AVR32114: Using the AVR32 LCD Controller

32-bit AVR®
Microcontrollers

Application Note

Features

- STN Panel Features
  Single and Dual Scan Color and Monochrome LCD Panels
  4-bit Single Scan, 8-bit Single or Dual Scan, 16-bit Dual Scan
- Interfaces
  Up to 16 Gray Levels for Monochrome and Up to 4096 Colors for Color Panel
  1 or 2 Bits per Pixel (Palletized), 4 Bits per Pixel (Non-palletized) for
- Monochrome
  1, 2, 4 or 8 bits per Pixel (Palletized), 16 Bits per Pixel (Non-palletized) for Color STN Display
- TFT Panel Features
  Single Scan Active TFT LCD Panel
  Up to 24-bit Single Scan Interfaces
  1, 2, 4 or 8 Bits per Pixel (Palletized), 16 or 24 Bits per Pixel (Non-palletized)
- Common Features
  Configurable Screen Size Up to 2048 x 2048
  DMA Controller for Reading the Display Data from an External
- Memory
  2K bytes Input FIF0
  2D Frame Buffer Addressing that allows panning

1 Introduction

This application note explains how to connect the Atmel® AVR®32AP7000 LCD Controller to a TFT display. It includes display selection guidelines, describes the hardware and software configurations and shows examples how to configure the LCD Controller.

2 Display Terminology

2.1 Naming Conventions and Definitions

The following definitions and naming conventions are used in this document.

- Bits per pixel (Bpp): Corresponds to the number of bits that describe the pixel color. (i.e., in 16 bits per pixel mode, each pixel can have 65536 different colors).
- Interface width: The Interface width corresponds to the number of I/O data lines (data bus width) used for the interface between the LCD controller and the display.
- Frame rate: This parameter is given by the display datasheet (in Hz) and describes how often the data should be updated. This rate must be respected to ensure the correct behavior of the display, thus avoiding bad image quality (flickering).
- Display size: Number of pixels on the display. Product of the pixel numbers in a line (horizontal) and a column (vertical) of the display.
- Pixel clock: Clock frequency on used to clock data into an attached display panel. This value depends on the display size, interface width and frame rate.

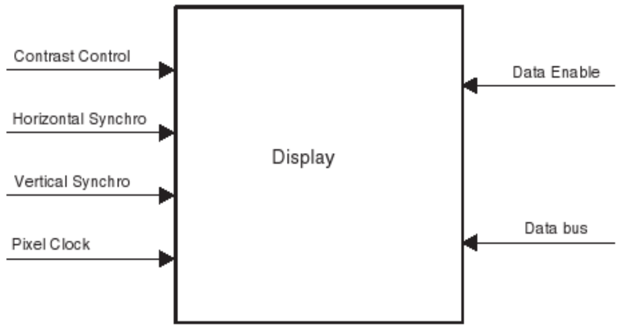
2.2 TFT and STN Technologies

The most commonly used display technologies are TFT (Thin Film Transistor) and STN (Super Twisted Nematic). STN displays use a passive matrix screen technology that has no active or controlling element inside the display cell. Pixels are controlled by energizing the appropriate row and column drive lines of the matrix from outside the display, resulting in a slow frame rate. STN screens have limited color range and viewing angle (~15 degrees max). With TFT LCDs, one to four transistors control each pixel. Typically, one transistor is used for each of the RGB color channels. Because of this direct control technique, TFT screens are sometimes called Active-Matrix LCDs. TFT technology provides more accurate color control, allowing it to display more colors. TFT screens also offer a wider viewing angle range (30 to 70 degrees) than the other popular types of LCDs.

2.3 Display Interface

The typical interface of a display is based on analog and digital inputs. The digital lines are a data bus, a pixel clock, vertical and horizontal synchronization signals and a data enable line. The voltage input is generally used for contrast control. The LCD Controller drives all these interface lines. If more inputs are required by the display, it may be managed by PIOs (power control, AC bias on some STN displays, backlight control, etc.)

