*Advance Information*

# SSD1901 TFT Graphics Controller

## 1 GENERAL DESCRIPTION

The SSD1901 is a TFT graphics controller with an embedded 80K byte SRAM display memory. It supports high resolution TFT panels with 8-bpp color depths, allowing up to 256 colors. The high integration of the SSD1901 provides a low cost, low power, single chip solution to meet the requirements of embedded systems, such as Office Automation equipment, Mobile Communications devices, and Hand-Held PCs where board size and battery life are major concerns.

The SSD1901supports Virtual, Split Screen and Floating Window display features to reduce the software manipulation. It also provides Fixed Window to simplify the display data update process. The above features, combines with the Operating System independence of the SSD1901 interface, make it the ideal solution for a wide variety of applications. It also provides advantage of single power supply. The SSD1901 is available in a 80 pin QFP package.

This document contains information on a new product. Specifications and information herein are subject to change without notice.

# 2 FEATURES

## 2.1 Integrated Display Memory

- Embedded 80K byte SRAM display memory.

## 2.2 CPU Interface

- Direct support of the following interfaces:
  - Motorola MC68K.
  - MPU bus interface using WAIT# signal.
- Direct memory mapping of internal registers into upper 128 bytes of 128K byte address space.
- The 80K byte display memory is directly and contiguously available through the 17-bit address bus.

## 2.3 Display Support

- 9/12 bits Active Matrix TFT interface
- Example resolutions: 160x160, 320x240

## 2.4 Display Modes

- 8 bit-per-pixel, 256-level color display.
- 256 simultaneous of 4096 colors on active matrix LCD panels.
- Virtual display support (displays images larger than the panel size through the use of panning and scrolling).
- Split screen display allows two different images to be simultaneously displayed.
- Fixed Window Mode simplifies the data update process for an image in a window area on the display.
- Floating Window Mode allows to display an image in a window area on the display without erasing the original image data.

## 2.5 Clock Source

- Maximum input clock (CLKI) frequency of 50MHz.
- Maximum operating clock (CLK) frequency of 25MHz.
- Operating clock (CLK) is derived from CLKI or BCLK input.
- Pixel Clock (PCLK) and Memory Clock (MCLK) are derived from CLK.

## 2.6 Miscellaneous

- Software Color Invert.
- Software Power Saving mode.
- Hardware Power Saving mode.
- LCD power-down sequencing.
- 4 General Purpose Input/Output pins are available.
- GPIO0 is available if Hardware Power Saving is not required.
- GPIO[4:2] are available if 9 bit TFT panel is selected.
- Single Supply : 3.0 volts to 3.6 volts.

# 3 ORDERING INFORMATION

**Table 3-1 Ordering Information**

| Ordering Part Number | Package Form |
|---|---|
| SSD1901QL5 | 80 pin 12 x 12 x 1.4 mm LQFP package |

# 4 BLOCK DIAGRAM

## 4.1 Functional Block Descriptions

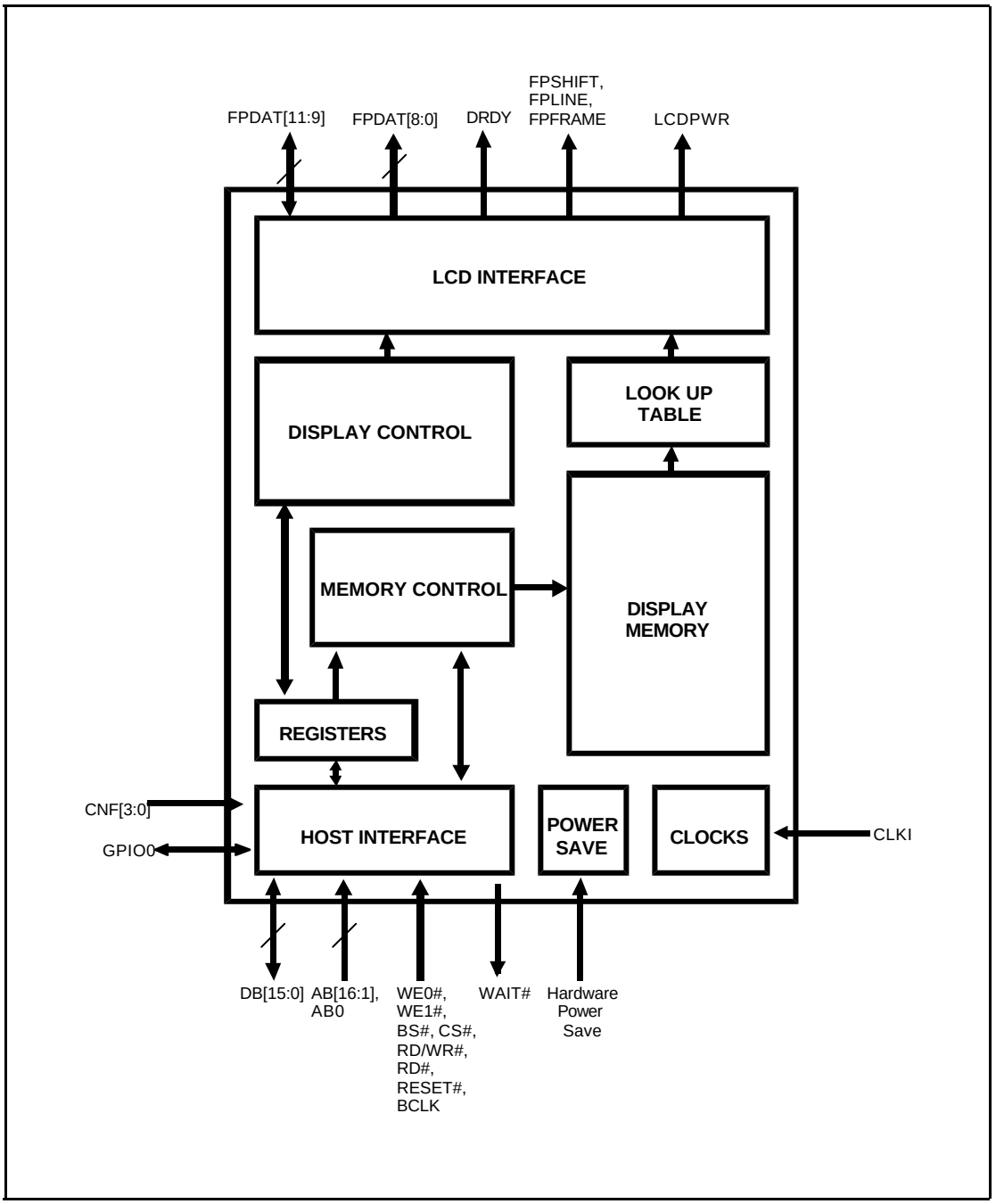


**Figure 4-1 System Block Diagram**

# 5 FUNCTIONAL BLOCK DESCRIPTION

## 5.1 Host Interface

The Host Interface provides the means for the CPU to communicate with the display memory and internal registers.

## 5.2 Memory Control

The Memory Control arbitrates between CPU accesses and display refresh accesses. It also generates the necessary signals to control the SRAM display memory.

## 5.3 Display Control

The Display Control manages data flow from the Memory Control through the Look-Up Table and to the LCD Interface. It also generates memory addresses for display refresh accesses.

Display memory represents one pixel on the display. Figure 5-1 "Display Data Memory Organization" shows the display data formats

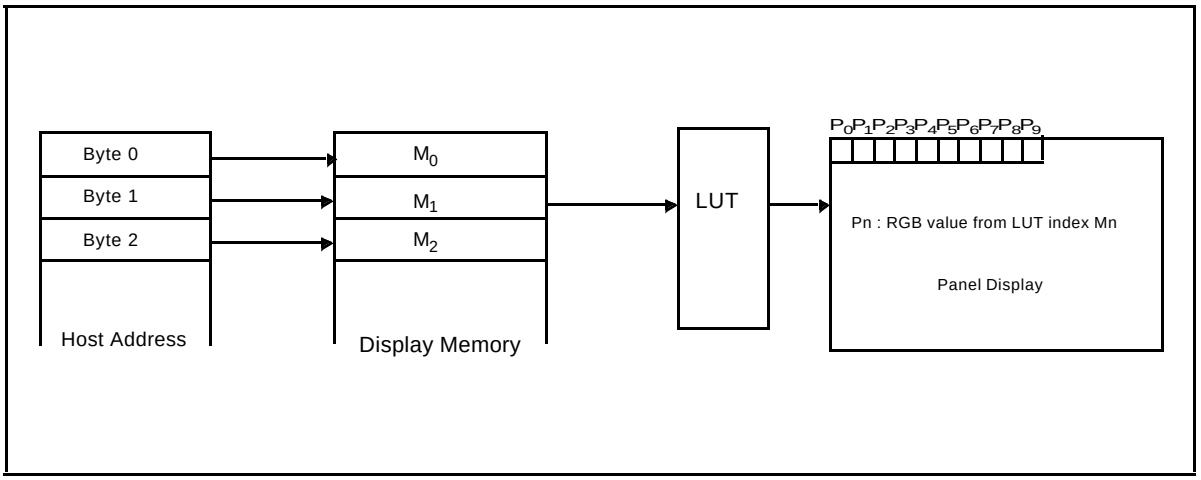


**Figure 5-1 Display Data Memory Organization**

## 5.4 Look - Up Table

The Look-Up Table contains three 256 entries, 4 bit wide Look-Up Tables or palettes, one for each primary color (red, green and blue).

The following figures are intended to show the display data output path only.

**Note**
When Software Color Invert is enabled, the display data inverts after the Look-Up Table.

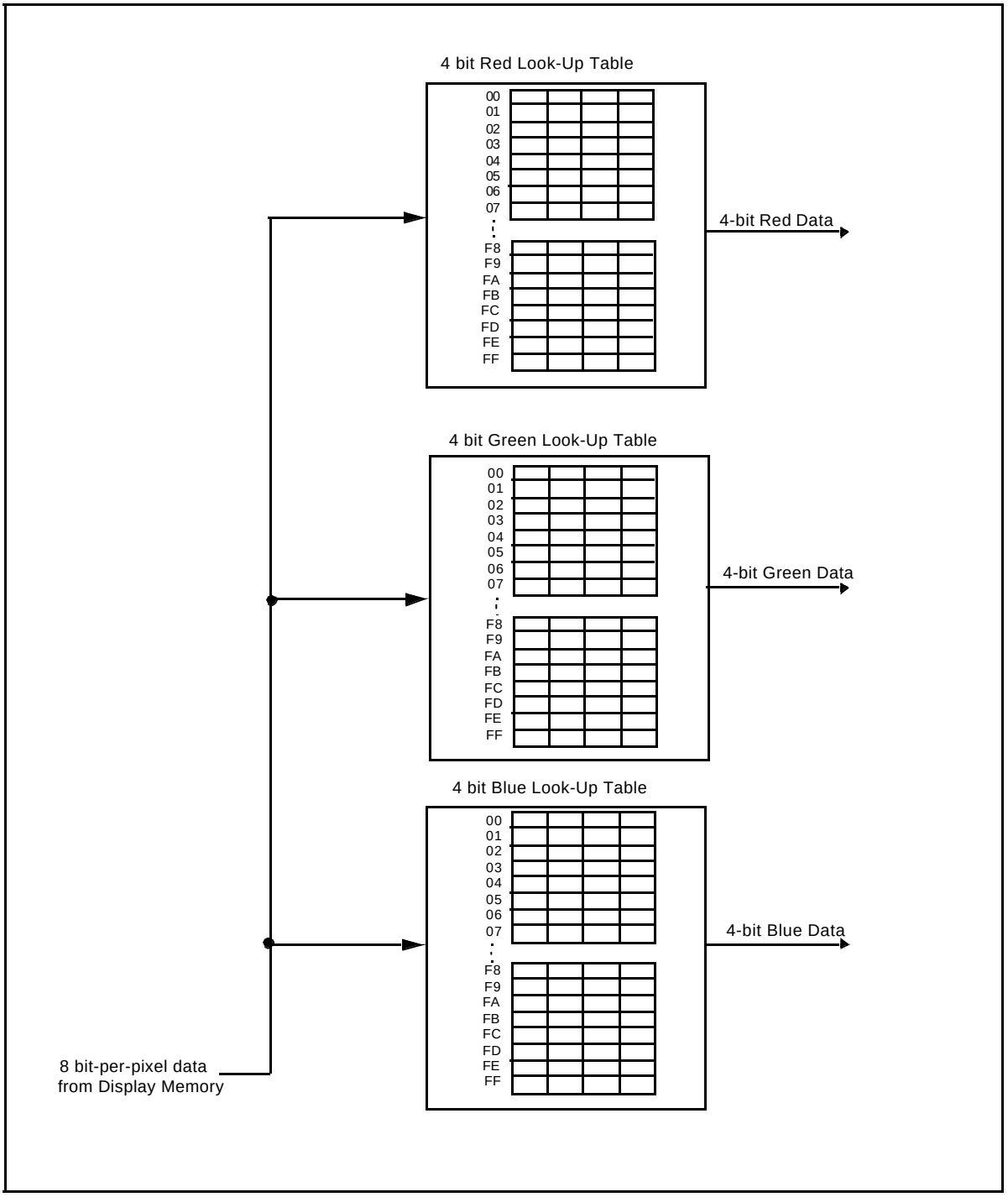


**Figure 5-2 Data Output Path for 12 bit TFT interface**

## 5.5 LCD Interface

The LCD Interface generates the display data format and timing control signals for TFT panels.

## 5.6 Power Saving

Enable the Power Saving Mode circuitry to initialize the power down sequence. When the power down sequence is completed, LCDPWR goes low to turn off the display. Refer to Figure 16 "Power Saving Modes" for Power Saving operation.

# 6 REGISTERS

## 6.1 Register Mapping

The SSD1901 registers are located in the upper 128 bytes of the 128K byte SSD1901 address range. The registers are accessible when CS# = 0 and AB[16:0] are in the range 1FF80h through 1FFFFh.

## 6.2 Register Descriptions

Unless specified otherwise, all register bits are reset to 0 during power up. All bits marked “0” must be programmed as zero. All bits marked “1” must be programmed as one.

| **REG[00h] Window Features Code Register**<br>Address = 1FF80h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Window features | Window mode selection | 0 | Window Blinking | Window Invert |

bits 4 Window features
When this bit = 0, Standard mode is selected. When this bit = 1, Window features is selected.

bits 3 Window mode selection
When this bit = 0, Fixed Window mode is selected. When this bit = 1, Floating Window mode is selected. This bit is effective in Window features is selected (REG[00h] bit 4 = 1)

bits 1 Window Blinking
When this bit = 0, Window Blinking for Window / Floating Window mode is disabled. When this bit = 1, Window Blinking for Window / Floating Window mode is enabled. This bit is effective in Winodw features is selected (REG[00h] bit 4 = 1).

bits 0 Window Invert
When this bit = 0, Window Invert for Window / Floating Window mode is disabled. When this bit = 1, Window Invert for Window / Floating Window mode is enabled. This bit is effective in Window features is selected (REG[00h] bit 4 = 1).

| **REG[01h] Window area Start Address Register (LSB)**<br>Address = 1FF81h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Window Start Address Bit 7 | Window Start Address Bit 6 | Window Start Address Bit 5 | Window Start Address Bit 4 | Window Start Address Bit 3 | Window Start Address Bit 2 | Window Start Address Bit 1 | Window Start Address Bit 0 |

| **REG[02h] Window area Start Address Register (MSB)**<br>Address = 1FF82h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Window Start Address Bit 15 | Window Start Address Bit 14 | Window Start Address Bit 13 | Window Start Address Bit 12 | Window Start Address Bit 11 | Window Start Address Bit 10 | Window Start Address Bit 9 | Window Start Address Bit 8 |

REG[01h] bits 7-0 Window Area Start Address Bits [15:0]
REG[02h] bits 7-0 These bits determine the **word address** of the start of window area. These registers are effective in Window features is selected (REG[00h] bit 4 = 1)

The start address is programmed as follows :
WindowStartAddress = (REG[02h],REG[01h]) Words

| **REG[03h] Floating Window Buffer Memory Start Address Register (LSB)**<br>Address = 1FF83h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Memory Start Address Bit 7 | Memory Start Address Bit 6 | Memory Start Address Bit 5 | Memory Start Address Bit 4 | Memory Start Address Bit 3 | Memory Start Address Bit 2 | Memory Start Address Bit 1 | Memory Start Address Bit 0 |

| **REG[04h] Floating Window Buffer Memory Start Address Register (MSB)**<br>Address = 1FF84h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Memory Start Address Bit 15 | Memory Start Address Bit 14 | Memory Start Address Bit 13 | Memory Start Address Bit 12 | Memory Start Address Bit 11 | Memory Start Address Bit 10 | Memory Start Address Bit 9 | Memory Start Address Bit 8 |

REG[03h] bits 7-0 Floating Window Buffer Memory Start Address Bits [15:0]
REG[04h] bits 7-0 These bits determine the**word address** of the start of Floating Window Buffer Memory for Floating Window mode. These registers are effective in Floating Window mode only (REG[00h] bit 4, 3 = 1)
The start address is programmed as follows :
MemoryStartAddress = (REG[04h],REG[03h]) Words

| **REG[05h] Words per line Register**<br>Address = 1FF85h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Words per line Bit 7 | Words per line Bit 6 | Words per line Bit 5 | Words per line Bit 4 | Words per line Bit 3 | Words per line Bit 2 | Words per line Bit 1 | Words per line Bit 0 |

| **REG[06h] Words per line Register (MSB)**<br>Address = 1FF86h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | Words per line Bit 8 |

