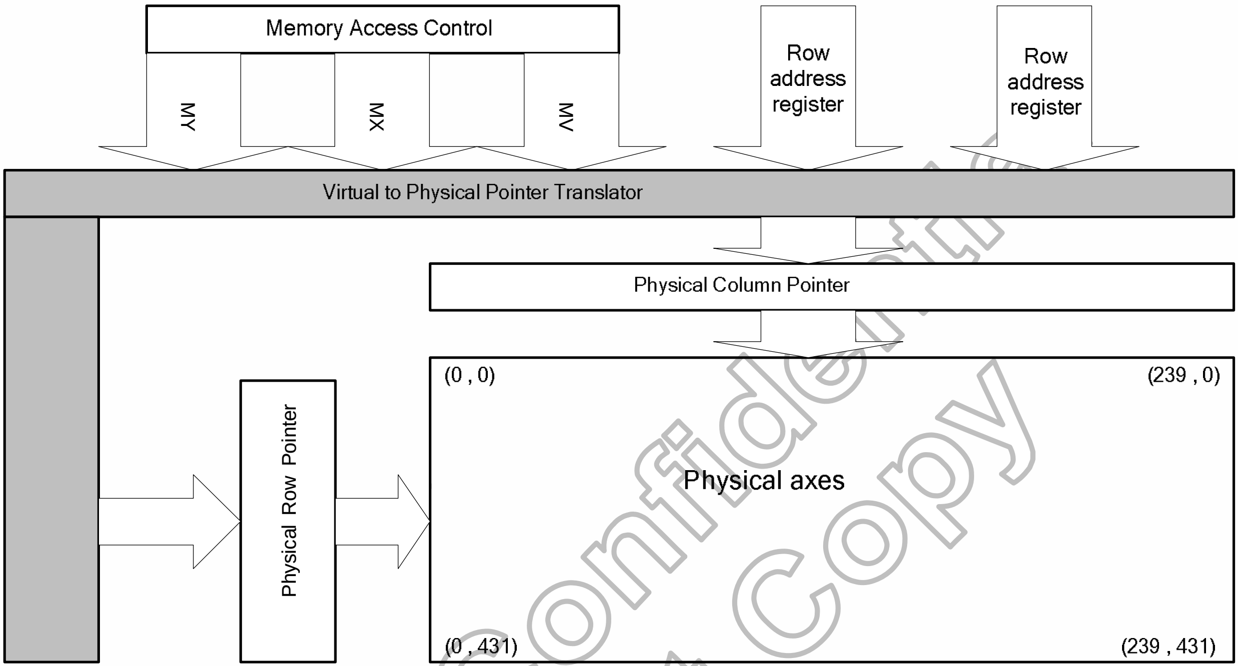


Figure 7. 2 MY, MX, MV Setting of 240RGB x 432 Dot

MY	MX	MV	CASET	PASET
0	0	0	Direct to Physical Column Pointer	Direct to Physical Page Pointer
0	0	1	Direct to Physical Column Pointer	Direct to (319-Physical Page Pointer)
0	1	0	Direct to (239-Physical Column Pointer)	Direct to Physical Page Pointer
0	1	1	Direct to (239-Physical Column Pointer)	Direct to (319-Physical Page Pointer)
1	0	0	Direct to Physical Page Pointer	Direct to Physical Column Pointer
1	0	1	Direct to (319-Physical Page Pointer)	Direct to Physical Column Pointer
1	1	0	Direct to Physical Page Pointer	Direct to (239-Physical Column Pointer)
1	1	1	Direct to (319-Physical Page Pointer)	Direct to (239-Physical Column Pointer)

Table 7. 1 MY, MX, MV Setting of 240RGB x 432 Dot

7.4 Interface Mode

7.4.1 Interface Mode Selection

IFSEL0	BS2-0	Register Data	Display Data
0	000, 001, 010, 011, 100, 101, 110	Command-Parameter interface (MPU interface)	GRAM
0	111	Command-Parameter interface (SPI + RGB interface)	Normal: RGB interface Partial: GRAM
1	000, 001, 010, 011, 100	Register-Content interface (MPU interface)	GRAM
1	101	MDDI interface	GRAM
1	11x	Register-Content interface (SPI + RGB interface)	Normal: RGB interface Partial: GRAM

Table 7. 2 Interface Mode Selection

7.4.2 Register-Content Interface Mode

P68	Input signal format selection
0	Format for I80 series MPU
1	Format for M68 series MPU

Table 7. 3 MPU selection in Register-content Interface Circuit

BS2	BS1	BS0	Interface	Transferring Method of RAM data	Transferring Method of Command
0	0	0	16-bit system interface	16-bit collective	8-bit collective
0	0	1		16-bit + 2 bit	
0	1	0	18-bit system interface	18-bit collective	
0	1	1	8-bit system interface	8-bit + 8-bit + 8-bit	
1	0	0		8-bit + 8-bit	
1	0	1	MDDI interface	18-bit	
1	1	x	Serial bus transfer interface	16 or 24-bit serial	8-bit serial

Table 7. 4 Interface Selection in Register-content Interface Mode

Parallel Bus System Interface

a. Data Pin Function for I80/M68 Series CPU

Operations	E_NWR	RW_NRD	DNC_SCL
Writes Indexes into IR	0	1	0
Reads internal status	1	0	0
Writes command into register or data into GRAM	0	1	1
Reads command from register or data from GRAM	1	0	1

Table 7. 5 Data Pin Function for I80 Series CPU

Operations	E_NWR	RW_NRD	DNC_SCL
Writes Indexes into IR	1	0	0
Reads internal status	1	1	0
Writes command into register or data into GRAM	1	0	1
Reads command from register or data from GRAM	1	1	1

Table 7. 6 Data Pin Function for M68 Series CPU

b. Bit mapping of one pixel data

Input Data (8-/16-/18-bit Interface) Written to GRAM through Write Data Register

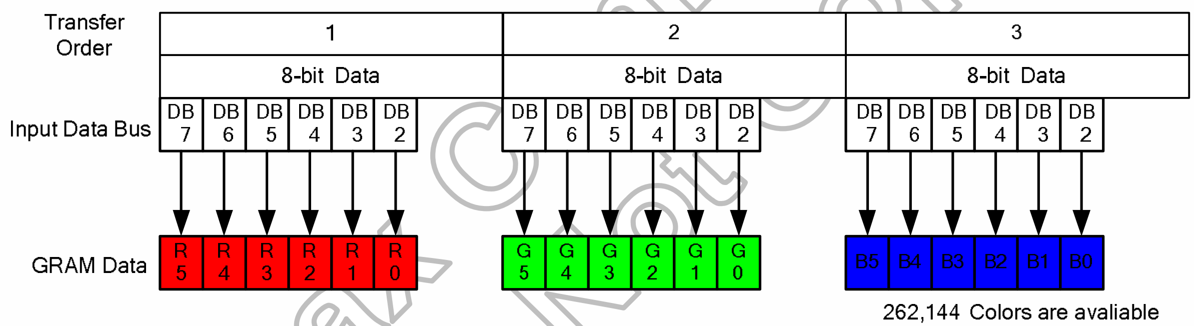


Figure 7. 3 Input Data Bus and GRAM Data Mapping in 8-Bit Bus System Interface with 18(6 + 6 + 6) Bit-Data Input (“BS2, BS1, BS0”=”011”)

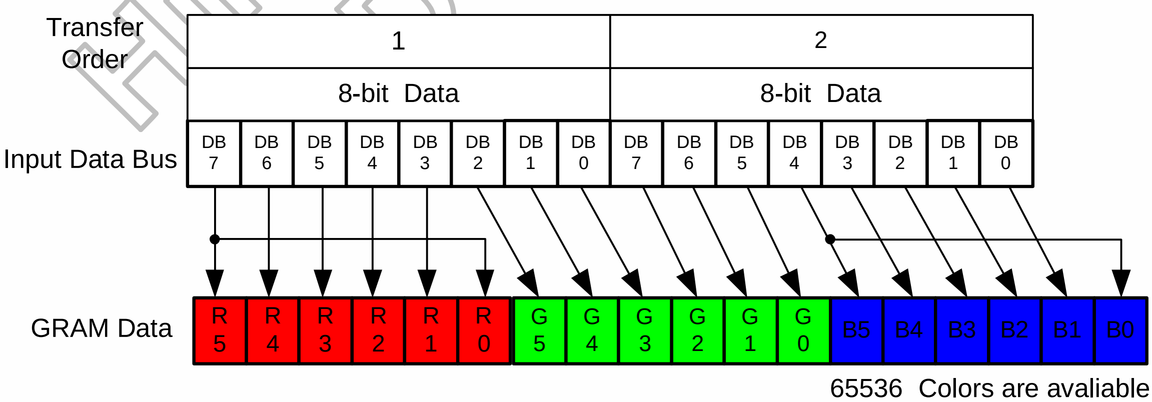


Figure 7. 4 Input Data Bus and GRAM Data Mapping in 8-Bit Bus System Interface with 16(5 + 6 + 5) Bit-Data Input (“BS2, BS1, BS0”=”100”)

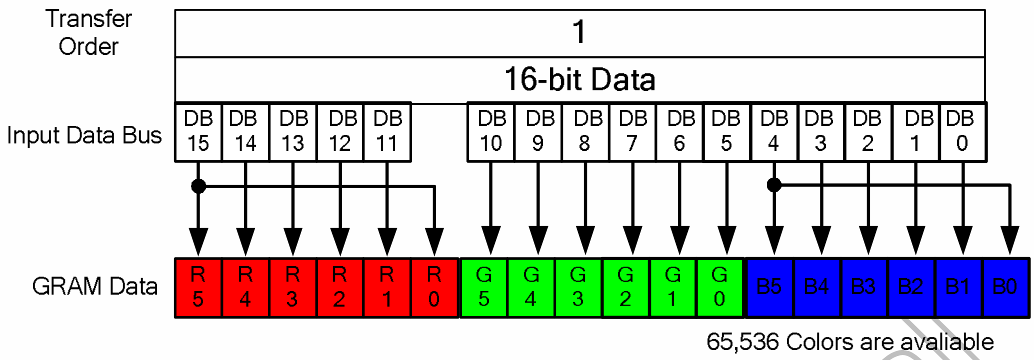


Figure 7. 5 Input Data Bus and GRAM Data Mapping in 16-Bit Bus System Interface with 16 Bit-Data Input ("BS2, BS1, BS0"="000")

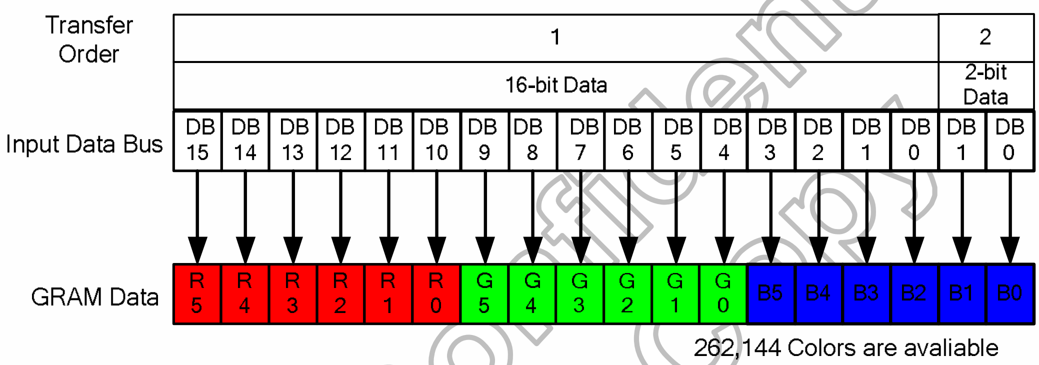


Figure 7. 6 Input Data Bus and GRAM Data Mapping in 16-Bit Bus System Interface with 18(16+2) Bit-Data Input ("BS2, BS1, BS0"="001")

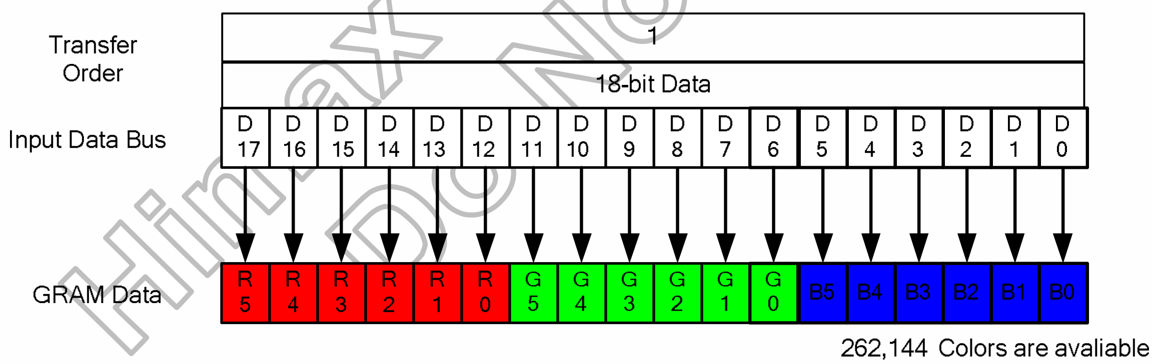


Figure 7. 7 Input Data Bus and GRAM Data Mapping in 18-Bit Bus System Interface ("BS2, BS1, BS0"="010")

i80- System Interface Timing

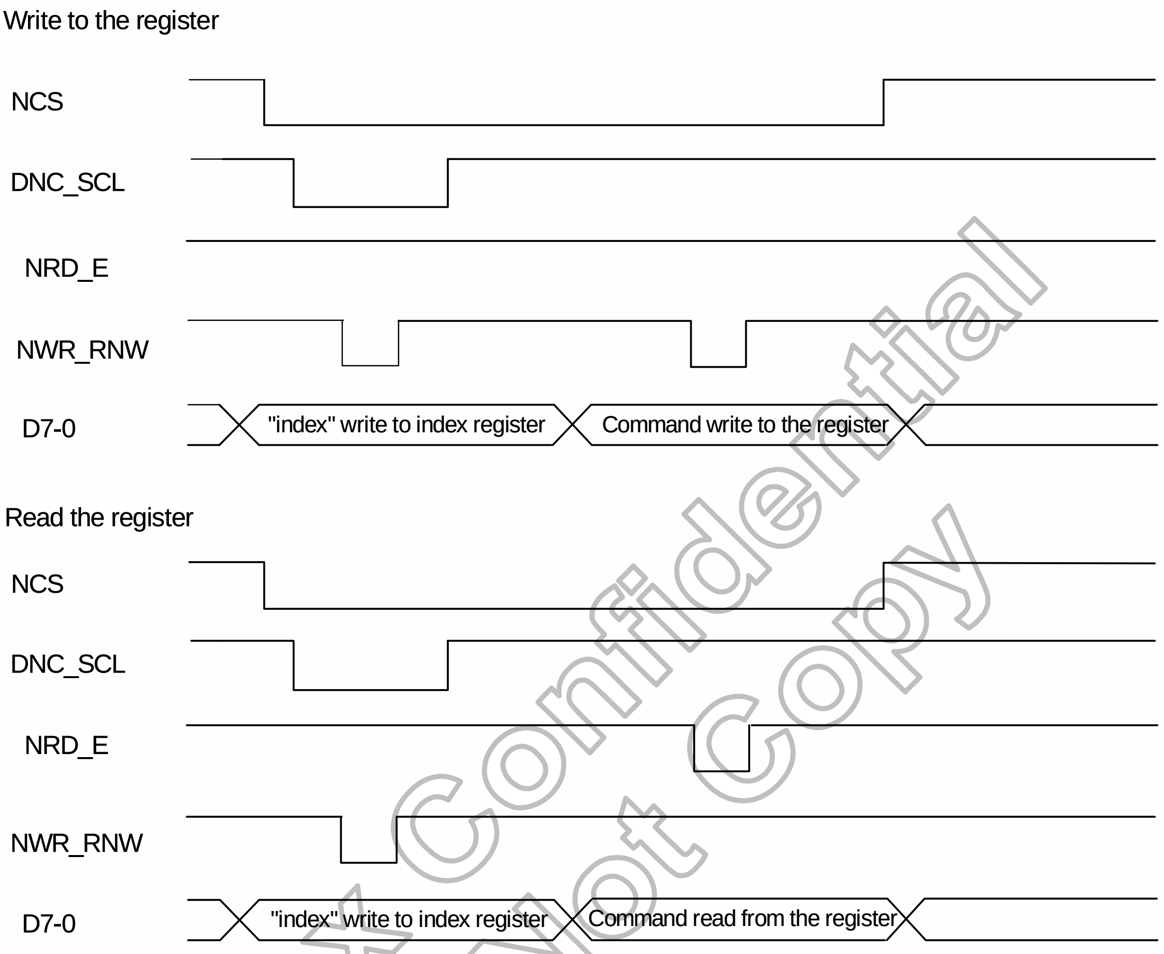


Figure 7. 8 Register read/write Timing in Parallel Bus System Interface (for I80 series MPU)

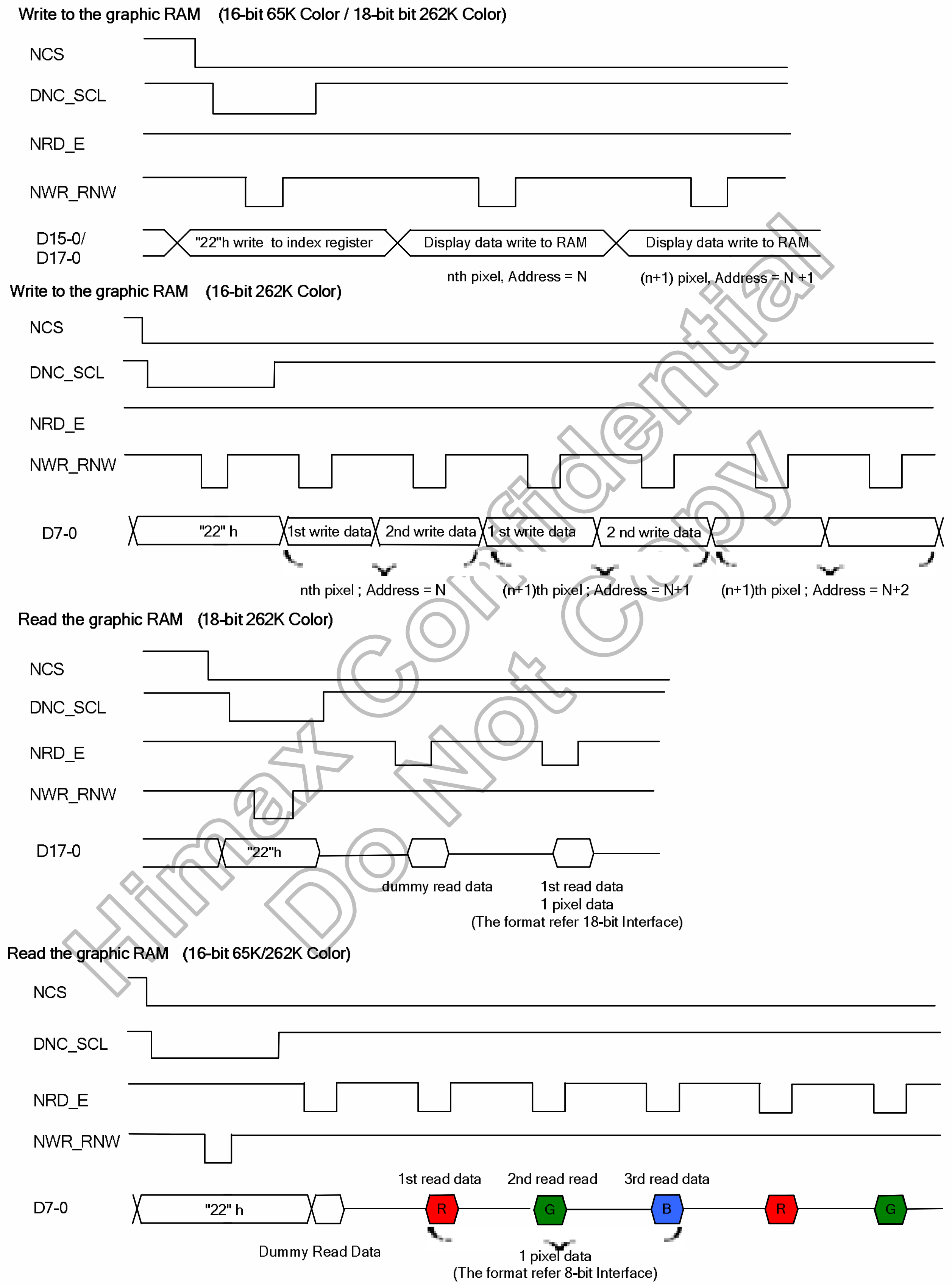


Figure 7. 9 GRAM read/write Timing in 16-/18-bit Parallel Bus System Interface (for I80 series MPU)