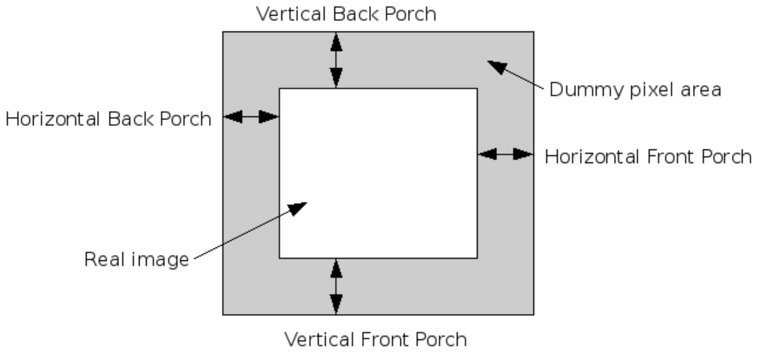
Figure 1. Display Interface



2.4 Blanking

For the internal synchronization mechanism, the display may need some dummy data at the beginning and/or end of a line, and at the beginning and/or end of a frame. This is called blanking. The dummy pixels/lines are not part of the frame buffer are therefore managed by the LCD Controller. Some delays must be introduced in vertical and horizontal timings to support it. These delays are often described as vertical/horizontal front/back porch delays in the display datasheets.

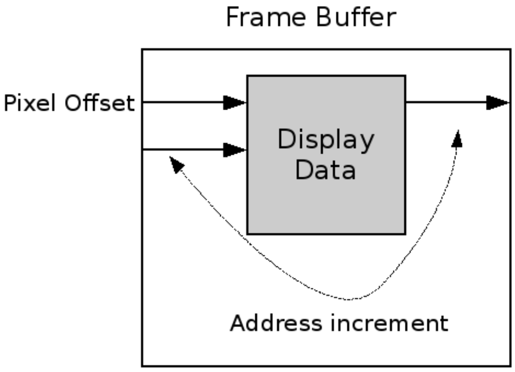
Figure 2. Blanking Parameters



2.5 2D Memory Addressing

The LCD Controller can be configured to work on a frame buffer larger than the actual screen size. By changing the values in a few registers, it is easy to move the displayed area along the frame buffer width and height. The LCD Controller increments the frame buffer pointer according to the pixel offset and the address increment to enable panning.

Figure 3. 2D Frame Buffer Addressing



3 Display Selection Guidelines

3.1 Display Type and Interface

In order to verify the compatibility of a display some points must be considered.

The LCD controller supports TFT RGB (up to 16M colors), color STN (up to 4096 colors) and Monochrome STN (up to 16 gray shades) displays. Other technologies may also be used if their interface is compatible.

The screen size is fully programmable. The maximum size supported is 2048 x 2048. Most of standard display resolution levels are compatible: VGA (640x480), QVGA (320 x 240), etc. In STN mode, single and double scan modes are supported. For a double scan mode, the maximum size for each panel is 1024 x 2048.

The interface must be 3.3V compliant. All the display lines should be present in the lines description list, see Table 3-1. If the display features more lines (i.e. gate command), the LCD Controller is not able to manage them. They could be managed by the PIO Controller or other peripherals. The polarity is programmable on PCLK, DVAL, VSYNC and HSYNC I/O lines.

Table 3-1. I/O Lines of the LCD controller

Name	**Description**	**Type**
CC	Contrast control signal	Output
HSYNC	Line synchronous signal (STN) or Horizontal synchronous signal (TFT)	Output
PCLK	LCD clock signal (STN/TFT)	Output
VSYNC	Frame synchronous signal (STN) or Vertical synchronization signal (TFT)	Output
DVAL	STN AC bias signal for the driver or Data enable signal (TFT)	Output
MOD	LCD Modulation signal	Output
PWR	LCD panel Power enable control signal	Output
GP[7:0]	LCD General purpose lines	Output
LCDD[23:0]	LCD Data Bus output	Output

The Controller supports the following interface configurations:

24-bit TFT single scan, 16-bit STN Dual Scan Mono (Color), 8-bit STN Dual (Single) Scan Mono (Color), 4-bit single scan Mono (Color).

4-bit single scan STN display

4 parallel data lines are used to shift data to successive single horizontal lines one at a time until the entire frame has been shifted and transferred. The 4 LSB pins of LCD Data Bus (LCDD [3:0]) can be directly connected to the LCD driver; the upper 20 bits of the bus (LCDD [23:4]) are not used.