REG[05h] bit 7-0 Words per line Bits [8:0]
REG[06h] bit 1 These bits determine words per line for Window /Floating Window features. This register is effective in Window features is selected (REG[00h] bit 4 = 1)
This register must be programmed with a value calculated as follows :

$$REG[06h], REG[05h] = \frac{HorizontalWindowSize(pixels)}{2} - 1$$

1FFh is the maximum value of this register for a horizontal resolution of 1024 pixels.

| **REG[07h] Vertical Window Size Register (LSB)**<br>Address = 1FF87h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Vertical Window Size Bit 7 | Vertical Window Size Bit 6 | Vertical Window Size Bit 5 | Vertical Window Size Bit 4 | Vertical Window Size Bit 3 | Vertical Window Size Bit 2 | Vertical Window Size Bit 1 | Vertical Window Size Bit 0 |

| **REG[08h] Vertical Window Size Register (MSB)**<br>Address = 1FF88h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | Vertical Window Size Bit 9 | Vertical Window Size Bit 8 |

REG[07h] bit 7-0
REG[08h] bit 1-0

Vertical Window Size Bits [9:0]
These bits determine the vertical resolution of the window area for Window features. This register is effective in Window features is selected (REG[00h] bit 4 = 1)
This register must be programmed with a value calculated as follows :

REG[08h],REG[07h] = VerticalWindowSize(lines) - 1

3FFh is the maximum value of this register for a vertical resolution of 1024 lines.

| **REG[09h] Look-Up Table Address for Blinking Register**<br>Address = 1FF89h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| LUT Address Bit 7 | LUT Address Bit 6 | LUT Address Bit 5 | LUT Address Bit 4 | LUT Address Bit 3 | LUT Address Bit 2 | LUT Address Bit 1 | LUT Address Bit 0 |

bits 7-0

LUT Address for Blinking Bits [7:0]
This register control the index of the Look-Up Tables (LUT) for Window Blinking. The SSD1901 has three 256-position, 4-bit wide LUTs, one for each red, green, and blue - refer to Figure 5.4 "Look - Up Table" for the architecture. This register is effective in Window Blinking only (REG[00h] bit 4,1 = 1)

| **REG[0Ah] Frame Period for Blinking Register**<br>Address = 1FF8Ah | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Frame Period Bit 7 | Frame Period Bit 6 | Frame Period Bit 5 | Frame Period Bit 4 | Frame Period Bit 3 | Frame Period Bit 2 | Frame Period Bit 1 | Frame Period Bit 0 |

bits 7-0

Frame Period for Blinking Bits [7:0]
This register control the number of frame period for Window Blinking. This register is effective in Window Blinking only (REG[00h] bit 4,1 = 1)
This register is programmed as follows :

REG[0Ah] = OnFramePeriod - 1 = OffFramePeriod - 1

| **REG[0Bh] Input Clock Register**<br>Address = 1FF8Bh | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| CLK Selection | Clock Divide Bit 6 | Clock Divide Bit 5 | Clock Divide Bit 4 | Clock Divide Bit 3 | Clock Divide Bit 2 | Clock Divide Bit 1 | Clock Divide Bit 0 |

bit 7 CLK Selection

This bit controls the source of Operating Clock (CLK). When this bit = 0, CLK input is come from CLKI pin and BCLK pin must be tied to VSS. When this bit = 1, CLK input is come from BCLK pin and CLKI pin must be tied to $IOV_{DD}$.

bits 6-0 Input Clock Bits [6:0]

This register further divides down the Operating Clock (CLK) on top of REG[62h] bit 4. These bits are effective when BCLK as the source of CLK (REG[0Bh] bit 7 = 1). The following table shows the selection of CLK.

| REG[62h] bit 4 | CLK |
|---|---|
| 0 | $\frac{CLK}{(REG[0Bh]bits\,6-0)+1}$ |
| 1 | $\frac{CLK}{2\times((REG[0Bh]bits\,6-0)+1)}$ |

**Table 6-1 Input Clock Register Selection**

| **REG[60h] Product Code Register**<br>Address = 1FFE0h | | | | | | | Read Only. |
|---|---|---|---|---|---|---|---|
| Product Code Bit 5 | Product Code Bit 4 | Product Code Bit 3 | Product Code Bit 2 | Product Code Bit 1 | Product Code Bit 0 | Revision Code Bit 1 | Revision Code Bit 0 |

bits 7-2 Product Code

This is a read-only register that indicates the product code of the chip. The product code is 100000.

bits 1-0 Revision Code

This is a read-only register that indicates the revision code of the chip. The revision code is 00.

| **REG[61h] Mode Register 0**<br>Address = 1FFE1h | | | | | | | Read/Write. |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | FPLine Polarity | FPFrame Polarity | Mask FPSHIFT | 0 | Data Width |

bit 4 FPLINE Polarity

This bit controls the polarity of FPLINE. When this bit = 0, FPLINE is active low. When this bit = 1, FPLINE is active high.

bit 3 FPFRAME Polarity

This bit controls the polarity of FPFRAME. When this bit = 0, FPFRAME is active low. When this bit = 1, FPFRAME is active high.

bit 2 Mask FPSHIFT

When this bit = 1, FPSHIFT is masked during non-display periods

bit 0 Data Width

This bit select the display data format. When this bit =0, 9-bit TFT panel is selected. When this bit =1, 12-bit TFT panel is selected.

| **REG[62h] Mode Register 1**<br>Address = 1FFE2h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | Operating Clock divide (CLK/2) | Display Blank | 0 | 0 | Software color Invert |

bit 4 Operating Clock Divide
This bit controls the value of Operating Clock (CLK). Refer to Table6-1, "Input Clock Register Selection," on page11 for CLK value.

PCLK=MCLK=CLK.

bit 3 Display Blank
This bit blanks the display image. When this bit = 1, the display is blanked (FPDAT lines to the panel are driven low). When this bit = 0, Standard Mode is selected.

bit 0 Software color Invert
When this bit = 1, Color Invert Mode is selected and display color is inverted. When this bit = 0, Standard Mode is selected.

**Note**
Display data is inverted after the Look-Up Table.

**Table 6-2 Display Features Operation**

| Bit 3 | Bit 0 | Display Data |
|---|---|---|
| 0 | 0 | Normal |
| 0 | 1 | Invert |
| 1 | X | Blank |

| **REG[63h] Mode Register 2**<br>Address = 1FFE3h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | LCDPWR Override | Hardware Power Saving Enable | Software Power Saving Bit 1 | Software Power Saving Bit 0 |

bit 3 LCDPWR Override
When this bit = 1, LCDPWR is forced inactive, by-passing the LCD power sequencing. The 127 frame delay between Hardware Power Saving and the LCD panel control signals is reduced to a single line. When this bit = 0, LCDPWR is controlled by the power sequencing logic within the SSD1901. See Table 9-7 "Power Down/Up Timing," for further information.

bit 2 Hardware Power Saving Enable
When this bit = 1, GPIO0 is used as the Hardware Power Saving input pin. When this bit = 0, GPIO0 op-

erates normally.

**Table 6-3 Hardware Power Saving/GPIO0 Operation**

| RESET# State | Hardware Power Saving Enable REG[63h] bit 2 | GPIO0 Config Reg[18h] bit 0 | GPIO0 Status/Control REG[79 h] bit 0 | GPIO0 Operation |
|---|---|---|---|---|
| 0 | X | X | X | |
| 1 | 0 | 0 | reads pin status | GPIO0 Input<br>(high impedance) |
| 1 | 0 | 1 | 0 | GPIO0 Output = 0 |
| 1 | 0 | 1 | 1 | GPIO0 Output = 1 |
| 1 | 1 | X | X | Hardware Power Saving Input<br>0 : Normal operation<br>1 : Power Saving mode |

bits 1-0 Software Power Saving Bits [1: 0]
These bits select the Power Saving Mode as shown in the following table.
.

**Table 6-4 Software Power Saving Mode Selection**

| Bit 1 | Bit 0 | Mode |
|---|---|---|
| 0 | 0 | Software Power Saving |
| 0 | 1 | reserved |
| 1 | 0 | reserved |
| 1 | 1 | Normal Operation |

Refer to Section 16 "Power Saving Modes" on page 53 for a complete description of the Power Saving Modes.

| **REG[64h] Horizontal Panel Size Register**<br>Address = 1FFE4h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | Horizontal Panel Size Bit 6 | Horizontal Panel Size Bit 5 | Horizontal Panel Size Bit 4 | Horizontal Panel Size Bit 3 | Horizontal Panel Size Bit 2 | Horizontal Panel Size Bit 1 | Horizontal Panel Size Bit 0 |

bits 6-0 Horizontal Panel Size Bits [6:0]
This register determines the horizontal resolution of the panel. This register must be programmed with a value calculated as follows:

$$\text{Horizontal Panel Size Register} = \frac{\text{Horizontal Panel Resolution(pixels)}}{8} - 1$$

**Note**
This register must not be set to a value less than 03h.
7F is the maximum value of this register for a horizontal resolution of 1024 pixels.

| **REG[65h] Vertical Panel Size Register (LSB)**<br>Address = 1FFE5h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Vertical Panel Size Bit 7 | Vertical Panel Size Bit 6 | Vertical Panel Size Bit 5 | Vertical Panel Size Bit 4 | Vertical Panel Size Bit 3 | Vertical Panel Size Bit 2 | Vertical Panel Size Bit 1 | Vertical Panel Size Bit 0 |

| **REG[66h] Vertical Panel Size Register (MSB)**<br>Address = 1FFE6h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | Vertical Panel Size Bit 9 | Vertical Panel Size Bit 8 |

REG[65h] bits 7-0 Vertical Panel Size Bits [9:0]
REG[66h] bits 1-0 This 10-bit register determines the vertical resolution of the panel. This register must programmed with a value calculated as follows.:

VerticalPanelSizeRegister = VerticalPanelResolution(lines)-1

3FFh is the maximum value of this register for a vertical resolution of 1024 lines.

| **REG[67h] FPLINE Start Position Register**<br>Address = 1FFE7h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | FPLINE Start Position Bit 4 | FPLINE Start Position Bit 3 | FPLINE Start Position Bit 2 | FPLINE Start Position Bit 1 | FPLINE Start Position Bit 0 |

bits 4-0 FPLINE Start Position
These bits are used to specify the position of the FPLINE pulse. These bits specify the delay, in 8-pixel resolution, from the end of a line of display data (FPDAT) to the leading edge of FPLINE. This register is programmed as follows:

FPLINEposition(pixels) = (REG[67h] + 2) x 8

The following constraint must be satisfied for the contents of this register :
REG[67h] ≤ REG[68h]

| **REG[68h] Horizontal Non-Display Period Register**<br>Address = 1FFE8h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Horizontal Non-Display Period Bit 4 | Horizontal Non-Display Period Bit 3 | Horizontal Non-Display Period Bit 2 | Horizontal Non-Display Period Bit 1 | Horizontal Non-Display Period Bit 0 |

bits 4-0 Horizontal Non-Display Period
These bits specify the horizontal non-display period in 8-pixel resolution. This register is programmed as follows :

HorizontalNonDisplayPeriod(pixels) = (REG[68h] + 4) x 8

| **REG[69h] FPFRAME Start Position Register**<br>Address = 1FFE9h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | FPFRAME Start Position Bit 5 | FPFRAME Start Position Bit 4 | FPFRAME Start Position Bit 3 | FPFRAME Start Position Bit 2 | FPFRAME Start Position Bit 1 | FPFRAME Start Position Bit 0 |

bits 5-0 FPFRAME Start Position

These bits are used to specify the position of the FPFRAME pulse. These bits specify the number of lines between the last line of display data (FPDAT) and the leading edge of FPFRAME. This register is programmed as follows:

FPFRAMEposition(lines) = REG[69h]

The following constraint must be satisified for the contents of this register :

1 <= REG[69h] <= REG[6Ah]

| **REG[6Ah] Vertical Non-Display Period Register**<br>Address = 1FFEAh | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Vertical Non-Display Status (RO) | 0 | Vertical Non-Display Period Bit 5 | Vertical Non-Display Period Bit 4 | Vertical Non-Display Period Bit 3 | Vertical Non-Display Period Bit 2 | Vertical Non-Display Period Bit 1 | Vertical Non-Display Period Bit 0 |

bit 7 Vertical Non-Display Status

This bit =1 during the Vertical Non-Display period.

bits 5-0 Vertical Non-Display Period

These bits specify the vertical non-display period. This register is programmed as follows:

VerticalNonDisplayPeriod(lines) = REG[6Ah] bits [5:0]

**Note**

This register should be set only while initialization.

| **REG[6Ch] Screen 1 Start Address Register (LSB)**<br>Address = 1FFECh | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Screen 1 Start Address Bit 7 | Screen 1 Start Address Bit 6 | Screen 1 Start Address Bit 5 | Screen 1 Start Address Bit 4 | Screen 1 Start Address Bit 3 | Screen 1 Start Address Bit 2 | Screen 1 Start Address Bit 1 | Screen 1 Start Address Bit 0 |

| **REG[6Dh] Screen 1 Start Address Register (MSB)**<br>Address = 1FFEDh | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Screen 1 Start Address Bit 15 | Screen 1 Start Address Bit 14 | Screen 1 Start Address Bit 13 | Screen 1 Start Address Bit 12 | Screen 1 Start Address Bit 11 | Screen 1 Start Address Bit 10 | Screen 1 Start Address Bit 9 | Screen 1 Start Address Bit 8 |

REG[6Dh] bits 7-0 Screen 1 Start Address Bits [15:0]

REG[6Ch] bits 7-0 These bits determine the **word address** of the start of Screen 1.

| **REG[6Eh] Screen 2 Start Address Register (LSB)**<br>Address = 1FFEEh | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Screen 2 Start Address Bit 7 | Screen 2 Start Address Bit 6 | Screen 2 Start Address Bit 5 | Screen 2 Start Address Bit 4 | Screen 2 Start Address Bit 3 | Screen 2 Start Address Bit 2 | Screen 2 Start Address Bit 1 | Screen 2 Start Address Bit 0 |

| **REG[6Fh] Screen 2 Start Address Register (MSB)**<br>Address = 1FFEFh | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Screen 2 Start Address Bit 15 | Screen 2 Start Address Bit 14 | Screen 2 Start Address Bit 13 | Screen 2 Start Address Bit 12 | Screen 2 Start Address Bit 11 | Screen 2 Start Address Bit 10 | Screen 2 Start Address Bit 9 | Screen 2 Start Address Bit 8 |

REG[6Fh] bits 7-0 Screen 2 Start Address Bits [15:0]
REG[6Eh] bits 7-0 These bits determine the **word address** of the start of Screen 2.

| **REG[71h] Memory Address Offset Register**<br>Address = 1FFF1h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Memory Address Offset Bit 7 | Memory Address Offset Bit 6 | Memory Address Offset Bit 5 | Memory Address Offset Bit 4 | Memory Address Offset Bit 3 | Memory Address Offset Bit 2 | Memory Address Offset Bit 1 | Memory Address Offset Bit 0 |

bits 7-0 Memory Address Offset Bits [7:0]
This register is used to create a virtual image by setting a word offset between the last address of one line and the first address of the following line. If this register is not equal to zero, then a virtual image is formed. The displayed image is a window into the larger virtual image. See Figure 6-1Screen-Register Relationship on page 17.

| **REG[72h] Screen 1 Vertical Size Register (LSB)**<br>Address = 1FFF2h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Screen 1 Vertical Size Bit 7 | Screen 1 Vertical Size Bit 6 | Screen 1 Vertical Size Bit 5 | Screen 1 Vertical Size Bit 4 | Screen 1 Vertical Size Bit 3 | Screen 1 Vertical Size Bit 2 | Screen 1 Vertical Size Bit 1 | Screen 1 Vertical Size Bit 0 |

| **REG[73h] Screen 1 Vertical Size Register (MSB)**<br>Address = 1FFF3h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| n/a | n/a | n/a | n/a | n/a | n/a | Screen 1 Vertical Size Bit 9 | Screen 1 Vertical Size Bit 8 |

REG[73h] bits 1-0 Screen 1 Vertical Size Bits [9:0]
REG[72h] bits 7-0 These bits determine the height (in lines) of Screen 1.
If this register is programmed with a value, n, where n is less than the Vertical Panel Size (REG[66h], REG[65h]), then lines 0 to n of the panel contain Screen 1 and lines n+1 to REG[66h], REG[65h] of the panel contain Screen 2. See Figure 6-1 Screen-Register Relationship on page 17.