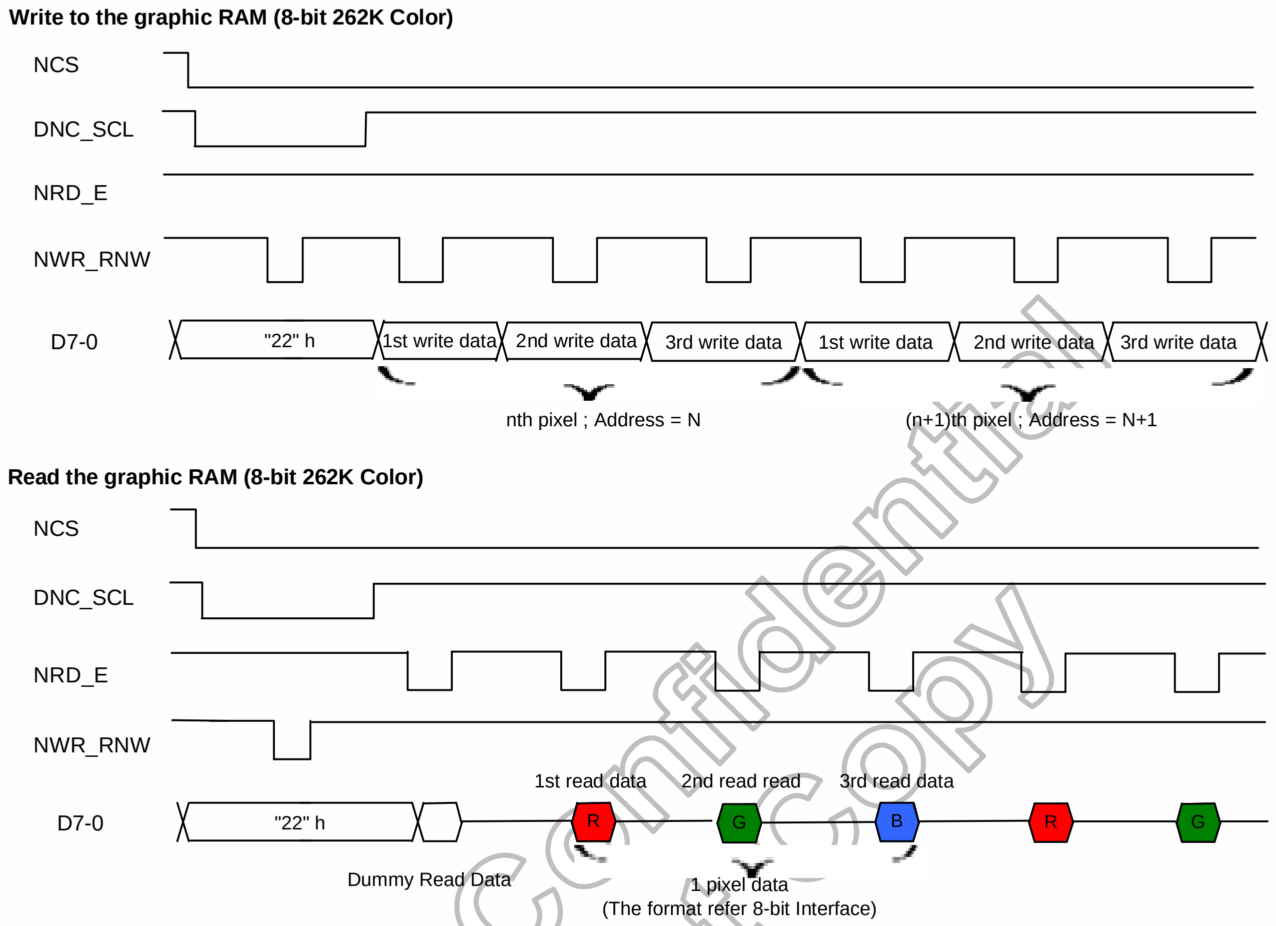


Figure 7. 10 GRAM read/write Timing in 8-bit Parallel Bus System Interface (for I80 series MPU)

m68- System Interface Timing

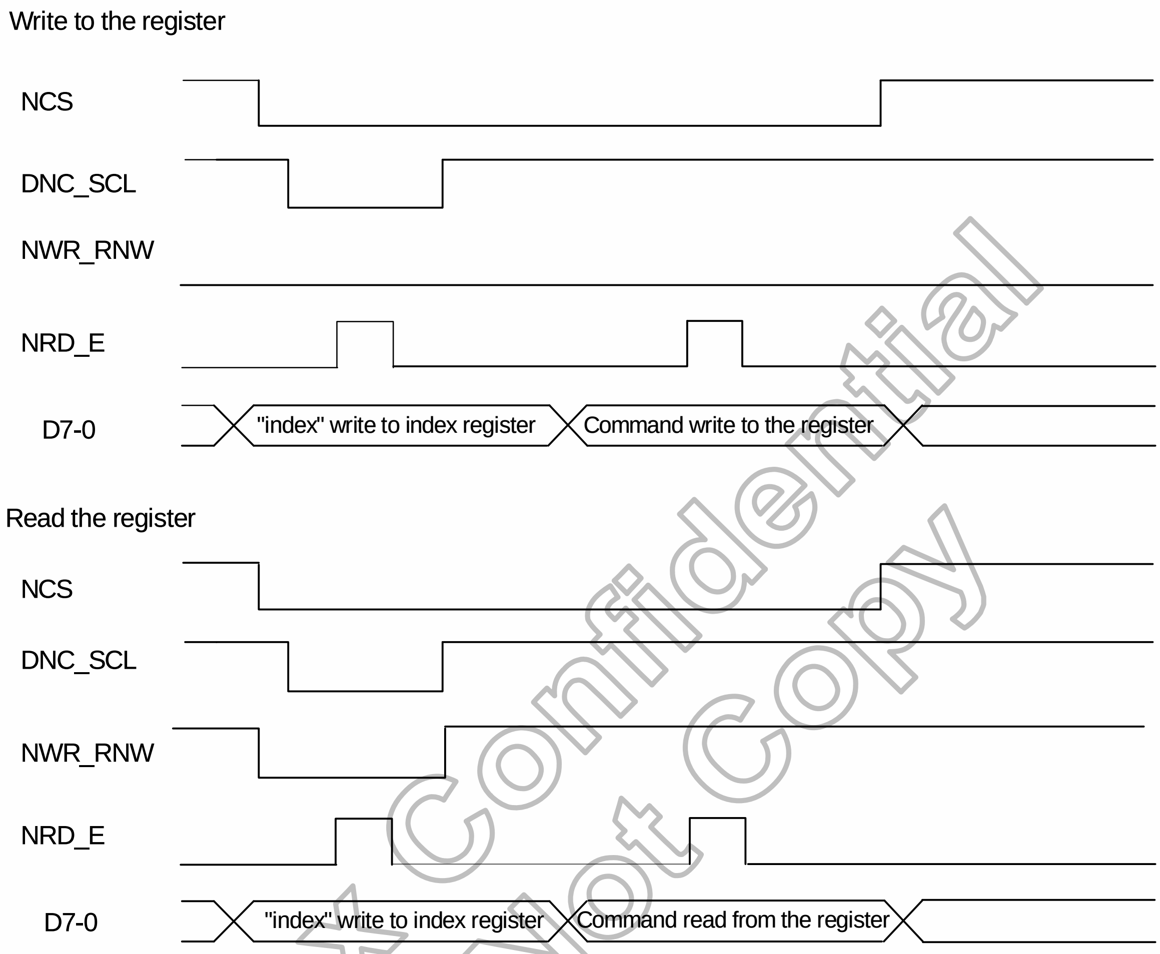
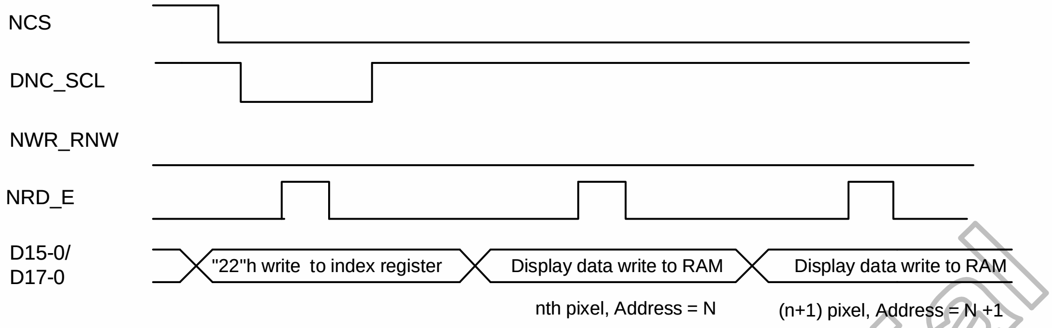
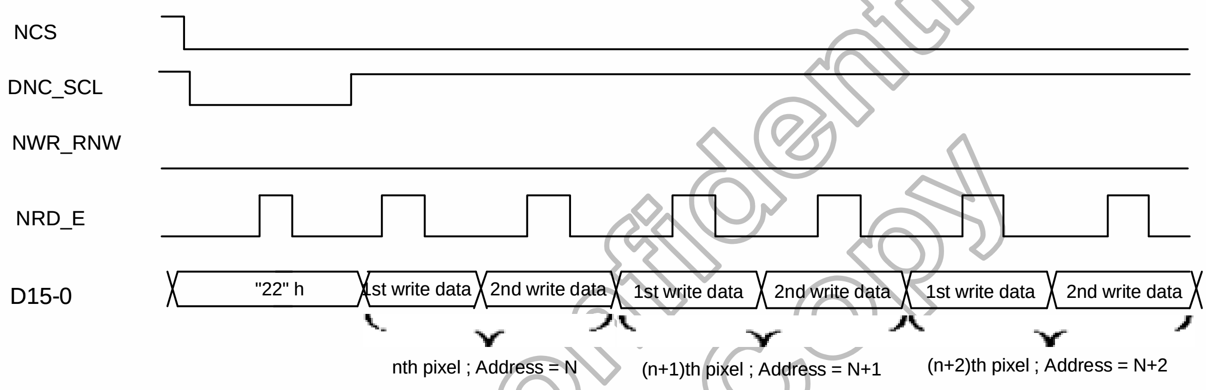


Figure 7. 11 Register read/write Timing in Parallel Bus System Interface (for M68 series MPU)

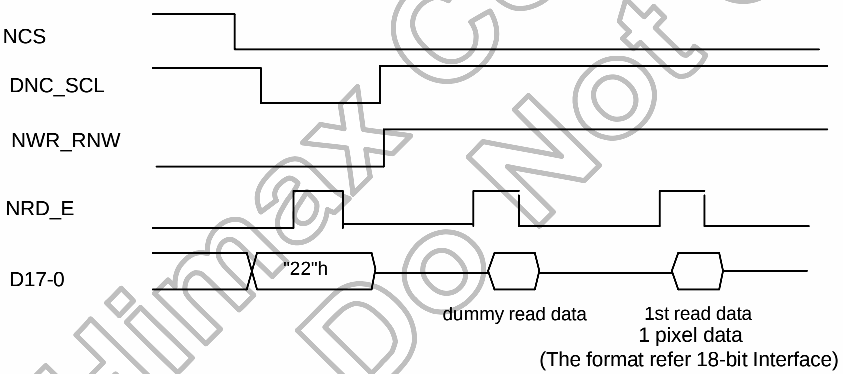
Write to the graphic RAM (16-bit 65K Color / 18-bit bit 262K Color)



Write to the graphic RAM (16+2-bit 262K Color)



Read the graphic RAM (18-bit bit 262K Color)



Read the graphic RAM (16-bit 65K/262K Color)

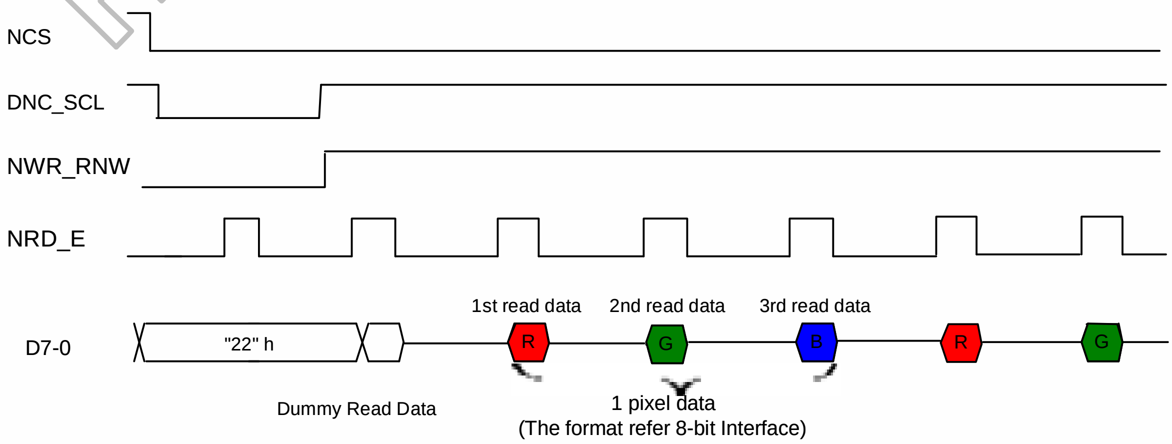


Figure 7. 12 GRAM read/write Timing in 16-/18-bit Parallel Bus System Interface (for M68 series MPU)

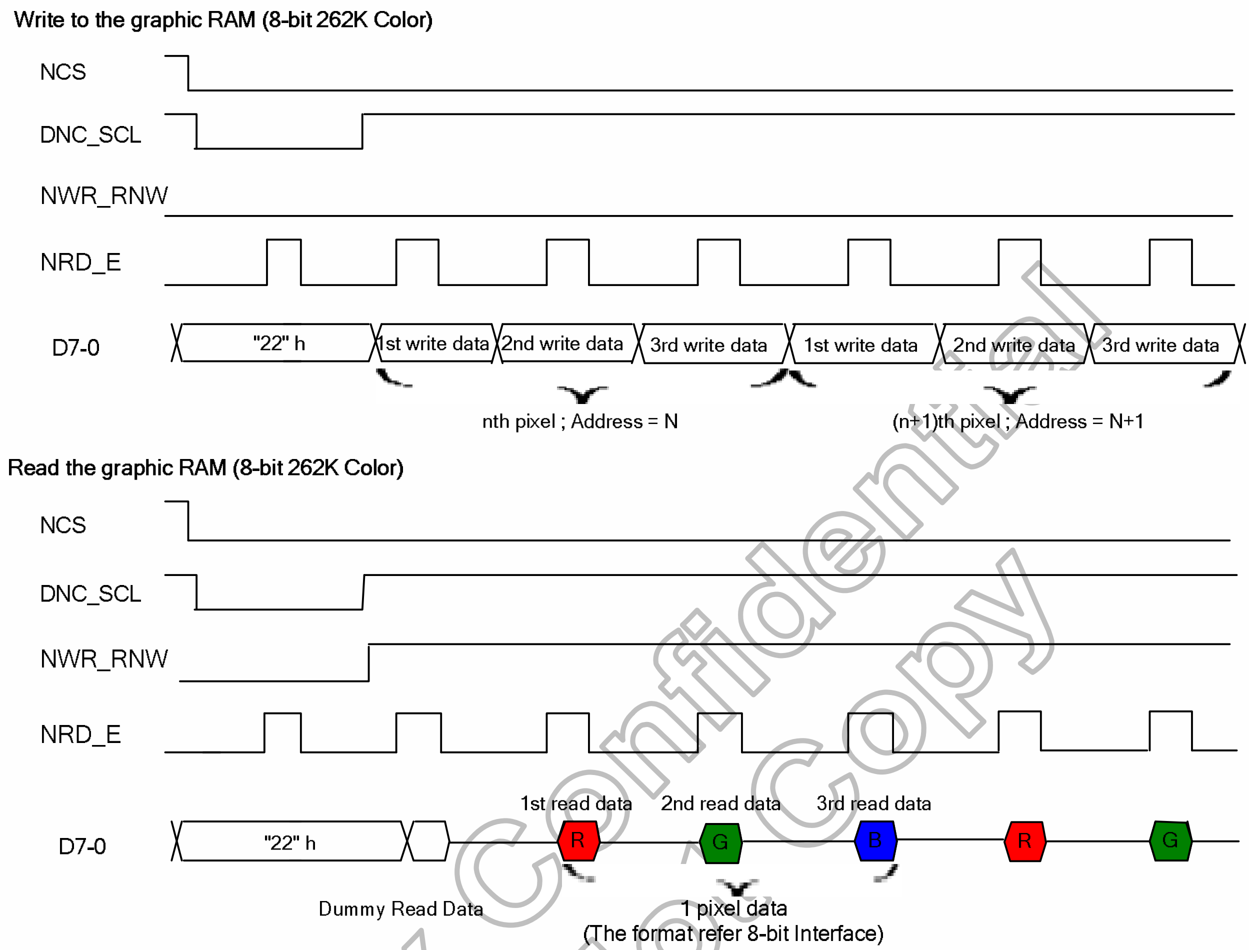


Figure 7. 13 GRAM read/write Timing in 8-bit Parallel Bus System Interface (for M68 series MPU)

7.4.3 Serial Data Transfer interface

RS	R/W	Function
0	0	Writes Indexes into IR
1	0	Writes command into register or data into GRAM
1	1	Reads command from register or data from GRAM

Table 7. 7 The Function of RS and R/W Bit bus

Serial Data Transfer interface Timing

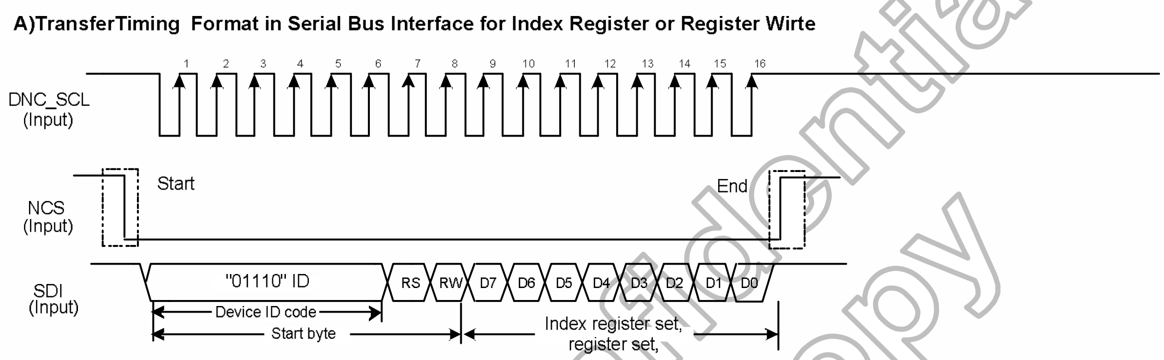


Figure 7. 14 Data Write Timing in Serial Bus System Interface

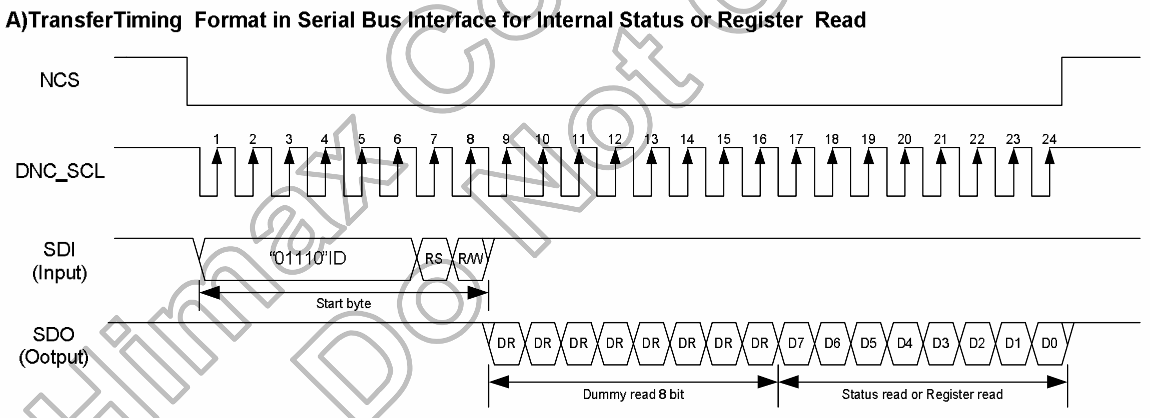


Figure 7. 15 Data Read Timing in Serial Bus System Interface

7.4.5 RGB Interface

EPL	ENABLE	Display
0	0	Disable
0	1	Enable
1	0	Enable
1	1	Disable

Table 7. 8 EPL bit setting and Valid ENABLE Signal

RGB Interface Timing

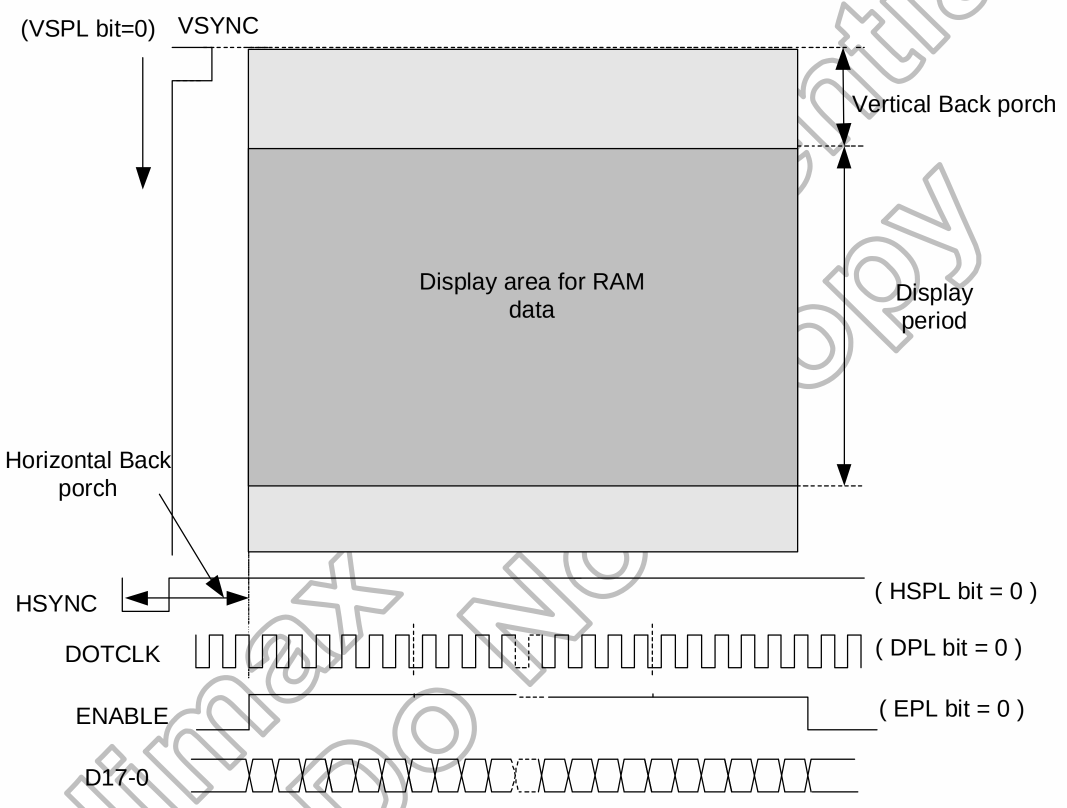


Figure 7. 16 RGB Interface Circuit Input Timing

(a) 18 bit/pixel color order (R 6-bit, G 6-bit, B 6-bit), 262,144 colors (CSEL(2-0) = "110")