8-bit single scan STN display

8 parallel data lines are used to shift data to successive single horizontal lines one at a time until the entire frame has been shifted and transferred. The 8 LSB pins of LCD Data Bus (LCDD [7:0]) can be directly connected to the LCD driver; the upper pins of the bus(LCDD [23:8]) are not used.

8-bit Dual Scan STN display

Two sets of 4 parallel data lines are used to shift data to successive upper and lower panel horizontal lines one at a time until the entire frame has been shifted and transferred. The bus LCDD[3:0] is connected to the upper panel data lines and the bus LCDD[7:4] is connected to the lower panel data lines. The rest of the LCD Data Bus lines (LCDD[23:8]) are not used.

16-bit Dual Scan STN display

Two sets of 8 parallel data lines are used to shift data to successive upper and lower panel horizontal lines one at a time until the entire frame has been shifted and transferred. The bus LCDD[7:0] is connected to the upper panel data lines and the bus LCDD[15:8] is connected to the lower panel data lines. The rest of the LCD Data Bus lines (LCDD[23:16]) are not used.

TFT single scan display

Up to 24 parallel data lines are used to shift data to successive horizontal lines one at a time until the entire frame has been shifted and transferred. The 24 data lines are divided in three bytes that define the color shade of each color component of each pixel. The LCDD bus is split as LCDD[23:16] for the blue component, LCDD[15:8] for the green component and LCDD[7:0] for the red component. If the LCD Module has lower color resolution (fewer bits per color component), only the most significant bits of each component are used.

3.2 Timings

The maximum and minimum timing supported by the LCD Controller are listed in Table 3-2. Thus if the display should be supported it should not exceed these timing parameters.

Table 3-2. Maximum and minimum timing parameters

Parameter	Min	Max	Unit
Vertical Front Porch	1	256	Line
Vertical Back Porch	1	256	Line
Horizontal Front Porch	1	2048	Pixel Clock Cycles
Horizontal Front Porch	1	256	Pixel Clock Cycles
Vertical Pulse Width	1	64	Line
Horizontal Pulse Width	1	64	Pixel Clock Cycles

3.3 Available Bandwidth

An attached display needs a lot of bandwidth from the expansion bus of the system. This must be considered in the display selection process. The bandwidth needed is calculated by following formula:

NeededBandwidth = DispaySize x BitsPerPixel x FrameRate

For the display on the STK®1000 (DisplaySize 320x240, BitsPerPixel 24, FrameRate 75Hz) the formula delivers following result.

Needed bandwidth = 320 x 240 x 24/8 x 75 Byte/s = 17MByte/s

The bus bandwidth should be at least twice as much as this value to provide enough bandwidth to a running application or even an operating system. To calculate the available bandwidth on the system following formula can be used:

AvailableBandwidth = BusSpeed x BusInterfaceWidth

On the STK1000 the maximum bus speed is 75MHz and the interface width to the memory is 32bit wide. This results in the following maximal available bandwidth:

Available bandwidth = 75 x 32/8 MB/s = 300 MB/s

4 TFT Display Selection Guide Example

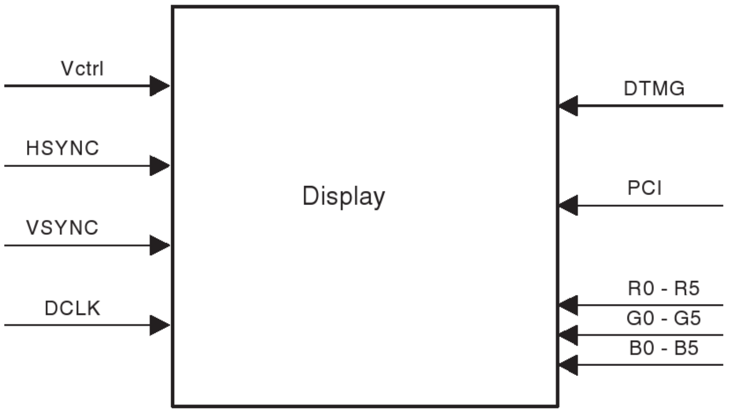
4.1 Display Description

An example display specification (TX09D71VM1CCA) lists the following parameters:

- TFT QVGA
- RGB 262K colors
- VCC [3.0 to 3.6V]
- Frame rate: 60 Hz
- Vertical Pulse Width: typ = 1 (in lines)
- Vertical Back Porch (VBP) : typ = 4 (in lines)
- Vertical Front Porch (VFP) : typ = 2 (in lines)
- Horizontal Pulse Width: typ = 5 (in pixels)
- Horizontal Back Porch (HBP): typ = 17 (in pixels)
- Horizontal Front Porch (HFP): typ = 11 (in pixels)

The Interface is defined as described in picture

Figure 4. Example Display Interface



4.2 Compatibility Test

To determine whether the example display is compatible, all points in the chapter Display Selection Guidelines must be evaluated.