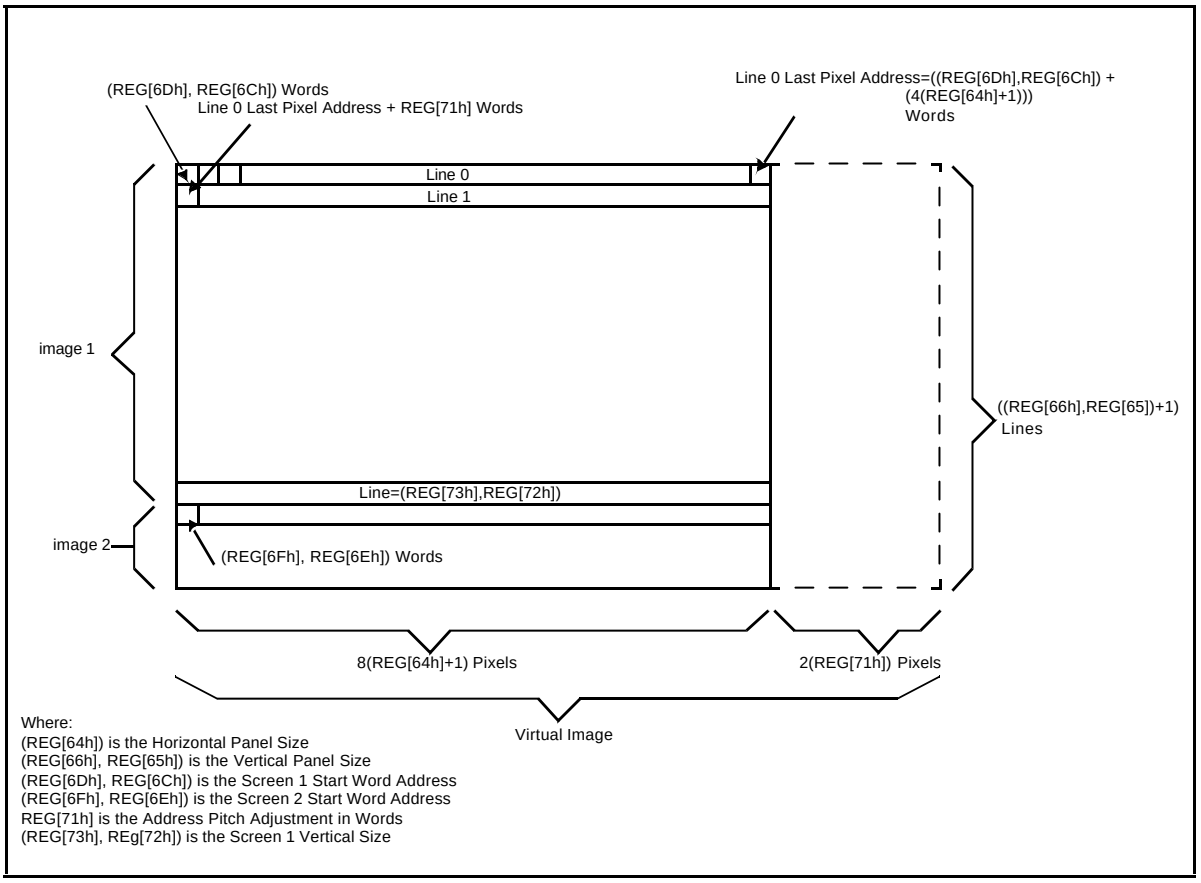


**Figure 6-1 Screen-Register Relationship**

Consider an example where REG[73h], REG[72] = 0CEh for a 320x240 display system. The upper 207 lines (CEh + 1) of the panel show an image from the Screen 1 Start Word Address. The remaining 33 lines show an image from the Screen 2 Start Word Address.

| **REG[75h] Look-Up Table Address Register**<br>Address = 1FFF5h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| LUT Address Bit 7 | LUT Address Bit 6 | LUT Address Bit 5 | LUT Address Bit 4 | LUT Address Bit 3 | LUT Address Bit 2 | LUT Address Bit 1 | LUT Address Bit 0 |

bits 7-0 LUT Address Bits [7:0]

These 8 bits control the pointer into the Look-Up Tables (LUT). The SSD1901 has three 256-position, 4-bit wide LUTs, one for each of red, green, and blue – refer to Section 5.4 "Look - Up Table" on page 5 for the architecture.

This register selects which LUT entry is read/write accessible through the LUT Data Register (REG[77h]). Writing the LUT Address Register automatically sets the pointer to the Red LUT. Accesses to the LUT Data Register automatically increment the pointer.

For example, writing 03h into the LUT Address Register sets the pointer to R[3]. A subsequent access to the LUT Data Register accesses R[3] and moves the pointer onto G[3]. Subsequent accesses to the LUT Data Register move the pointer onto B[3], R[4], G[4], B[4], R[5], etc.

| **REG[77h] Look-Up Table Data Register**<br>Address = 1FFF7h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| LUT Data Bit 3 | LUT Data Bit 2 | LUT Data Bit 1 | LUT Data Bit 0 | n/a | n/a | n/a | n/a |

bits 7-4 LUT Data Bits [3:0]

This register is used to read/write the RGB Look-Up Tables. This register accesses the entry at the pointer controlled by the Look-Up Table Address Register (REG[75h]).

Data Bits [2:0] will be used for 9 bit TFT panel (REG[61h] bit 0 = 0).

Access to the Look-Up Table Data Register automatically increases the value of the pointer.

**Note**
The RGB data is inserted into the LUT after the Blue data is written, i.e. all three colors must be written before the LUT is updated. Other registers must not be accessed unless all three colors are accessed.

| **REG[78h] GPIO Configuration Control Register**<br>Address = 1FFF8h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | GPIO4 Pin IO Configu-ration | GPIO3 Pin IO Configu-ration | GPIO2 Pin IO Configu-ration | 0 | GPIO0 Pin IO Configu-ration |

bits 4-0 GPIO[4:2] and GPIO0 Pin IO Configuration

These bits determine the direction of the GPIOn pins.
When the GPIOn Pin IO Configuration bit = 0, the corresponding GPIOn pin is configured as an input. The input can be read at the GPIOn Status/Control Register bit (REG[79h]).

When the GPIOn Pin IO Configuration bit = 1, the corresponding GPIOn pin is configured as an output. The output can be controlled by writing the GPIOn Status/Control Register bit (REG[79h]).

**Note**
GPIO0 Configuration Bit has no effect when the Hardware Power Saving mode is enabled.
GPIO[4:2] Configuration Bits have no effect for 12-bit TFT operation.

When the GPIOs configured as input, all unused pins must be tied to $IOV_{DD}$ .

| **REG[79h] GPIO Status/Control Register**<br>Address = 1FFF9h | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| n/a | n/a | n/a | GPIO4 Pin IO Status | GPIO3 Pin IO Status | GPIO2 Pin IO Status | n/a | GPIO0 Pin IO Status |

bits 4-0 GPIO[4:2] and GPIO0 Pin IO Status

When the GPIOn pins are configured as an input, the corresponding GPIO Status bit is used to read the pin input. See REG[78h] above.

When the GPIOn pin is configured as an output, the corresponding GPIO Status bit is used to control the pin output. See REG[78h] above.

| **REG[7Ah] Scratch Pad Register**<br>Address = 1FFFAh | | | | | | | Read/Write |
|---|---|---|---|---|---|---|---|
| Scratch Bit 7 | Scratch Bit 6 | Scratch Bit 5 | Scratch Bit 4 | Scratch Bit 3 | Scratch Bit 2 | Scratch Bit 1 | Scratch Bit 0 |

bits 7-0 Scratch Pad Register
This register contains general use read/write bits. These bits have no effect on hardware configuration.

**REG[6Bh], REG[70h], REG[7Bh], REG[7Ch], REG[7Eh] and REG[7Fh]**
These registers are reserved for factory testing and should not be written. Any value written to these registers may result in damage to the SSD1901 and/or any panel connected to the SSD1901.

## 6.3 Register Accesibility

All registers listed in Table 6-5 Registers accessible only at Software Power Saving Mode on page 19 can be programmed at Software Power Saving Mode only.

**Table 6-5 Registers accessible only at Software Power Saving Mode**

| Register Name | Register Number |
|---|---|
| Input Clock Register | REG[0Bh] |
| Mode Register 0 | REG[61h] |
| Horizontal Panel Size Register | REG[64h] |
| Vertical Panel Size Register (LSB) | REG[65h] |
| Vertical Panel Size Register (MSB) | REG[66h] |
| FPLINE Start Position Register | REG[67h] |
| Horizontal Non-Display Period Register | REG[68h] |
| FPFRAME Start Position Register | REG[69h] |
| Vertical Non-Display Period Register | REG[6Ah] |

**Note:**
Improper access to those registers may cause damage to panel.

# 7 PIN DESCRIPTION

## 7.1 Pinout Diagram

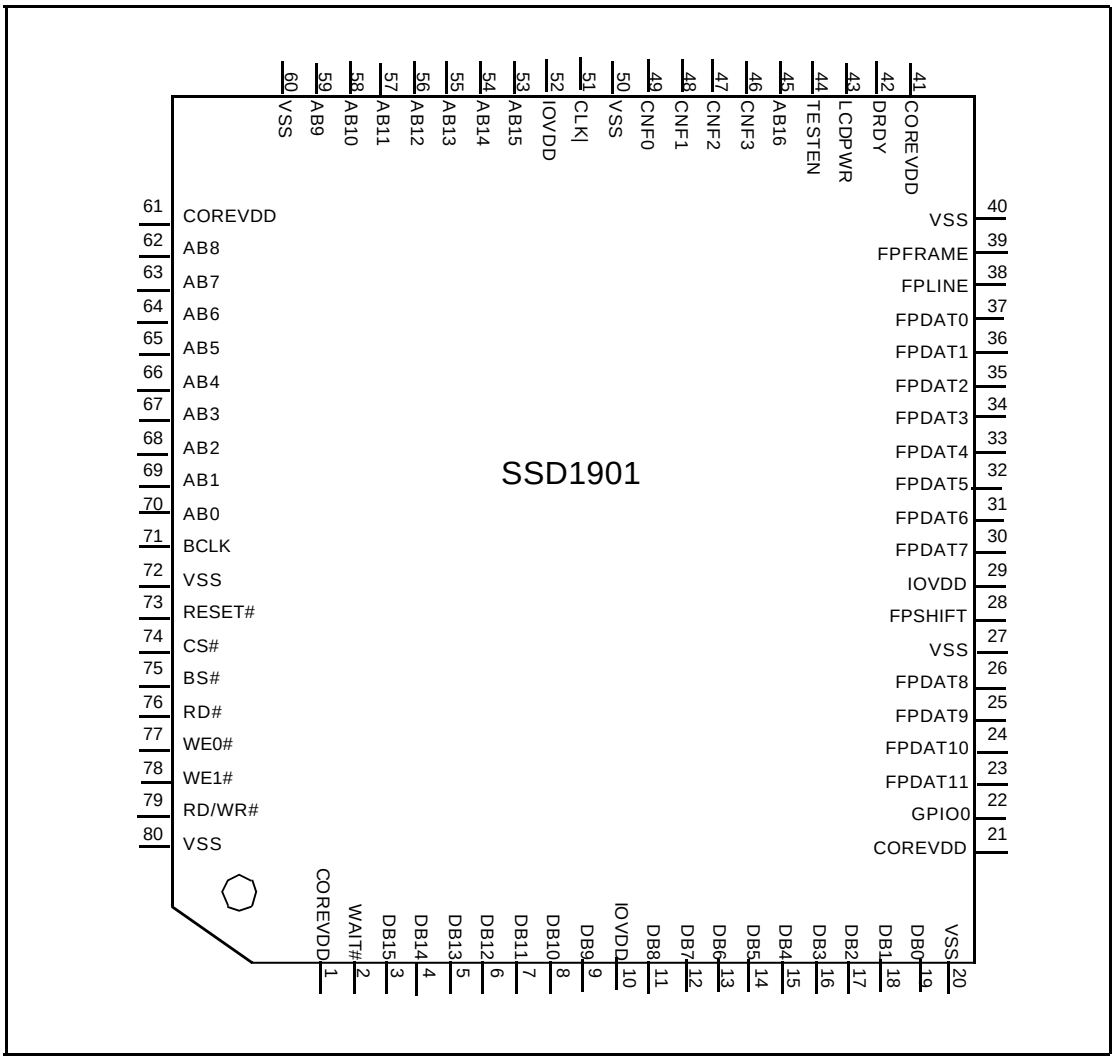


**Figure 7-1 Pinout Diagram**

**Note**
Package type: 80 pin surface mount LQFP

| Pin # | Signal Name | Pin # | Signal Name | Pin # | Signal Name | Pin # | Signal Name |
|---|---|---|---|---|---|---|---|
| 1 | COREVDD | 21 | COREVDD | 41 | COREVDD | 61 | COREVDD |

| Pin # | Signal Name | Pin # | Signal Name | Pin # | Signal Name | Pin # | Signal Name |
|---|---|---|---|---|---|---|---|
| 2 | WAIT# | 22 | GPIO0 | 42 | DRDY | 62 | AB8 |
| 3 | DB15 | 23 | FPDAT11 | 43 | LCDPWR | 63 | AB7 |
| 4 | DB14 | 24 | FPDAT10 | 44 | TESTEN | 64 | AB6 |
| 5 | DB13 | 25 | FPDAT9 | 45 | AB16 | 65 | AB5 |
| 6 | DB12 | 26 | FPDAT8 | 46 | CNF3 | 66 | AB4 |
| 7 | DB11 | 27 | VSS | 47 | CNF2 | 67 | AB3 |
| 8 | DB10 | 28 | FPSHIFT | 48 | CNF1 | 68 | AB2 |
| 9 | DB9 | 29 | IOVDD | 49 | CNF0 | 69 | AB1 |
| 10 | IOVDD | 30 | FPDAT7 | 50 | VSS | 70 | AB0 |
| 11 | DB8 | 31 | FPDAT6 | 51 | CLKI | 71 | BCLK |
| 12 | DB7 | 32 | FPDAT5 | 52 | IOVDD | 72 | VSS |
| 13 | DB6 | 33 | FPDAT4 | 53 | AB15 | 73 | RESET# |
| 14 | DB5 | 34 | FPDAT3 | 54 | AB14 | 74 | CS# |
| 15 | DB4 | 35 | FPDAT2 | 55 | AB13 | 75 | BS# |
| 16 | DB3 | 36 | FPDAT1 | 56 | AB12 | 76 | RD# |
| 17 | DB2 | 37 | FPDAT0 | 57 | AB11 | 77 | WE0# |
| 18 | DB1 | 38 | FPLINE | 58 | AB10 | 78 | WE1 |
| 19 | DB0 | 39 | FPFRAME | 59 | AB9 | 79 | RD/WR# |
| 20 | VSS | 40 | VSS | 60 | VSS | 80 | VSS |

## 7.2 Pin Description

**Key:**
I =Input
O =Output
IO = Bi-Directional (Input/Output)
P = Power pin
C = CMOS Input
CS = Schmitt input level
CO = Output driver (with driver type=3/-1.5mA)
TSx = Tri-state output driver, x denotes driver type (1=3/-1.5mA, 2=6/-3mA)
CN = Low-noise output driver (with driver type =12/-6mA)

**Table 7-1 Host Interface**

| Pin Names | Type | Pin# | Cell | RESET State | Description |
|---|---|---|---|---|---|
| AB0 | I | 70 | CS | Input | This pin has multiple functions.<br>• For MC68K #1, this pin inputs the lower data strobe (LDS#).<br>• For MC68K #2, this pin inputs system address bit 0 (A0).<br>• For Generic #1, this pin inputs system address bit 0 (A0).<br>• For Generic #2, this pin inputs system address bit 0 (A0).<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary. |
| AB[16:1] | I | 45,53,54, 55,56,57, 58,59,62, 63,64,65, 66,67,68, 69 | C | Input | These pins input the system address bits 16 through 1 (A[16:1]). |
| DB[15:0] | IO | 3,4,5,6,7,8, 9,11,12,13, 14,15,16,1 7,18,19 | C/TS2 | Hi-Z | These pins have multiple functions.<br>• For MC68K #1, these pins are connected to D[15:0].<br>• For MC68K #2, these pins are connected to D[31:16] for a 32-bit device (e.g. MC68030) or D[15:0] for a 16-bit device (e.g. MC68340).<br>• For Generic #1, these pins are connected to D[15:0].<br>• For Generic #2, these pins are connected to D[15:0].<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary. |
| WE0# | I | 77 | CS | Input | This pin has multiple functions.<br>• For MC68K #1, this pin must be tied to IO $V_{DD}$<br>• For MC68K #2, this pin inputs the bus size bit 0 (SIZ0).<br>• For Generic #1, this pin inputs the write enable signal for the lower data byte (WE0#).<br>• For Generic #2, this pin inputs the write enable signal (WE#)<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary |
| WE1# | I | 78 | CS | Input | This pin has multiple functions.<br>• For MC68K #1, this pin inputs the upper data strobe (UDS#).<br>• For MC68K #2, this pin inputs the data strobe (DS#).<br>• For Generic #1, this pin inputs the write enable signal for the upper data byte (WE1#).<br>• For Generic #2, this pin inputs the byte enable signal for the high data byte (BHE#).<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary. |
| CS# | I | 74 | C | Input | This pin inputs the chip select signal. |
| BCLK | I | 71 | C | Input | System bus clock. This input must be connected to Vss if CLKI as clock source. |
| BS# | I | 75 | CS | Input | This pin has multiple functions.<br>• For MC68K #1, this pin inputs the address strobe (AS#).<br>• For MC68K #2, this pin inputs the address strobe (AS#).<br>• For Generic #1, this pin must be tied to $V_{SS}$.<br>• For Generic #2, this pin must be tied to IO $V_{DD}$.<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary. |
| RD/WR# | I | 79 | CS | Input | This pin has multiple functions.<br>• For MC68K #1, this pin inputs the R/W# signal.<br>• For MC68K #2, this pin inputs the R/W# signal.<br>• For Generic #1, this pin inputs the read command for the upper data byte (RD1#).<br>• For Generic #2, this pin must be tied to IO $V_{DD}$.<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary. |