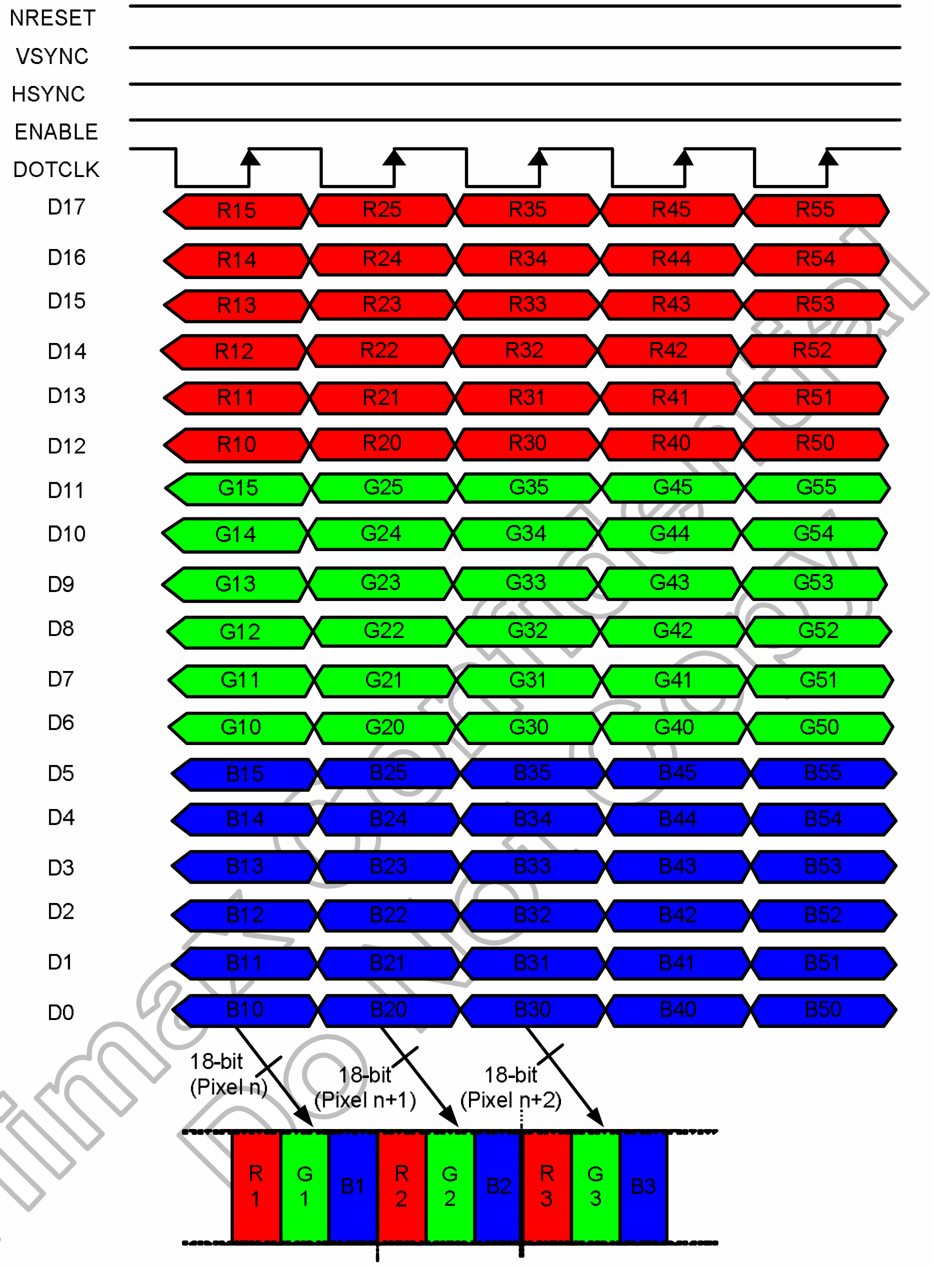


Figure 7. 17 18 bit / pixel Data Input of RGB Interface

(b) 16 bit/pixel color order (R 5-bit, G 6-bit, B 5-bit), 65,536 colors (CSEL(2-0) = "101")

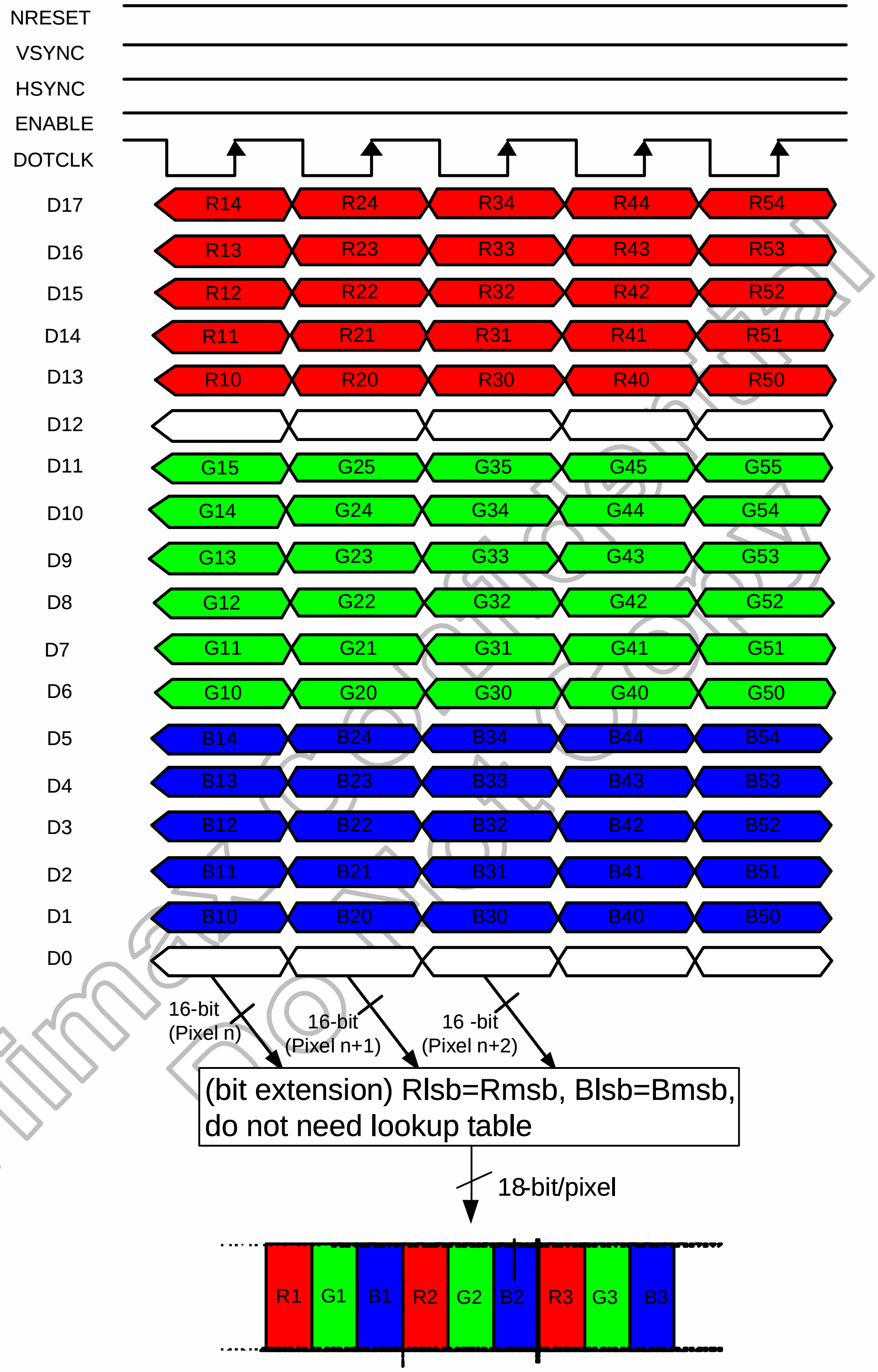


Figure 7. 18 16 bit / pixel Data Input of RGB Interface

7.4.6 MDDI Interface

The HX-8352-A support MDDI, which is a differential serial interface with high-speed, low voltage swing characteristics. Both command and display image data can be transferred by MDDI. The devices connected by Data and STB link are host and client part.

Host transfer data to client in “forward” direction, client transfer data to host in “reverse” direction. The Data line is Dual direction, both command and image data are all send through the Data line. The STB line send strobe signal from host to client.

Data transferred in MDDI link are encoded as packet type.

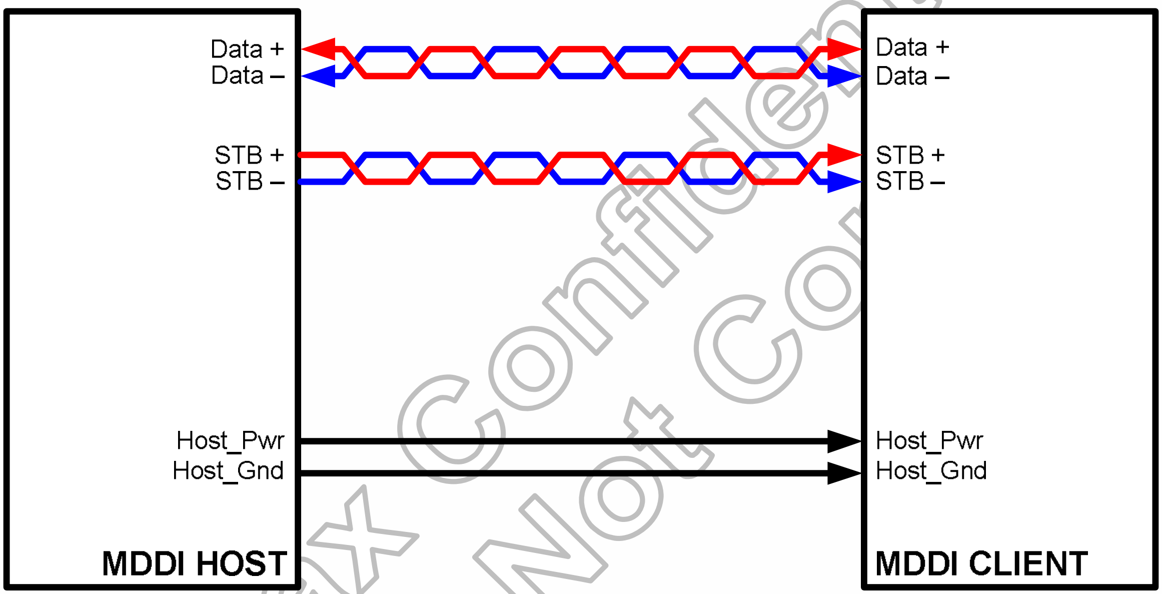


Figure 7. 19 Physical Connection of MDDI Host and Client

7.4.6.1 Terminology

The devices connected by the MDDI link are called the host and client. Data going from the host to the client travels in the ***forward*** direction, and data from the client to the host travels in the ***reverse*** direction.

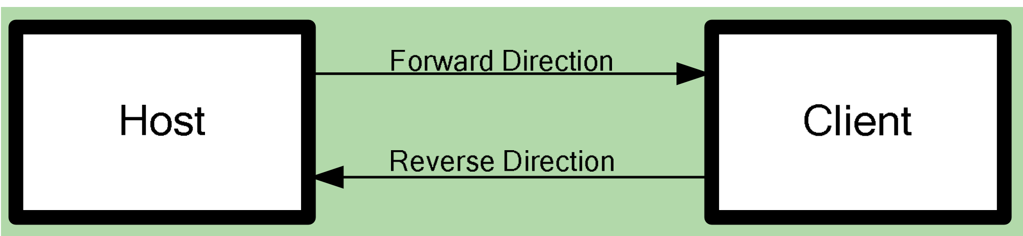


Figure 7. 20 MDDI Terminology

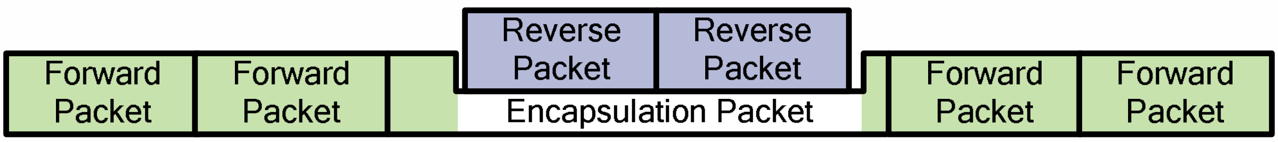


Figure 7. 21 Example of Bi-Directional MDDI Communication

7.4.6.2 MDDI Packet

Data transmitted over the MDDI link is grouped into packets. Several packets format is supported in HX8352-A. Most packets are in forward direction, transferred from host to client; but reverse encapsulation packet is in reverse direction, transferred from MDDI client to host. A number of packets, started by sub-frame header packet, construct one sub frame.

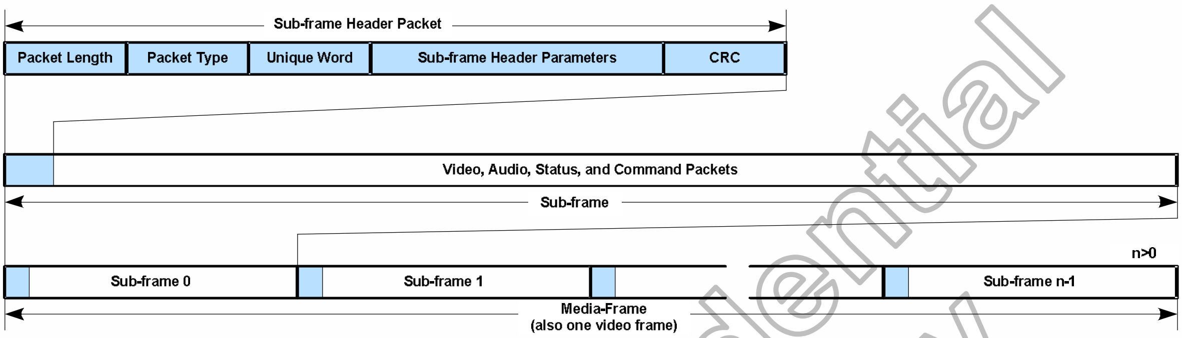


Figure 7. 22 MDDI Packet Structure

Refer to MDDI frame structure, sub-frame header packet is placed in front of a sub-frame, and some sub-frames make up a media-frame.

HX8352-A support 9 types of packets, which described in the table below.

Packet	Function	Direction
Sub-frame header packet	Header of each sub frame	Forward
Register access packet	Register setting	Forward
Video stream packet	Video data transfer	Forward
Filler packet	Fill empty packet space	Forward
Reverse link encapsulation packet	Reverse data packet	Reverse
Round-trip delay measurement packet	Host->client->host delay check	Forward/Reverse
Client capability packet	Capability of client check	Reverse
Client request and status packet	Information about client status	Reverse
Link shutdown packet	End of frame	Forward

Figure 7. 23 List of Supported MDDI Packet

Sub-frame Header Packet

Packet Length	Packet type =0x3bffh	Unique word =0x005a	Reserved 1	Sub-frame length	Protocol Version	Sub-frame Count	Media-frame Count	CRC
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	2 bytes	2 bytes	4 bytes	2 bytes

packet length	:	total number of bytes in the packet not including the packet length field
packet type	:	packet type, 0x3bffh for sub-frame header packet
unique word	:	link packet type to form a 32-bit unique word for good autocorrelation.
reserved 1	:	not used(all zero)
sub-frame length	:	specifies number of bytes per sub-frame
protocol version	:	set all zero
sub-frame count	:	specifies number of sub-frame header packet.
media frame count	:	specifies number of media frame
CRC	:	error check

Register Access Packet

Packet Length	Packet type =146	bClient ID	Read/Write Info	Register Address	Parameter CRC	Register Data list	Register Data CRC
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	2 bytes	2 bytes	4 bytes

packet length : total number of bytes in the packet not including the packet length field
packet type : packet type, 146(decimal) for register access packet
bClient ID : set all zero
Read/Write Info : when write value to register, bit[15:14] = “00”
when request data from register, bit[15:14]=“10”
when data from register, bit[15:14] = “11”
bit[13:0] : 00_0000_0000_0001
register address : Register address is set written here.
parameter CRC : To error check from packet length to register address
register data list : Paramter data is written here.
register data CRC : To error check register data list.

Video Stream Packet

Packet Length	Packet type =16	bClient ID	video data format descriptor	pixel data attributes	X left edge	Y top edge	X right edge	Y bottom dedge	X start	Y start
2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes

pixel count	parameter CRC	pixel data	pixel data CRC
2 bytes	2 bytes	packet length - 26 bytes	2 bytes

packet length total number of bytes in the packet not including the packet length field
packet type packet type, 16 (decimal) for register access packet
bClient ID set all zero
video data format descriptor bits[15:13]=010, raw RGB format (fixed value)
bit[12]=1,only packed type is available (fixed value)
bits[11:0]=0110_0110_0110, 18bit pixel
bits[11:0]=0101_0110_0101, 16bit pixel
 bits[1:0]=11, displayed both eyes (fixed value)
pixel data attributes bit[5]=1, X left edge .. Y start is not defined.(fixed value)
others are all zero
X left edge Non used in HX8352-A, set all zero
X top edge Non used in HX8352-A, set all zero
X right edge Non used in HX8352-A, set all zero
Y bottom edge Non used in HX8352-A, set all zero
X start Non used in HX8352-A, set all zero
Y start Non used in HX8352-A, set all zero
Pixel count Write number of pixel
Patameter CRC To error check from packet length to pixel count
pixel data pixel data info. Number of pixel data must not be over 65509
pixel data CRC To pixel data error check.

Filler Packet

Packet Length	Packet type =0	filler bytes (all zero)	CRC
2 bytes	2 bytes	packet length – 4 bytes	2 bytes

packet length : total number of bytes in the packet not including the packet length field
packet type : packet type, 16 (decimal) for register access packet
filler bytes : set to all zero (The size is under packet length available)
CRC : To error check

Link Shutdown Packet

Packet Length	Packet type =69	CRC	All zeros
2 bytes	2 bytes	2 bytes	16 bytes

packet length total number of bytes in the packet not including the packet length field
packet type packet type, 16 (decimal) for register access packet
CRC To error check
All zeros write all zero (size is 16 bytes, because MDDI for HX8352-A is type 1)

Fixed Value

7.4.6.3 Hibernation / Wake up

HX8352-A supports hibernation mode for reduce current consumption. The MDDI link can enter the hibernation state quickly and wake up from hibernation quickly. This allows the system to force the MDDI link into hibernation frequently to reduce power consumption. In hibernation mode, hi-speed drivers and receivers are disabled and low-speed & low-power receivers are enabled to detect wake-up sequence.

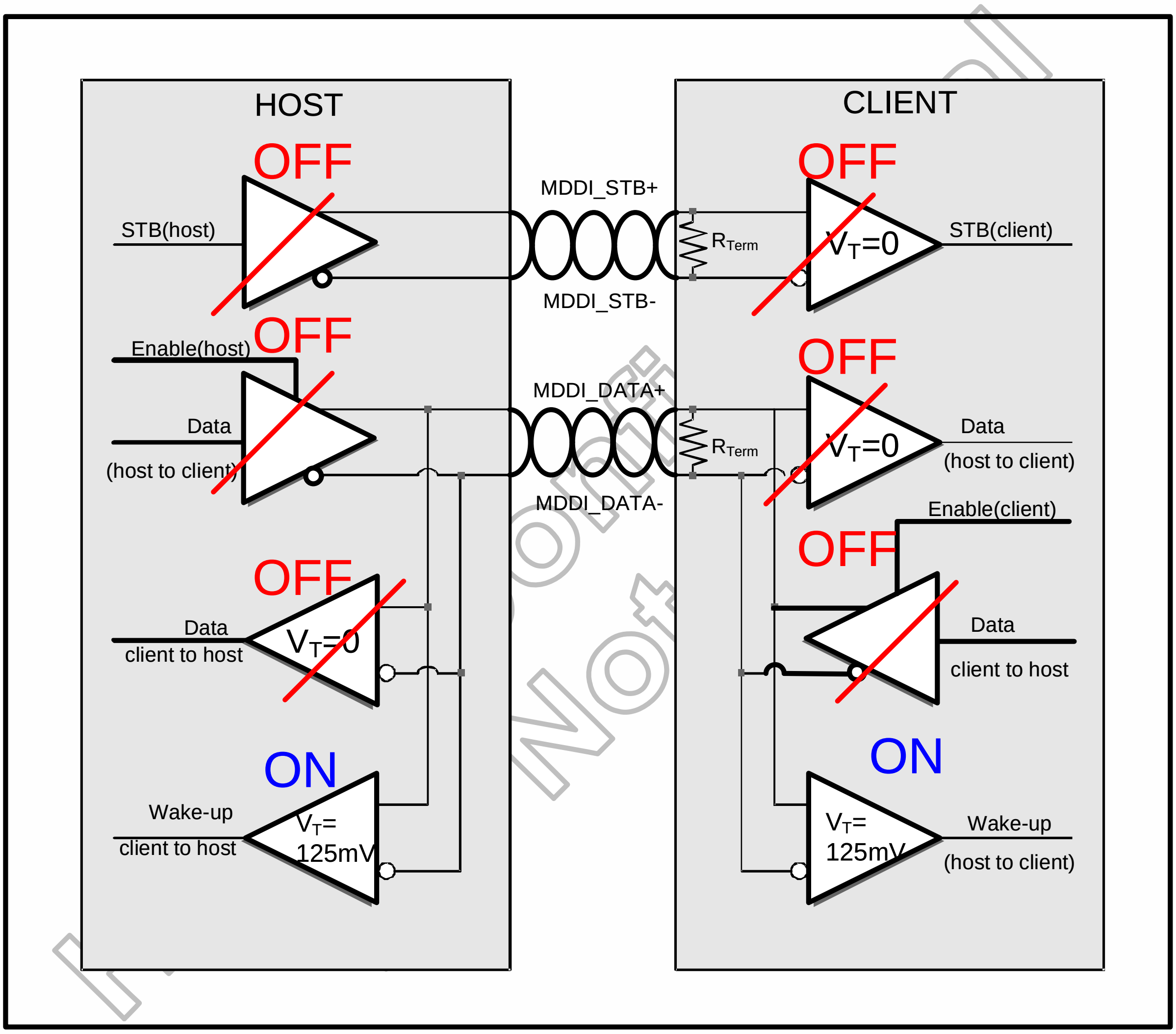


Figure 7. 24 MDDI Transceiver / Receiver State in Hibernation

When the link wakes up from hibernation the host and client exchange a sequence of pulses. These pulsed can be detected using low-speed line receivers that consume only a fraction of the current as the differential receivers required to receive the signals at the maximum link operating speed.

Either the host or client can wake up the link, which is supported in HX8352-A: Host-Initial Wakeup & Client-Initial Wakeup.

7.4.6.4 MDDI Link Wakeup Sequence

Figure below provide a host-initiated wake-up is described below without contention from the client trying to wake up at the same time. The labeled events are:

Host-Initiated Wake-up

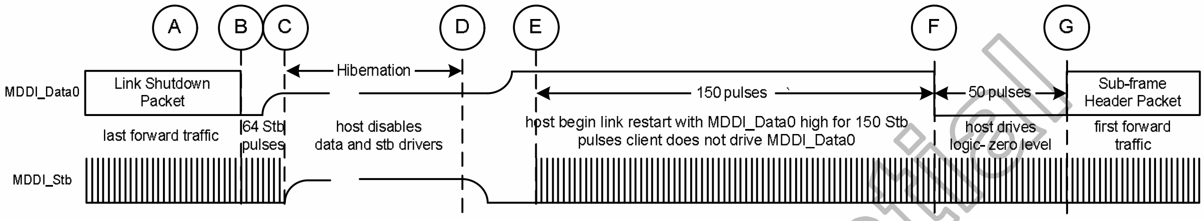


Figure 7. 25 Host-initiated Link Wakeup Sequence

A. The host sends a Link Shutdown Packet to inform the client that the link will transition to the low-power hibernation state.