- As the display is a TFT the technology is supported.
- The display size is QVGA (320 x 240) and thus supported by the LCD Controller.
- The LCD Controller can control all pins except the PCI pin. A GPIO pin can be used for the PCI pin. The DTMG signal is controlled by DVAL of the LCD Controller.
- The display has a 3.3V interface as required by the LCD Controller interface.
- The needed bandwidth is (320 x 240 x 24/8 x 60 Byte/s = 13.8MB/s) 13.8MB/s, when using a frame buffer with 24 bits per pixel. This is feasible by the AVR32AP7000 as it can handle up to 300MB/s on the external bus interface.
- All timings are in the range of the maximum and minimum specification.

5 Hardware Connection

5.1 18-bit TFT Display

As the LCD Controller just supports 16 and 24 bit pixel sizes one of these modes must be selected for an 18-bit TFT display. In 24-bit mode the full color space is available (262144 colors) but more memory for the frame buffer is needed. This leads also to a lower performance of the system, as more data has to be transferred from memory to the LCD Controller. In 16-bit mode are fewer colors available but the overhead of the 24-bit mode is removed.

5.1.1 24-bit Mode

Table 5-1. Connection of an 18-bit TFT display to the LCD Controller in 24-bit mode

Memory Bits	LCD Controller RGB Format	LCD Controller interface	Display Interface
0	B0	D0	--
1	B1	D1	R0
2	B2	D2	R1

Memory Bits	LCD Controller RGB Format	LCD Controller interface	Display Interface
3	B3	D3	R2
4	B4	D4	R3
5	B5	D5	R4
6	B6	D6	R5
7	B7	D7	R6
8	G0	D8	--
9	G1	D9	G0
10	G2	D10	G1
11	G3	D11	G2
12	G4	D12	G3
13	G5	D13	G4
14	G6	D14	G5
15	G7	D15	G6
16	B0	D16	--
17	R1	D17	B0
18	R2	D18	B1
19	R3	D19	B2
20	R4	D20	B3
21	R5	D21	B4
22	R6	D22	B5
23	R7	D23	B6

5.1.2 16-bit Mode

In 16-bit mode the LCD Controller uses a 1-5-5-5-pixel format. Table 5-2 shows how this could be mapped to a RGB 6-5-6-pixel format. The 5 bits for red and blue are connected on the MSB pins of the display. B0 and R0 are connected to ground and the data lines D10 and D2 are not used. R5 is connected to a intensity bit. Take a look at the table 35-12 in the device datasheet for pinout details of the LCD controller.

Table 5-2. Connection of an 18-bit display to the LCD Controller in 16-bit mode

Memory Bits	LCD Controller RGB Format	LCD Controller interface	Display interface
0	B0	D4	R1
1	B1	D5	R2
2	B2	D6	R3
3	B3	D7	R4
4	B4	D2	R5
5	G0	D11	G0
6	G1	D12	G1
7	G2	D13	G2

Memory Bits	LCD Controller RGB Format	LCD Controller interface	Display interface
8	G3	D14	G3
9	G4	D15	G4
10	R0	D3	G5
11	R1	D20	B1
12	R2	D21	B2
13	R3	D22	B3
14	R4	D23	B4
15	I	D18	B5

5.2 24-bit TFT Display

A 24-bit TFT display uses all the data lines of the LCD Controller. Table 5-3 shows the pin mapping.