**Table 7-1 Host Interface**

| Pin Names | Type | Pin# | Cell | RESET State | Description |
|---|---|---|---|---|---|
| RD# | I | 76 | CS | Input | This pin has multiple functions.<br>• For MC68K #1, this pin must be tied to IO $V_{DD}$ .<br>• For MC68K #2, this pin inputs the bus size bit 1 (SIZ1).<br>• For Generic #1, this pin inputs the read command for the lower data byte (RD0#).<br>• For Generic #2, this pin inputs the read command (RD#).<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary. |
| WAIT# | O | 2 | TS2 | Hi-Z | This pin has multiple functions.<br>• For MC68K #1, this pin outputs the data transfer acknowledge signal (DTACK#).<br>• For MC68K #2, this pin outputs the data transfer and size acknowledge bit 1 (DSACK1#).<br>• For Generic #1, this pin outputs the wait signal (WAIT#).<br>• For Generic #2, this pin outputs the wait signal (WAIT#).<br>See Table 7-7 Host Bus Interface Pin Mapping on page 25 for summary. |
| RESET# | I | 73 | CS | 0 | Active low input to set all internal registers to the default state and to force all signals to their inactive states. |

**Table 7-2 LCD Interface**

| Pin Name | Type | Pin # | Cell | Reset# State | Description |
|---|---|---|---|---|---|
| FPDAT[8:0] | O | 26,30,31,32, 33,34,35,36, 37 | CN | 0 | Panel Data bits [8:0] |
| FPDAT[11:9] | O,IO | 23,24,25 | CN | Input | These pins have multiple functions.<br>• Panel Data bits [11:9] for 12 bits TFT panels.<br>• General Purpose Input/Output pins GPIO[4:2].<br>These pins should be connected to IO $V_{DD}$ when unused.<br>See Table 7-16 LCD Interface Pin Mapping on page 28 for summary |
| FPFRAME | O | 39 | CN | 0 | Frame Pulse |
| FPLINE | O | 38 | CN | 0 | Line Pulse |
| FPSHIFT | O | 28 | CN | 0 | Shift Clock |
| LCDPWR | O | 43 | CO | 0 | Active high LCD Power Control |
| DRDY | O | 42 | CN | 0 | Display Enable |

**Table 7-3 Clock Input**

| Pin Name | Type | Pin # | Driver | Description |
|---|---|---|---|---|
| CLKI | I | 51 | C | Input Clock. This input pin must be connected to IOVDD if BCLK as clock source. |

**Table 7-4 Miscellaneous**

| Pin Name | Type | Pin # | Cell | RESET# STATE | Description |
|---|---|---|---|---|---|
| CNF[3:0] | I | 46,47, 48,49 | C | As set by hardware | These inputs are used to configure the SSD1901 - see Table 7-6 "Summary of Power On/Reset Options," .<br>Must be connected directly to IO $V_{DD}$ or $V_{SS}$ |
| GPIO0 | IO,I | 22 | CS/TS1 | Input | This pin has multiple functions - see REG[63h] bit2.<br>• General Purpose Input/Output pin.<br>• Hardware Power Saving Input pin. |
| TESTEN | I | 44 | C | Hi-Z | Test Enable input. This input must be connected to Vss. |

**Table 7-5 Power Supply**

| Pin Name | Type | Pin # | Driver | Description |
|---|---|---|---|---|
| COREVDD | P | 1,21,41,61 | P | COREVDD pins are regulated supply pins of the chip. They are generated by the internal regulator and voltage at these pins are for internal reference only. They cannot be used for driving external circuitry.<br>0.1uF capacitors to VSS must be connected on each pin. |
| IOVDD | P | 10,29,52 | P | IO $V_{DD}$ |
| VSS | P | 20,27,40,50, 60,72,80 | P | Common $V_{SS}$ |

## 7.3 Summary of Configuration Options

**Table 7-6 Summary of Power On/Reset Options**

| Configuration Pin | Power On/Reset State | |
|---|---|---|
| | 1 (Connected to IOVDD) | 0 (Connected to VSS) |
| CNF3 | Big Endian | Little Endian |
| CNF[2:0] | Select host bus interface as follows:<br>**CNF2 CNF1 CNF0 BS# Host Bus**<br>0 0 X X reserved<br>0 1 0 X reserved<br>0 1 1 X MC68K #1, 16-bit<br>1 0 0 X reserved<br>1 0 1 X MC68K #2, 16-bit<br>1 1 0 X reserved<br>1 1 1 0 Generic #1, 16-bit<br>1 1 1 1 Generic #2, 16-bit | |

## 7.4 Host Bus Interface Pin Mapping

**Table 7-7 Host Bus Interface Pin Mapping**

| SSD1901 Pin Names | MC68K #1 | MC68K #2 | Generic #1 | Generic #2 |
|---|---|---|---|---|
| AB[16:1] | A[16:1] | A[16:1] | A[16:1] | A[16:1] |
| AB0 | LDS# | A0 | A0 | A0 |
| DB[15:0] | D[15:0] | D[31:16] | D[15:0] | D[15:0] |
| WE1# | UDS# | DS# | WE1# | BHE# |
| CS# | External Decode | External Decode | External Decode | External Decode |
| BCLK | CLK | CLK | BCLK | BCLK |
| BS# | AS# | AS# | connect to $V_{SS}$ | connect to IO $V_{DD}$ |
| RD/WR# | R/W# | R/W# | RD1# | connect to IO $V_{DD}$ |
| RD# | connect to IO $V_{DD}$ | SIZ1 | RD0# | RD# |
| WE0# | connect to IO $V_{DD}$ | SIZ0 | WE0# | WE# |
| WAIT# | DTACK# | DSACK1# | WAIT# | WAIT# |
| RESET# | RESET# | RESET# | RESET# | RESET# |

## 7.5 Pin State Summary

There are two data bus architectures, little endian and big endian. Following tables describes the pin state summary in difference interfaces.

### 7.5.1 MC68K#1 Pin State

**Table 7-8 MC68K#1 Pin State in Big Endian Mode (CNF3 = 1)**

| Operation | D15 - D8 | D7 - D0 | UDS# | LDS# |
|---|---|---|---|---|
| Address 2n Byte Read/Write | DATA[7:0] | - | 0 | 1 |
| Address 2n + 1 Byte Read/Write | - | DATA[7:0] | 1 | 0 |
| Address 2n Word Read/Write | DATA[15:8] | DATA[7:0] | 0 | 0 |
| Address 2n + 1 Word Read/Write | Not allow | Not allow | Not allow | Not allow |

**Table 7-9 MC68K#1 Pin State in Little Endian Mode (CNF3 = 0)**

| Operation | D15 - D8 | D7 - D0 | UDS# | LDS# |
|---|---|---|---|---|
| Address 2n Byte Read/Write | - | DATA[7:0] | 1 | 0 |
| Address 2n + 1 Byte Read/Write | DATA[7:0] | - | 0 | 1 |
| Address 2n Word Read/Write | DATA[15:8] | DATA[7:0] | 0 | 0 |
| Address 2n + 1 Word Read/Write | Not allow | Not allow | Not allow | Not allow |

### 7.5.2 MC68K#2 Pin State

**Table 7-10 MC68K#2 Pin State in Big Endian Mode (CNF3 = 1)**

| Operation | D15 - D8 | D7 - D0 | SIZ1 | SIZ0 | A0 |
|---|---|---|---|---|---|
| Address 2n Byte Read/Write | DATA[7:0] | - | 0 | 1 | 0 |
| Address 2n + 1 Byte Read/Write | - | DATA[7:0] | 0 | 1 | 1 |
| Address 2n Word Read/Write | DATA[15:8] | DATA[7:0] | 1 | 0 | 0 |
| Address 2n + 1 Word Read/Write* | - | DATA[15:8] | 1 | 0 | 1 |
| *This is a misalign operation where DATA[7:0] will be asseted in the next memory cycle. | | | | | |

**Table 7-11 MC68K#2 Pin State in Little Endian Mode (CNF3 = 0)**

| Operation | D15 - D8 | D7 - D0 | SIZ1 | SIZ0 | A0 |
|---|---|---|---|---|---|
| Address 2n Byte Read/Write | - | DATA[7:0] | 0 | 1 | 0 |
| Address 2n + 1 Byte Read/Write | DATA[7:0] | - | 0 | 1 | 1 |
| Address 2n Word Read/Write | DATA[15:8] | DATA[7:0] | 1 | 0 | 0 |
| Address 2n + 1 Word Read/Write* | DATA[7:0] | - | 1 | 0 | 1 |
| *This is a misalign operation where DATA[15:8] will be asseted in the next memory cycle. | | | | | |

### 7.5.3 Generic #1 Pin State

**Table 7-12 Generic #1 Pin State in Big Endian Mode (CNF3 = 1)**

| Operation | D15 - D8 | D7 - D0 | WE1# | WE0# | RD1# | RD0# |
|---|---|---|---|---|---|---|
| Address 2n Byte Write | DATA[7:0] | - | 0 | 1 | 1 | 1 |
| Address 2n + 1 Byte Write | - | DATA[7:0] | 1 | 0 | 1 | 1 |
| Address 2n Word Write | DATA[15:8] | DATA[7:0] | 0 | 0 | 1 | 1 |
| Address 2n + 1 Word Write | Not allow | Not allow | Not allow | Not allow | Not allow | Not allow |
| Address 2n Byte Read | DATA[7:0] | - | 1 | 1 | 0 | 1 |
| Address 2n + 1 Byte Read | - | DATA[7:0] | 1 | 1 | 1 | 0 |
| Address 2n Word Read | DATA[15:8] | DATA[7:0] | 1 | 1 | 0 | 0 |
| Address 2n + 1 Word Read | Not allow | Not allow | Not allow | Not allow | Not allow | Not allow |

**Table 7-13 Generic #1 Pin State in Little Endian Mode (CNF3 = 0)**

| Operation | D15 - D8 | D7 - D0 | WE1# | WE0# | RD1# | RD0# |
|---|---|---|---|---|---|---|
| Address 2n Byte Write | - | DATA[7:0] | 1 | 0 | 1 | 1 |
| Address 2n + 1 Byte Write | DATA[7:0] | - | 0 | 1 | 1 | 1 |
| Address 2n Word Write | DATA[15:8] | DATA[7:0] | 0 | 0 | 1 | 1 |
| Address 2n + 1 Word Write | Not allow | Not allow | Not allow | Not allow | Not allow | Not allow |
| Address 2n Byte Read | - | DATA[7:0] | 1 | 1 | 1 | 0 |
| Address 2n + 1 Byte Read | DATA[7:0] | - | 1 | 1 | 0 | 1 |
| Address 2n Word Read | DATA[15:8] | DATA[7:0] | 1 | 1 | 0 | 0 |
| Address 2n + 1 Word Read | Not allow | Not allow | Not allow | Not allow | Not allow | Not allow |

### 7.5.4 Generic #2 Pin State

**Table 7-14 Generic #2 Pin State in Big Endian Mode (CNF3 = 1)**

| Operation | D15 - D8 | D7 - D0 | BHE# | A0 |
|---|---|---|---|---|
| Address 2n Byte Read/Write | DATA[7:0] | - | 1 | 0 |
| Address 2n + 1 Byte Read/Write | - | DATA[7:0] | 0 | 1 |
| Address 2n Word Read/Write | DATA[15:8] | DATA[7:0] | 0 | 0 |
| Address 2n + 1 Word Read/Write* | - | DATA[15:8] | 0 | 1 |
| *This is a misalign operation where DATA[7:0] will be asseted in the next memory cycle. | | | | |

**Table 7-15 Generic #2 Pin State in Little Endian Mode (CNF3 = 0)**

| Operation | D15 - D8 | D7 - D0 | BHE# | A0 |
|---|---|---|---|---|
| Address 2n Byte Read/Write | - | DATA[7:0] | 1 | 0 |
| Address 2n + 1 Byte Read/Write | DATA[7:0] | - | 0 | 1 |
| Address 2n Word Read/Write | DATA[15:8] | DATA[7:0] | 0 | 0 |
| Address 2n + 1 Word Read/Write* | DATA[7:0] | - | 0 | 1 |
| *This is a misalign operation where DATA[15:8] will be asseted in the next memory cycle. | | | | |

## 7.6 LCD Interface Pin Mapping

**Table 7-16 LCD Interface Pin Mapping**

| SSD1901 Pin Name | 9-bit | 12-bit |
|---|---|---|
| FPFRAME | FPFRAME | |
| FPLINE | FPLINE | |
| FPSHIFT | FPSHIFT | |
| DRDY | DRDY | |
| FPDAT0 | R2 | R3 |
| FPDAT1 | R1 | R2 |
| FPDAT2 | R0 | R1 |
| FPDAT3 | G2 | G3 |
| FPDAT4 | G1 | G2 |
| FPDAT5 | G0 | G1 |
| FPDAT6 | B2 | B3 |
| FPDAT7 | B1 | B2 |
| FPDAT8 | B0 | B1 |
| FPDAT9 | GPIO2 | R0 |
| FPDAT10 | GPIO3 | G0 |
| FPDAT11 | GPIO4 | B0 |
| LCDPWR | LCDPWR | |

**Note**

Unused GPIO pins must be connected to IO $V_{DD}$

# 8 D.C. CHARACTERISTICS

**Table 8-1 Absolute Maximum Ratings**

| Symbol | Parameter | Rating | Units |
|---|---|---|---|
| IO $V_{DD}$ | Supply Voltage | $V_{SS}$ - 0.3 to 4.0 | V |
| $V_{IN}$ | Input Voltage | $V_{SS}$ - 0.3 to IO $V_{DD}$ + 0.5 | V |
| $V_{OUT}$ | Output Voltage | $V_{SS}$ - 0.3 to IO $V_{DD}$ + 0.5 | V |
| $T_{STG}$ | Storage Temperature | -65 to 150 | °C |

**Table 8-2 Recommended Operating Conditions for IO VDD = 3.0V ± 10%**

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| IO $V_{DD}$ | Supply Voltage | 3.0 | 3.3 | 3.6 | V |
| $V_{IN}$ | Input Voltage | $V_{SS}$ | | 5.0 | V |
| $T_{OPR}$ | Operating Temperature | -30 | 25 | 85 | °C |

**Table 8-3 Input Specifications**

| Symbol | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|
| $P_{SAVE}$ | Power Consumption at Power Saving mode | IOVDD = 3.0V, BCLK and CLKI inactive | | 700 | µW |
| $P_{DISPLAY}$ | Power Consumption at Display Mode | RAM with checker-board, CLKI = 4MHz, BCLK inactive, IOVDD = 3.0V No Loading | | 11 | mW |
| $V_{IL}$ | Low Level Input Voltage | | | 0.8 | V |
| $V_{IH}$ | High Level Input Voltage | | IO $V_{DD}$ - 0.8 | | V |
| $V_{T+}$ | Positive-going Threshold Schmitt inputs | | 1.1 | | V |
| $V_{T-}$ | Negative-going Threshold Schmitt inputs | | | 0.94 | V |
| $I_{IZ}$ | Input Leakage Current | $V_{DD}$ = Max<br>$V_{IH} = V_{DD}$<br>$V_{IL} = V_{SS}$ | -1 | 1 | µA |
| $C_{IN}$ | Input Pin Capacitance | | | 10 | pF |

**Table 8-4 Output Specifications**

| Symbol | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|
| $V_{OL}$ | Low Level output Voltage | | | 0.4 | V |
| $V_{OH}$ | High Level Output Voltage | | IO $V_{DD}$ - 0.4 | | V |
| $I_{OZ}$ | Output Leakage Current | $V_{DD}$ = MAX | -1 | 1 | µA |
| $V_{OUT}$ | Output Pin Capacitance | | | 10 | pF |
| $C_{BID}$ | Bidirectional Pin Capacitance | | | 10 | pF |

# 9 A.C. CHARACTERISTICS

Conditions: IO $V_{DD}$ = 3.0V - 3.6V
$T_A$ = -30°C - 85°C
$T_{rise}$ and $T_{fall}$ for all inputs must be $\leq$ 5 nsec (10% ~ 90%)
$C_L$ = 60pF (Bus/CPU Interface)
$C_L$ = 60pF (LCD Panel Interface)