B. Following the CRC of the Link Shutdown Packet the host toggles MDDI_Stb for 64 cycles to allow processing in the client to finish before it stops MDDI_Stb from toggling which stops the recovered clock in the client device. Also during this interval the host initially sets MDDI_Data0 to a logic-zero level, and then disables the MDDI_Data0 output in the range of 16 to 48 MDDI_Stb cycles (including output disable propagation delays) after the CRC. It may be desirable for the client to place its high-speed receivers for MDDI_Data0 and MDDI_Stb into a low power state any time after 48 MDDI_Stb cycles after the CRC and before point C.

C. The host enters the low-power hibernation state by disabling the MDDI_Data0 and MDDI_Stb drivers and by placing the host controller into a low-power hibernation state. It is also allowable for MDDI_Stb to be driven to a logic-zero level or to continue toggling during hibernation. The client is also in the low-power hibernation state.

D. After a while, the host begins the link restart sequence by enabling the MDDI_Data0 and MDDI_Stb driver outputs. The host drives MDDI_Data0 to a logic-one level and MDDI_Stb to a logic-zero level for at least the time it takes for the drivers to fully enable their outputs. The host shall wait at least 200 nsec after MDDI_Data0 reaches a valid logic-one level and MDDI_Stb reaches a valid logic-zero level before driving pulses on MDDI_Stb. This gives the client sufficient time to prepare to receive high-speed pulses on MDDI_Stb. The client first detects the wake-up pulse using a low-power differential receiver having a +125mV input offset voltage.

E. The host drivers are fully enabled and MDDI_Data0 is being driven to a logic-one level. The host begins to toggle MDDI_Stb in a manner consistent with having a logic-zero level on MDDI_Data0 for a duration of 150 MDDI_Stb cycles.

F. The host drives MDDI_Data0 to a logic-zero level for 50 MDDI_Stb cycles. The client begins to look for the Sub-frame Header Packet after MDDI_Data0 is at a logic-zero level for 40 MDDI_Stb cycles.

G. The host begins to transmit data on the forward link by sending a Sub-frame Header Packet. Beginning at point G the MDDI host generates MDDI_Stb based on the logic level on MDDI_Data0 so that proper data-strobe encoding commences from point G.

An example of a typical client-initiated service request event with no contention is illustrated in below figure. The labeled events are：

Client-Initiated Wake-up

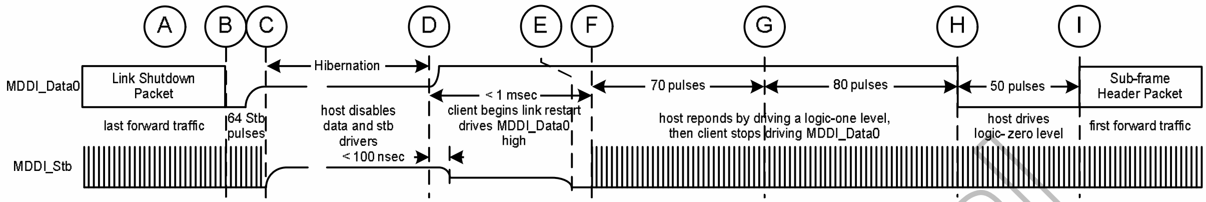


Figure 7. 26 Client-initiated Link Wake-up Sequence

A. The host sends a Link Shutdown Packet to inform the client that the link will transition to the low-power hibernation state.

B. Following the CRC of the Link Shutdown Packet the host toggles MDDI_Stb for 64 cycles to allow processing in the client to finish before it stops MDDI_Stb from toggling which stops the recovered clock in the client device. Also during this interval the host initially sets MDDI_Data0 to a logic-zero level, and then disables the MDDI_Data0 output in the range of 16 to 48 MDDI_Stb cycles (including output disable propagation delays) after the CRC. It may be desirable for the client to place its high-speed receivers for MDDI_Data0 and MDDI_Stb into a low power state any time after 48 MDDI_Stb cycles after the CRC and before point C.

C. The host enters the low-power hibernation state by disabling its MDDI_Data0 and MDDI_Stb driver outputs. It is also allowable for MDDI_Stb to be driven to a logic-zero level or to continue toggling during hibernation. The client is also in the low-power hibernation state.

D. After a while, the client begins the link restart sequence by enabling the MDDI_Stb receiver and also enabling an offset in its MDDI_Stb receiver to guarantee the state of the received version of MDDI_Stb is a logic-zero level in the client before the host enables its MDDI_Stb driver. The client will need to enable the offset in MDDI_Stb immediately before enabling its MDDI_Stb receiver to ensure that the MDDI_Stb receiver in the client is always receiving a valid differential signal and to prevent erroneous received signals from propagating into the client. After that, the client enables its MDDI_Data0 driver while driving MDDI_Data0 to a logic-one level. It is allowed for MDDI_Data0 and MDDI_Stb to be enabled simultaneously if the time to enable the offset and enable the standard MDDI_Stb differential receiver is less than 200 nsec.

E. Within 1 msec the host recognizes the service request pulse, and the host begins the link restart sequence by enabling the MDDI_Data0 and MDDI_Stb driver outputs. The host drives MDDI_Data0 to a logic-one level and MDDI_Stb to a logic-zero level for at least the time it takes for the drivers to fully enable their outputs. The host shall wait at least 200 nsec after MDDI_Data0 reaches a valid logic-one level and MDDI_Stb reaches a valid fully driven logic-zero level before driving pulses on MDDI_Stb. This gives the client sufficient time to prepare to receive high-speed pulses on MDDI_Stb.

F. The host begins outputting pulses on MDDI_Stb and shall keep MDDI_Data0 at a logic-one level for a total duration of 150 MDDI_Stb pulses through point H. The host generates MDDI_Stb in a manner consistent with sending a logic-zero level on MDDI_Data0. When the client recognizes the first pulse on MDDI_Stb it shall disable the offset in its MDDI_Stb receiver.

G. The client continues to drive MDDI_Data0 to a logic-one level for 70 MDDI_Stb pulses, and the client disables its MDDI_Data0 driver at point G. The host continues to drive MDDI_Data0 to a logic-one level for a duration of 80 additional MDDI_Stb pulses, and at point H drives MDDI_Data0 to a logic-zero level.

H. The host drives MDDI_Data0 to a logic-zero level for 50 MDDI_Stb cycles. The client begins to look for the Sub-frame Header Packet after MDDI_Data0 is at a logic-zero level for 40 MDDI_Stb cycles.

I. After asserting MDDI_Data0 to a logic-zero level and driving MDDI_Stb for a duration of 50 MDDI_Stb pulses the host begins to transmit data on the forward link at point I by sending a Sub-frame Header Packet. The client begins to look for the Sub-frame Header Packet after MDDI_Data0 is at a logic-zero level for 40 MDDI_Stb cycles.

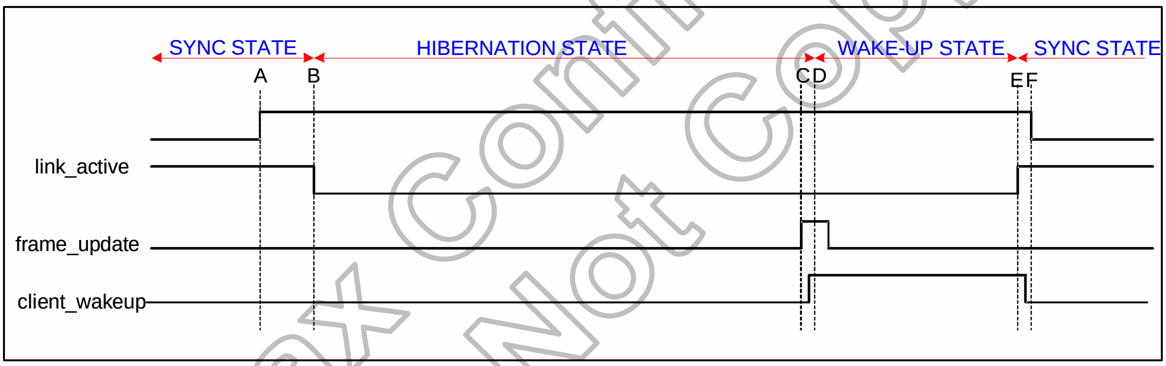
7.4.6.5 Sequence for the Client to Wake up the Link

The HX8352-A supports two link wake up mode for the client based on VSYNC or GPIO. Only in hibernation mode, the client can wake up the link. User should configure the register for a wakeup before link is shut down.

Link wakeup based on VSYNC

When all display data finishes being displayed in display mode, a data request is sent to the MDDI host for new video data. The MDDI link is normally in hibernation mode for reducing power dissipation by the interface. Before the on-chip RAM is updated, the MDDI link must be woken up. In that case, you can use a link wakeup by the client as a data request. When the link wakeup register VWAKE_EN is set in VSYNC mode, the link is woken up by the client synchronously with a vertical sync signal generated in the HX8352-A. If the interface speed and the wakeup period are well known, link wakeup based on VSYNC can be used to attain consistent display.

Figure below shows detailed timing on a link wakeup based on VSYNC.



The Detailed descriptions for labeled events are as follows:

A. MDDI host writes to the VSYNC based link wakeup register to enable a wake-up based on internal vertical-sync signal.

B. link_active goes low when the host puts in the link into hibernation after no more data needs to be sent to the HX8352-A

C. frame_update, the internal vertical-sync signal goes high indicating that update pointer has wrapped around and is now reading from the beginning of the frame buffer. Link wake-up point can be set using WKF and WKL (4Ah, 4Bh) registers. WKF specifies the number of frame before wake-up; WKL specifies the number of lines before wake-up.

D. client_wakeup input to the MDDI client goes high to start the client initiated link wake-up.

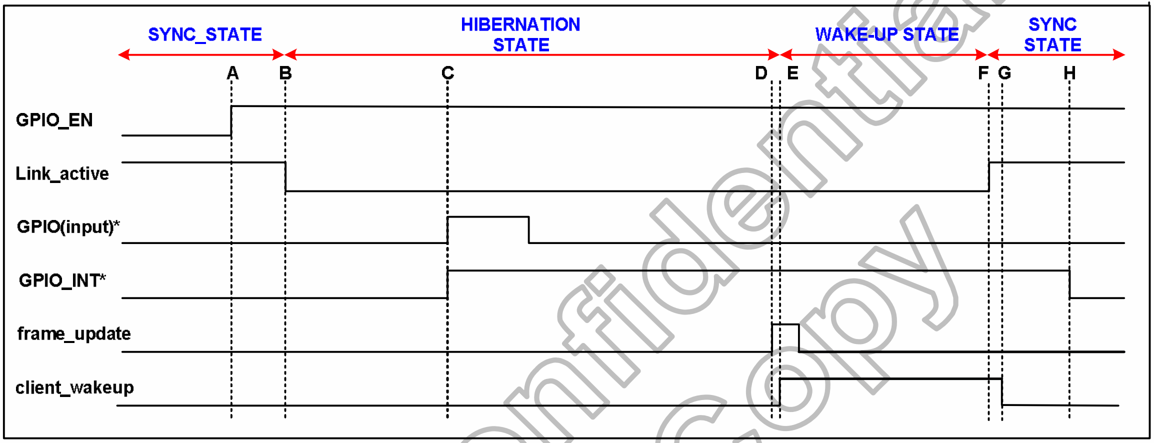
E. link_active goes high after the host brings the link out of hibernation.

F. After link wake-up, client_wakeup signal and the VWAKE register are cleared automatically.

GPIO Based Link Wake-up

In VSYNC-based link wake-up, wake-up enable register setting prior to link shut-down. GPIO based Link wake-up is enabled by interrupt from outside of the IC. For GPIO based link wake-up, GPIO interrupt enable and GPIO PAD mode (to input mode) setting must be set. Once HX8352-A receive interrupt, internal GPIO base link wake-up flag set to high, and the following procedure is similar to that of VSYNC based link wake-up.

The following figure shows detailed timing for GPIO based link wake-up.



The Detailed descriptions for labeled events are as follows:

A. Host send register access packet to sets GPIO clear interrupt register to disable clear interrupt (R4Ch:GPIO_CLR) and GPIO interrupt enable register (4Eh: GPIO_EN) for a particular GPIO.

B. After host sending all data, Link goes into hibernation (and link_active goes low) .

C. GPIO input goes high, and the GPIO interrupt (GPIO_INT) is latched.

D. Frame_update signal goes high indicating that the display has wrapped around. Link wake-up point can be set using WKF and WKL (4Ah, 4Bh) registers.

E. Client_wakeup signal of the MDDI client goes high to start the client initiated link wake-up.

F. Link_active goes high after the host make link leaving hibernation.

G. After link wake-up, client_wakeup signal is reset to low.

H. MDDI host clears the interrupt by writing '0' to the register with the bit set for that particular interrupt (GPCLR: 4Fh). Between point G and H the host will have read the GPIO_INT values to see what interrupts are active.

7.4.6.6 Sub Panel Interface

The HX8352-A supports the Sub Panel interface which connected to Sub Panel driver IC with Parallel Interface. When HX8352-A receive MDDI packets from host device, the HX8352-A will convert MDDI packet to parallel data and send to sub panel driver IC.

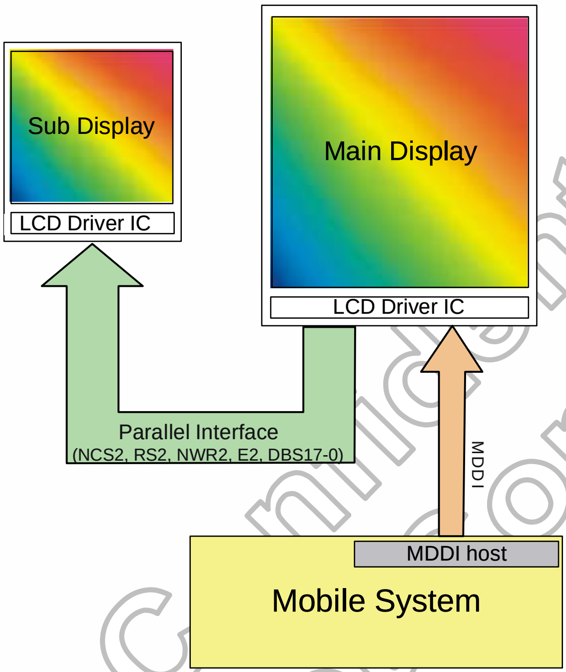


Figure 7. 27 Sub Panel Interface

Sub Panel Function

When the register access packet (R53h='00h') is received, then following register access packets or video stream packets are transferred to the sub panel via Sub Panel Interface. Sub panel selection address (R53h) can be changed by setting register SUB_SEL. The SUB_SEL value must be set to unused address in both Main/Sub Panel Driver IC. If video data is transferred to the sub panel driver IC via the Sub Panel Interface, additional RAM Index (default 0022h) is automatically generated by HX8352-A.

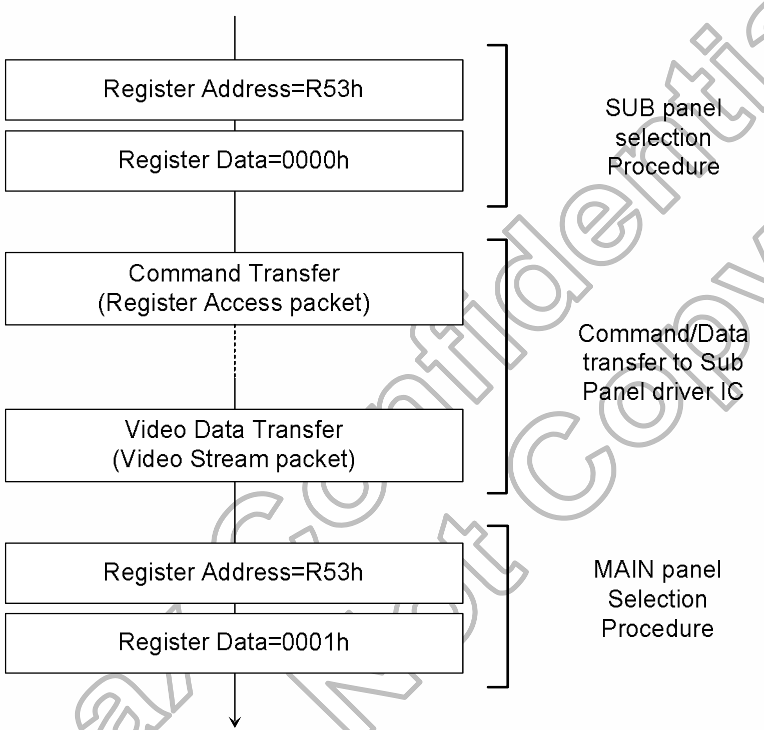


Figure 7. 28 Main/Sub Panel Selection Procedure

Sub Panel Interface Timing

The HX8352-A's Sub Panel Interface is supports two type panel (TFT and STN) and offer 18-/16-/9-/8-bit interface format (i80 and m68 system).

TFT Type Sub Panel Timing

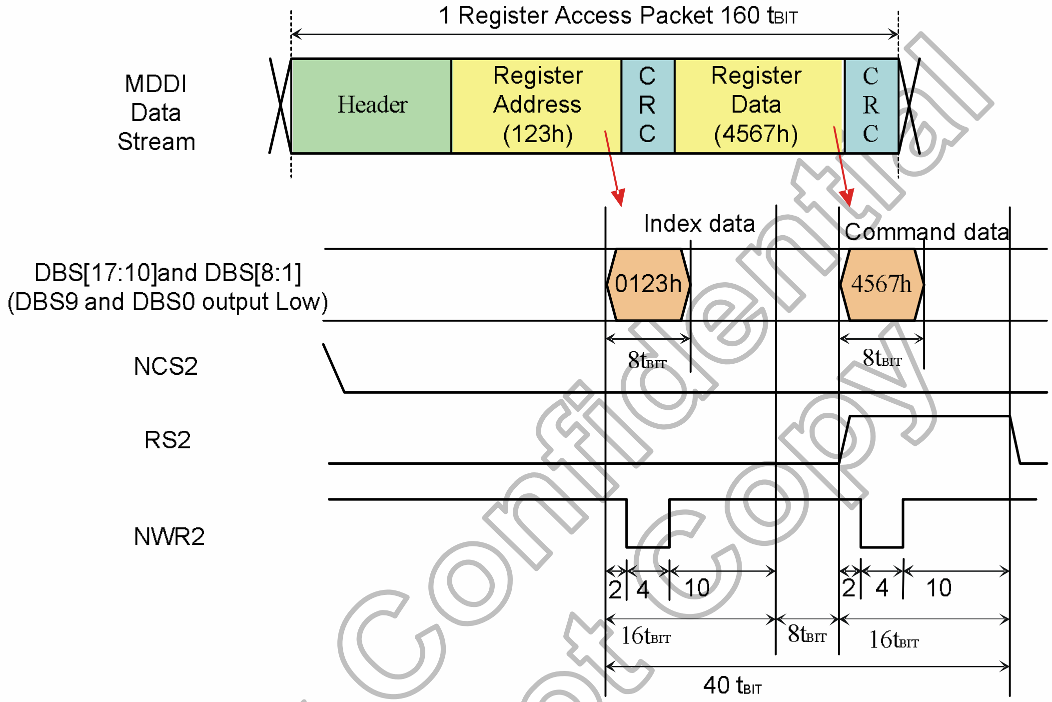


Figure 7. 29 18-/16-Bit Sub Panel Interface Register Access Data Timing for i80 Series TFT Sub Panel

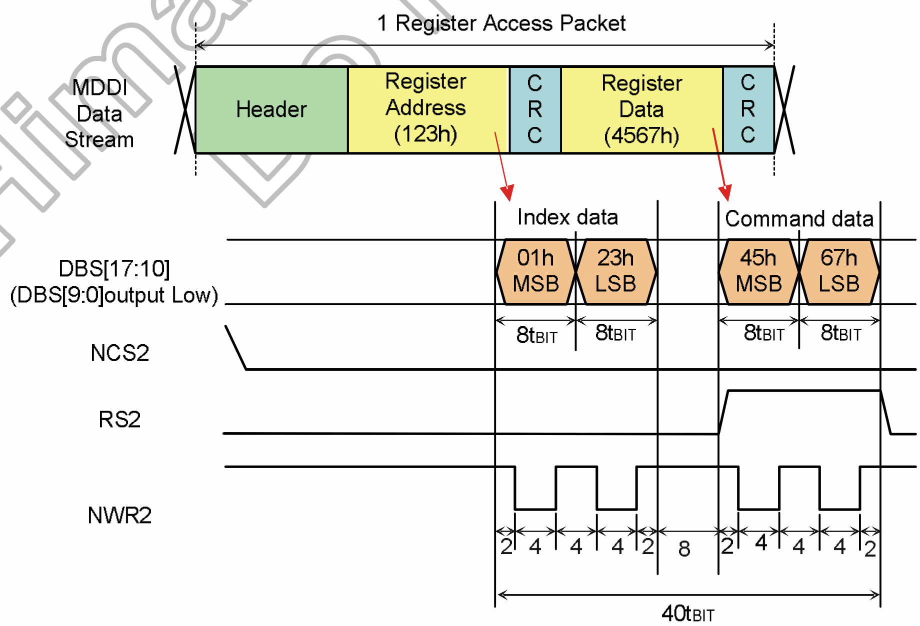


Figure 7. 30 9-/8-Bit Sub Panel Interface Register Access Data Timing for i80 Series TFT Sub Panel