Table 5-3. Connection of a 24-bit TFT display to the LCD Controller in 24-bit mode

Memory Bits	LCD Controller RGB Format	LCD Controller interface	Display Interface
0	B0	D0	R0
1	B1	D1	R1
2	B2	D2	R2
3	B3	D3	R3
4	B4	D4	R4
5	B5	D5	R5
6	B6	D6	R6
7	B7	D7	R7
8	G0	D8	G0
9	G1	D9	G1
10	G2	D10	G2
11	G3	D11	G3
12	G4	D12	G4
13	G5	D13	G5
14	G6	D14	G6
15	G7	D15	G7
16	B0	D16	B0
17	R1	D17	B1
18	R2	D18	B2
19	R3	D19	B3
20	R4	D20	B4
21	R5	D21	B5
22	R6	D22	B6

Memory Bits	LCD Controller RGB Format	LCD Controller interface	Display Interface
23	R7	D23	B7

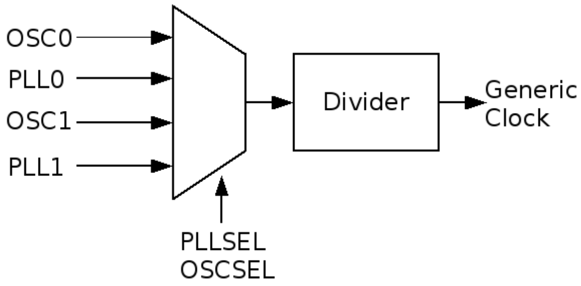
6 Software Initialization Sequence

Following steps are needed to configure and activate the LCD Controller.

6.1 Power Manager

The LCD Controller needs two clocks. One clock is derived from the High Speed Bus Matrix and can be activated in the HSBMASK register in the power manager. The other clock is a generic clock and the source of this clock can be OSC0, OSC1, PLL0 and PLL1. Especially when a high and accurate pixel clock is needed it should be considered to set one PLL or oscillator to this frequency, as the divider might not be able to deliver an acceptable value. In addition can an accurate pixel clock lead to better system performance as this minimizes the needed bandwidth. Before using the LCD Controller, the programmer must ensure that these two clocks are enabled.

Figure 5. Generic Clock selection



6.2 LCD Controller Pins

The pins used for interfacing the LCD Controller are multiplexed with PIO lines. The programmer must first program the PIO Controller to assign the pins to their peripheral function. If some I/O lines of the LCD Controller are not used by the application, the PIO Controller can use them for other purposes.

6.3 Disabling LCD and DMA Controller

To be sure that initialization is correct, first disable the LCD Controller and DMA by clearing the LCD_PWR and the DMAEN bits in the PWRCON and the DMACON registers, respectively.

6.4 Pixel Clock

The pixel clock is used to clock the pixel data out of the module into the attached display. The frequency of this clock affects the frame rate and all timing parameters and should therefore be set carefully.

To estimate the needed (theoretical) pixel clock use following formulas according to the display type.

STN Mono: PixelClockTheo = FrameRate x DisplaySize / IfWidth

STN Color: PixelClockTheo = FrameRate x DisplaySize x 3 / IfWidth

TFT: PixelClockTheo = FrameRate x DisplaySize

As the LCD Controller pixel clock divider can only divide by 1 or 2*(n+1), this clock source should be delivered as a multiple of this divisor or at least close to it to match the desired pixel clock rate. As mentioned before an inaccurate pixel clock leads to lower performance and should therefore be avoided.

If the bypass option is set in the LCDCON1 register the pixel clock is equal to the generic clock set up in the power manager. Otherwise following rule is used to compute the pixel clock.

PixelClock = GenericClock / (CLKVAL + 1) x 2

Where

- the GenericClock is the source clock of the divider and
- CLKVAL is the divider value in the LCDCON1 register.

The calculation of an adequate divider requires following steps.

CLKVALTheo = PixelClockTheo /((2 x GenericClock) -1)

With

- CLKVALTheo: Theoretical divider value
- PixelClockTheo: Desired pixel clock

Round the CLKVALTheo to the next smaller integer number to get the real CLKVAL value and check your resulting pixel clock:

PixelClock = GenericClock /(2 x (CLKVAL + 1))

Example:

A TFT QVGA display has the following features: 320 x 240, 60 Hz, 262K colors, 18-bit data bus. LCD Controller clock is 100 MHz.

PixelClockTheo = 320 x 240 x 60 = 4.608 MHz

CLKVALTheo = 100 / (2 x 4.608) - 1 = 9.85

CLKVAL = 9

PixelClock = 100 / (2 x (9 + 1)) = 5 MHz.

6.5 Display Size

The display size is set in the Frame Control Register (LCDFRMCTRL) register. Two values must be calculated according to the display type and size.