## 9.1 Bus Interface Timing

### 9.1.1 Motorola MC68K #1 Interface Timing

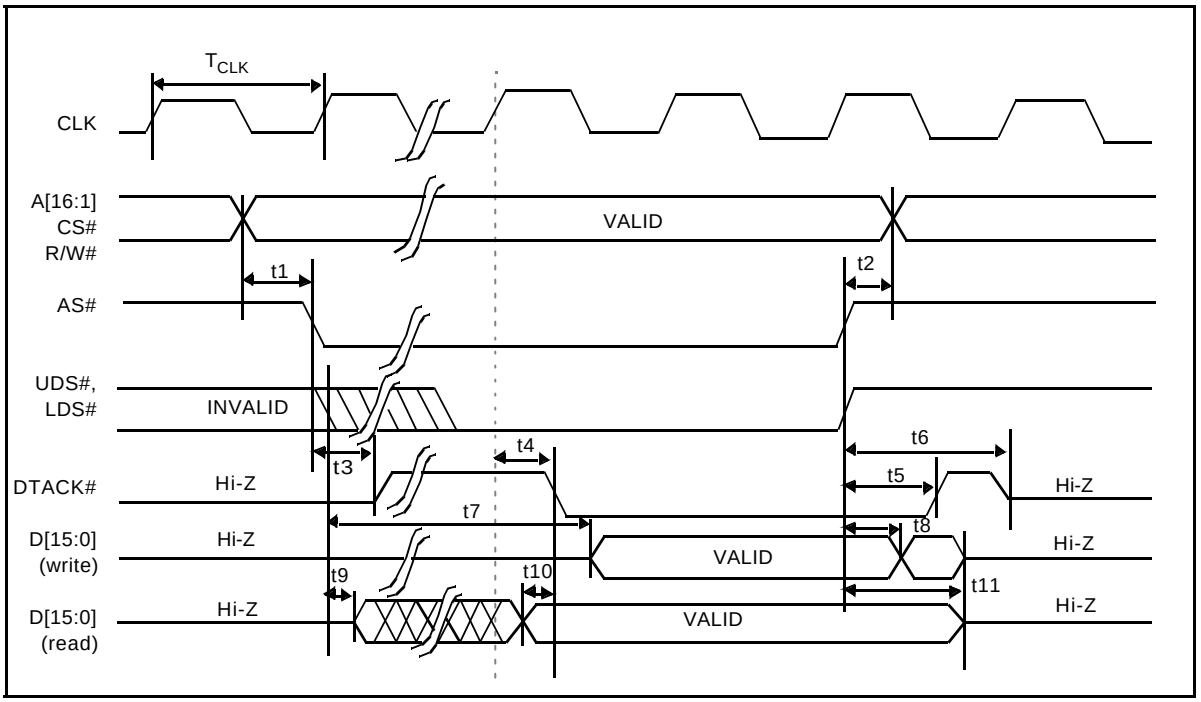


**Figure 9-1 MC68K #1 Bus Timing (MC68000)**

**Table 9-1 MC68K #1 Bus Timing (MC68000)**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $f_{CLK}$ | Bus Clock Frequency | 0 | 33 | MHz |
| $T_{CLK}$ | Bus Clock period | $1/f_{CLK}$ | | |
| t1 | A[16:1], CS# valid before AS# falling edge | 0 | | ns |
| t2 | A[16:1], CS# hold from AS# rising edge | 0 | | ns |
| t3 | AS# low to DTACK# driven high | | 16 | ns |
| t4 | CLK to DTACK# low | | 15 | ns |
| t5 | AS# high to DTACK# high | | 20 | ns |
| t6 | AS# high to DTACK# high impedance | | $T_{CLK}$ | |
| t7 | UDS#, LDS# falling edge to D[15:0] valid (write cycle) | | $T_{CLK}$ | |
| t8 | D[15:0] hold from AS# rising edge (write cycle) | 0 | | ns |
| t9 | UDS#, LDS# falling edge to D[15:0] driven (read cycle) | | 15 | ns |
| t10 | D[15:0] valid to DTACK# falling edge (read cycle) | 0 | | ns |
| t11 | UDS#, LDS# rising edge to D[15:0] high impedence | | 10 | ns |

### 9.1.2 Motorola MC68K #2 Interface Timing

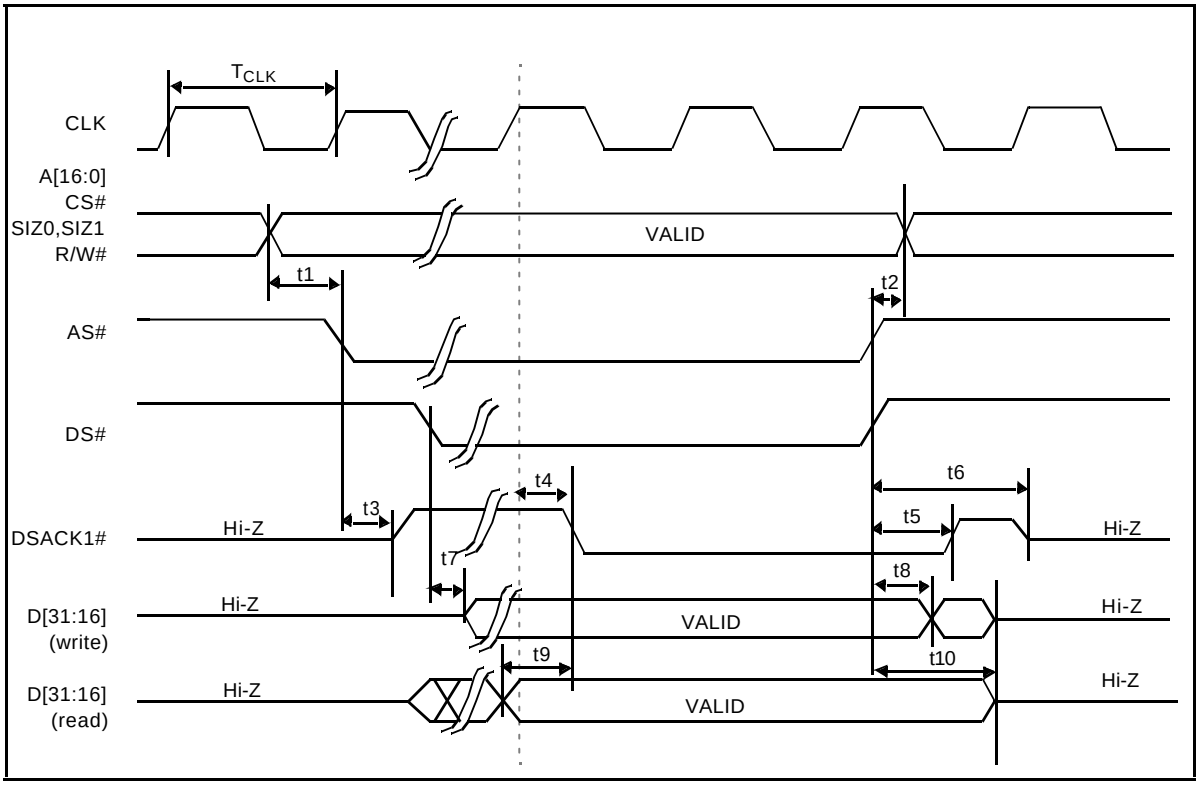


**Figure 9-2 MC68K #2 Timing (MC68030)**

**Table 9-2 MC68K #2 Timing (MC68030)**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $f_{CLK}$ | Bus Clock Frequency | 0 | 33 | MHz |
| $T_{CLK}$ | Bus Clock period | $1/f_{CLK}$ | | |
| t1 | A[16:0],CS#, SIZ0, SIZ1 valid before AS# falling edge | 0 | | ns |
| t2 | A[16:0], CS#, SIZ0, SIZ1 hold from AS# rising edge | 0 | | ns |
| t3 | AS# low to DSACK1# driven high | | 22 | ns |
| t4 | CLK to DSACK1# low | | 18 | ns |
| t5 | AS# high to DSACK1# high | | 20 | ns |
| t6 | AS# high to DSACK1# high impedance | | $T_{CLK}$ | |
| t7 | DS# falling edge to D[31:16] valid (write cycle) | | $T_{CLK}/2$ | |
| t8 | AS#, DS# rising edge to D[31:16] invalid (write cycle) | 0 | | ns |
| t9 | D[31:16] valid to DSACK1# low (read cycle) | 0 | | ns |
| t10 | AS#, DS# rising edge to D[31:6] high impedance | | 20 | ns |

### 9.1.3 Generic #1 Interface Timing

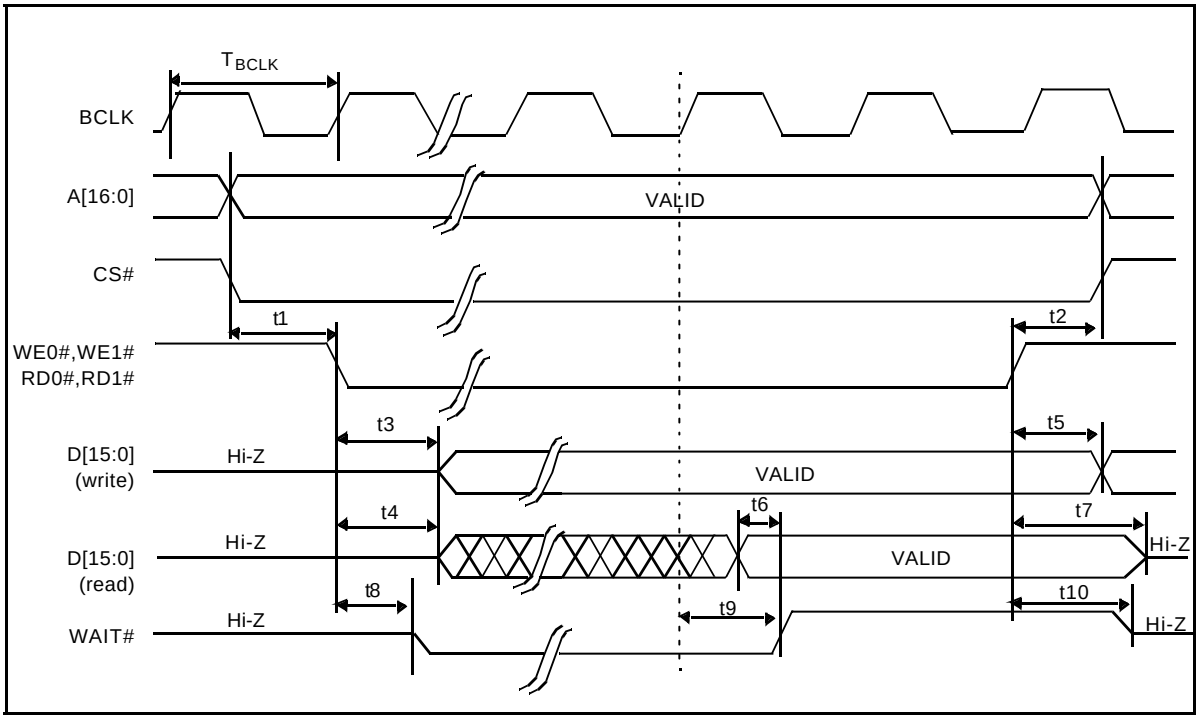


**Figure 9-3 Generic #1 Timing**

**Table 9-3 Generic #1 Timing**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $f_{BCLK}$ | Bus Clock Frequency | 0 | 50 | MHz |
| $T_{BCLK}$ | Bus Clock period | $1/f_{BCLK}$ | | |
| t1 | A[16:0],CS# valid to WE0#, WE1# low (write cycle) or RD0#, RD1# low (read cycle) | 0 | | ns |
| t2 | WE0#, WE1# high (write cycle) or RD0#, RD1# high (read cycle) to A[16:0], CS# invalid | 0 | | ns |
| t3 | WE0#, WE1# low to D[15:0] valid (write cycle) | | $T_{BCLK}$ | ns |
| t4 | RD0#, RD1# low to D[15:0] driven (read cycle) | | 17 | ns |
| t5 | WE0#, WE1# high to D[15:0] invalid (write cycle) | 0 | | ns |
| t6 | D[15:0] valid to WAIT# high (read cycle) | 0 | | ns |
| t7 | RD0#, RD1# high to D[15:0] high impedance (read cycle) | | 10 | ns |
| t8 | WE0#, WE1# low (write cycle) or RD0#, RD1# low (read cycle) to WAIT# driven low | | 16 | ns |
| t9 | BCLK to WAIT# high | | 16 | ns |
| t10 | WE0#, WE1# high (write cycle) or RD0#, RD1# high (read cycle) to WAIT# high impedance | | 16 | ns |

### 9.1.4 Generic #2 Interface Timing

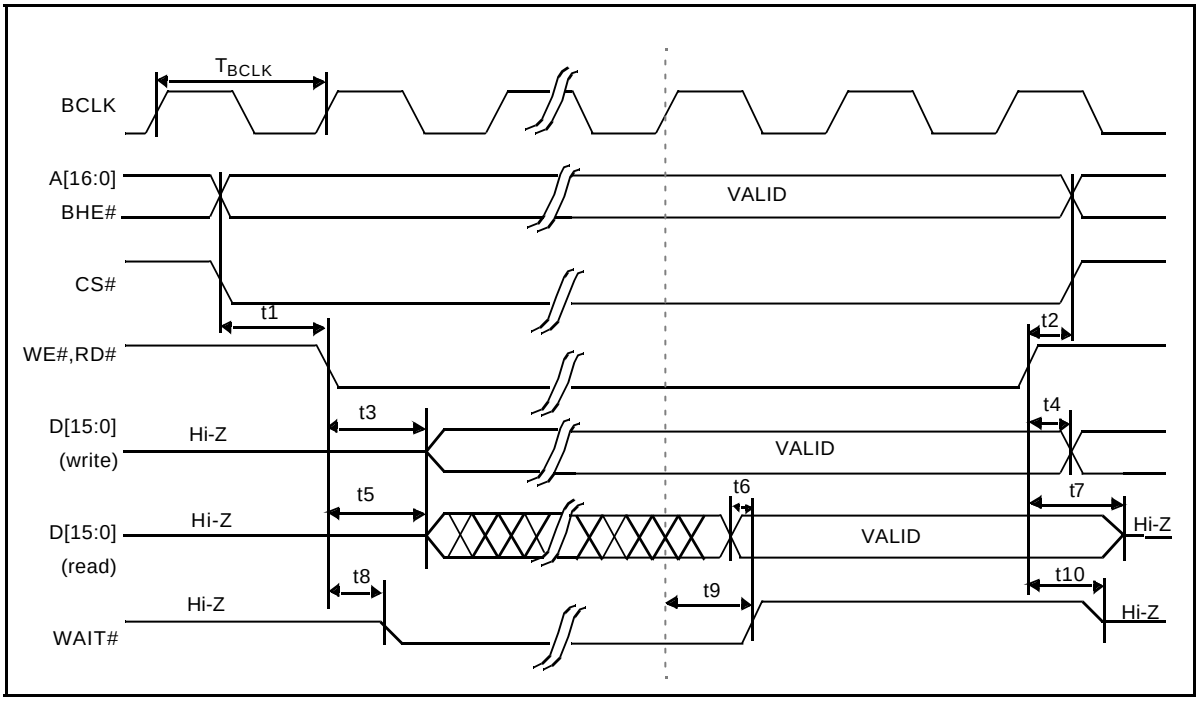


**Figure 9-4 Generic #2 Timing**

**Table 9-4 Generic #2 Timing**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $f_{BCLK}$ | Bus Clock frequency | 0 | 50 | MHz |
| $T_{BCLK}$ | Bus Clock period | $1/f_{BCLK}$ | | |
| t1 | A[16:0], BHE#, CS# valid to WE#, RD# low | 0 | | ns |
| t2 | WE#, RD# high to A[16:0], BHE#, CS# invalid | 0 | | ns |
| t3 | WE# low to D[15:0] valid (write cycle) | | $T_{BCLK}$ | |
| t4 | WE# high to D[15:0] invalid (write cycle) | 0 | | ns |
| t5 | RD# low to D[15:0] driven (read cycle) | | 16 | ns |
| t6 | D[15:0] valid to WAIT# high (read cycle) | 0 | | ns |
| t7 | RD# high to D[15:0] high impedance (read cycle) | | 10 | ns |
| t8 | WE#, RD# low to WAIT# driven low | | 14 | ns |
| t9 | BCLK to WAIT# high | | 10 | ns |
| t10 | WE#,RD# high to WAIT# high impedance | | 11 | ns |

## 9.2 Clock Input Requirements

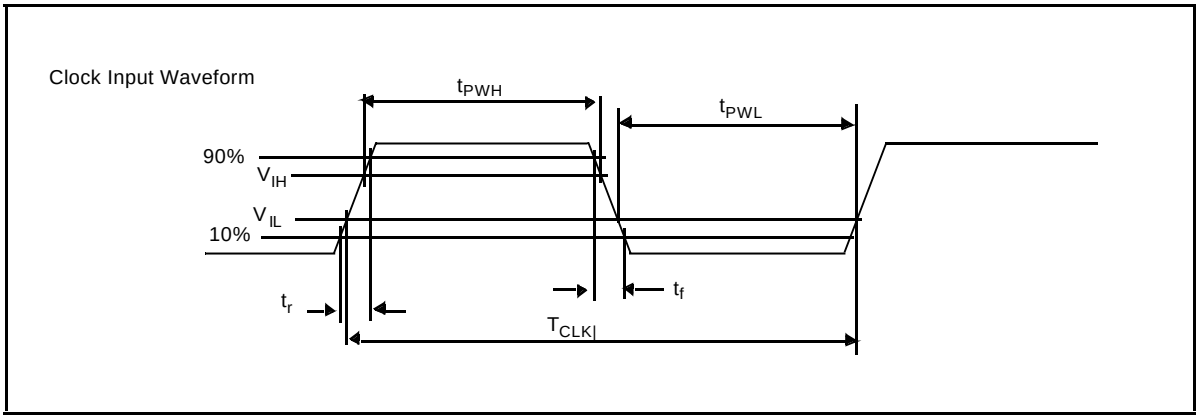


**Figure 9-5 Clock Input (CLKI) Requirements**

**Table 9-5 Clock Input (CLKI) Requirements**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $f_{CLKI}$ | Input Clock Frequency | 0 | 50 | MHz |
| $T_{CLKI}$ | Input Clock period | $1/f_{CLKI}$ | | |
| $t_{PWH}$ | Input Clock Pulse Width High | 8 | | ns |
| $t_{PWL}$ | Input Clock Pulse Width Low | 8 | | ns |
| $t_f$ | Input clock Fall Time (10%-90%) | | 5 | ns |
| $t_r$ | Input Clock Rise Time (10% - 90%) | | 5 | ns |

**Note**

Where CLKI is > 25 MHz it must be divided by 2 (REG[62h] bit 4 = 1).