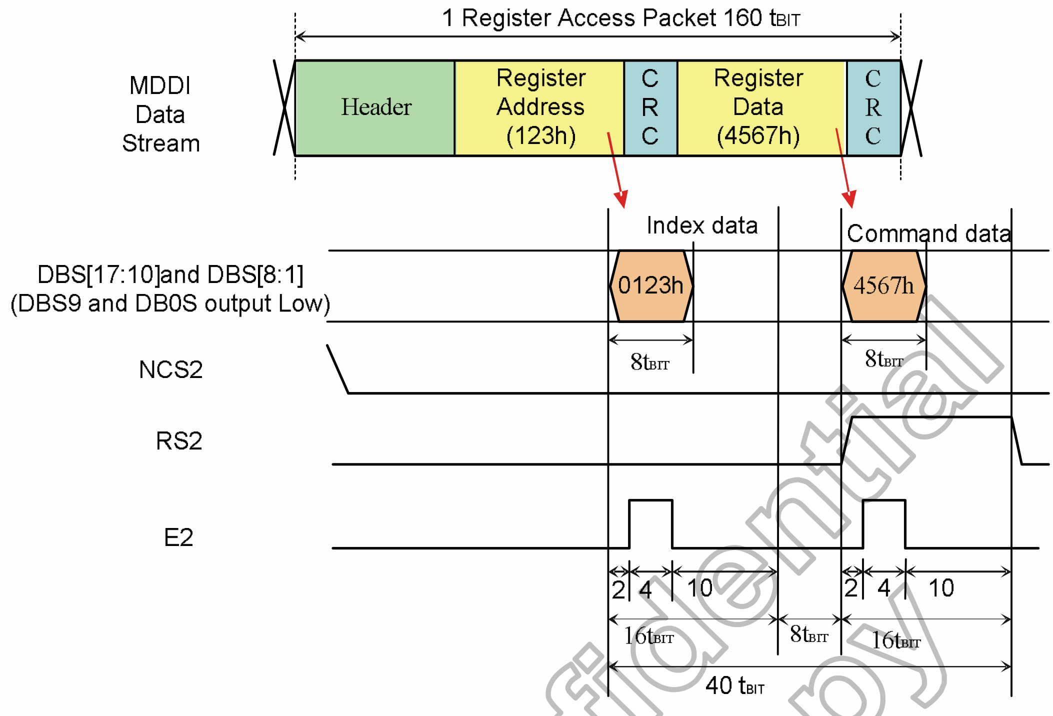


Figure 7. 31 18-/16-Bit Sub Panel Interface Register Access Data Timing for i80 Series TFT Sub Panel

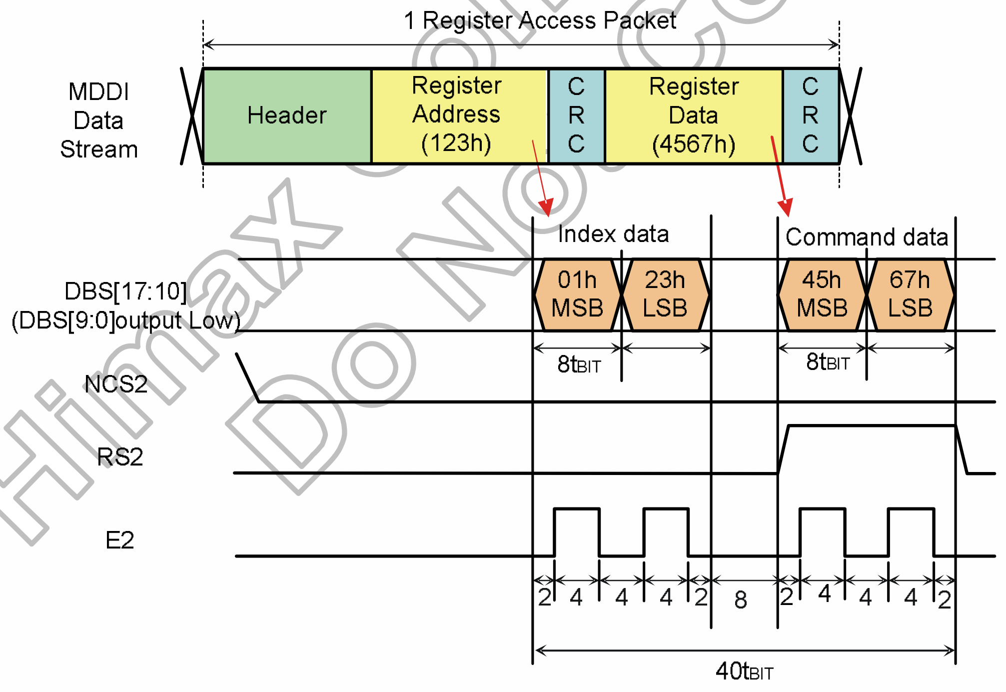


Figure 7. 32 9-/8-Bit Sub Panel Interface Register Access Data Timing for m68 Series TFT Sub Panel

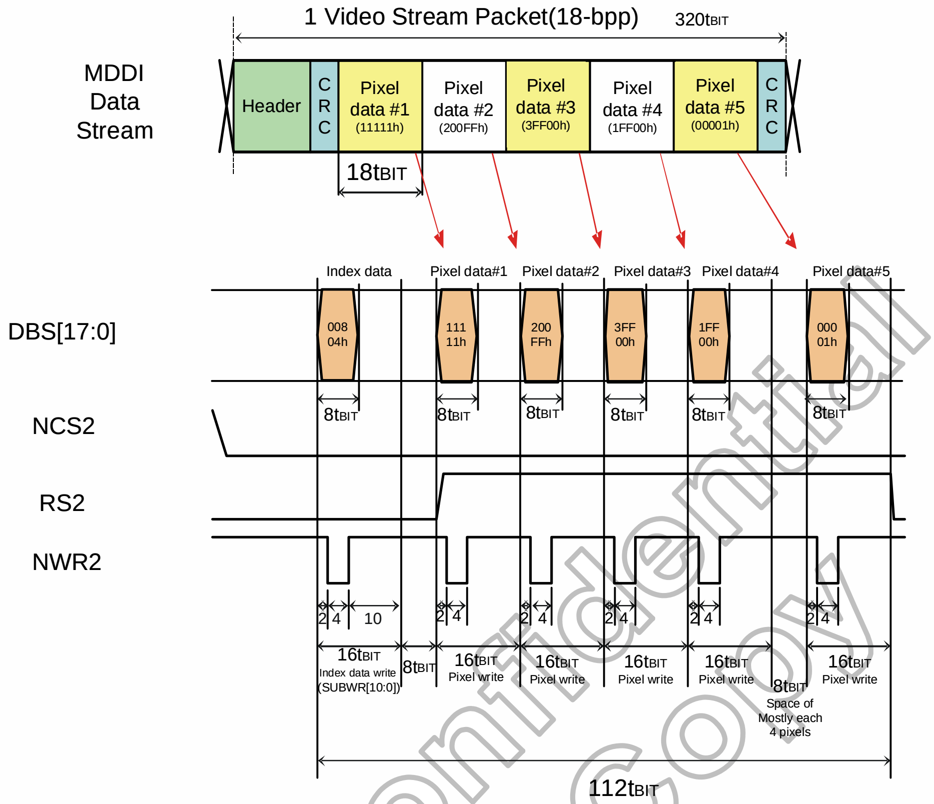


Figure 7. 33 18-Bit Sub Panel Interface Video Data Timing for i80 Series TFT Sub Panel

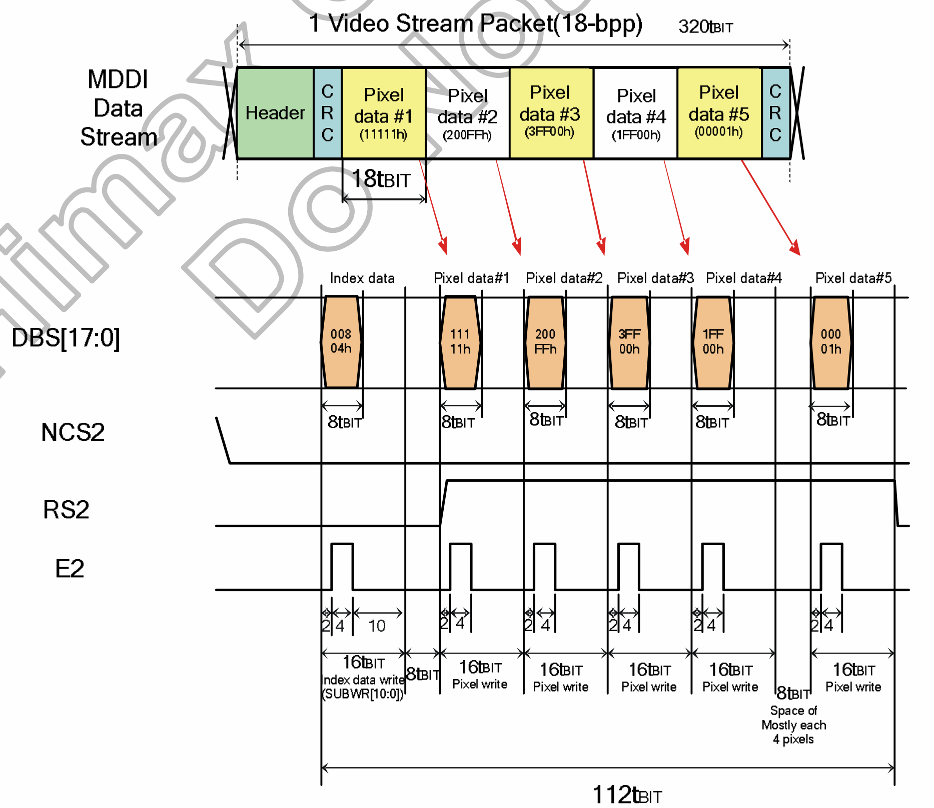


Figure 7. 34 18-Bit Sub Panel Interface Video Data Timing for M68 Series TFT Sub Panel

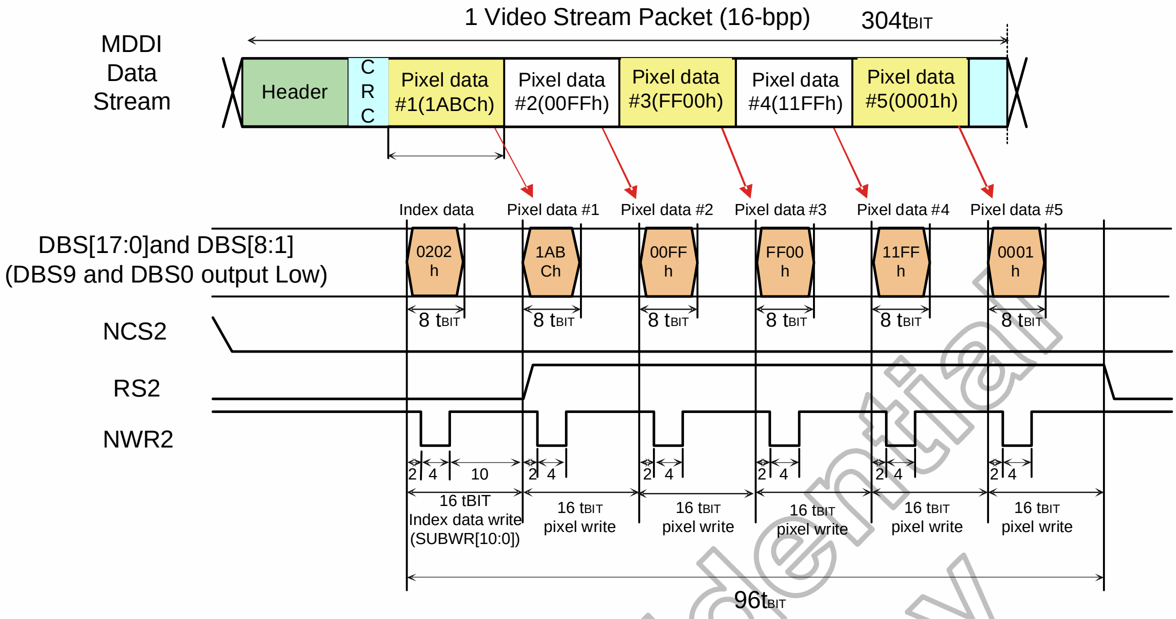


Figure 7. 35 16-Bit Sub Panel Interface Video Data Timing for I80 Series TFT Sub Panel

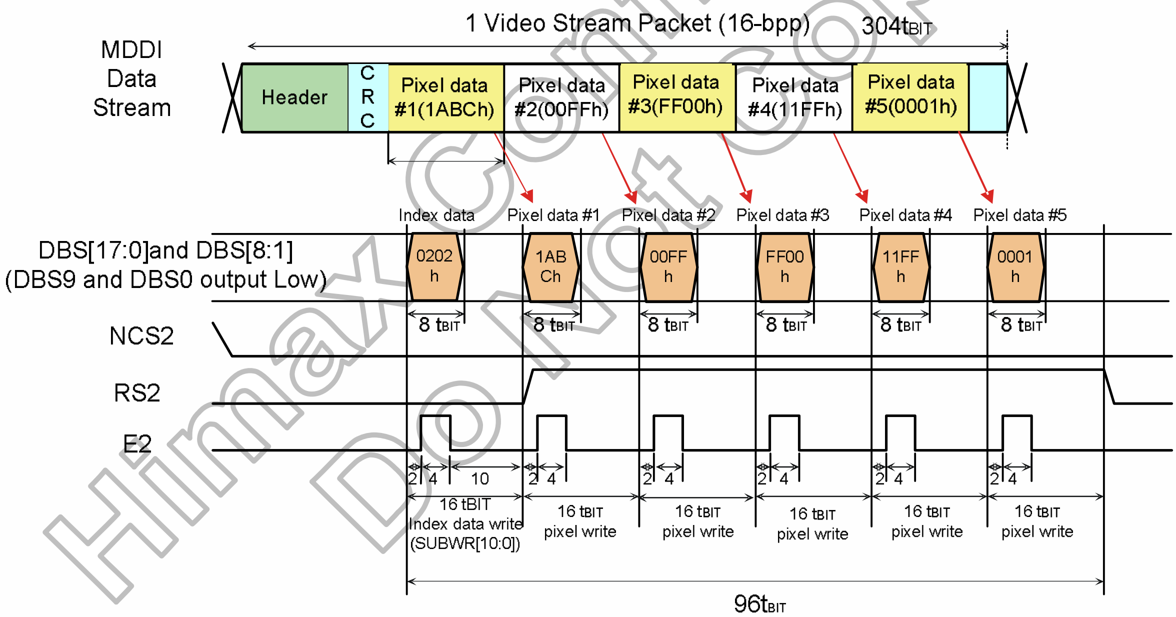


Figure 7. 36 16-Bit Sub Panel Interface Video Data Timing for m68 Series TFT Sub Panel

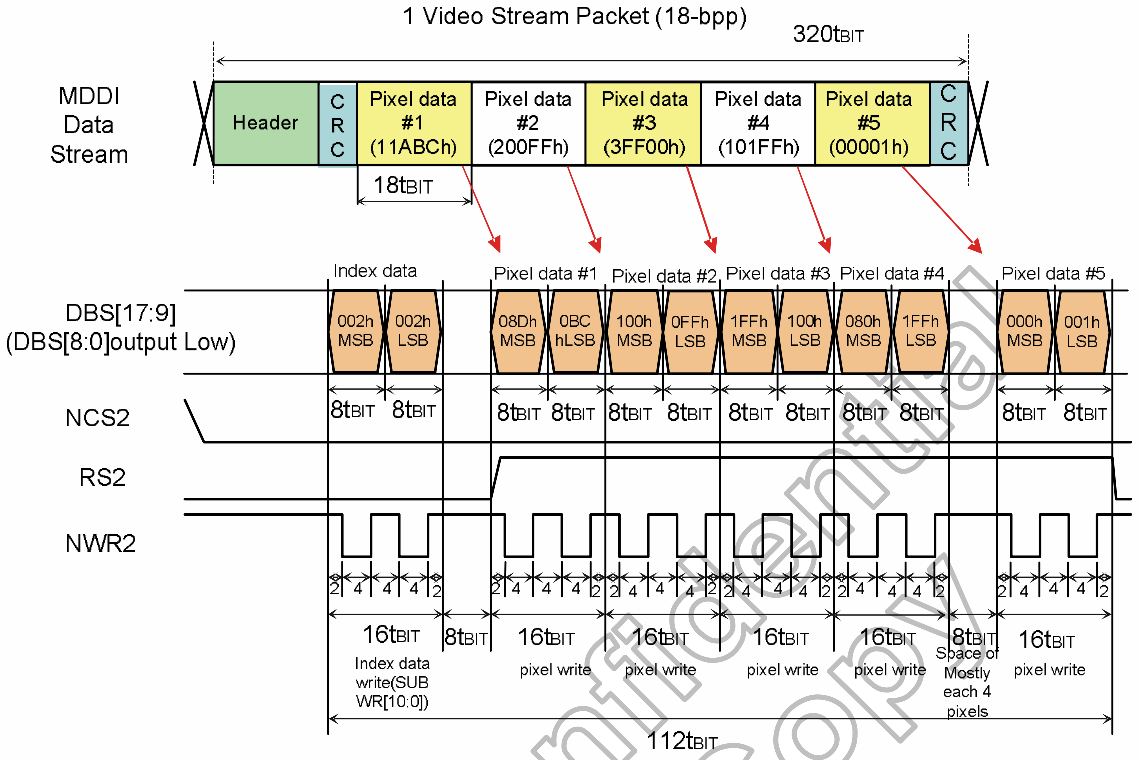


Figure 7. 37 9-Bit Sub Panel Interface Video Data Timing for i80 Series TFT Sub Panel

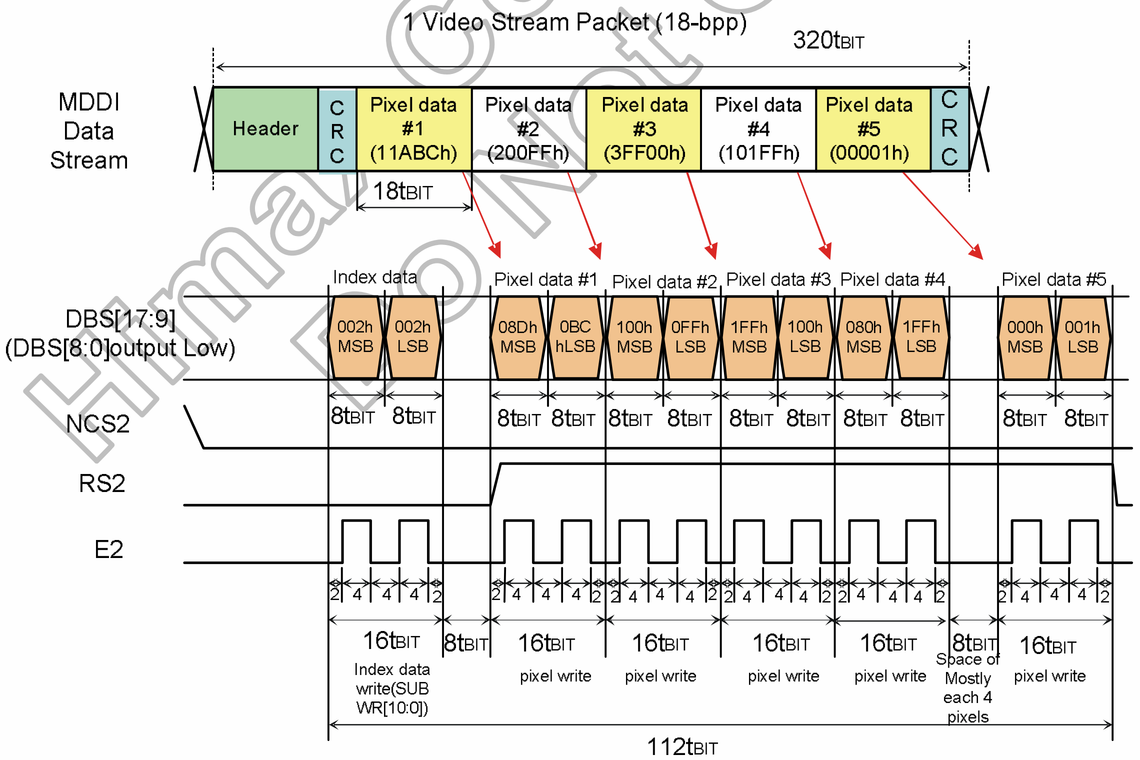


Figure 7. 38 9-Bit Sub Panel Interface Video Data Timing for m68 Series TFT Sub Panel

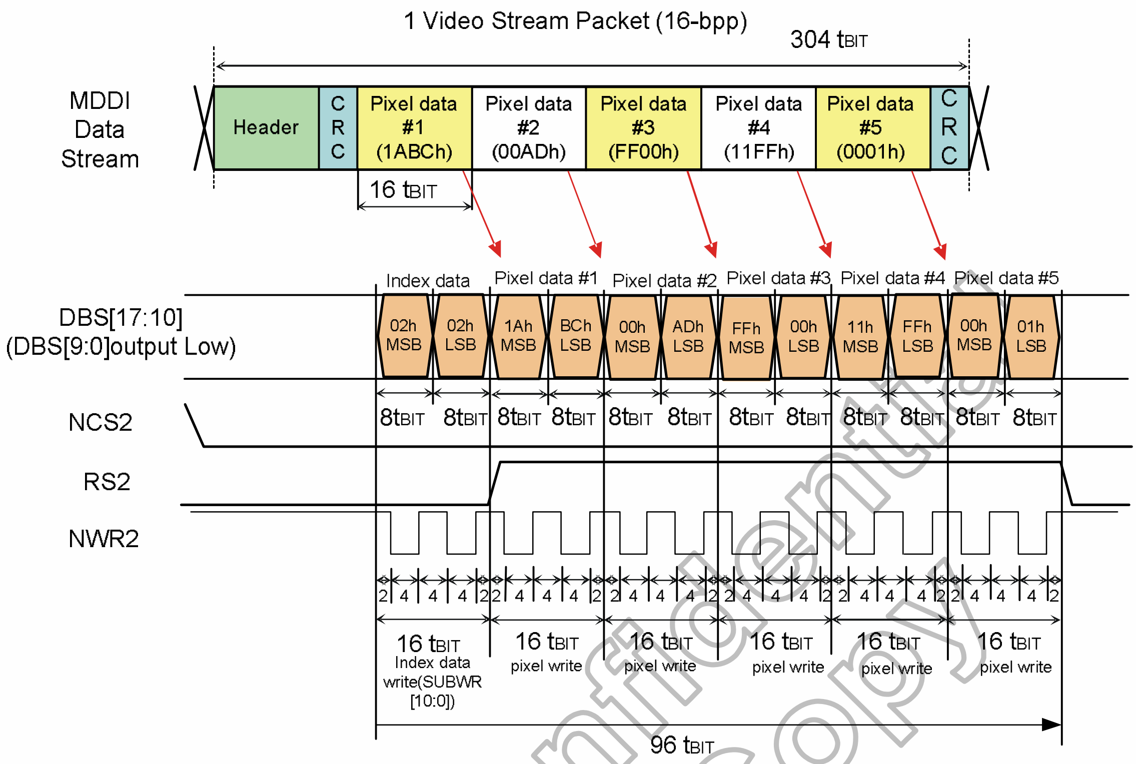


Figure 7. 39 8-Bit Sub Panel Interface Video Data Timing for i80 Series TFT Sub Panel