LINEVAL: Vertical size of the display. For all display types this value is the vertical display size in pixels minus one. In dual scan mode, vertical display size refers to the size of each panel.

HOZVAL: Horizontal size of the LCD display depending on the number of horizontal pixels, the interface width and the display type. Valid settings are:

STN Monochrome Mode: HOZVAL = (HorizontalDisplaySize / IfWidth) – 1

STN Color Mode: HOZVAL = HorizontalDisplaySize = 3 x NumberOfHorizontalPixels:

Note:

- Valid interface width in 4-bit single scan or 8-bit dual scan STN display mode is 4
- Valid interface width in 8-bit single scan or 16-bit dual scan STN display mode is 8
- If the value calculated for HOZVAL with the above formula is not an integer, it must be rounded up to the next integer value.

In TFT Mode: HOZVAL = HorizontalDisplaySize - 1

6.6 Timings

The vertical and horizontal pulse width is configurable through the VPW and HPW fields in the LCDTIM1 and LCDTIM2 registers. It is defined in number of pixel clock cycles. These values should be taken from the display datasheet.

6.7 Blanking

The blanking times, Vertical Front Porch, Vertical Back Porch, Horizontal Front Porch and Horizontal Back Porch must be set according the display datasheet. These values can be set in the registers LCDTIM1 and LCDTIM2.

6.8 Interrupt

The LCD Controller interrupts are configured as the other AVR32 peripherals through the Enable/Disable/Mask/Status/Clear registers.

6.9 Contrast

The contrast PWM Signal is programmable from 0x0 to 0xFF in the CVAL field of the CONTRAST_VAL register. Polarity and frequency are also programmable in the PS and POL fields of the CONTRAST_CTR register. To enable/disable it, set or clear the ENA bit.

6.10 DMA Base Address

In TFT and STN Single Scan mode, the frame buffer address is stored in the DMABADDR1 register. In STN Double Scan mode, the second frame buffer address is stored in the DMABADDR2 register. If the frame buffer is located in the SDRAM, it is recommended to store it in a different bank than the application in order to optimize the SDRAM access.

6.11 DMA Frame Configuration

Two parameters are needed to configure the DMA. These are the burst length and the frame size. The frame size for the DMA transfer measured in 32bit words and can be calculated by following formula.

FRMSIZE = DisplaySize x Bpp / 32

Note that the Bpp parameter to use is the one used for the LCD Controller configuration. If it is an 18-bit display, the configuration is 16 bits or 24 bits. In this formula, Bpp is 16 or 24, not 18.

6.12 FIFO

A FIFO block buffers the input data read by the DMA module. It contains two input FIFOs to be used in dual scan configuration that are configured as a single FIFO when used in single scan configuration.

The upper threshold of the FIFOs can be configured in the FIFOTH field of the LCDFIFO register. The LCD Controller core will request a DMA transfer when the number of words in each FIFO is less than FIFOTH words. To avoid overwriting in the FIFO and to maximize the FIFO utilization, the FIFOTH should be programmed with:

- FIFOTH = LCDFIFOSize - (2 x DMABurstLength + 3)

Where:

- LCDFIFOSize is the effective size of the FIFO. It is the total FIFO memory size in single scan mode and half that size in dual scan mode. On the AP7000 this is 512 in single scan mode and 256 in dual scan mode.
- DMABurstLength is the burst length of the transfers made by the DMA

6.13 Enabling DMA and LCD Controller

Reset and enable the DMA by setting the DMARST and DMAEN bits in the DMACON register. It must be done in 2 steps.

Enable the LCD Controller by setting the LCD_PWR bit in the PWRCON register. At this moment, the LCD Controller is running and is sending frames to the displays via the LCD Controller interface. Depending on the display interface, other IOs may be configured to enable the display, or for backlight activation.

7 Optimizations

If the LCD Controller has to transfer a lot of data, as this is required when a large display is used, some optimizations may be needed to avoid a buffer under resulting in de-synchronization of the timing generator. To get the maximum available bandwidth the HSB bus matrix should run at the maximum frequency.

7.1 LCD Controller As Fixed Master

A High Speed Bus Matrix slave can only be connected to one master at a time. After the master has finished, the slave device has three possibilities that can be configured:

- If no other master request is pending, the slave disconnects the master.
- If no other master request is pending, the slave stays connected to the current master.
- If no other master request is pending, the slave connects to the fixed master that has been specified in the configuration.