## 9.3 Display Interface

### 9.3.1 Power On/Reset Timing

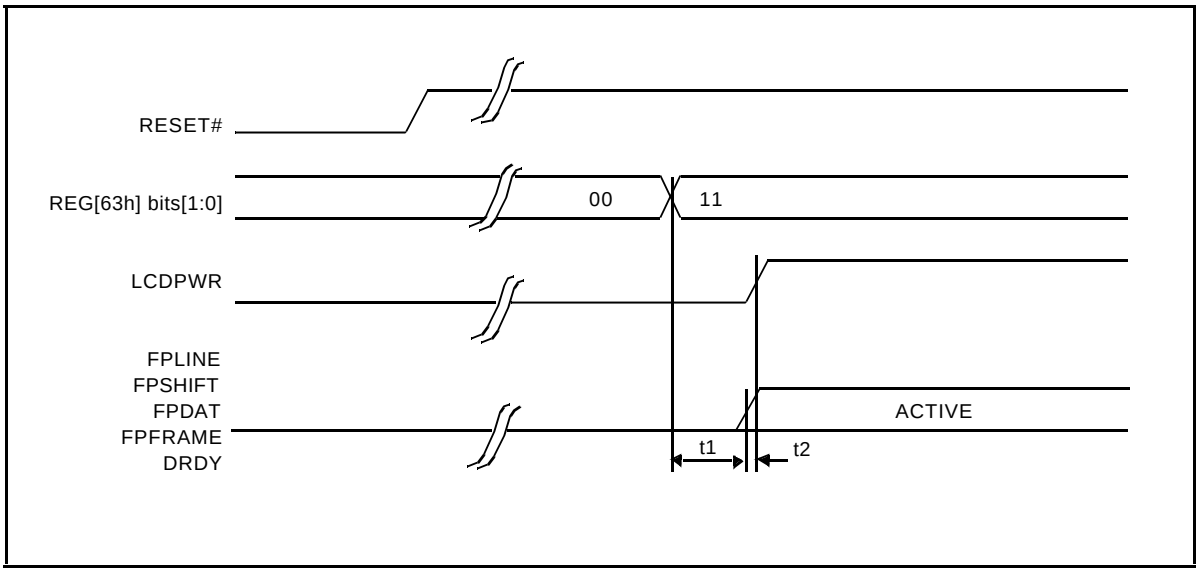


**Figure 9-6 LCD Panel Power On/Reset Timing**

**Table 9-6 LCD Panel Power On/Reset Timing**

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| t1 | REG[63h] to FPLINE, FPFRME, FPSHIFT, FPDAT, DRDY active | | | $T_{FPFRAME}$ | ns |
| t2 | FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY active to LCDPWR | | 0 | | Frame |

**Note**

Where $T_{FPFAME}$ is the period of FPFRAME

### 9.3.2 Power Down/Up Timing

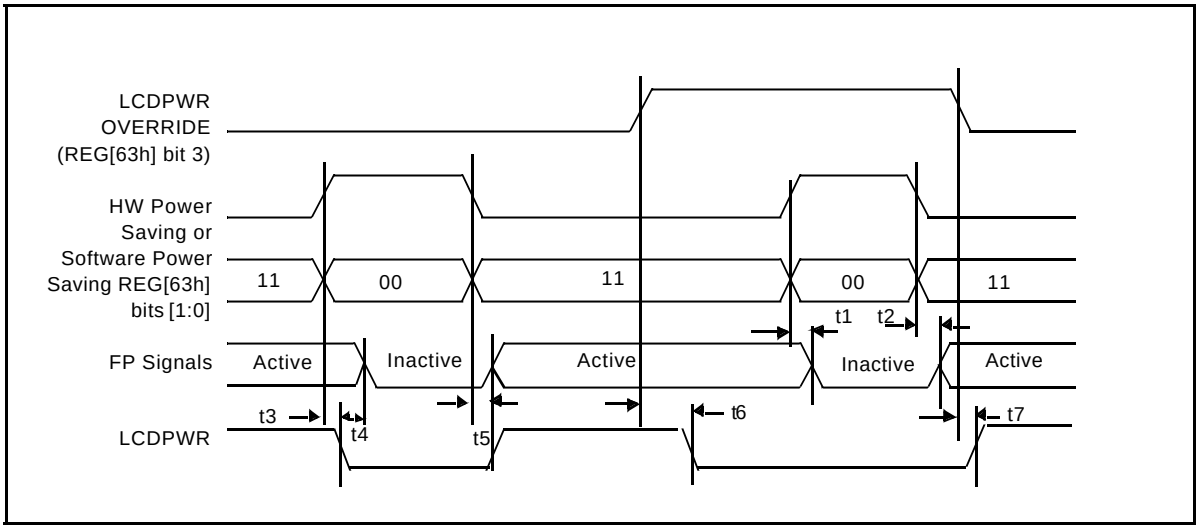


**Figure 9-7 Power Down/Up Timing**

**Table 9-7 Power Down/Up Timing**

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| t1 | HW Power Saving active to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY inactive - LCDPWR Override = 1 | | | 1 | Frame |
| t2 | HW Power Saving inactive to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY active - LCDPWR Override = 1 | | | 1 | Frame |
| t3 | HW Power Saving active to LCDPWR low - LCDPWR Override = 0 | | | 1 | Frame |
| t4 | LCDPWR low to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY inactive - LCDPWR Override = 0 | | 127 | | Frame |
| t5 | HW Power Saving inactive to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY, LCDPWR active - LCDPWR Override = 0 | | 0 | | Frame |
| t6 | LCDPWR Override active (1) to LCDPWR inactive | | | 1 | Frame |
| t7 | LCDPWR Override inactive (1) to LCDPWR active | | | 1 | Frame |

### 9.3.3 9/12-Bit TFT Panel Timing

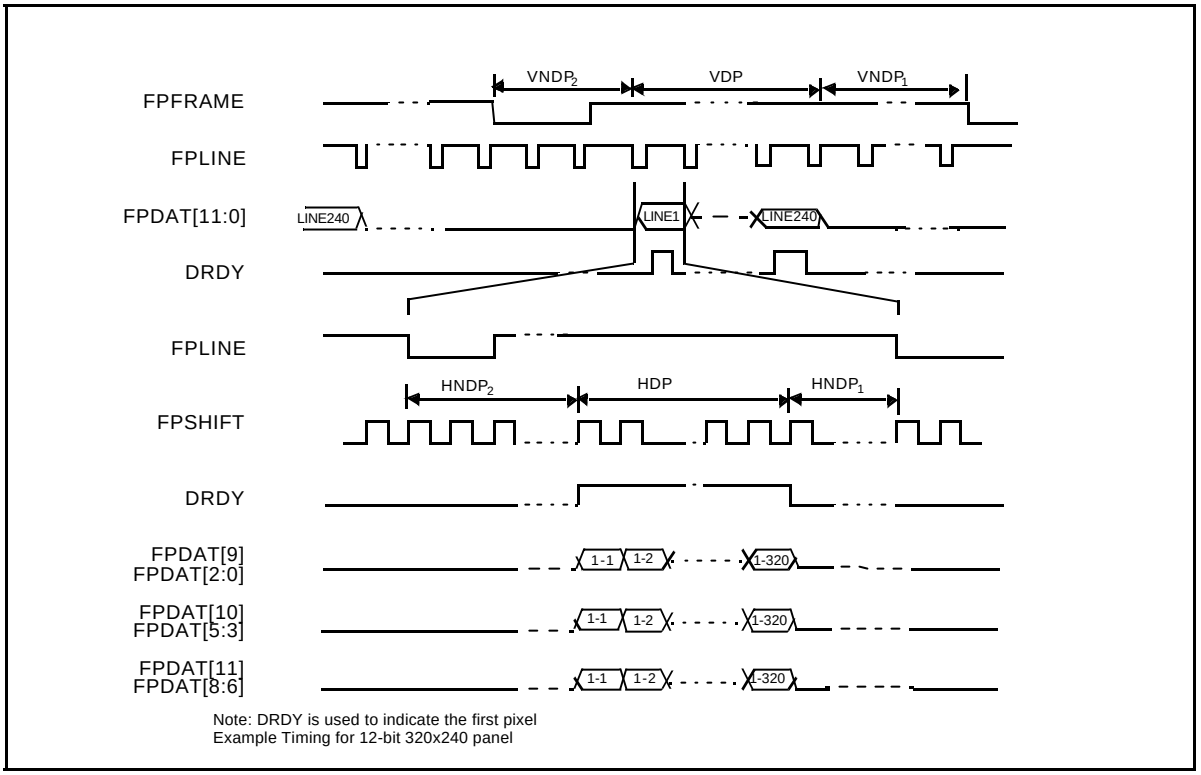


**Figure 9-8 9/12-Bit TFT Panel Timing**

| | | |
|---|---|---|
| VDP | = Vertical Display Period | = (REG[66h] bits 1-0, REG[65h] bits 7-0) + 1 Lines |
| VNDP | = Vertical Non-Display Period | = VNDP1 + VNDP2 = (REG[6Ah] bits 5-0) Lines |
| VNDP1 | = Vertical Non-Display Period 1 | = (REG[69h] bits 5-0) Lines |
| VNDP2 | = Vertical Non-Display Period 2 | = (REG[6Ah] bits 5-0) - (REG[69h] bits 5-0) Lines |
| HDP | = Horizontal Display Period | = ((REG[64h] bits 6-0) + 1) x 8Ts |
| HNDP | = Horizontal Non-Display Period | = HNDP1 + HNDP2 = (REG[68h] bits 4-0 + 4) x 8Ts |
| HNDP1 | = Horizontal Non-Display Period 1 | = ((REG[67h] bits 4-0) x 8) + 16Ts |
| HNDP2 | = Horizontal Non-Display Period 2 | = (((REG[68h] bits 4-0) - (REG[67h] bits 4-0)) x 8) + 16$T_S$ |
| $T_S$ | = Pixel Clock Period | |

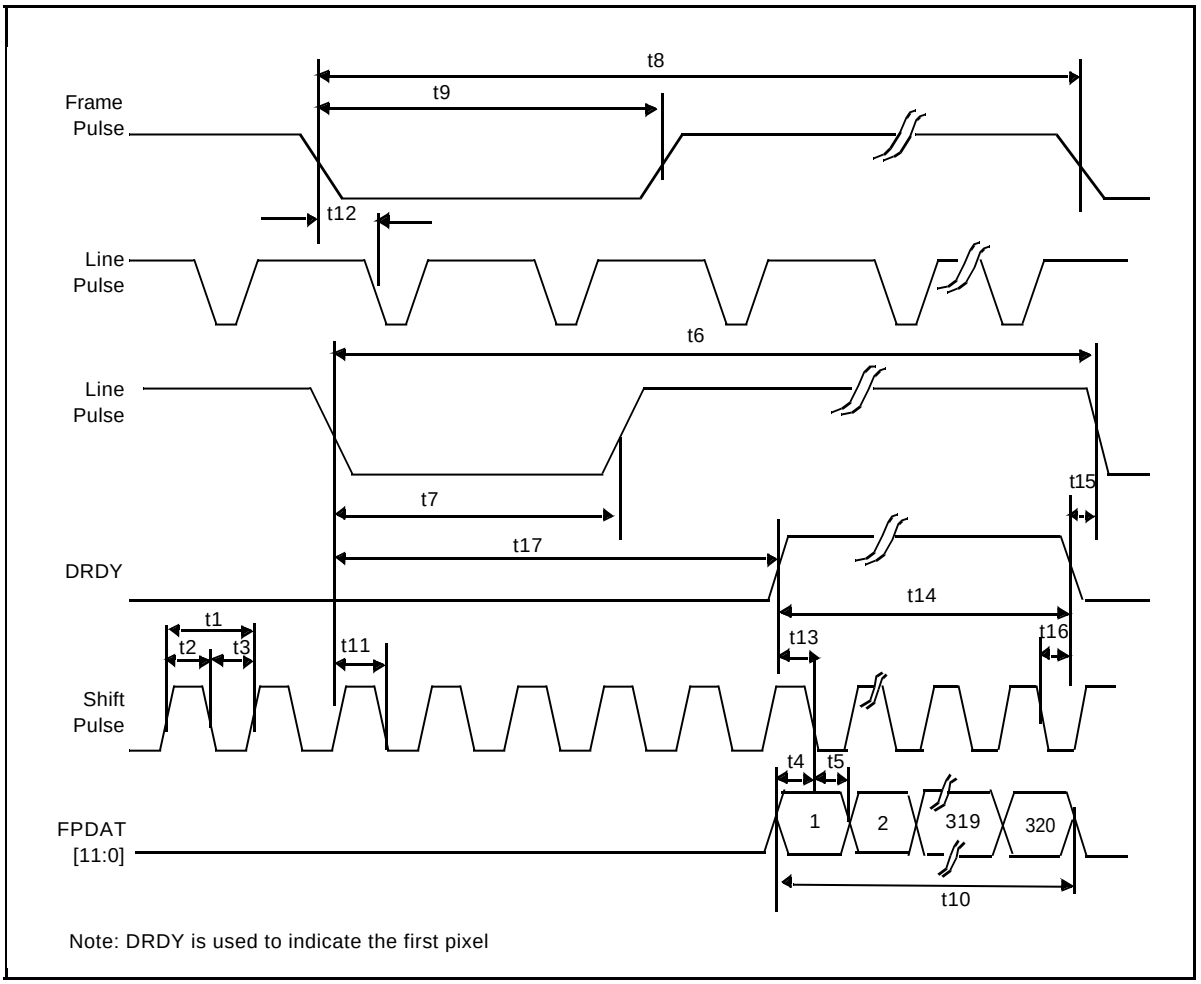


**Figure 9-9 TFT A.C. Timing**

**Table 9-8 TFT A.C. Timing**

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| t1 | Shift Pulse period | 1 | | | Ts<br>note 1 |
| t2 | Shift Pulse width high | 0.5 | | | Ts |
| t3 | Shift Pulse width low | 0.5 | | | Ts |
| t4 | Data setup to Shift Pulse falling edge | 0.5 | | | Ts |
| t5 | Data hold from Shift Pulse falling edge | 0.5 | | | Ts |
| t6 | Line Pulse cycle time | note 2 | | | |
| t7 | Line Pulse pulse width low | 9 | | | Ts |

**Table 9-8 TFT A.C. Timing**

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| t8 | Frame Pulse cycle time | note 3 | | | |
| t9 | Frame Pulse pulse width low | 2t6 | | | |
| t10 | Horizontal display period | note 4 | | | |
| t11 | Line Pulse setup to Shift Pulse falling edge | 0.5 | | | Ts |
| t12 | Frame Pulse falling edge to Line Pulse falling edge phase difference | t6 - 18Ts | | | |
| t13 | DRDY to Shift Pulse falling edge setup time | 0.5 | | | Ts |
| t14 | DRDY pulse width | note 5 | | | |
| t15 | DRDY falling edge to Line Pulse falling edge | note 6 | | | |
| t16 | DRDY hold from Shift Pulse falling edge | 0.5 | | | Ts |
| t17 | Line Pulse Falling edge to DRDY active | note 7 | | 250 | Ts |

**Note:**

1. Ts = pixel clock period
2. t6min = [((REG[64h] bits 6-0) + 1) x 8 + ((REG[68h] bits 4-0) + 4) x 8] Ts
3. t8min = [((REG[66h] bits 1-0, REG[65h] bits 7-0) + 1) + (REG[6Ah] bits 5-0)] Lines
4. t10min = [((REG[64h] bits 6-0) + 1) x 8] Ts
5. t14min = [((REG[64h] bits 6-0) + 1) x 8] Ts
6. t15min = [(REG[67h] bits 4-0) x 8 + 16] Ts
7. t17min = [((REG[68h] bits 4-0) - (REG[67] bits 4-0)) x 8 + 16] Ts

# 10 Frame Rate Calculation

The following formula is used to calculate the display frame rate.

$$\text{FrameRate} = \frac{f_{PCLK}}{(HDP + HNDP) \times (VDP + VNDP)}$$

Where: $f_{PCLK}$ = Pixel Clock (PCLK) frequency (Hz)
HDP = Horizontal Display Period = ((REG[64h] bits 6-0) + 1) x 8 Pixels
HNDP = Horizontal Non-Display Period = ((REG[68h] bits 4-0) + 4) x 8 Pixels
VDP = Vertical Display Period = ((REG[66h] bits 1-0, REG[65h] bits 7-0) + 1) Lines
VNDP = Vertical Non-Display Period = (REG[6Ah] bits 5-0) Lines

# 11 Clock Requirements

The following table shows the required clock for the different function in the SSD1901. The operating clock (CLK) is derived from either CLKI or BCLK input. REG[0Bh] is used to select the clock source.

| Function | CLK |
|---|---|
| Register Read/Write | Not Needed |
| Mermory Read/Write | Needed |
| Look-Up Table Register Read/Write | Not Needed |
| Software Power Saving | Can be stopped after 128 frames from entering Software Power Saving, i.e. after REG[63h] bits 1-0 = 11b |
| Hardware Power Saving | Can be stopped after 128 frames from entering Hardware Power Saving |

**Figure 11-1 SSD1901 Internal Clock Requirements**

# 12 Virtual Display Mode

Virtual Display Mode refers to the situation where the image area to be viewed is larger than the physical display area. The difference can be in the horizontal, vertical or both dimensions. To view the image, the display is used as a window into the display memory. At any given time only a portion of the image is visible. Panning and scrolling are used to view the full image.
The Memory Address Offset register determines the number of horizontal pixels in the virtual image area. The offset register can be used to specify from 0 to 255 additional words for each scan line, i.e. span an additional 510 pixels.

The maximum possible vertical size of the virtual image area is the result of dividing 81920 bytes of display memory by the number of bytes on each line.