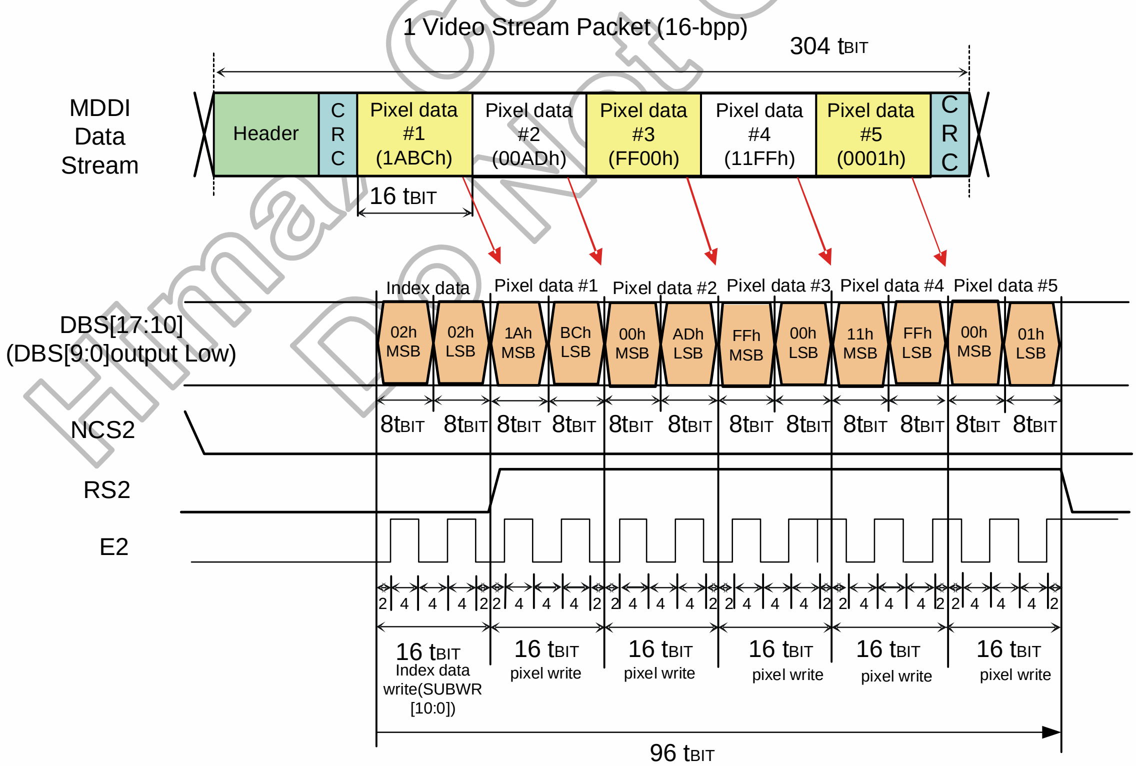


Figure 7. 40 8-Bit Sub Panel Interface Video Data Timing for m68 Series TFT Sub Panel

STN Type Sub Panel Timing

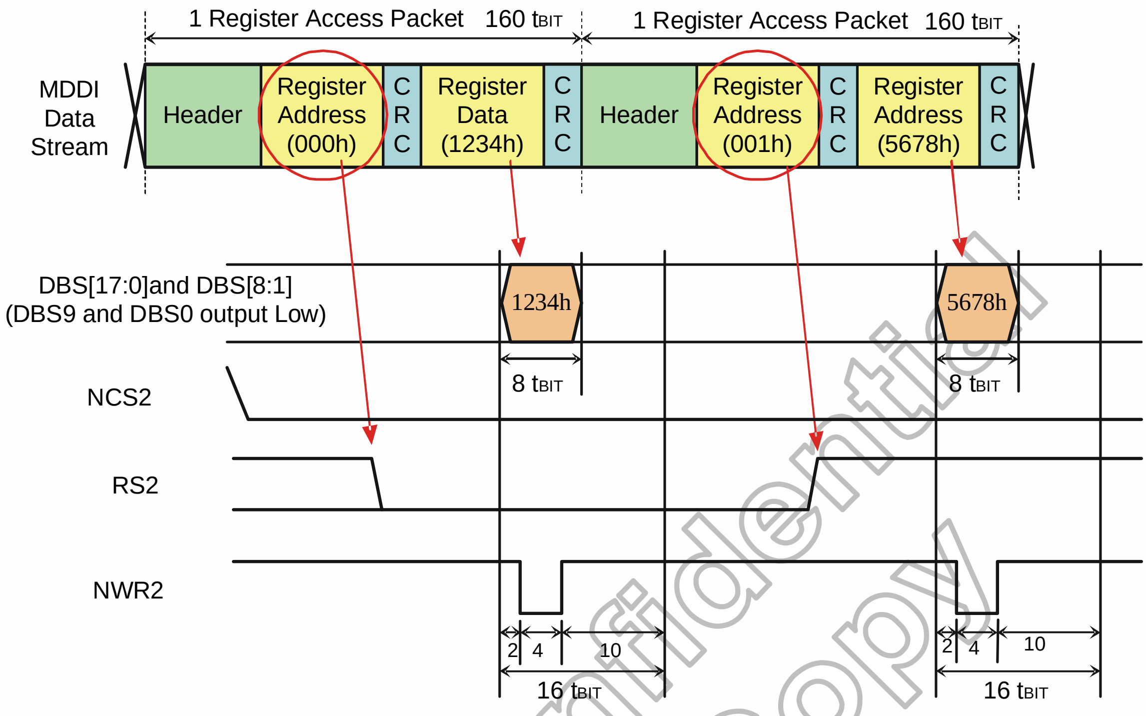


Figure 7. 41 18-/16-Bit Sub Panel Interface Register Access Data Timing for i80 Series STN Sub Panel

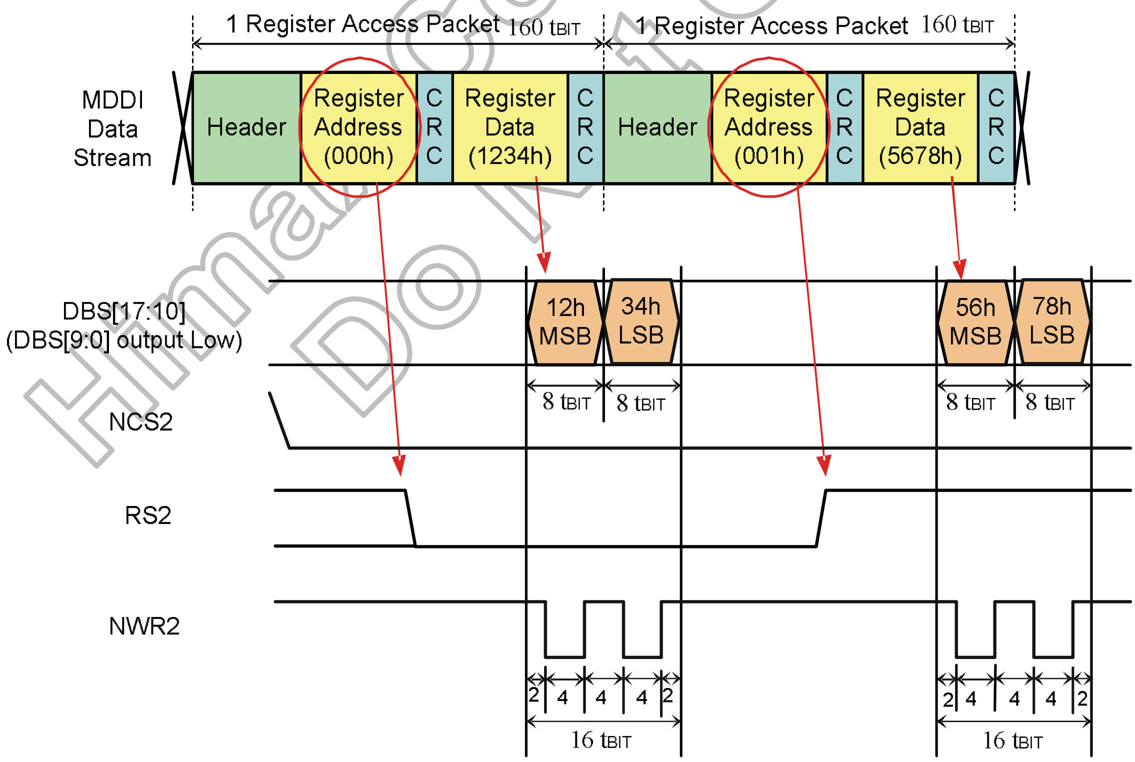


Figure 7. 42 9-/8-Bit Sub Panel Interface Register Access Data Timing for i80 Series STN Sub Panel

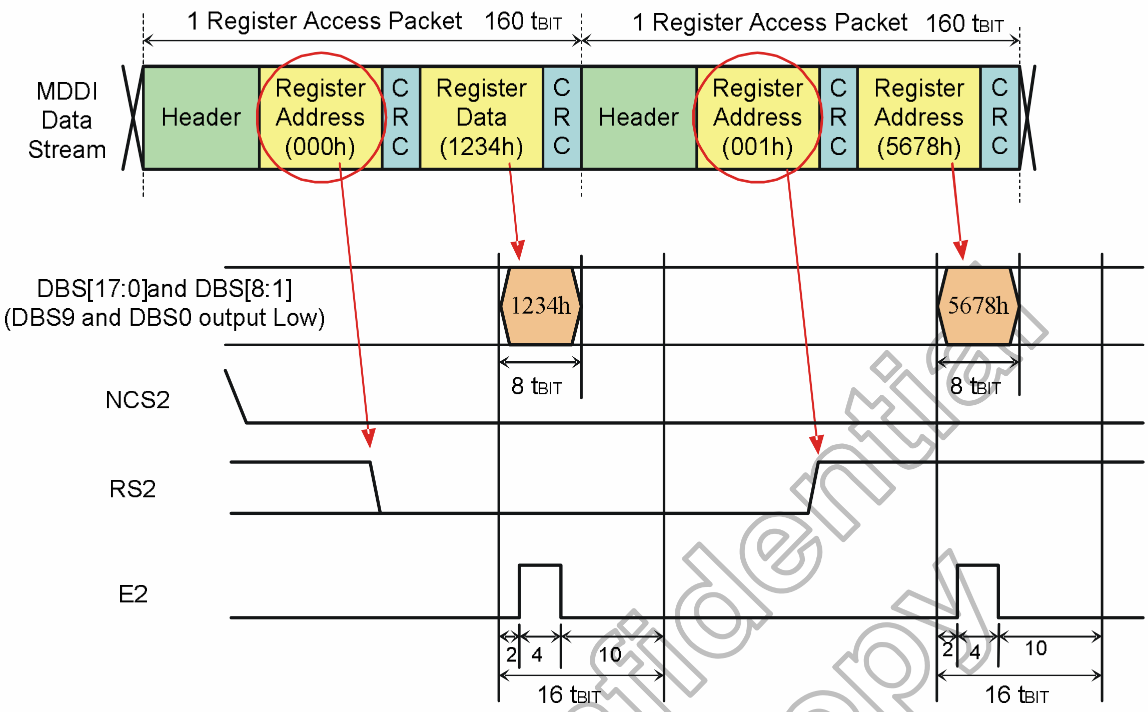


Figure 7. 43 18-/16-Bit Sub Panel Interface Register Access Data Timing for m68 Series STN Sub Panel

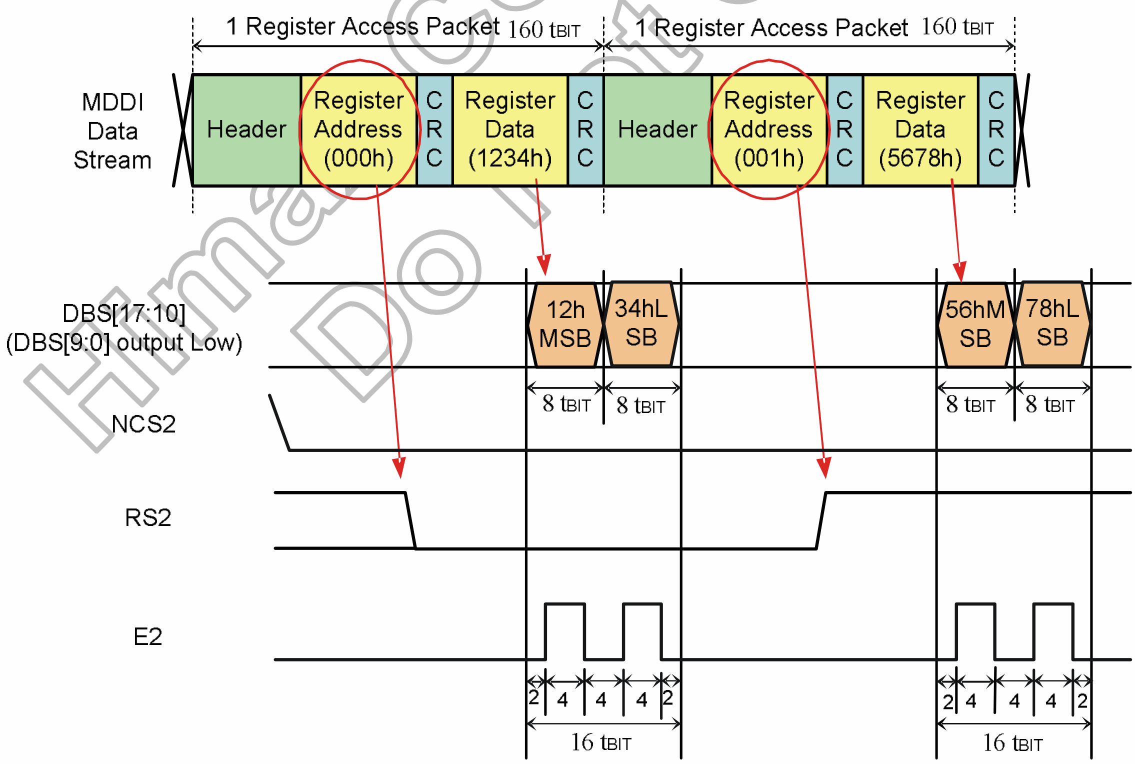
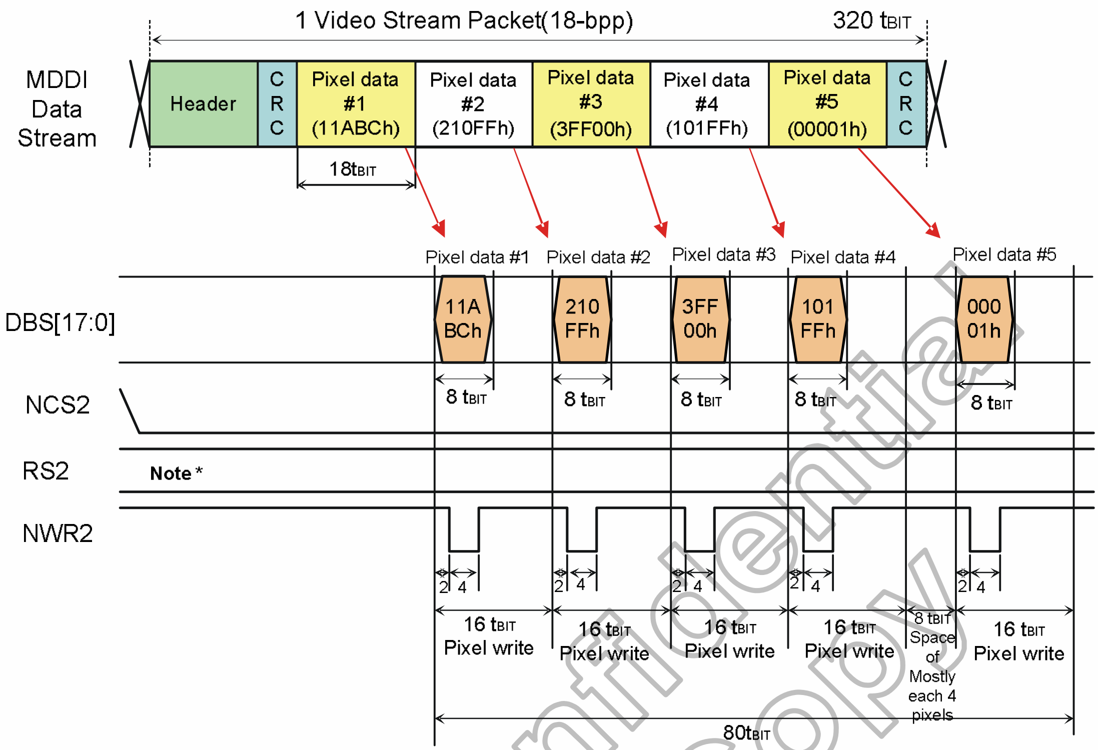
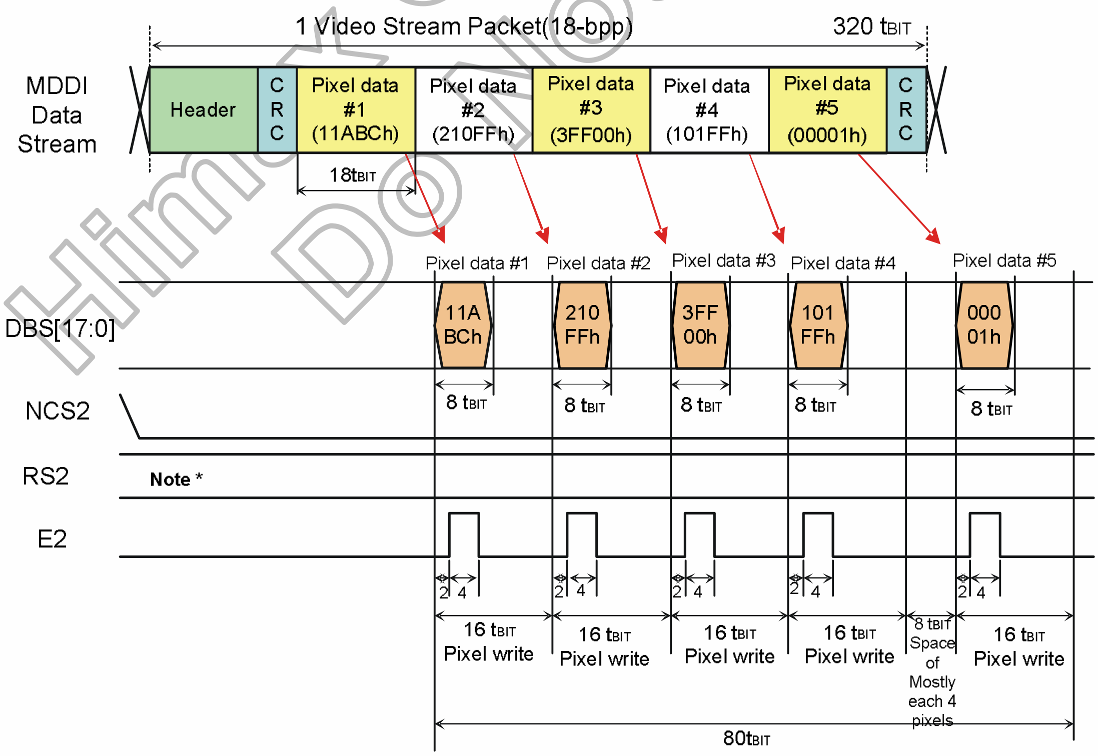


Figure 7. 44 9-/8-Bit Sub Panel Interface Register Access Data Timing for m68 Series STN Sub Panel



Note * : The status RS2 output is specified by SUBRS[1:0] bit of index:020h

Figure 7. 45 18-Bit Sub Panel Interface Video Data Timing for i80 Series STN Sub Panel



Note * : The status RS2 output is specified by SUBRS[1:0] bit of index:020h

Figure 7. 46 18-Bit Sub Panel Interface Video Data Timing for m68 Series STN Sub Panel

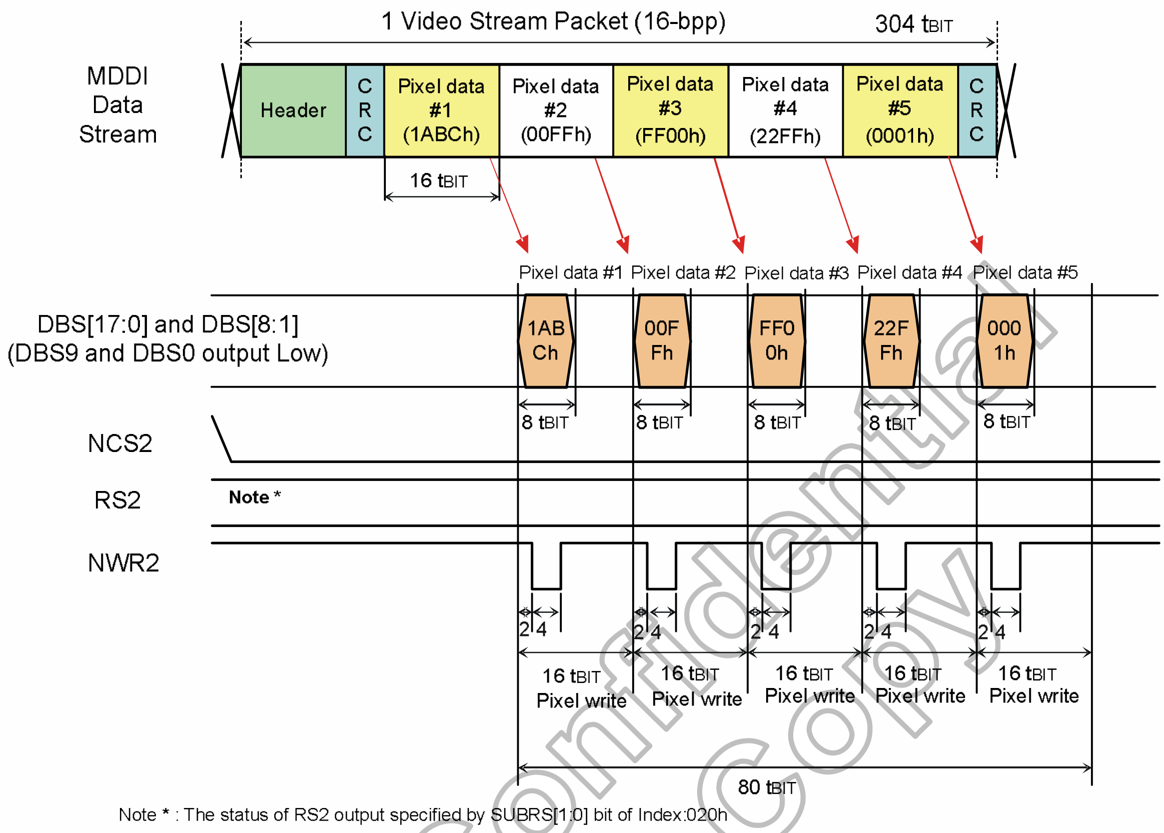


Figure 7. 47 16-Bit Sub Panel Interface Video Data Timing for i80 Series STN Sub Panel