The DMA Controller of the LCD Controller can be set as a fixed master for the external bus interface. This means after another master has accessed the external bus interface it sets the bus back to the LCD Controller. So no switching has to be done when new data has to be fetched from a frame buffer in the external memory. This results in a faster access time.

To set up the LCD Controller as default master of the external bus interface the fields DEFMSTR_TYPE and FIXED_DEFMSTR have to be set up in the SCFG4 register.

Please take a look at chapter “HSB Bus Matrix” and chapter “Memories” of the users manual for a detailed description.

7.2 Priority-Based Arbitration

The slave, here the external bus interface, uses a round robin algorithm by default to schedule accesses of masters to its self. This can be critical if other masters request access to the slave and thus reducing the bandwidth for the LCD Controller. This can lead to a buffer underflow in the LCD Controller, resulting in loss of synchronization. A typical symptom of a buffer underflow is when the image on the screen is shifted vertically/horizontally. To resynchronize the timing generator and thus getting back a correct view, the LCD Controller and its DMA engine have to be restarted.

A possible solution to guarantee access to the external bus interface for the LCD Controller is to enable a priority based arbitration algorithm for this slave. If the LCD Controller gets the highest priority for the slave it is guaranteed that it gets access before all other masters. A major draw back is that all other masters will have a reduced access or even in a worst case no access at all to the slave.

Chapter “HSB Bus Matrix” of the device datasheet describes how to activate the priority-based arbitration and the setting of priorities.

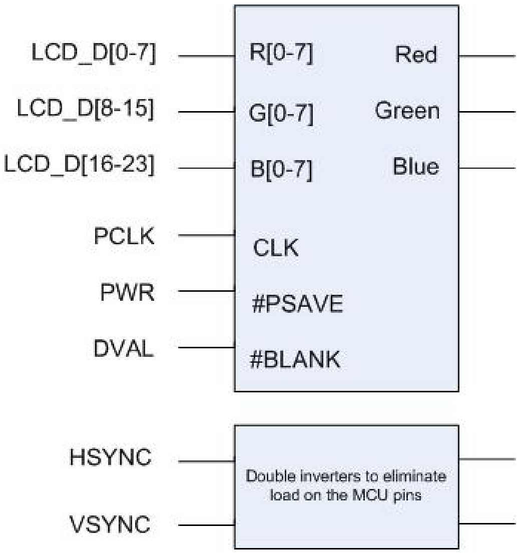
8 Attaching An Analog Monitor To The LCD Controller

The LCD Controller can drive a monitor with an analog interface, but for this purpose are some external components, such as a video DAC needed. This circuitry is already available on the STK1000 development board.

8.1 Hardware Setup

A basic view of the needed hardware to connect an analog monitor shows image … This solution is based on a video DAC used on the STK1000 development board. The schematics of this board is available on the Atmel website. Take a look at this document for a detailed description of the circuitry.

Figure 8-1. Block diagram of the hardware



- **BLANK**: The output on the red/green/blue signals is forced to the blanking level. This pin is controlled by the data valid signal of the LCD Controller (DVAL).
- **PSAVE**: Sets the video DAC in a power save mode.
- **R[0-7]/G[0-7]/B[0-7]**: Input for the pixel data from the LCD Controller. If less than 24-bits per pixel are used the lower pins of each color may be ignored.
- **Red/Green/Blue**: These are differential, high impedance current sources. These drive the video levels into a 75 Ohm terminated load.
- **CLK**: Input for the pixel clock from the LCD Controller (PCLK).
- **HSYNC and VSYNC**: These signals are routed through two inverters to eliminate any load on the CPU pins.

As monitors select their synchronization set up also according to the VSYNC frequency and pulse width an additional external circuitry may be needed to ensure the minimal pulse width accepted by the monitor, as the LCD Controller cannot generate long vertical synchronization pulses. See table … for the maximum possible pulse width.

8.2 Timings

A correct timing set up of the video signal is an essential part in the configuration of the LCD Controller. The timings have to match the parameters given by the monitor's specification. If this condition is not met following results are possible:

- Data is displayed incorrect.
- No data at all is shown, as the monitor does not recognize the format.
- The monitor's synchronization circuitry could be damaged.