Figure 12-1 "Display Area Inside a Virtual Display" depicts a typical use of a virtual display area. The display panel is 240x160 pixels, an image of 320x240 pixels can be viewed by navigating a 240x160 pixel display area around the image using panning and scrolling.

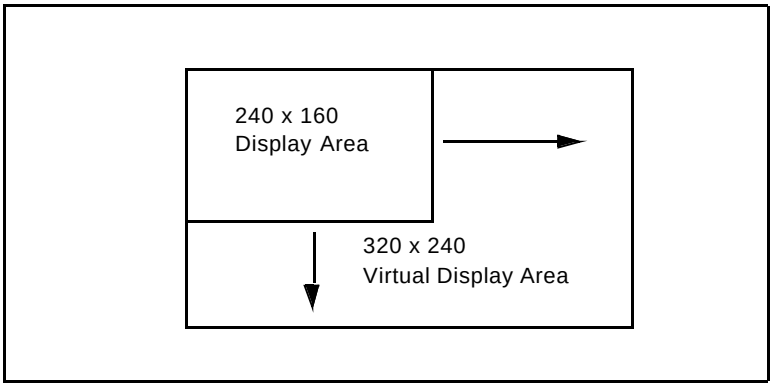


**Figure 12-1 Display Area Inside a Virtual Display**

Panning and scrolling are the operations of moving a physical display area about a virtual image in order to view the

entire image a portion at time. For example, after setting up the previous example (virtual display) and drawing an image into it we would only be able to view one portion of the image. Panning and scrolling are used to view the rest of the image.
Panning describes the horizontal (side to side) motion of the display area. When panning to the right the image in the display area appears to slide to the left. When panning to the left the image to appears to slide to the right. Scrolling describes the vertical (up and down) motion of the display area. Scrolling down causes the image to appear to slide up and scrolling up causes the image to appear to slide down.

Both panning and scrolling are performed by modifying the start address registers. The start address registers are a word offset to the data to be displayed in the top left corner of a frame. Changing the start address by one means a change on the display of the number of pixels in one word. There are two pixels in each word. So 2 pixels represents the smallest step we can pan to the left or right.

## 12.1 Registers

Virtual Display is performed by manipulating two register sets. The Memory Address Offset determine the word offset of the display. The Screen 1 Start Address determine the starting memory of Screen 1.

### 12.1.1 Memory Address Offset Register

REG[71h] forms an 8-bit value called the Memory Address Offset. This offset is the number of additional words on each line of the display. If the offset is set to zero there is no virtual width.

**Note**
This value does not represent the number of words to be shown on the display. The display width is set in the Horizontal Display Width register.

### 12.1.2 Screen 1 Start Address Registers

REG[6Ch] and REG[6Dh] form the 16 bits screen 1 start address. Screen 1 is displayed starting at the top left corner of the display. These registers form the word offset to the first byte in display memory to be displayed in the upper left corner of the screen. Changing these registers by one will shift the display image 2 pixels.

# 13 Split Screen Display

Occasionally the need arises to display two different but related images. Take, for example, a game where the main play area requires rapid updates and game status, displayed at the bottom of the screen, requires infrequent updates. The Split Screen feature allows a programmer to setup a display in such a manner. With proper configuration, the programmer has to update the main area on a regular basis only. Occasionally, the secondary area is updated when needed.

Figure 13-1 "240x160 Single Panel For Split Screen" illustrates how a 240x160 panel may be configured to have one image displaying from scan line 0 to scan line 99 and image 2 displaying from scan line 100 to scan line 159. Although this example picks specific values, the split between image 1 and image 2 may occur at any line of the display.

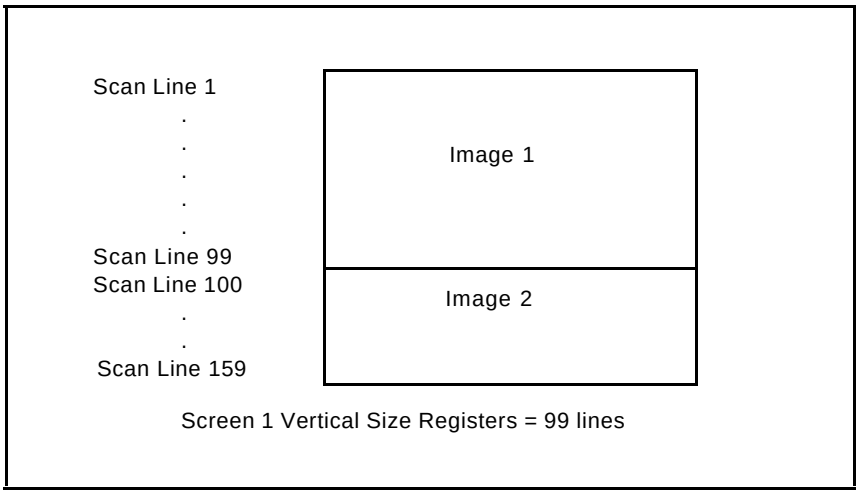


**Figure 13-1 240x160 Single Panel For Split Screen**

In split screen operation "Image 1" is taken from the display memory location pointed to by the Screen 1 Start Address registers and is always located at the top of the screen. "Image 2" is taken from the display memory location pointed to by the Screen 2 Start Address registers. The line where "Image 1" end and "Image 2" begins is determined by the Screen 1 Vertical Size register.

## 13.1 Registers

Split screen operation is performed by manipulating three register sets. Screen 1 Start Address and Screen 2 Start Address determine the display memory of the first and second images will be taken from. The Vertical Size registers determine how many lines Screen 1 will use. The following is a description of the registers used to do split screen.

### 13.1.1 Screen 1 Vertical Size

REG[72h] and REG[73h] form a ten bit value which determines the size of screen 1. When the vertical size is equal to or greater than the physical number of lines being displayed there is no visible effect on the display. When the vertical size value is less than the number of physical display lines, operation is like this:

1. From the beginning of a frame to the number of lines indicated by vertical size the display data will come from the memory area pointed to by the Screen 1 Display Start Address.
2. After vertical size lines have been displayed, the system will begin displaying data from the memory area pointed by Screen 2 Display Start Address.

One thing that must be pointed out here is that Screen 1 display is always displayed at the top of the screen followed

by screen 2 display. This relationship holds regardless of where in display memory Screen 1 Start Address and Screen 2 Start Address are pointing.
For instance, Screen 2 Start Address may point to offset zero of display memory while Screen 1 Start Address points to a location several thousand bytes higher. Screen 1 will still be shown first on the display. While not particularly useful, it is even possible to set screen 1 and screen 2 to the same address.

### 13.1.2 Screen 2 Start Address Registers

REG[6Eh] and REG[6Fh] form the 16 bits Screen 2 Start Address. Screen 2 is always displayed immediately following the screen 1 data and will begin at the left-most pixel on a line. Keep in mind that if the Screen 1 Vertical Size is equal to or greater than the physical display then Screen 2 will not be shown.
These registers form the word offset to the first byte in display memory to be displayed. Changing these registers by one will shift the display image 2 pixels.

Screen 1 Start Address registers, REG[6Ch] and REG[6Dh] are discussed in Section 12.1.2 "Screen 1 Start Address Registers" on page 46.

# 14 Fixed Window Mode

Fixed Window Mode simplifies the data update process for an image in a window area on the main display area. Data read/write accesses under Fixed Window Mode are implemented by memory address translation between the external address bus and the internal memory. Address on external address bus represents the window address which is re-mapped to the corresponding memory address. Any address outside the window area will be ignored except the Fixed Window Mode disabled (REG[00h] bit 4 = 0). Bus speed will be slowed down if this mode is enabled. It also provides blinking and invert functions for the fixed window area.

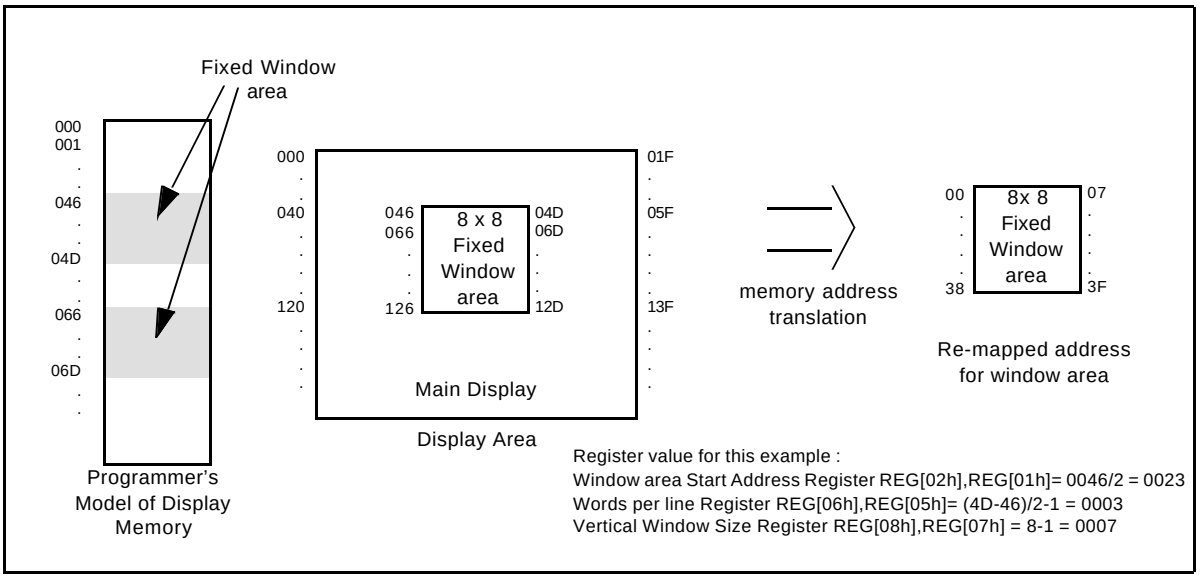


**Figure 14-1 Description for Window Mode**

## 14.1 Registers

Fixed Window Mode operation is performed by manipulating six register sets. These registers provide functions as followings :

Fixed Window Read / Write
- Select the window area start address, words per line and vertical window size for the Fixed window area.
- Enable the Fixed Window mode for data read / write accesses. Address on address bus remaps to the corresponding memory address.
- Disable the Fixed Window Mode, return normal operation. Address on address bus directly points to the memory address.

Fixed Window Blinking
- Select the window area start address, words per line and vertical window size for the Fixed Window area.
- Select the number of frame periods for blinking.
- Select the blinking color as the address of LUT.
- Enable the Window Blinking. Then the window area periodically changes to the blinking color. Data in memory does not change.
- Disable the Window Blinking, return to normal operation.

Fixed Window Invert
- Select the window area start address, words per line and vertical window size for the Fixed Window area.
- Enable the Window Invert. Then the display data within the window area inverts after the LUT. Data in memory does not change.
- Disable the Window Invert, return normal operation.

### 14.1.1 Window Features Code Register

REG[00h] is used to enable the window feature, select the Window Mode feature, enable blinking and invert functions for Window Mode.

### 14.1.2 Window Area Start Addres Register

REG[01h] and REG[02h] form the 16 bits value to determine the window area start address. Changing these registers by one will shift the window area 2 pixels.

### 14.1.3 Words per line Register

REG[05h] and REG[06h] form the 16 bits value to determine the number of words per line for Window Mode.

### 14.1.4 Vertical Window Size Register

REG[07h] and REG[08h] form 10 bits value to determine vertical resolution for Window Mode.

### 14.1.5 Look-Up Table Address for Blinking Register

REG[09h] represents the LUT address pointer for Window Blinking.

### 14.1.6 Frame Period for Blinking Register

REG[0Ah] controls the number of frame period for Window Blinking.

# 15 Floating Window Mode

Floating Window Mode allows to display an image in a floating window area within the main display without erasing the original image. There is separated memory that will be allocated for the Floating Window. Contents of Floating Window area is fetched from the Floating Window Buffer Memory. It also provide blinking and invert functions for the floating window area.

**Note :**
- Fixed Window Mode and Floating Window Mode cannot be enabled concurrently.

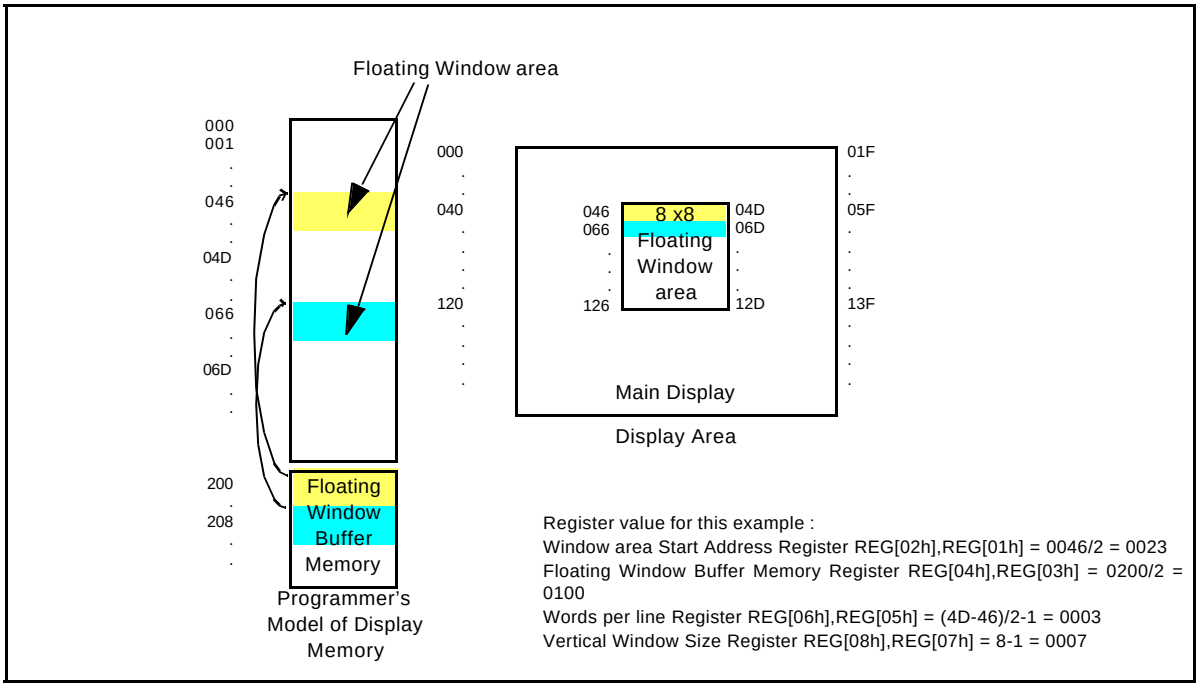


**Figure 15-1 Description for Floating Window Mode**

## 15.1 Registers

Floating Window Mode operation is performed by manipulating seven register sets. These registers provide functions as followings :

Floating Window Display
- Select the window area start address, start address for Floating Window Buffer Memory, words per line and vertical window size for the Floating Window area.
- Enable the Floating Window mode. Then the contents of floating window buffer memory is fetched for floating window area.
- Disable the Floating Window mode, return to normal operation.

Floating Window Blinking
- Select the window area start address, start address for Floating Window Buffer Memory, words per line and vertical window size for the Floating Window area.
- Select the number of frame periods for blinking.
- Select the blinking color as the address of LUT.
- Enable the Window Blinking. Then the window area periodically changes to the blinking color.
- Disable the Window Blinking, return to normal operation.

Floating Window Invert

- Select the window area start address, start address for Floating Window Buffer Memory, words per line and vertical window size for the Floating Window area.
- Enable the Window Invert. Then the display data withing the window are inverts after the LUT.
- Disable the Window Invert, return to normal operation.

### 15.1.1 Floating Window Buffer Memory Start Addres Register

REG[03h] and REG[04h] form the 16 bits value to determine the start address for Floating Window Buffer Memory. These buffer memory is a separated memory that will be allocated for the Floating Window.

Window Features Code register, REG[00h] is discussed in Section 14.1.1 "Window Features Code Register" on page 50.

Window Area Start Address register, REG[01h] and REG[02h] are discussed in Section 14.1.2 "Window Area Start Addres Register" on page 50.

Words per line registers, REG[05h] and REG[06h] are discussed in Section 14.1.3 "Words per line Register" on page 50.

Veritcal Window Size register, REG[07h] and REG[08h] are discussed in Section 14.1.4 "Vertical Window Size Register" on page 50.

Look-Up Table Address for Blinking register REG[09h] is discussed in Section 14.1.5 "Look-Up Table Address for Blinking Register" on page 50.

Frame Period for Blinking Register, REG[0Ah] is discussed in Section 14.1.6 "Frame Period for Blinking Register" on page 50.

# 16 Power Saving Modes

Two Power Saving Modes have been incorporated into the SSD1901. These modes are enabled as follows:

**Table 16-1 Power Saving Mode Selection**

| Hardware Power Saving Mode | Software Power Saving Bit 1 | Software Power Saving Bit 0 | Mode |
|---|---|---|---|
| 0 | 0 | 0 | Software Power Saving Mode |
| 0 | 0 | 1 | reserved |
| 0 | 1 | 0 | reserved |
| 0 | 1 | 1 | Normal Operation |
| 1 | X | X | Hardware Power Saving Mode |

Software Power Saving Mode saves power by powering down the panel and stopping display refresh access to the display memory.

Hardware Power Saving Mode saves power by powering down the panel, stopping access to the display memory and registers, and disabling the Host Bus Interface.