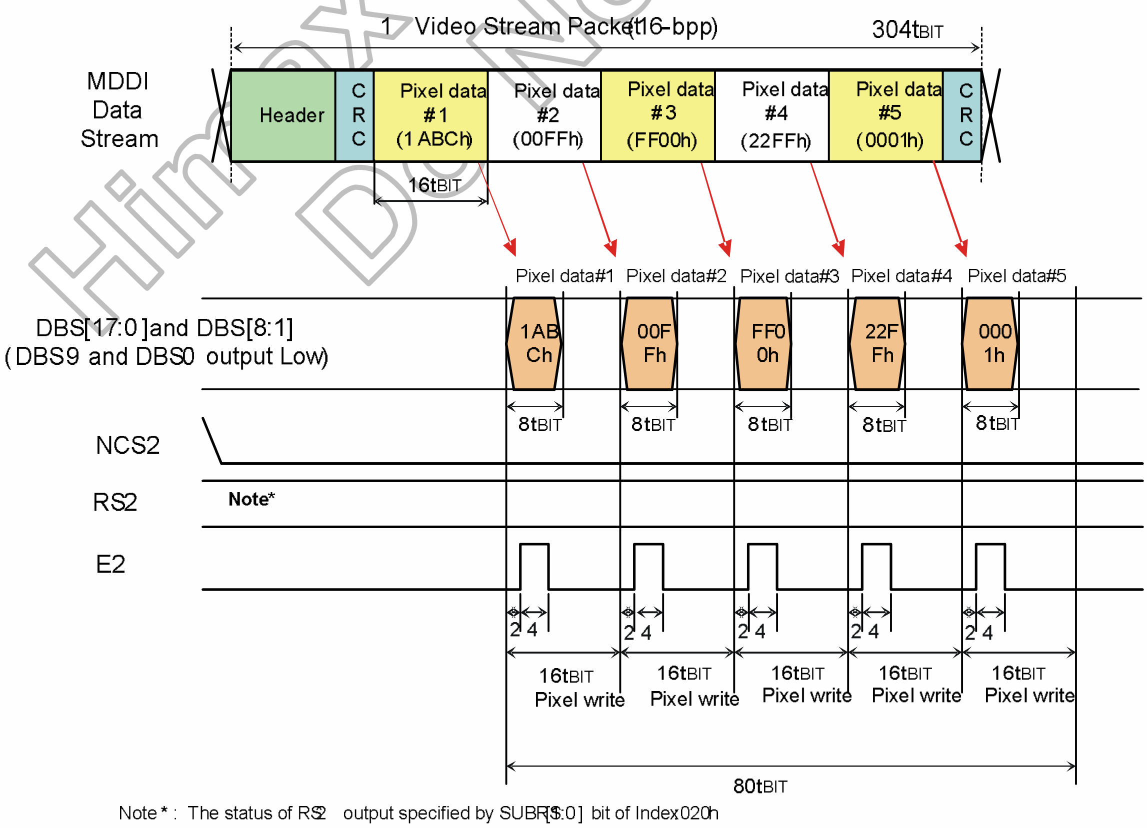


Figure 7. 48 16-Bit Sub Panel Interface Video Data Timing for m68 Series STN Sub Panel

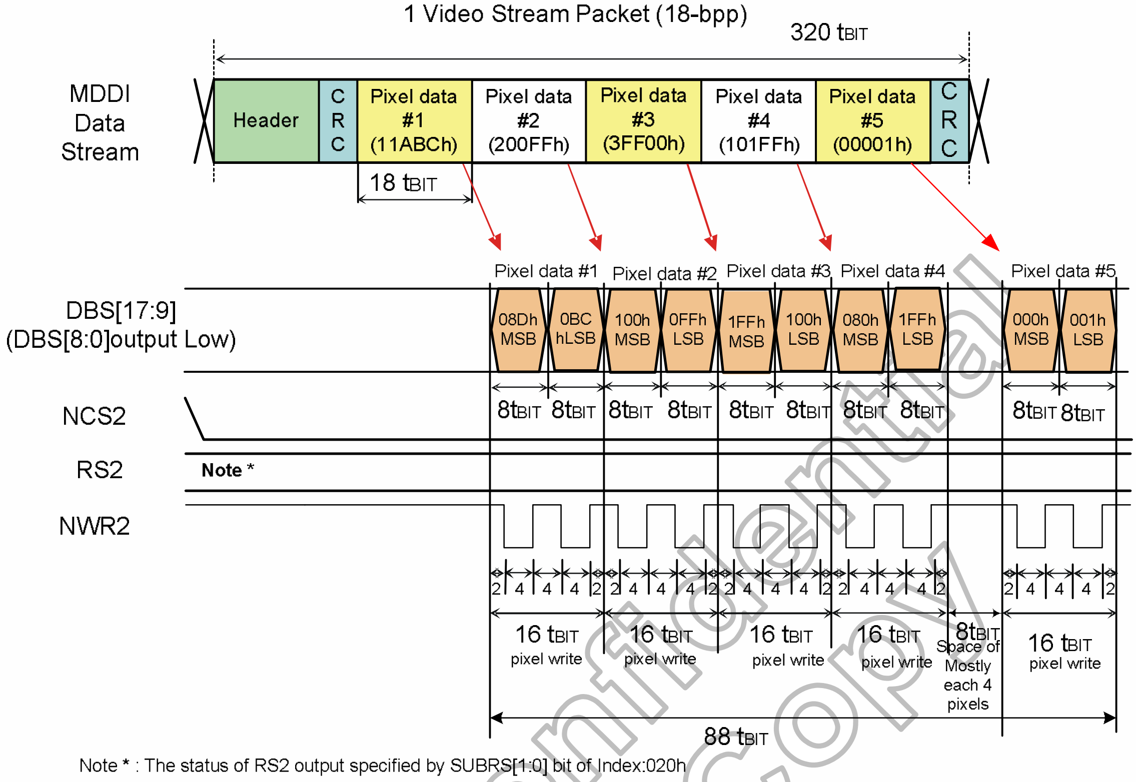


Figure 7. 49 9-Bit Sub Panel Interface Video Data Timing for i80 Series STN Sub Panel

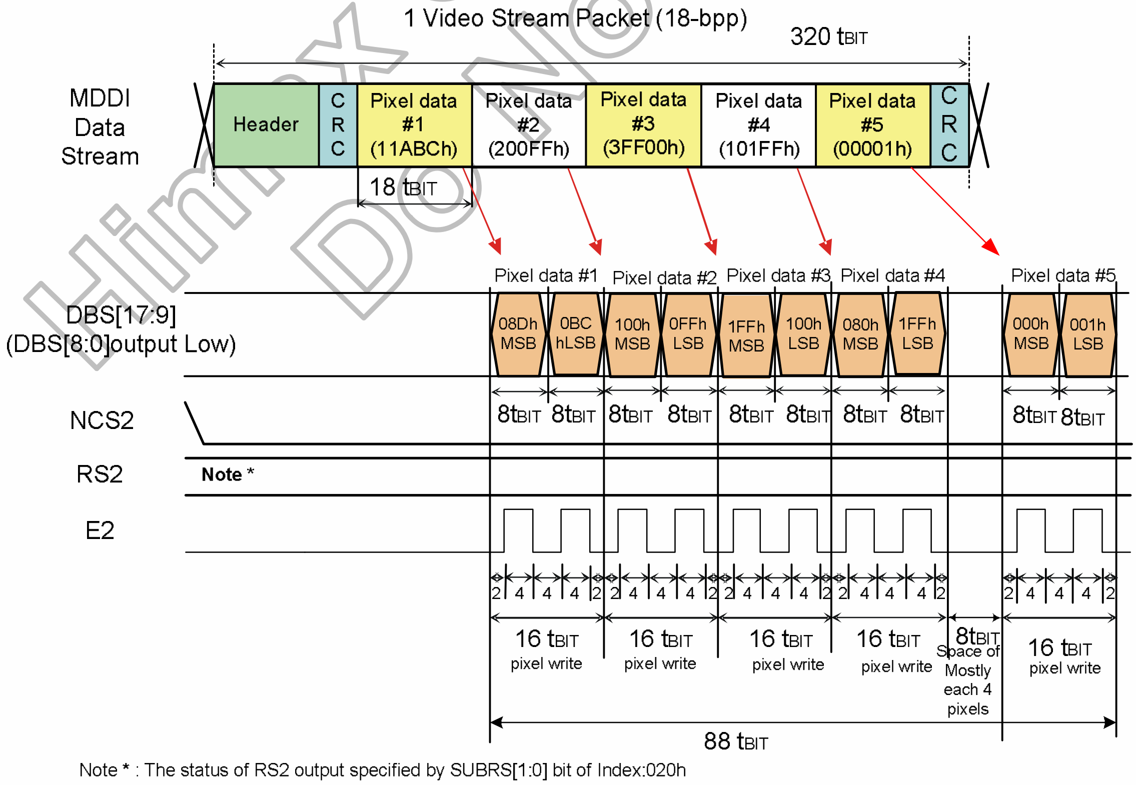
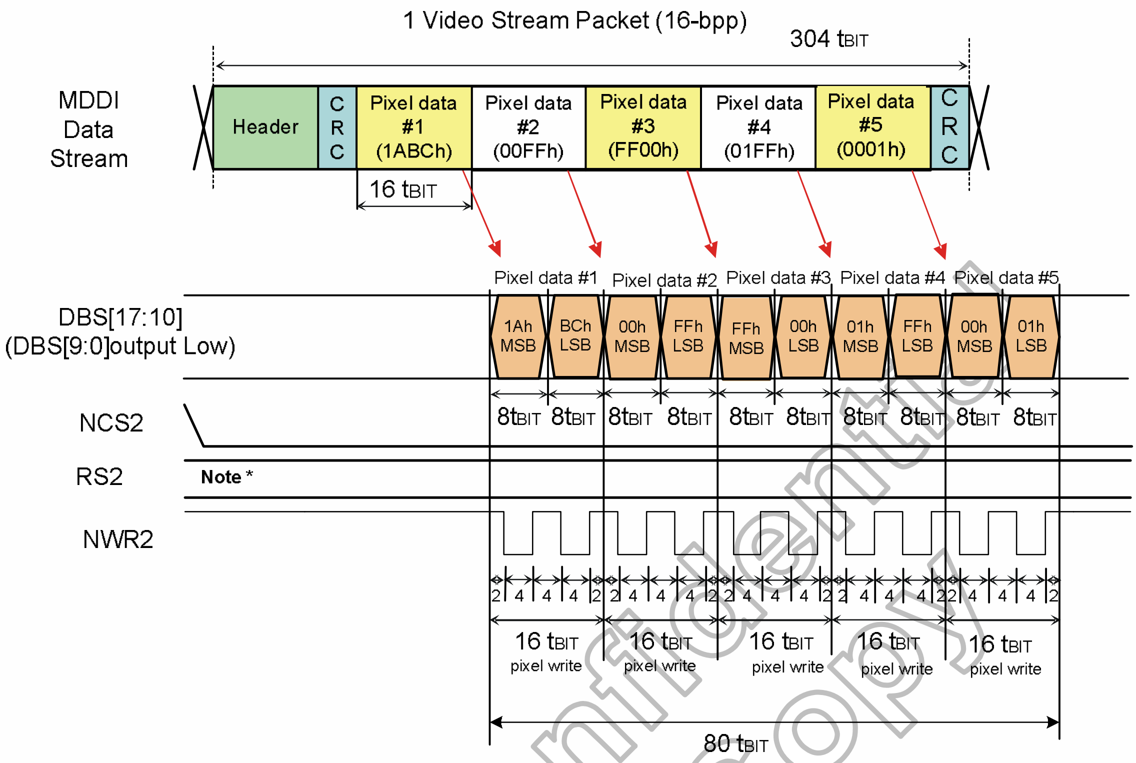
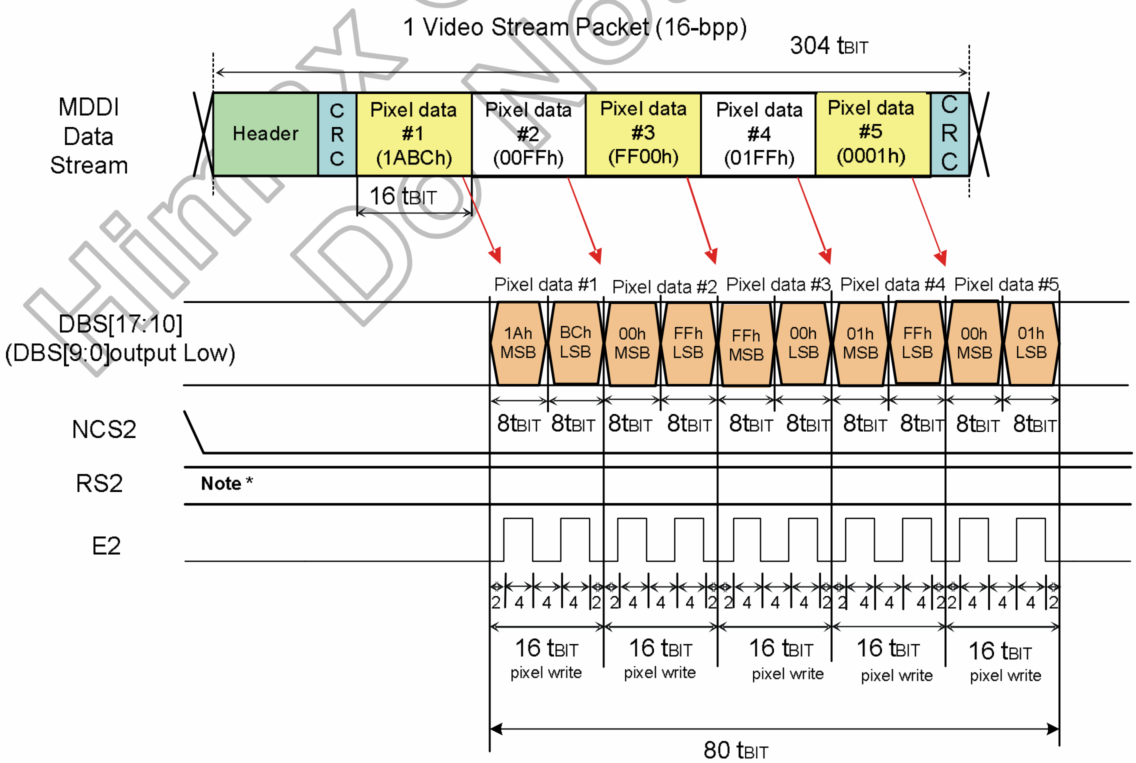


Figure 7. 50 9-Bit Sub Panel Interface Video Data Timing for m68 Series STN Sub Panel



Note * : The status of RS2 output specified by SUBRS[1:0] bit of Index:020h

Figure 7. 51 8-Bit Sub Panel Interface Video Data Timing for i80 Series STN Sub Panel



Note * : The status of RS2 output specified by SUBRS[1:0] bit of Index:020h

Figure 7. 52 8-Bit Sub Panel Interface Video Data Timing for m68 Series STN Sub Panel