Many monitors support the timings specified by the ®VESA organization. These timings are available and free of charge on http://www.vesa.org.

8.2.1 Horizontal Timing

A monitor detects the horizontal synchronization pulses on the HSYNC line provided from the LCD Controller and then decides based on the polarity, frequency, and/or duration of those pulses how to set up its internal synchronization engine. Thus to get this working it is important to know the supported horizontal synchronization frequencies of the monitor, as well as the acceptable width of the according synchronization pulses. If the width of pulse is incorrect, it can make the displayed information too large or too small, as well as possibly preventing the monitor from synchronizing to the generated signal. The acceptable range of sync pulse width and polarity for a given frequency should be given in the specifications for the monitor. If this is not the case it is recommended to contact the manufacturer.

8.2.2 Vertical Timing

The vertical timing controls the vertical movements of the display scan. The monitor detects the vertical synchronization pulses on the VSYNC line and equivalent to the horizontal timing sets up its internal synchronization engine based on polarity, frequency, and/or duration of those pulses. Therefore it is also necessary to know the supported vertical synchronization frequency ranges, acceptable pulse width and signal polarity for the given monitor.

9 Implementation

9.1 Driver files

The LCD Controller driver consists of two files “lcdc.c” and “lcdc.h”. The “lcdc.c” file contains all source code needed to initialize the LCD Controller. The header file “lcdc.h” declares the initialization function and the configuration for the LCD Controller.

Before the LCD Controller can be used all needed clocks must be activated. The next step is to pass a valid configuration to the initialization function. This function sets up the LCD Controller as described in the chapter Software Initialization Sequence. If the Controller was set up correct the frame buffer can now be used to display data.

9.2 Example code

9.2.1 Test Screen Example

This example shows how to set up the LCD controller for the TFT display on the STK1000 and shows a simple test pattern on the screen.

9.2.2 2D Addressing Mode Example

This example shows the use of the 2D addressing mode on the LCD Controller. It uses the TFT display on the STK1000 to show a part of the "virtual" frame buffer. The virtual frame buffer (640x480) is four times the size of the LCD viewing area (320x240). A bitmap picture (320x240) is read from the flash, enlarged by four and then written to the frame buffer. So only a part of the picture will be visible on the screen. Setting the base pointer of the frame buffer with switches on the STK1000 enables panning.

Before you run this application program the picture (AVR32.bmp) into the flash at address 0x00400000. Use for instance the avr32program application for this purpose.

```
avr32program program -F bin -O 0x00400000 -v -e -f cfi@0 AVR32.bmp
```

To program the application into the flash use:

```
avr32program program -v -e -f cfi@0 lcdc_example.elf
```

If you intend to program the picture to another location change the define BITMAP_FILE_ADDRESS accordingly in the source. Feel free to use your own bitmap file.

The input (switches) header marked J25, used for moving the viewing area, must be connected to the header labeled J1 (PORTB[0..7]).

To move the viewing area use following switches:

- Switch0: Move viewing area 10 pixels to the right
- Switch1: Move viewing area 10 pixels to the left
- Switch2: Move viewing area 10 lines up
- Switch3: Move viewing area 10 lines down

9.2.3 VGA, SVGA, XVGA Example

This example shows how to set up the LCD Controller in combination with an external analog monitor. Three configurations for the resolutions VGA 648x480, SVGA

800x600 and XVGA 1024X768 are available. Check your monitor specifications if it supports one of these configurations. If not, the parameters can be adjusted (i.e. blanking, polarity and synchronization pulse width).

To choose a configuration uncomment one of the defines in the example code.

- CONF_640_480_60 -> VGA 640x480@60Hz
- CONF_800_600_60 -> SVGA 800x600@60Hz
- CONF_1024_768_60 -> XVGA 1024x768@60Hz

Before you run this application program the picture (AVR32.bmp) into the flash at address 0x00400000. Use for instance the avr32program application for this purpose.

```
avr32program program -F bin -O 0x00400000 –v –e –fcfi@0 AVR32.bmp
```

To program the application into the flash use:

```
avr32program program -v -e -f cfi@0 lcdc_example.elf
```

If this picture is not available a blank rectangle will be visible in the upper left corner. The pictures resolution is 320x240 (QVGA).

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr32@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.