## 16.1 Power Saving Mode Function Summary

**Table 16-2 Power Saving Mode Function Summary**

| | Hardware Power Saving | Software Power Saving | Normal |
|---|---|---|---|
| Register Access Possible? | No | Yes | Yes |
| Memory Access Possible? | No | Yes | Yes |
| Look-Up Table registers Access Possible? | No | Yes | Yes |
| Display Control Running? | No | No | Yes |
| Display Active? | No | No | Yes |
| LCDPWR | Inactive | Inactive | Active |
| FPDAT[11:0], FPSHIFT (see note) | Forced low | Forced low | Active |
| FPLINE, FPFRAME, DRDY | Forced low | Forced low | Active |

**Note**
When FPDAT[11:9] are designated as GPIO these pins are not forced low during Power Saving Mode. Unused GPIO pins must be tied to IO $V_{DD}$ - see Table 7-16 LCD Interface Pin Mapping on page 28.

## 16.2 Panel Power Up/Down Sequence

After chip reset or when entering/exiting a Power Saving Mode, the Panel Interface signals follow a power on/off sequence shown below. This sequence is essential to prevent damage to the LCD panel.

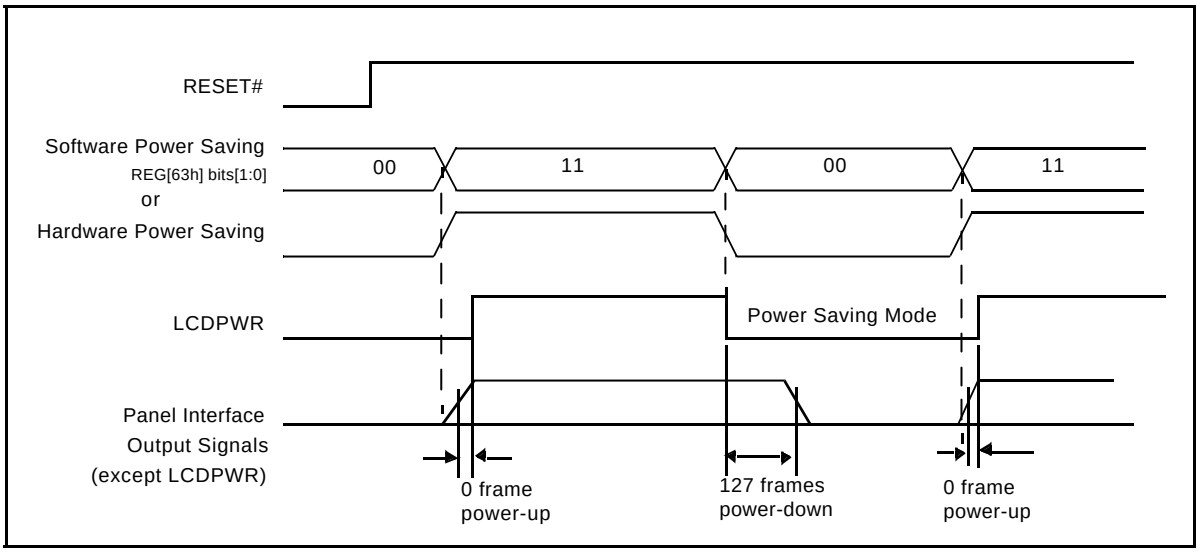


**Figure 16-1 Panel On/Off Sequence**

After chip reset, LCDPWR is inactive and the rest of the panel interface output signals are held “low”. Software initializes the chip (i.e. programs all registers) and then programs REG[63h] bits [1:0] to 11b. This starts the power-up sequence as shown. The power-up/power-down sequence delay is 127 frames. The Look-Up Table registers may be programmed any time after REG[63h] bits[1:0] = 11b.

The power-up/power-down sequence also occurs when exiting/entering Software Power Saving Mode.

# 17 APPLICATION EXAMPLE

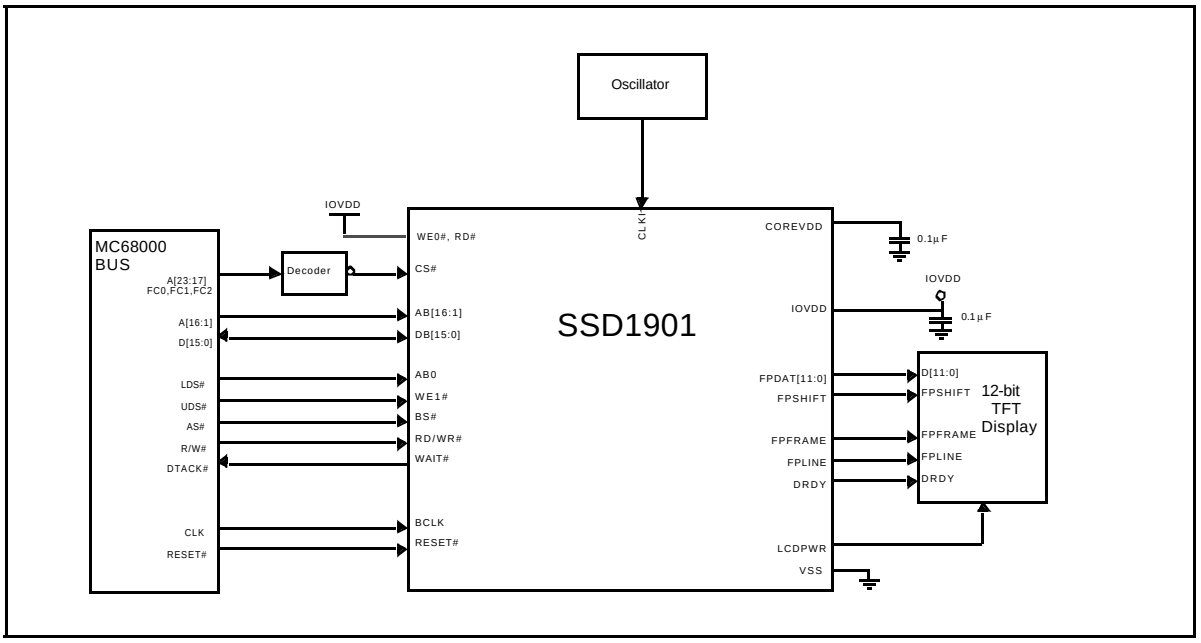


**Figure 17-1 Typical System Diagram (MC68K#1 Bus)**

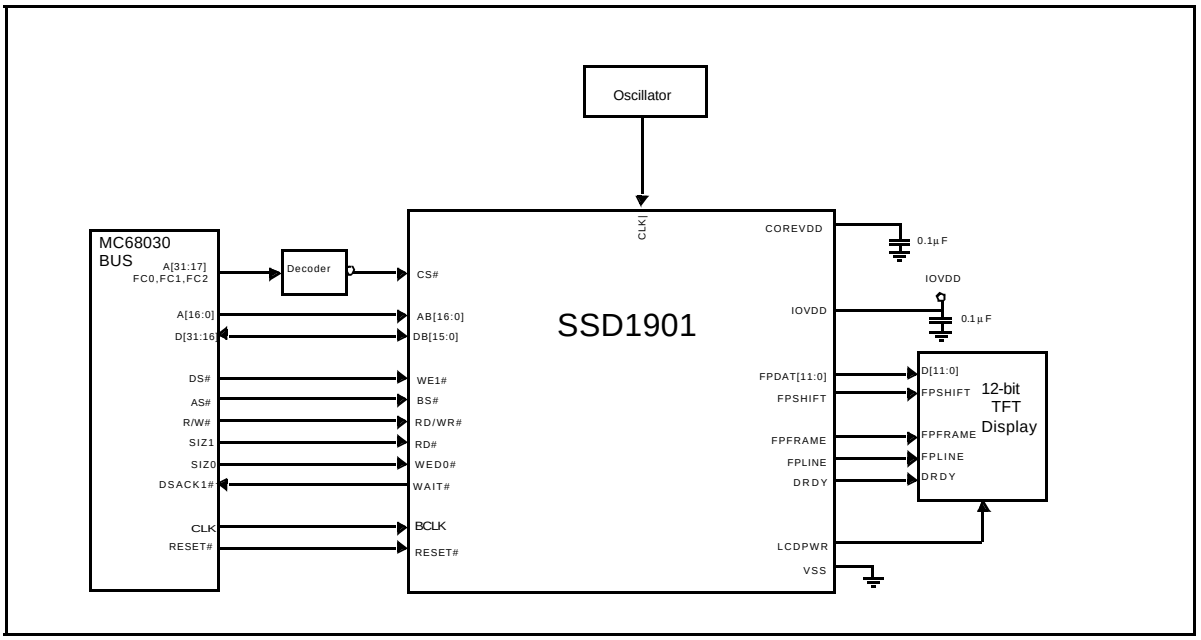


**Figure 17-2 Typical System Diagram (MC68K#2 Bus)**

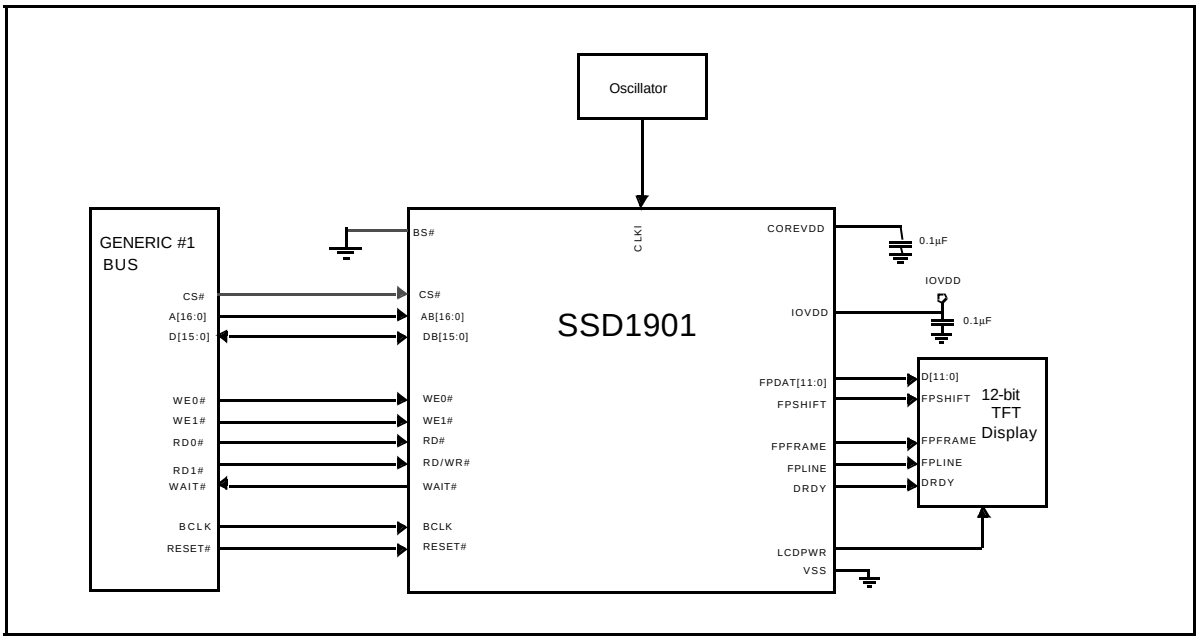


**Figure 17-3 Typical System Diagram (Generic #1 Bus)**

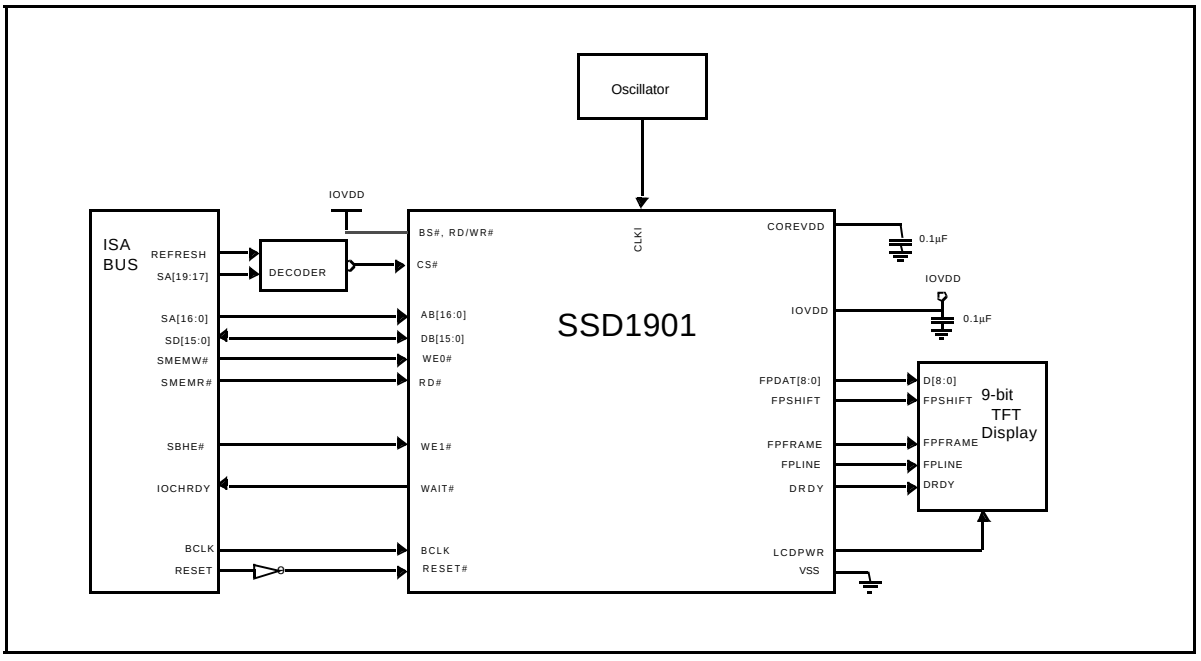


**Figure 17-4 Typical System Diagram (Generic #2 Bus)**

# 18 Appendix

## 18.1 Package Drawing

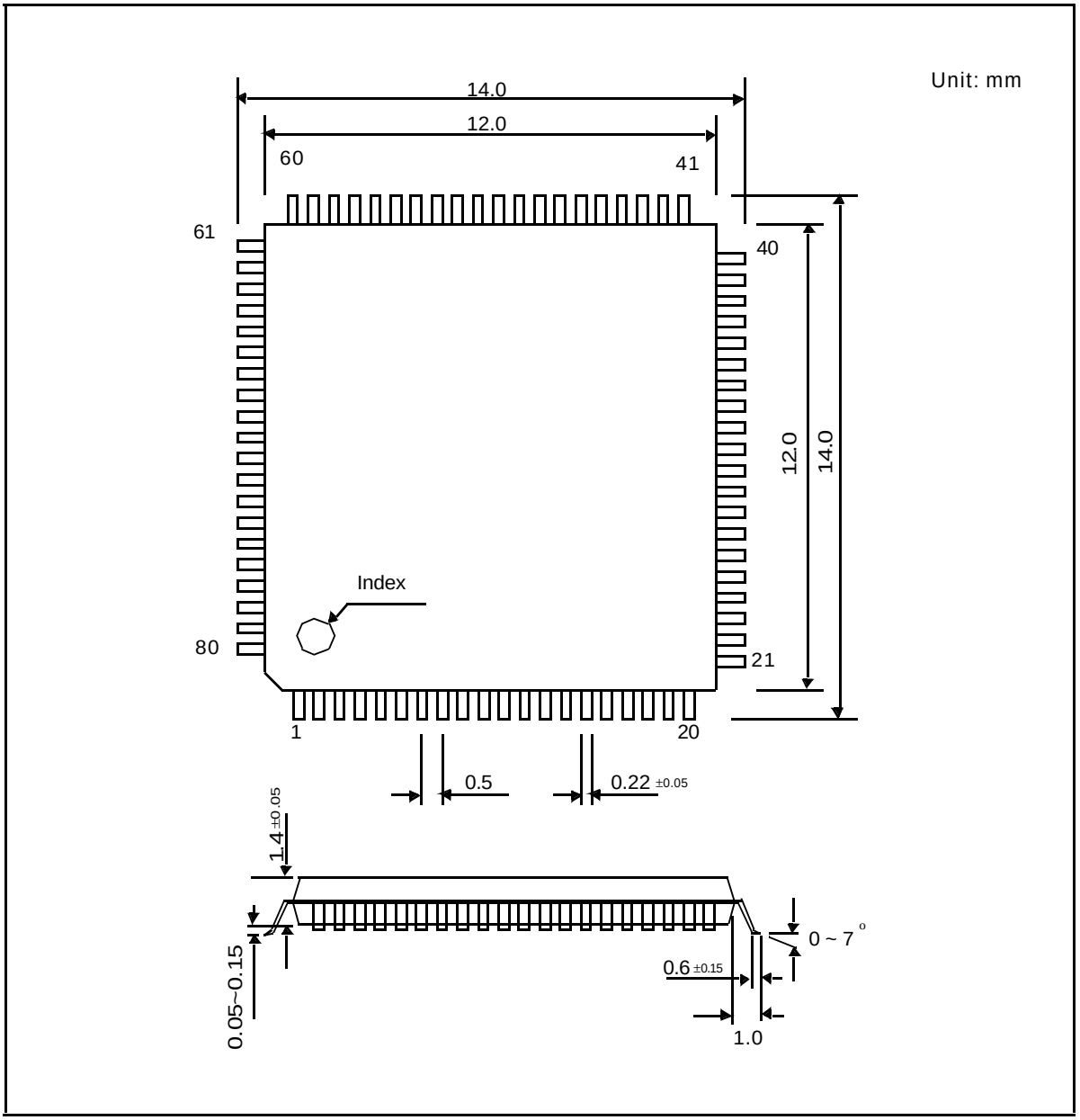


**Figure 18-1 Mechanical Drawing 80 pin LQFP**