7.5 Initial Procedure

7.5.1 Power Supply Setting Flow

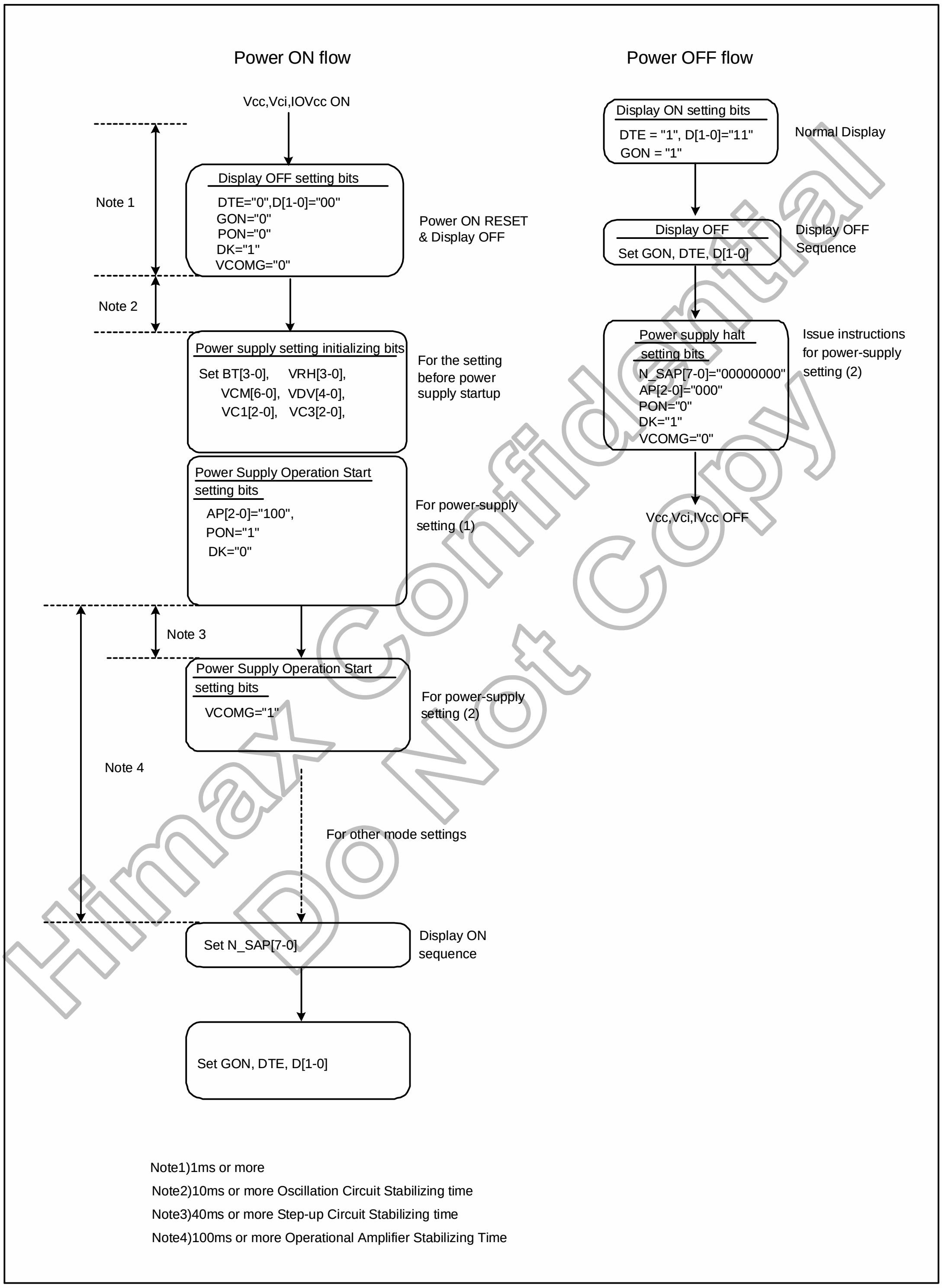


Figure 7. 53 Power Supply Setting Flow

7.5.2 Display on/off Setting Flow

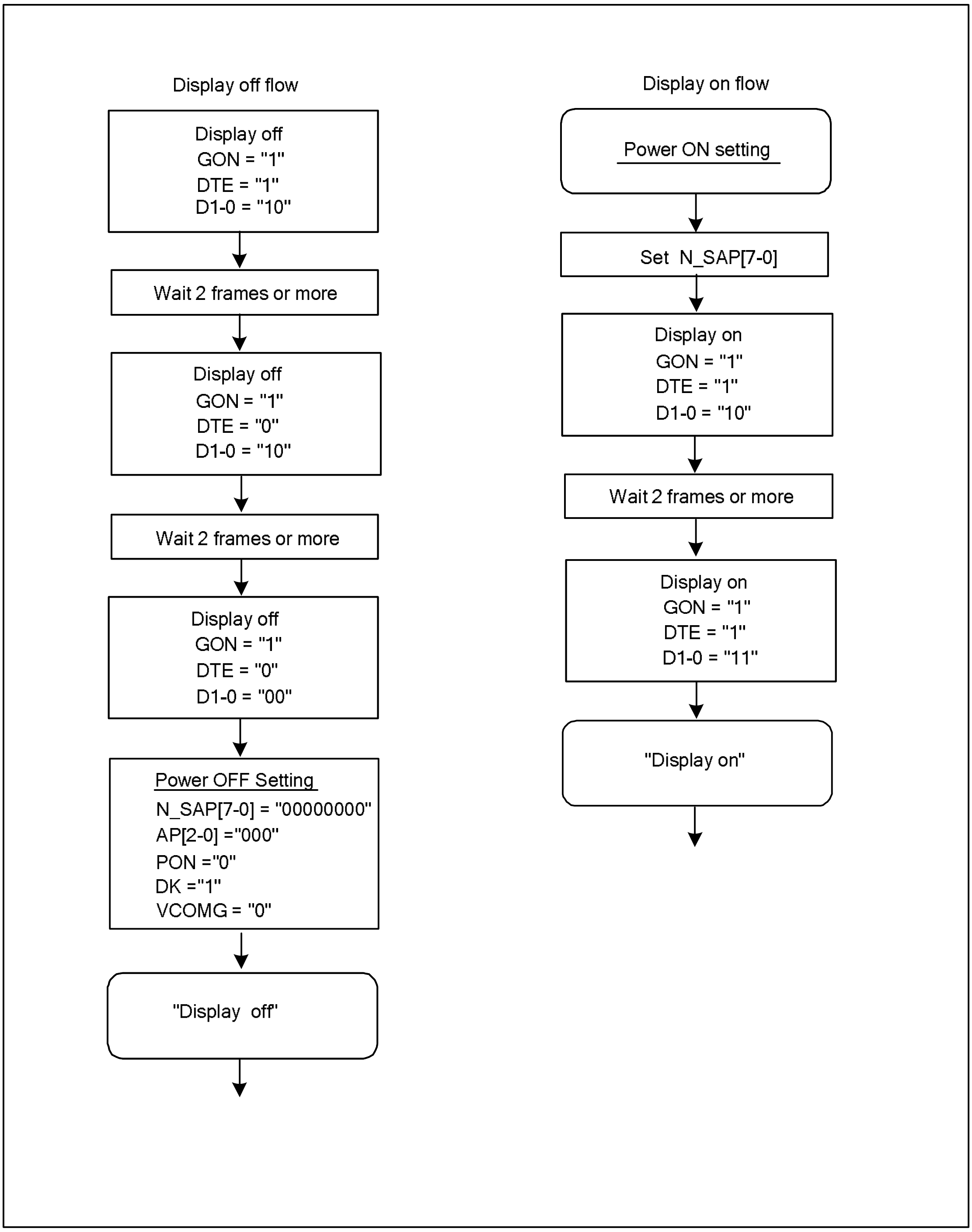


Figure 7. 54 Display On/Off Setting Flow

7.5.3 Standby Mode Setting Flow

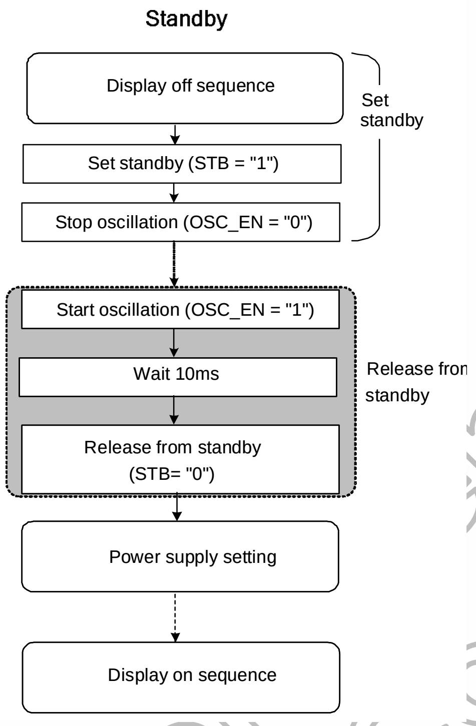


Figure 7. 55 Standby Mode Setting Flow

7.6 Initial code for reference
7.6.1 The reference setting of Normal Display for Register-Content Interface Mode
7.6.1.1 The reference setting of CMO 3.0" Panel

```
void HX8352_Init_CMO30(void)
{
      RESET();
      DelayX1ms(150);

        Set_LCD_8B_REG(0x83,0x02);            // TESTM=1
        Set_LCD_8B_REG(0x85,0x03);            // VDC_SEL=011.
        Set_LCD_8B_REG(0x8B,0x00);            // STBA[15:8]=0x00
        Set_LCD_8B_REG(0x8C,0x93);            // STBA[7]=1, STBA[5:4]=01, STBA[1:0]=11
        Set_LCD_8B_REG(0x91,0x01);            // DCDC_SYNC=1
        Set_LCD_8B_REG(0x83,0x00);            // TESTM=0

// Gamma Setting
        Set_LCD_8B_REG(0x3E,0xF0);
        Set_LCD_8B_REG(0x3F,0x07);
        Set_LCD_8B_REG(0x40,0x00);
        Set_LCD_8B_REG(0x41,0x43);
        Set_LCD_8B_REG(0x42,0x16);
        Set_LCD_8B_REG(0x43,0x16);
        Set_LCD_8B_REG(0x44,0x43);
        Set_LCD_8B_REG(0x45,0x77);
        Set_LCD_8B_REG(0x46,0x00);
        Set_LCD_8B_REG(0x47,0x1E);
        Set_LCD_8B_REG(0x48,0x0F);
        Set_LCD_8B_REG(0x49,0x00);

// Power Supply Setting
        Set_LCD_8B_REG(0x17,0x91);            // RADJ=0110, OSC_EN=1
        Set_LCD_8B_REG(0x23,0x01);            // TE 0n
        Set_LCD_8B_REG(0x2B,0xF9);            // N_DCDC=0xF9.

        DelayX1ms(10);

        Set_LCD_8B_REG(0x1B,0x14);            // BT=0001, AP=100
        Set_LCD_8B_REG(0x1A,0x11);            // VC3=001, VC1=001 (VLCD/DDVDH)=6.45V)
        Set_LCD_8B_REG(0x1C,0x0D);            // VRH=1110 (VREG1=6.0V)
        Set_LCD_8B_REG(0x1F,0x27);            // VCM=010_1011
        DelayX1ms(20);

        Set_LCD_8B_REG(0x19,0x0A);            // GASENB=0, PON=0, DK=1, XDK=0,
                                              // VLCD_TRI=1, STB=0
        Set_LCD_8B_REG(0x19,0x1A);            // GASENB=0, PON=1, DK=1, XDK=0,
                                              // VLCD_TRI=1, STB=0
        DelayX1ms(40);

        Set_LCD_8B_REG(0x19,0x12);            // GASENB=0, PON=1, DK=0, XDK=0,
                                              // VLCD_TRI=1, STB=0
                                              // VLCD=2XVCI by 2 CAPs
        DelayX1ms(40);
        Set_LCD_8B_REG(0x1E,0x2C);            // VCOMG=1, VDV=0_1110
        DelayX1ms(100);
```

```
// DGC Function Enable
        Set_LCD_8B_REG(0x5A,0x01);
        DGC_PA_REG(0x5C);

// Display ON Setting
        Set_LCD_8B_REG(0x3C,0xC0);              // N_SAP=1100 000
        Set_LCD_8B_REG(0x3D,0xC0);              // I_SAP =1100 0000
        Set_LCD_8B_REG(0x34,0x38);              // EQS=1000 0111
        Set_LCD_8B_REG(0x35,0x38);              // EQP=0011 1000
        Set_LCD_8B_REG(0x24,0x38);              // GON=1, DTE=1, D=10
        DelayX1ms(40);
        Set_LCD_8B_REG(0x24,0x3C);              // GON=1, DTE=1, D=11

        Set_LCD_8B_REG(0x16,0x08);              // BGR=1
        Set_LCD_8B_REG(0x01,0x02);              // INVON=0, NORNO=1
        Set_LCD_8B_REG(0x55,0x00);
}

void DGC_PA_REG(unsigned char ADDR)
{
        unsigned char i;

        M51_CTRL_LCD_nCS = 0;
        M51_CTRL_LCD_RS = 0;
        WR_8B_FORMAT(ADDR);                     //ADDR=0x5C.
        M51_CTRL_LCD_RS = 1;

      for( i=0; i<=2; i++)
      {
          WR_8B_FORMAT(0x01 );
            WR_8B_FORMAT(0x01 );
            WR_8B_FORMAT(0x01 );
            WR_8B_FORMAT(0x01 );
            WR_8B_FORMAT(0x07 );
            WR_8B_FORMAT(0x11 );
            WR_8B_FORMAT(0x17 );
            WR_8B_FORMAT(0x1B );
            WR_8B_FORMAT(0x1E );
            WR_8B_FORMAT(0x22 );
            WR_8B_FORMAT(0x27 );
            WR_8B_FORMAT(0x2C );
            WR_8B_FORMAT(0x30 );
            WR_8B_FORMAT(0x35 );
            WR_8B_FORMAT(0x39 );
            WR_8B_FORMAT(0x3D );
            WR_8B_FORMAT(0x41 );
            WR_8B_FORMAT(0x45 );
            WR_8B_FORMAT(0x49 );
            WR_8B_FORMAT(0x4D );
            WR_8B_FORMAT(0x51 );
            WR_8B_FORMAT(0x54 );
            WR_8B_FORMAT(0x58 );
            WR_8B_FORMAT(0x5B );
            WR_8B_FORMAT(0x5F );
            WR_8B_FORMAT(0x62 );
            WR_8B_FORMAT(0x66 );
```

```
        WR_8B_FORMAT(0x69 );
        WR_8B_FORMAT(0x6D );
        WR_8B_FORMAT(0x71 );
        WR_8B_FORMAT(0x74 );
        WR_8B_FORMAT(0x78 );
        WR_8B_FORMAT(0x7C );
        WR_8B_FORMAT(0x80 );
        WR_8B_FORMAT(0x84 );
        WR_8B_FORMAT(0x88 );
        WR_8B_FORMAT(0x8D );
        WR_8B_FORMAT(0x91 );
        WR_8B_FORMAT(0x96 );
        WR_8B_FORMAT(0x9A );
        WR_8B_FORMAT(0x9F );
        WR_8B_FORMAT(0xA4 );
        WR_8B_FORMAT(0xA9 );
        WR_8B_FORMAT(0xAD );
        WR_8B_FORMAT(0xB0 );
        WR_8B_FORMAT(0xB4 );
        WR_8B_FORMAT(0xB7 );
        WR_8B_FORMAT(0xBB );
        WR_8B_FORMAT(0xBF );
        WR_8B_FORMAT(0xC3 );
        WR_8B_FORMAT(0xC7 );
        WR_8B_FORMAT(0xCC );
        WR_8B_FORMAT(0xD0 );
        WR_8B_FORMAT(0xD6 );
        WR_8B_FORMAT(0xDB );
        WR_8B_FORMAT(0xDF );
        WR_8B_FORMAT(0xE3 );
        WR_8B_FORMAT(0xE7 );
        WR_8B_FORMAT(0xEB );
        WR_8B_FORMAT(0xEE );
        WR_8B_FORMAT(0xF1 );
        WR_8B_FORMAT(0xF5 );
        WR_8B_FORMAT(0xF7 );
        WR_8B_FORMAT(0xFC );
    }

    M51_CTRL_LCD_nCS = 1;
}
```

7.6.1.2 The reference setting of CMO 2.8" Panel

```
void HX8352_Init_CMO28(void)
{
    RESET();
    DelayX1ms(150);

        Set_LCD_8B_REG(0x83,0x02);          // TESTM=1
        Set_LCD_8B_REG(0x85,0x03);          // VDC_SEL=011.
        Set_LCD_8B_REG(0x8B,0x01);          // STBA[15:8]=0x01
        Set_LCD_8B_REG(0x8C,0x93);          // STBA[7]=1, STBA[5:4]=01,
                                            //STBA[1:0=11
        Set_LCD_8B_REG(0x91,0x01);          // DCDC_SYNC=1
        Set_LCD_8B_REG(0x83,0x00);          // TESTM=0

// Gamma Setting
        Set_LCD_8B_REG(0x3E,0xA5);
        Set_LCD_8B_REG(0x3F,0x52);
        Set_LCD_8B_REG(0x40,0x00);
        Set_LCD_8B_REG(0x41,0x36);
        Set_LCD_8B_REG(0x42,0x00);
        Set_LCD_8B_REG(0x43,0x77);
        Set_LCD_8B_REG(0x44,0x15);
        Set_LCD_8B_REG(0x45,0x76);
        Set_LCD_8B_REG(0x46,0x01);
        Set_LCD_8B_REG(0x47,0x00);
        Set_LCD_8B_REG(0x48,0x00);
        Set_LCD_8B_REG(0x49,0x02);

 // Power Supply Setting
        Set_LCD_8B_REG(0x17,0x91);          // RADJ=1100, OSC_EN=1
        Set_LCD_8B_REG(0x23,0x01);          // TE 0n
        Set_LCD_8B_REG(0x2B,0xF9);          // N_DCDC=0xF9.
        DelayX1ms(10);

        Set_LCD_8B_REG(0x1B,0x14);          // BT=0001, AP=100
        Set_LCD_8B_REG(0x1A,0x11);          // VC3=001, VC1=001 (VLCD=6V)
        Set_LCD_8B_REG(0x1C,0x0D);          // VRH=1101 (VREG1=5.5V)
        Set_LCD_8B_REG(0x1F,0x3A);          // VCM=011_1010
        DelayX1ms(20);

        Set_LCD_8B_REG(0x19,0x0A);          // GASENB=0, PON=0, DK=1, XDK=0,
                                            // VLCD_TRI=1, STB=0
        Set_LCD_8B_REG(0x19,0x1A);          // GASENB=0, PON=1, DK=1, XDK=0,
                                            //VLCD_TRI=1, STB=0
        DelayX1ms(40);

        Set_LCD_8B_REG(0x19,0x12);          // GASENB=0, PON=1, DK=0, XDK=0,
                                            //VLCD_TRI=1, STB=0,
                                            //VLCD=2XVCI by 2 CAPs
        DelayX1ms(40);
        Set_LCD_8B_REG(0x1E,0x2D);          // VCOMG=1, VDV=0_1101
        DelayX1ms(100);
```

```
// DGC Function Enable
        Set_LCD_8B_REG(0x5A,0x01);
        DGC_PA_REG(0x5C);

// Display ON Setting
        Set_LCD_8B_REG(0x3C,0xC0);     // N_SAP=1100 0000
        Set_LCD_8B_REG(0x3D,0xC0);     // I_SAP=1100 0000
        Set_LCD_8B_REG(0x34,0x38);     // EQS=1000 0111
        Set_LCD_8B_REG(0x35,0x38);     // EQP=0011 1000
        Set_LCD_8B_REG(0x24,0x38);     //GON=1, DTE=1, D=10
        DelayX1ms(40);
        Set_LCD_8B_REG(0x24,0x3C);     //GON=1, DTE=1, D=11

        Set_LCD_8B_REG(0x16,0x08);     //BGR=1
        Set_LCD_8B_REG(0x01,0x06);     //INVON=1, NORNO=1
        Set_LCD_8B_REG(0x55,0x00);
}

void DGC_PA_REG(unsigned char ADDR)
{
        unsigned char i;

        M51_CTRL_LCD_nCS = 0;
        M51_CTRL_LCD_RS = 0;
        WR_8B_FORMAT(ADDR);                        //ADDR=0x5C.
        M51_CTRL_LCD_RS = 1;

    for( i=0; i<=2; i++)
    {
            WR_8B_FORMAT(0x00);
            WR_8B_FORMAT(0x00);
            WR_8B_FORMAT(0x00);
            WR_8B_FORMAT(0x00);
            WR_8B_FORMAT(0x00);
            WR_8B_FORMAT(0x00);
            WR_8B_FORMAT(0x0D);
            WR_8B_FORMAT(0x14);
            WR_8B_FORMAT(0x19);
            WR_8B_FORMAT(0x1D);
            WR_8B_FORMAT(0x21);
            WR_8B_FORMAT(0x28);
            WR_8B_FORMAT(0x2F);
            WR_8B_FORMAT(0x34);
            WR_8B_FORMAT(0x39);
            WR_8B_FORMAT(0x3E);
            WR_8B_FORMAT(0x42);
            WR_8B_FORMAT(0x46);
            WR_8B_FORMAT(0x4A);
            WR_8B_FORMAT(0x4D);
            WR_8B_FORMAT(0x50);
            WR_8B_FORMAT(0x56);
            WR_8B_FORMAT(0x5B);
            WR_8B_FORMAT(0x60);
            WR_8B_FORMAT(0x64);
            WR_8B_FORMAT(0x69);
            WR_8B_FORMAT(0x6D);
```

```
        WR_8B_FORMAT(0x71);
        WR_8B_FORMAT(0x75);
        WR_8B_FORMAT(0x79);
        WR_8B_FORMAT(0x7D);
        WR_8B_FORMAT(0x81);
        WR_8B_FORMAT(0x84);
        WR_8B_FORMAT(0x88);
        WR_8B_FORMAT(0x8C);
        WR_8B_FORMAT(0x8F);
        WR_8B_FORMAT(0x92);
        WR_8B_FORMAT(0x96);
        WR_8B_FORMAT(0x99);
        WR_8B_FORMAT(0x9D);
        WR_8B_FORMAT(0xA0);
        WR_8B_FORMAT(0xA4);
        WR_8B_FORMAT(0xA7);
        WR_8B_FORMAT(0xAB);
        WR_8B_FORMAT(0xAE);
        WR_8B_FORMAT(0xB2);
        WR_8B_FORMAT(0xB5);
        WR_8B_FORMAT(0xB9);
        WR_8B_FORMAT(0xBD);
        WR_8B_FORMAT(0xC1);
        WR_8B_FORMAT(0xC4);
        WR_8B_FORMAT(0xC8);
        WR_8B_FORMAT(0xCD);
        WR_8B_FORMAT(0xD1);
        WR_8B_FORMAT(0xD6);
        WR_8B_FORMAT(0xDB);
        WR_8B_FORMAT(0xDE);
        WR_8B_FORMAT(0xE2);
        WR_8B_FORMAT(0xE6);
        WR_8B_FORMAT(0xEA);
        WR_8B_FORMAT(0xEE);
        WR_8B_FORMAT(0xF2);
        WR_8B_FORMAT(0xF6);
        WR_8B_FORMAT(0xFE);
    }

    M51_CTRL_LCD_nCS = 1;}
```

7.6.2 The reference setting of into Standby mode for Register-Content Interface Mode

```
void HX8352A_STB_INTO (void)
{
// Display Off
        Set_LCD_8B_REG(0x24,0x38);      //GON=1, DTE=1, D=10
        DelayX1ms (40);
        Set_LCD_8B_REG(0x24,0x28);      //GON=1, DTE=0, D=10
        DelayX1ms (40);
        Set_LCD_8B_REG(0x24,0x00);      //GON=0, DTE=0, D=00

    // Power Off
        Set_LCD_8B_REG(0x1E,0x14);      // VCOMG=0, VDV=1_0100
        DelayX1ms(10);
        Set_LCD_8B_REG(0x19,0x02);      // GASENB=0, PON=0, DK=0,
                                        // XDK=0, VLCD_TRI=1, STB=0
        DelayX1ms(10);
        Set_LCD_8B_REG(0x19,0x0A);      // GASENB=0, PON=0, DK=1,
                                        // XDK=0, VLCD_TRI=1, STB=0

        DelayX1ms(10);
        Set_LCD_8B_REG(0x1B,0x40);      // AP=000
        DelayX1ms(10);
         Set_LCD_8B_REG(0x3C,0x00);         // N_SAP=1100 0000
        DelayX1ms(10);

    // Into STB mode
        Set_LCD_8B_REG(0x19,0x0B);      // GASENB=0, PON=0, DK=0,
                                         // XDK=0, VLCD_TRI=1, STB=1
        DelayX1ms(10);

    // Stop Oscillation
        Set_LCD_8B_REG(0x17,0x90);          // RADJ=1001, OSC_EN=0
}
```

7.6.3 The reference setting of exit Standby mode for Register-Content Interface Mode

void HX8352A_STB_EXIT_3.0" (void)

```
{
// Start Oscillation
        Set_LCD_8B_REG(0x17,0x91);          // RADJ=1001, OSC_EN=1
        DelayX1ms(10);

// Exit STB mode

        Set_LCD_8B_REG(0x19,0x0A);      // GASENB=0, PON=0, DK=0,
                                        // XDK=0, VLCD_TRI=1, STB=0

// Power Supply Setting

        Set_LCD_8B_REG(0x1B,0x14);          // BT=0001, AP=100
        Set_LCD_8B_REG(0x1A,0x11);          // VC3=001, VC1=001 (VLCD/DDVDH)=6.45V)
        Set_LCD_8B_REG(0x1C,0x0D);          // VRH=1110 (VREG1=6.0V)
        Set_LCD_8B_REG(0x1F,0x27);          // VCM=010_1011
        DelayX1ms(20);

        Set_LCD_8B_REG(0x19,0x0A);          // GASENB=0, PON=0, DK=1, XDK=0,
                                            //VLCD_TRI=1, STB=0
        Set_LCD_8B_REG(0x19,0x1A);          // GASENB=0, PON=1, DK=1, XDK=0,
                                            //VLCD_TRI=1, STB=0
        DelayX1ms(40);

        Set_LCD_8B_REG(0x19,0x12);          // GASENB=0, PON=1, DK=0, XDK=0,
                                            //VLCD_TRI=1, STB=0,
                                            // VLCD=2XVCI by 2 CAPs
        DelayX1ms(40);

        Set_LCD_8B_REG(0x1E,0x2C);          // VCOMG=1, VDV=0_1110
        DelayX1ms(100);

// Display ON Setting
        Set_LCD_8B_REG(0x3C,0xC0);     // N_SAP=1100 0000
        Set_LCD_8B_REG(0x3D,0xC0);     // I_SAP=1100 0000
        Set_LCD_8B_REG(0x34,0x38);     // EQS=0011 1000
        Set_LCD_8B_REG(0x35,0x38);     // EQP=0011 1000
        Set_LCD_8B_REG(0x24,0x38);     //GON=1, DTE=1, D=10
        DelayX1ms(40);

        Set_LCD_8B_REG(0x24,0x3C);     //GON=1, DTE=1, D=11
}
```

```
void HX8352A_STB_EXIT_2.8" (void)
{

// Start Oscillation
        Set_LCD_8B_REG(0x17,0x91);          // RADJ=1001, OSC_EN=1
        DelayX1ms(10);

// Exit STB mode

        Set_LCD_8B_REG(0x19,0x0A);      // GASENB=0, PON=0, DK=0,
                                        // XDK=0, VLCD_TRI=1, STB=0

// Power Supply Setting

        Set_LCD_8B_REG(0x1B,0x14);                  // BT=0001, AP=100
        Set_LCD_8B_REG(0x1A,0x11);                  // VC3=001, VC1=001 (VLCD=6V)
        Set_LCD_8B_REG(0x1C,0x0D);                  // VRH=1101 (VREG1=5.5V)
        Set_LCD_8B_REG(0x1F,0x3A);                  // VCM=011_1010 ==>VcomH
        DelayX1ms(20);

        Set_LCD_8B_REG(0x19,0x0A);           // GASENB=0, PON=0, DK=1, XDK=0,
                                              //VLCD_TRI=1, STB=0
        Set_LCD_8B_REG(0x19,0x1A);           // GASENB=0, PON=1, DK=1, XDK=0,
                                              //VLCD_TRI=1, STB=0
        DelayX1ms(40);

        Set_LCD_8B_REG(0x19,0x12);           // GASENB=0, PON=1, DK=0, XDK=0,
                                             //VLCD_TRI=1,
                                           // STB=0,VLCD=2XVCI by 2 CAPs
        DelayX1ms(40);

        Set_LCD_8B_REG(0x1E,0x2D);                   // VCOMG=1, VDV=0_1101
        DelayX1ms(100);



// Display ON Setting
        Set_LCD_8B_REG(0x3C,0xC0);     // N_SAP=1100 0000
        Set_LCD_8B_REG(0x3D,0xC0);     // I_SAP=1100 0000
        Set_LCD_8B_REG(0x34,0x38);     // EQS=0011 1000
        Set_LCD_8B_REG(0x35,0x38);     // EQP=0011 1000
        Set_LCD_8B_REG(0x24,0x38);     //GON=1, DTE=1, D=10
        DelayX1ms(40);

        Set_LCD_8B_REG(0x24,0x3C);     //GON=1, DTE=1, D=11
}
```

8. Revision History

Version	Date	Description of changes
01	2007/09/11	New setup
	2007/11/28	Updated initial code in P64 ~ P72.
	2008/02/24	Updated initial code in P64 ~ P72.
	2008/03/12	Updated initial code in P64 ~ P72.